

## Modelarts

# Usermanual

Date 2024-04-30

# **Contents**

1 Service Overview	1
1.1 Infographics	1
1.1.1 What Is ModelArts	2
1.2 What Is ModelArts?	
1.3 Functions	5
1.4 Basic Knowledge	6
1.4.1 Introduction to the AI Development Lifecycle	6
1.4.2 Basic Concepts of AI Development	7
1.4.3 Common Concepts of ModelArts	9
1.4.4 Introduction to Development Tools	
1.5 AI frameworks supported by ModelArts	
1.6 Related Services	
1.7 How Do I Access ModelArts?	
2 Preparations	20
2.1 Configuring Access Authorization (Global Configuration)	
2.2 Creating an OBS Bucket	
3 ExeML	27 27 28 28 28 29
<b>3 ExeML</b> 3.1 Introduction to ExeML 3.2 Image Classification. 3.2.1 Preparing Data. 3.2.2 Creating a Project. 3.2.3 Labeling Data.	27 
<b>3 ExeML</b> 3.1 Introduction to ExeML 3.2 Image Classification 3.2.1 Preparing Data 3.2.2 Creating a Project 3.2.3 Labeling Data 3.2.4 Training a Model	<b>27</b> 28 28 28 29 32 35
<b>3 ExeML</b>	<b>27</b> 
<b>3 ExeML</b> 3.1 Introduction to ExeML 3.2 Image Classification 3.2.1 Preparing Data. 3.2.2 Creating a Project. 3.2.3 Labeling Data. 3.2.4 Training a Model. 3.2.5 Deploying a Model as a Service. 3.3 Object Detection.	<b>27</b> 28 28 28 29 32 35 37 39
<b>3 ExeML</b>	<b>27</b> 28 28 29 32 35 37 39 39
<b>3 ExeML</b>	<b>27</b> 28 28 29 32 35 37 39 39 42
<b>3 ExeML</b>	<b>27</b> 28 28 29 32 35 37 39 39 42 45
<b>3 ExeML</b>	<b>27</b> 
<b>3 ExeML</b> 3.1 Introduction to ExeML 3.2 Image Classification 3.2.1 Preparing Data 3.2.2 Creating a Project 3.2.3 Labeling Data 3.2.4 Training a Model 3.2.5 Deploying a Model as a Service 3.3 Object Detection 3.3.1 Preparing Data 3.3.2 Creating a Project 3.3 Labeling Data 3.3.3 Labeling Data 3.3.4 Training a Model 3.3.5 Deploying a Model as a Service	<b>27</b> 28 28 29 32 35 35 37 39 42 42 45 48 50
<b>3 ExeML</b>	<b>27</b> 

3.4.2 Creating a Project	55
3.4.3 Training a Model	59
3.4.4 Deploying a Model as a Service	60
3.5 Tips	62
3.5.1 How Do I Quickly Create an OBS Bucket and a Folder When Creating a Project?	63
3.5.2 Where Are Models Generated by ExeML Stored? What Other Operations Are Supported?	63
4 Workflow	65
4.1 MLOps Overview	
4.2 What Is Workflow?	67
4.3 How to Use a Workflow?	69
4.3.1 Using a Workflow Subscribed to From AI Hub	
4.3.2 Configuring a Workflow	69
4.3.2.1 Configuration Entries	69
4.3.2.2 Runtime Configurations	70
4.3.2.3 Resource Configurations	71
4.3.2.4 Tab Configuration	71
4.3.2.5 Input and Output Configurations	72
4.3.2.6 Phase Parameters	72
4.3.2.7 Saving Configurations	73
4.3.3 Starting, Stopping, Searching for, Copying, or Deleting a Workflow	73
4.3.4 Viewing Workflow Execution Records	76
4.3.5 Retrying, Stopping, or Proceeding a Phase	78
4.3.6 Partial Execution	
5 Data Management	80
5.1 Introduction to Data Preparation	
5.2 Getting Started	81
5.3 Introduction to Data Preparation	
5.4 Creating a Dataset	87
5.4.1 Dataset Overview	87
5.4.2 Creating a Dataset	90
5.4.3 Modifying a Dataset	95
5.5 Importing Data	
5.5.1 Introduction to Data Importing	96
5.5.2 Importing Data from OBS	
5.5.2.1 Introduction to Importing Data from OBS	98
5.5.2.2 Importing Data from an OBS Path	100
5.5.2.3 Specifications for Importing Data from an OBS Directory	103
5.5.2.4 Importing a Manifest File	108
5.5.2.5 Specifications for Importing a Manifest File	110
5.5.3 Importing Data from Local Files	127
5.6 Data Analysis and Preview	128
5.6.1 Processing Data	129

5.6.2 Auto Grouping	
5.6.3 Data Filtering	
5.6.4 Data Feature Analysis	
5.7 Labeling Data	
5.8 Publishing Data	
5.8.1 Introduction to Data Publishing	
5.8.2 Publishing a Data Version	
5.8.3 Managing Data Versions	
5.9 Exporting Data	
5.9.1 Introduction to Exporting Data	
5.9.2 Exporting Data to a New Dataset	
5.9.3 Exporting Data to OBS	
5.10 Introduction to Data Labeling	
5.11 Manual Labeling	
5.11.1 Creating a Labeling Job	
5.11.2 Image Labeling	
5.11.2.1 Image Classification	
5.11.2.2 Object Detection	
5.11.2.3 Image Segmentation	
5.11.3 Text Labeling	
5.11.3.1 Text Classification	
5.11.3.2 Named Entity Recognition	
5.11.3.3 Text Triplet	
5.11.4 Audio Labeling	
5.11.4.1 Sound Classification	
5.11.4.2 Speech Labeling	
5.11.4.3 Speech Paragraph Labeling	
5.11.5 Video Labeling	
5.11.6 Viewing Labeling Jobs	
5.11.6.1 Viewing My Created Labeling Jobs	
5.11.6.2 Viewing My Participated Labeling Jobs	
5.12 Auto Labeling	
5.12.1 Creating an Auto Labeling Job	
5.13 Team Labeling	
5.13.1 Team Labeling Overview	
5.13.2 Creating and Managing Teams	
5.13.2.1 Managing Teams	
5.13.2.2 Managing Team Members	
5.13.3 Creating a Team Labeling Job	
5.13.4 Logging In to ModelArts	
5.13.5 Starting a Team Labeling Job	
5.13.6 Reviewing Team Labeling Results	

5.13.7 Accepting Team Labeling Results	220
6 Devenviron	223
6.1 Introduction to DevEnviron	
6.2 Application Scenarios	
6.3 Managing Notebook Instances	
6.3.1 Creating a Notebook Instance	
6.3.2 Accessing a Notebook Instance	229
6.3.3 Searching for, Starting, Stopping, or Deleting a Notebook Instance	
6.3.4 Changing a Notebook Instance Image	231
6.3.5 Changing the Flavor of a Notebook Instance	231
6.3.6 Selecting Storage in DevEnviron	232
6.3.7 Dynamically Expanding EVS Disk Capacity	235
6.3.8 Modifying the SSH Configuration for a Notebook Instance	
6.3.9 Viewing the Notebook Instances of All IAM Users Under One Tenant Account	
6.4 JupyterLab	
6.4.1 Operation Process in JupyterLab	
6.4.2 JupyterLab Overview and Common Operations	
6.4.3 Code Parametrization Plug-in	
6.4.4 Using ModelArts SDK	
6.4.5 Using the Git Plug-in	
6.4.6 Visualized Model Training	
6.4.6.1 Introduction to Training Job Visualization	255
6.4.6.2 MindInsight Visualization Jobs	
6.4.6.3 TensorBoard Visualization Jobs	
6.4.7 Uploading and Downloading Data in Notebook	
6.4.7.1 Uploading Files to JupyterLab	
6.4.7.1.1 Scenarios	
6.4.7.1.2 Uploading Files from a Local Path to JupyterLab	
6.4.7.1.3 Cloning an Open-Source Repository in GitHub	
6.4.7.1.4 Uploading OBS Files to JupyterLab	279
6.4.7.1.5 Uploading Remote Files to JupyterLab	
6.4.7.2 Downloading a File from JupyterLab to a Local Path	
6.5 Local IDE	
6.5.1 Operation Process in a Local IDE	
6.5.2 Local IDE (PyCharm)	
6.5.2.1 Connecting to a Notebook Instance Through PyCharm Toolkit	
6.5.2.1.1 PyCharm Toolkit	286
6.5.2.1.2 Downloading and Installing PyCharm Toolkit	
6.5.2.1.3 Connecting to a Notebook Instance Through PyCharm Toolkit	
6.5.2.2 Manually Connecting to a Notebook Instance Through PyCharm	295
6.5.2.3 Submitting a Training Job Using PyCharm Toolkit	
6.5.2.3.1 Submitting a Training Job (New Version)	

C C D D D Characing a Training Joh	205
6.5.2.3.2 Stopping a Training Job	
6.5.2.3.3 Viewing Training Logs	
6.5.2.4 Uploading Data to a Notebook Instance Using PyCharm	
6.5.3 Local IDE (VS Code)	
6.5.3.1 Connecting to a Notebook Instance Through VS Code	
6.5.3.2 Installing VS Code	
6.5.3.3 Connecting to a Notebook Instance Through VS Code With One Click	
6.5.3.4 Connecting to a Notebook Instance Through VS Code Toolkit	
6.5.3.5 Manually Connecting to a Notebook instance Through VS Code	
6.5.3.6 Remotely Debugging In VS Code	
6.5.3.7 Optoading and Downloading Files in VS Code	
6.5.4 Local IDE (Accessed Using SSH)	
6.6 Using Notebook to Develop Ascend Operators	
6.7 ModelArts CLI Command Reference	
6.7.1 ModelArts CLI Overview	
6.7.2 (Optional) Installing ma-cli Locally	
6.7.3 Autocompletion for ma-cli Commands	
6.7.4 ma-cli Authentication	
6.7.5 ma-cli Image Building Command	
6.7.5.1 ma-cli Image Building Command	
6.7.5.2 Obtaining an Image Creation Template	
6.7.5.3 Loading an Image Creation Template	
6.7.5.4 Obtaining Registered ModelArts Images	
6.7.5.5 Creating an Image in ModelArts Notebook	
6.7.5.6 Obtaining Image Creation Caches in ModelArts Notebook	
6.7.5.7 Clearing Image Creation Caches in ModelArts Notebook	
6.7.5.8 Registering SWR Images with ModelArts Image Management	
6.7.5.9 Deregistering a Registered Image from ModelArts Image Management	
6.7.5.10 Debugging an SWR Image on an ECS	361
6.7.6 Using the <b>ma-cli ma-job</b> Command to Submit a ModelArts Training Job	
6.7.6.1 ma-cli ma-job Command Overview	
6.7.6.2 Obtaining ModelArts Training Jobs	
6.7.6.3 Submitting a ModelArts Training Job	
6.7.6.4 Obtaining ModelArts Training Job Logs	
6.7.6.5 Obtaining ModelArts Training Job Events	
6.7.6.6 Obtaining ModelArts AI Engines for Training	
6.7.6.7 Obtaining ModelArts Resource Specifications for Training	
6.7.6.8 Stopping a ModelArts Training Job	
6.7.7 Using ma-cli to Copy OBS Data	
7 Training Management	377
7.1 Introduction to Model Development	
7.2 Preparing Data	

7.3 Preparing Algorithms	380
7.3.1 Introduction to Algorithm Preparation	380
7.3.2 Using a Preset Image (Custom Script)	381
7.3.2.1 Overview	381
7.3.2.2 Developing a Custom Script	382
7.3.2.3 Creating an Algorithm	384
7.3.3 Using Custom Images	388
7.3.4 Viewing Algorithm Details	391
7.3.5 Searching for an Algorithm	392
7.3.6 Deleting an Algorithm	392
7.4 Performing a Training	392
7.4.1 Creating a Training Job	393
7.4.2 Viewing Training Job Details	407
7.4.3 Viewing Training Job Events	409
7.4.4 Training Job Logs	411
7.4.4.1 Introduction to Training Job Logs	411
7.4.4.2 Common Logs	412
7.4.4.3 Ascend Logs	413
7.4.4.4 Viewing Training Job Logs	417
7.4.4.5 Locating Faults by Analyzing Training Logs	418
7.4.5 Cloud Shell	419
7.4.5.1 Logging In to a Training Container Using Cloud Shell	419
7.4.5.2 Keeping a Training Job Running	421
7.4.5.3 Preventing Cloud Shell Session from Disconnection	422
7.4.5.4 Analyzing the Call Stack of the Suspended Process Using the py-spy Tool and Locating the Suspended Problem By Analyzing Code	422
7.4.6 Viewing the Resource Usage of a Training Job	423
7.4.7 Evaluation Results	424
7.4.8 Viewing Fault Recovery Details	429
7.4.9 Viewing Environment Variables of a Training Container	429
7.4.10 Stopping, Rebuilding, or Searching for a Training Job	433
7.4.11 Releasing Training Job Resources	434
7.5 Training Experiment	434
7.5.1 Introduction to Experiment	434
7.5.2 Adding a Training Job to an Experiment	434
7.5.3 Viewing an Experiment	436
7.5.4 Deleting an Experiment	437
7.6 Advanced Training Operations	437
7.6.1 Selecting a Training Mode	437
7.6.2 Automatic Recovery from a Training Fault	440
7.6.2.1 Training Fault Tolerance Check	440
7.6.2.2 Fault Dying Gasp	444
7.6.3 Resumable Training and Incremental Training	446

7.6.4 Detecting Training Job Suspension	447
7.6.5 Priority of a Training Job	448
7.6.6 Permission to Set the Highest Job Priority	
7.7 Distributed Training	
7.7.1 Distributed Training	450
7.7.2 Single-Node Multi-Card Training Using DataParallel	451
7.7.3 Multi-Node Multi-Card Training Using DistributedDataParallel	453
7.7.4 Distributed Debugging Adaptation and Code Example	454
7.7.5 Sample Code of Distributed Training	458
8 Inference Deployment	
8.1 Introduction to Inference	465
8.2 Managing AI Applications	
8.2.1 Introduction to AI Application Management	
8.2.2 Creating an AI Application	
8.2.2.1 Importing a Meta Model from a Training Job	
8.2.2.2 Importing a Meta Model from OBS	
8.2.2.3 Importing a Meta Model from a Container Image	473
8.2.3 Viewing the AI Application List	
8.2.4 Viewing Details About an AI Application	479
8.2.5 Managing AI Application Versions	
8.2.6 Viewing Events of an AI Application	
8.3 Deploying an AI Application as a Service	487
8.3.1 Deploying AI Applications as Real-Time Services	
8.3.1.1 Deploying as a Real-Time Service	487
8.3.1.2 Viewing Service Details	492
8.3.1.3 Testing the Deployed Service	498
8.3.1.4 Accessing Real-Time Services	499
8.3.1.4.1 Accessing a Real-Time Service	499
8.3.1.4.2 Authentication Mode	500
8.3.1.4.3 Access Mode	503
8.3.1.4.4 Accessing a Real-Time Service Through WebSocket	510
8.3.1.4.5 Server-Sent Events	513
8.3.1.5 Cloud Shell	
8.3.2 Deploying AI Applications as Batch Services	515
8.3.2.1 Deploying as a Batch Service	515
8.3.2.2 Viewing the Batch Service Prediction Result	521
8.3.3 Deploying AI Applications as Edge Services	521
8.3.3.1 Deploying an Edge Service	
8.3.3.2 Accessing an Edge Service Deployed on IEF Edge Nodes	524
8.3.3.3 Accessing an Edge Service Deployed in a ModelArts Edge Resource Pool	527
8.3.3.4 Load Balancing	529
8.3.3.5 Installing and Configuring NFS	532

8.3.4 Upgrading a Service	534
8.3.5 Starting, Stopping, Deleting, or Restarting a Service	536
8.3.6 Viewing Service Events	537
8.4 Edge Resource Pool	540
8.4.1 Overview	540
8.4.2 Node	
8.4.3 Resource Pool	548
8.4.4 Enabling LTS	551
8.5 Inference Specifications	552
8.5.1 Model Package Specifications	553
8.5.1.1 Introduction to Model Package Specifications	553
8.5.1.2 Specifications for Editing a Model Configuration File	554
8.5.1.3 Specifications for Writing Model Inference Code	570
8.5.2 Examples of Custom Scripts	576
8.5.2.1 TensorFlow	576
8.6 ModelArts Monitoring on Cloud Eye	582
8.6.1 ModelArts Metrics	582
8.6.2 Setting Alarm Rules	584
8.6.3 Viewing Monitoring Metrics	586
9 Resource Management	588
9.1 Resource Pool	
9.2 Elastic Cluster	
9.2.1 Comprehensive Upgrades to ModelArts Resource Pool Management Functions	589
9.2.2 Creating a Resource Pool	591
9.2.3 Viewing Details About a Resource Pool	594
9.2.4 Resizing a Resource Pool	597
9.2.5 Migrating the Workspace	
9.2.6 Changing Job Types Supported by a Resource Pool	599
9.2.7 Upgrading a Resource Pool Driver	600
9.2.8 Deleting a Resource Pool	600
9.2.9 Abnormal Status of a Dedicated Resource Pool	601
9.2.10 ModelArts Network	606
9.3 Elastic Server	608
9.3.1 Overview	608
9.3.2 Preparations	609
9.3.3 Getting Started	611
9.3.4 Managing an Elastic Server	614
9.3.4.1 Creating an Elastic Server	615
9.3.4.2 Viewing Instance Details	616
9.3.4.3 Using SSH to Remotely Log In to an Instance	617
9.3.4.4 Starting or Stopping an Instance	619

9.3.4.6 Deleting an Instance	620
9.3.5 Configuring the Network as an Administrator	620
9.4 Monitoring Resources	621
9.4.1 Overview	621
9.4.2 Using Grafana to View AOM Monitoring Metrics	622
9.4.2.1 Procedure	
9.4.2.2 Installing and Configuring Grafana	622
9.4.2.2.1 Installing and Configuring Grafana on Windows	622
9.4.2.2.2 Installing and Configuring Grafana on Linux	623
9.4.2.2.3 Installing and Configuring Grafana on a Notebook Instance	626
9.4.2.3 Configuring a Grafana Data Source	630
9.4.2.4 Using Grafana to Configure Dashboards and View Metric Data	635
9.4.3 Viewing All ModelArts Monitoring Metrics on the AOM Console	642
10 AI Hub	679
10.1 AI Hub	
10.2 Registering with AI Hub	
10.3 Management Center	
10.4 Subscription & Use	
10.4.1 Searching for and Adding an Asset to Favorites	
10.4.2 Subscribing to an Algorithm	
10.4.3 Subscribing to a Model	
10.4.4 Downloading Datasets	
10.4.5 Subscribing to a Workflow	690
10.5 Publish & Share	
10.5.1 Publishing an Algorithm	
10.5.2 Publishing a Model	697
10.5.3 Publishing Data	
11 Custom Images	
11.1 Image Management	
11.2 Introduction to Preset Images (Mainstream Images)	
11.2.1 Preset Images	
11.2.2 Preset MindSpore Images on Arm	711
11.2.3 Preset TensorFlow Images on Arm	
11.2.4 Preset PyTorch Images on Arm	719
11.3 Using Custom Images in Notebook Instances	
11.3.1 Registering an Image in ModelArts	720
11.3.2 Creating a Custom Image	721
11.3.3 Saving a Notebook Instance as a Custom Image	721
11.3.3.1 Saving a Notebook Environment Image	721
11.3.3.2 Using a Custom Image to Create a Notebook Instance	722
11.3.4 Creating and Using a Custom Image in Notebook	722
11.3.4.1 Application Scenarios and Process	

11.3.4.2 Step 1 Creating a Custom Image	723
11.3.4.3 Step 2 Registering a New Image	725
11.3.4.4 Step 3 Using a New Image to Create a Development Environment	726
11.4 Using a Custom Image to Train Models (Model Training)	728
11.4.1 Overview	728
11.4.2 Example: Creating a Custom Image for Training	730
11.4.2.1 Example: Creating a Custom Image for Development and Training (MindSpore + Ascend)	730
11.4.2.1.1 Scenarios	731
11.4.2.1.2 Step 1 Creating an OBS Bucket and Folder	731
11.4.2.1.3 Step 2 Preparing Script Files and Uploading Them to OBS	732
11.4.2.1.4 Step 3 Creating a Custom Image	742
11.4.2.1.5 Step 4 Uploading the Image to SWR	745
11.4.2.1.6 Step 5 Creating and Debugging a Notebook Instance on ModelArts	747
11.4.2.1.7 Step 6 Creating a Training Job on ModelArts	747
11.4.3 Preparing a Training Image	748
11.4.3.1 Specifications for Custom Images for Training Jobs	748
11.4.3.2 Migrating an Image to ModelArts Training	749
11.4.3.3 Using a Base Image to Create a Training Image	750
11.4.4 Creating an Algorithm Using a Custom Image	751
11.4.5 Using a Custom Image to Create a CPU- or GPU-based Training Job	755
11.4.6 Using a Custom Image to Create an Ascend-based Training Job	760
11.4.7 Troubleshooting Process	762
11.5 Using a Custom Image to Create AI applications for Inference Deployment	763
11.5.1 Custom Image Specifications for Creating AI Applications	763
11.5.2 Creating a Custom Image and Using It to Create an AI Application	765
11.6 FAQs	769
11.6.1 How Can I Log In to SWR and Upload Images to It?	769
11.6.2 How Do I Configure Environment Variables for an Image?	771
11.6.3 How Do I Use Docker to Start an Image Saved Using a Notebook Instance?	771
11.6.4 How Do I Configure a Conda Source in a Notebook Development Environment?	772
11.6.5 What Are Supported Software Versions for a Custom Image?	773
12 Permissions Management	
12.1 Basic Concents	774
12.2 Permission Management Mechanisms	780
12.2.1 IAM	781
12.2.2 Agencies and Dependencies	789
12.2.3 Workspace	810
12.3. Configuration Practices in Typical Scenarios	810
12.3.1 Assigning Permissions to Individual Users for Using ModelArts	811
12.3.2 Separately Assigning Permissions to Administrators and Developers	814
12.3.3 Viewing the Notebook Instances of All IAM Users Linder One Tenant Account	821
12.3.4 Logging In to a Training Container Using Cloud Shell	822
	022

12.3.5 Prohibiting a User from Using a Public Resource Pool	824
12.4 FAQ	826
12.4.1 What Do I Do If a Message Indicating Insufficient Permissions Is Displayed When I Use M	IodelArts? 826
13 Best Practices	
13.1 Migrating a Locally Developed MindSpore Model to the Cloud for Training	
13.2 Creating an Al Application Using a Custom Engine	
13.3 Using a Large Model to Create an AI Application and Deploying a Real-Time Service	
13.4 Importing a Model from OBS to Create an AI Application and Deploying a Real-Time Service	ce 857
14 Full-Process Development of WebSocket Real-Time Services	860
15 FAQs	866
15.1 General Issues	866
15.1.1 What Is ModelArts?	
15.1.2 What Are the Relationships Between ModelArts and Other Services?	
15.1.3 What Are the Differences Between ModelArts and DLS?	
15.1.4 Which Ascend Chips Are Supported?	
15.1.5 How Do I Obtain an Access Key?	
15.1.6 How Do I Upload Data to OBS?	
15.1.7 What Do I Do If the System Displays a Message Indicating that the AK/SK Pair Is Unavail	able? 868
15.1.8 What Do I Do If a Message Indicating Insufficient Permissions Is Displayed When I Use N	IodelArts?
15.1.9 How Do I Use ModelArts to Train Models Based on Structured Data?	
15.1.10 How Do I View All Files Stored in OBS on ModelArts?	
15.1.11 Where Are Datasets of ModelArts Stored in a Container?	
15.1.12 Which AI Frameworks Does ModelArts Support?	
15.1.13 What Are the Functions of ModelArts Training and Inference?	
15.1.14 Can AI-assisted Identification of ModelArts Identify a Specific Label?	
15.1.15 Why Is the Job Still Queued When Resources Are Sufficient?	
15.2 Data Management (Old Version)	
15.2.1 Are There Size Limits for Images to be Uploaded?	
15.2.2 What Do I Do If Images in a Dataset Cannot Be Displayed?	
15.2.3 How Do I Integrate Multiple Object Detection Datasets into One Dataset?	
15.2.4 What Do I Do If Importing a Dataset Failed?	
15.2.5 Can a Table Dataset Be Labeled?	
15.2.6 What Do I Do to Import Locally Labeled Data to ModelArts?	
15.2.7 Why Does Data Fail to Be Imported Using the Manifest File?	
15.2.8 Where Are Labeling Results Stored?	
15.2.9 How Do I Download Labeling Results to a Local PC?	
15.2.10 Why Cannot Team Members Receive Emails for a Team Labeling Task?	
15.2.11 Can Two Accounts Concurrently Label One Dataset?	

15.2.12 Can I Delete an Annotator from a Labeling Team with a Labeling Task Assigned? What Is the Impact on the Labeling Result After Deletion? If the Annotator Cannot Be Deleted, Can I Separate the Annotator's Labeling Result?	. 882
15.2.13 How Do I Define a Hard Example in Data Labeling? Which Samples Are Identified as Hard Examples?	. 882
15.2.14 Can I Add Multiple Labeling Boxes to an Object Detection Dataset Image?	882
15.2.15 How Do I Merge Two Datasets?	883
15.2.16 Does Auto Labeling Support Polygons?	883
15.2.17 What Do the Options for Accepting a Team Labeling Task Mean?	883
15.2.18 Why Are Images Displayed in Different Angles Under the Same Account?	.883
15.2.19 Do I Need to Train Data Again If New Data Is Added After Auto Labeling Is Complete?	884
15.2.20 Why Does the System Display a Message Indicating My Label Fails to Save on ModelArts?	. 884
15.2.21 Can One Label By Identified Among Multiple Labels?	884
15.2.22 Why Are Newly Added Images Not Automatically Labeled After Data Amplification Is Enabled	? 885
15.2.23 Why Cannot Videos in a Video Dataset Be Displayed or Played?	.885
15.2.24 Why All the Labeled Samples Stored in an OBS Bucket Are Displayed as Unlabeled in ModelAr After the Data Source Is Synchronized?	rts .885
15.2.25 How Do I Use Soft-NMS to Reduce Bounding Box Overlapping?	885
15.2.26 Why ModelArts Image Labels Are Lost?	885
15.2.27 How Do I Add Images to a Validation or Training Dataset?	885
15.2.28 Can I Customize Labels for an Object Detection Dataset?	.886
15.2.29 What ModelArts Data Management Can Be Used for?	886
15.2.30 Will My Old-Version Datasets Be Cleared After the Old Version Is Discontinued? The existing datasets and the ones newly created in the old version will be retained after the old version is	
alscontinued	888
15.2.22 How Do L View the Size of a Dataset?	.000
15.2.22 How Do I View Labeling Details of a New Dataset?	000
15.2.33 How Do I Export Labeled Data?	880
15.2.35 Why Cannot L Find My Newly Created Dataset?	880
15.2.36 What Do I Do If the Database Quota Is Incorrect?	880
15.2.37 How Do I Split a Dataset?	890
15.2.38 How Do I Delete a Dataset Image?	890
15.2.39 Why Is There No Sample in the ModelArts Dataset Downloaded from AI Gallery and Then an OBS Bucket?	890
15.3 Notebook (New Version)	892
15.3.1 Constraints	892
15.3.1.1 Is sudo Privilege Escalation Supported?	. 892
15.3.1.2 Does ModelArts Support apt-get?	.892
15.3.1.3 Is the Keras Engine Supported?	.892
15.3.1.4 Does ModelArts Support the Caffe Engine?	893
15.3.1.5 Can I Install MoXing in a Local Environment?	. 893
15.3.1.6 Can Notebook Instances Be Remotely Logged In?	.893

15.3.2 Data Upload or Download	. 893
15.3.2.1 How Do I Upload a File from a Notebook Instance to OBS or Download a File from OBS to a Notebook Instance?	. 893
15.3.2.2 How Do I Upload Local Files to a Notebook Instance?	. 895
15.3.2.3 How Do I Import Large Files to a Notebook Instance?	. 895
15.3.2.4 Where Will the Data Be Uploaded to?	. 895
15.3.2.5 How Do I Download Files from a Notebook Instance to a Local Computer?	. 895
15.3.2.6 How Do I Copy Data from Development Environment Notebook A to Notebook B?	.896
15.3.2.7 What Can I Do If a File Fails to Be Uploaded to a Notebook Instance?	896
15.3.2.8 Failed to View the Local Mount Point of a Dynamically Mounted OBS Parallel File System in JupyterLab of a Notebook Instance	. 897
15.3.3 Data Storage	. 898
15.3.3.1 How Do I Rename an OBS File?	. 898
15.3.3.2 Do Files in /cache Still Exist After a Notebook Instance is Stopped or Restarted? How Do I Ava a Restart?	oid 898
15.3.3.3 How Do I Use the pandas Library to Process Data in OBS Buckets?	. 898
15.3.4 Environment Configurations	. 898
15.3.4.1 How Do I Check the CUDA Version Used by a Notebook Instance?	.898
15.3.4.2 How Do I Enable the Terminal Function in DevEnviron of ModelArts?	.899
15.3.4.3 How Do I Install External Libraries in a Notebook Instance?	. 899
15.3.4.4 How Do I Obtain the External IP Address of My Local PC?	. 900
15.3.4.5 How Can I Resolve Abnormal Font Display on a ModelArts Notebook Accessed from iOS?	. 900
15.3.4.6 Is There a Proxy for Notebook? How Do I Disable It?	902
15.3.5 Notebook Instances	. <mark>90</mark> 2
15.3.5.1 What Do I Do If I Cannot Access My Notebook Instance?	. 902
15.3.5.2 What Should I Do When the System Displays an Error Message Indicating that No Space Left After I Run the pip install Command?	.904
15.3.5.3 What Do I Do If "Read timed out" Is Displayed After I Run pip install?	. 904
15.3.5.4 What Do I Do If the Code Can Be Run But Cannot Be Saved, and the Error Message "save error Is Displayed?	or" . 905
15.3.5.5 When the SSH Tool Is Used to Connect to a Notebook Instance, Server Processes Are Cleared, the GPU Usage Is Still 100%	, but 905
15.3.6 Code Execution	. 905
15.3.6.1 What Do I Do If a Notebook Instance Won't Run My Code?	. 905
15.3.6.2 Why Does the Instance Break Down When dead kernel Is Displayed During Training Code Running?	. 906
15.3.6.3 What Do I Do If cudaCheckError Occurs During Training?	906
15.3.6.4 What Should I Do If DevEnviron Prompts Insufficient Space?	. 907
15.3.6.5 Why Does the Notebook Instance Break Down When opency.imshow Is Used?	. 907
15.3.6.6 Why Cannot the Path of a Text File Generated in Windows OS Be Found In a Notebook Instar	nce? . 907
15.3.6.7 What Do I Do If Files Fail to Be Saved in JupyterLab?	.908
15.3.7 Failures to Access the Development Environment Through VS Code	. 908
15.3.7.1 What Do I Do If the VS Code Window Is Not Displayed?	.908

15.3.7.2 What Do I Do If a Remote Connection Failed After VS Code Is Opened?	909
15.3.7.3 Basic Problems Causing the Failures to Access the Development Environment Through VS Co	de 912
15.3.7.4 What Do I Do If Error Message "Could not establish connection to xxx" is Displayed During a	· - · -
Remote Connection?	. 914
15.3.7.5 What Do I Do If the Connection to a Remote Development Environment Remains in "Setting	up
SSH Host xxx: Downloading VS Code Server locally" State for More Than 10 Minutes?	914
15.3.7.6 What Do I Do If the Connection to a Remote Development Environment Remains in the Stat "Setting up SSH Host xxx: Downloading VS Code Server locally" for More Than 10 Minutes?	e of 917
15.3.7.7 What Do I Do If the Connection to a Remote Development Environment Remains in the Stat	e of
"ModelArts Remote Connect: Connecting to instance xxx" for More Than 10 Minutes?	918
15.3.7.8 What Do I Do If a Remote Connection Is in the Retry State?	918
15.3.7.9 What Do I Do If Error Message "The VS Code Server failed to start" Is Displayed?	920
15.3.7.10 What Do I Do If Error Message "Permissions for 'x:/xxx.pem' are too open" Is Displayed?	921
15.3.7.11 What Do I Do If Error Message "Bad owner or permissions on C:\Users\Administrator/.ssh/ config" or "Connection permission denied (publickey)" Is Displayed?	922
15.3.7.12 What Do I Do If Error Message "ssh: connect to host xxx.pem port xxxxx: Connection refuse Displayed?	d" Is 924
15.3.7.13 What Do I Do If Error Message "ssh: connect to host ModelArts-xxx port xxx: Connection tir out" Is Displayed?	ned 924
15.3.7.14 What Do I Do If Error Message "Load key "C:/Users/xx/test1/xxx.pem": invalid format" Is Displayed?	925
15.3.7.15 What Do I Do If Error Message "An SSH installation couldn't be found" or "Could not estab	lish
15.2.7.16 What Do L Do If Error Mossago "no such identity" C://Jears/yy /test nom: No such file or	520
directory" Is Displayed?	928
15.3.7.17 What Do I Do If Error Message "Host key verification failed" or "Port forwarding is disabled Displayed?	" Is 929
15.3.7.18 What Do I Do If Error Message "Failed to install the VS Code Server" or "tar: Error is not recoverable: exiting now" Is Displayed?	931
15.3.7.19 What Do I Do If Error Message "XHR failed" Is Displayed When a Remote Notebook Instance Accessed Through VS Code?	ce Is 931
15.3.7.20 What Do I Do for an Automatically Disconnected VS Code Connection If No Operation Is	
Performed for a Long Time?	932
15.3.7.21 What Do I Do If It Takes a Long Time to Set Up a Remote Connection After VS Code Is	934
15.3.7.22 What Do I Do If Error Message "Connection reset" Is Displayed During an SSH Connection?	935
15.3.7.22 What Do I bo If a Notebook Instance is Frequently Disconnected or Stuck After Like	555
MobaXterm to Connect to the Notebook Instance in SSH Mode?	935
15.3.8 Others	937
15.3.8.1 How Do I Use Multiple Ascend Cards for Debugging in a Notebook Instance?	937
15.3.8.2 Why Is the Training Speed Similar When Different Notebook Flavors Are Used?	938
15.3.8.3 How Do I Perform Incremental Training When Using MoXing?	938
15.3.8.4 How Do I View GPU Usage on the Notebook?	940
15.3.8.5 How Can I Obtain GPU Usage Through Code?	942
15.3.8.6 Which Real-Time Performance Indicators of an Ascend Chin Can I View?	944
15.3.8.7 What Are the Relationships Between Files Stored in JupyterLab. Terminal. and OBS?	944

15.3.8.8 How Do I Migrate Data from an Old-Version Notebook Instance to a New-Version One?	.944
15.3.8.9 How Do I Use the Datasets Created on ModelArts in a Notebook Instance?	. 947
15.3.8.10 pip and Common Commands	.947
15.3.8.11 What Are Sizes of the /cache Directories for Different Notebook Specifications in DevEnviror	ו? 948
15.3.8.12 What Is the Impact of Resource Overcommitment on Notebook Instances?	. 948
15.4 Training Jobs	949
15.4.1 Functional Consulting	. 949
15.4.1.1 What Are the Solutions to Underfitting?	. 949
15.4.1.2 What Are the Precautions for Switching Training Jobs from the Old Version to the New Versio	n? . 949
15.4.1.3 How Do I Obtain a Trained ModelArts Model?	.951
15.4.1.4 What Is TensorBoard Used for in Model Visualization Jobs?	.951
15.4.1.5 How Do I Obtain RANK_TABLE_FILE on ModelArts for Distributed Training?	.951
15.4.1.6 How Do I Obtain the CUDA and cuDNN Versions of a Custom Image?	.952
15.4.1.7 How Do I Obtain a MoXing Installation File?	. 952
15.4.1.8 In a Multi-Node Training, the TensorFlow PS Node Functioning as a Server Will Be Continuou Suspended. How Does ModelArts Determine Whether the Training Is Complete? Which Node Is a	sly
Worker?	.952
15.4.1.9 How Do Finstall Moxing for a Custom image of a Training Job?	. 952
15.4.2 Reduing Data During Training	952
15.4.2.1 How Do I Compute the input and Output Data for fraining Models on ModelArts	.952
15.4.2.2 Now Do Finiprove fraining Efficiency is Low When a Large Number of Data Files Are Read During	. 900
Training?	954
15.4.2.4 How Do I Define Path Variables When Using MoXing?	. 955
15.4.3 Compiling the Training Code	. 955
15.4.3.1 How Do I Create a Training Job When a Dependency Package Is Referenced by the Model to I Trained?	Be . 955
15.4.3.2 What Is the Common File Path for Training Jobs?	. 956
15.4.3.3 How Do I Install a Library That C++ Depends on?	.956
15.4.3.4 How Do I Check Whether a Folder Copy Is Complete During Job Training?	. 957
15.4.3.5 How Do I Load Some Well Trained Parameters During Job Training?	. 957
15.4.3.6 How Do I Obtain Training Job Parameters from the Boot File of the Training Job?	. 958
15.4.3.7 Why Can't I Use os.system ('cd xxx') to Access the Corresponding Folder During Job Training?	. 958
15.4.3.8 How Do I Invoke a Shell Script in a Training Job to Execute the .sh File?	. 958
15.4.3.9 How Do I Obtain the Dependency File Path to be Used in Training Code?	. 959
15.4.3.10 What Is the File Path If a File in the model Directory Is Referenced in a Custom Python Package?	. 959
15.4.4 Creating a Training Job	. 959
15.4.4.1 What Can I Do If the Message "Object directory size/quantity exceeds the limit" Is Displayed When I Create a Training Job?	. 960
15.4.4.2 What Are Sizes of the /cache Directories for Different Resource Specifications in the Training Environment?	. 960
15.4.4.3 Is the /cache Directory of a Training Job Secure?	.961

15.4.4.4 Why Is a Training Job Always Queuing?	961
15.4.4.5 What Determines the Hyperparameter Directory (/work or /ma-user) When Creating a Trainin Job?	ng . 961
15.4.5 Managing Training Job Versions	.962
15.4.5.1 Does a Training Job Support Scheduled or Periodic Calling?	.962
15.4.6 Viewing Job Details	. 962
15.4.6.1 How Do I Check Resource Usage of a Training Job?	962
15.4.6.2 How Do I Access the Background of a Training Job?	962
15.4.6.3 Is There Any Conflict When Models of Two Training Jobs Are Saved in the Same Directory of a Container?	э . 963
15.4.6.4 Only Three Valid Digits Are Retained in a Training Output Log. Can the Value of <b>loss</b> Be Changed?	. 963
15.4.6.5 Can a Trained Model Be Downloaded or Migrated to Another Account? How Do I Obtain the Download Path?	.963
15.5 Service Deployment	963
15.5.1 Model Management	. 963
15.5.1.1 Importing Models	.964
15.5.1.1.1 How Do I Import the .h5 Model of Keras to ModelArts?	964
15.5.1.1.2 How Do I Edit the Installation Package Dependency Parameters in a Model Configuration Fi When Importing a Model?	ile . 964
15.5.1.1.3 What Do I Do If Error ModelArts.0107 Is Reported When I Use MindSpore to Create an AI Application?	. 966
15.5.1.1.4 How Do I Change the Default Port to Create a Real-Time Service Using a Custom Image?	.966
15.5.1.1.5 Does ModelArts Support Multi-Model Import?	967
15.5.1.1.6 Restrictions on the Size of an Image for Importing an AI Application	. 967
15.5.2 Service Deployment	967
15.5.2.1 Functional Consulting	. 967
15.5.2.1.1 What Types of Services Can Models Be Deployed as on ModelArts?	.967
15.5.2.1.2 What Are the Differences Between Real-Time Services and Batch Services?	968
15.5.2.1.3 What Is the Maximum Size of a Prediction Request Body?	.968
15.5.2.1.4 How Do I Select Compute Node Specifications for Deploying a Service?	. 968
15.5.2.1.5 What Is the CUDA Version for Deploying a Service on GPUs?	. 969
15.5.2.2 Real-Time Services	. 969
15.5.2.2.1 What Do I Do If a Conflict Occurs in the Python Dependency Package of a Custom Prediction Script When I Deploy a Real-Time Service?	on 969
15.5.2.2.2 What Is the Format of a Real-Time Service API?	969
15.5.2.2.3 Why Did My Service Deployment Fail with Proper Deployment Timeout Configured?	.970
15.6 API/SDK	970
15.6.1 Can ModelArts APIs or SDKs Be Used to Download Models to a Local PC?	.970
15.6.2 What Installation Environments Do ModelArts SDKs Support?	.970
15.6.3 Does ModelArts Use the OBS API to Access OBS Files over an Intranet or the Internet?	.971
15.6.4 How Do I Obtain a Job Resource Usage Curve After I Submit a Training Job by Calling an API?	.971
15.6.5 How Do I View the Old-Version Dedicated Resource Pool List Using the SDK?	. 971
15.7 Using PyCharm Toolkit	.971

15.7.1 What Should I Do If an Error Occurs During Toolkit Installation?	971
15.7.2 What Should I Do If an Error Occurs When I Edit a Credential in PyCharm Toolkit?	972
15.7.3 Why Cannot I Start Training?	974
15.7.4 What Should I Do If Error "xxx isn't existed in train_version" Occurs When a Training Job Is	074
15.7.5 What Should I Do If Error "Invalid OBS path" Occurs When a Training Job Is Submitted?	974
15.7.6 What Should I Do If Error "NoSuchKey" Occurs When PyCharm Toolkit Is Used to Submit a	
Training Job?	975
15.7.7 What Should I Do If an Error Occurs During Service Deployment?	976
15.7.8 How Do I View Error Logs of PyCharm Toolkit?	977
15.7.9 How Do I Use PyCharm ToolKit to Create Multiple Jobs for Simultaneous Training?	977
15.7.10 What Should I Do If "Error occurs when accessing to OBS" Is Displayed When PyCharm Too Used?	lKit Is 977
16 Troubleshooting	978
16.1 General Issues	978
16.1.1 Incorrect OBS Path on ModelArts	978
16.2 ExeML	980
16.2.1 Preparing Data	980
16.2.1.1 Failed to Publish a Dataset Version	980
16.2.1.2 Invalid Dataset Version	983
16.2.2 Training a Model	983
16.2.2.1 Failed to Create an ExeML-powered Training Job	983
16.2.2.2 ExeML-powered Training Job Failed	983
16.2.3 Deploying a Model	987
16.2.3.1 Failed to Submit the Real-time Service Deployment Task	987
16.2.3.2 Failed to Deploy a Real-time Service	987
16.2.4 Publishing a Model	988
16.2.4.1 Failed to Submit the Model Publishing Task	988
16.2.4.2 Failed to Publish a Model	988
16.3 DevEnviron	989
16.3.1 Environment Configuration Faults	989
16.3.1.1 Disk Space Used Up	990
16.3.1.2 An Error Is Reported When Conda Is Used to Install Keras 2.3.1 in Notebook	992
16.3.1.3 Error "HTTP error 404 while getting xxx" Is Reported During Dependency Installation in a Notebook	993
16.3.1.4 The numba Library Has Been Installed in a Notebook Instance and Error "import numba ModuleNotFoundError: No module named 'numba'" Is Reported	993
16.3.2 Instance Faults	994
16.3.2.1 Failed to Create a Notebook Instance and JupyterProcessKilled Is Displayed in Events	994
16.3.2.2 What Do I Do If I Cannot Access My Notebook Instance?	995
16.3.2.3 What Should I Do When the System Displays an Error Message Indicating that No Space Le	eft 997
16.3.2.4 What Do I Do If the Code Can Be Run But Cannot Be Saved, and the Error Message "save e	error"
Is Displayed?	997

16.3.2.5 ModelArts.6333 Error Occurs	997
16.3.2.6 What Can I Do If a Message Is Displayed Indicating that the Token Does Not Exist or Is Lost	
When I Open a Notebook Instance?	998
16.3.3 Code Running Failures	. 998
16.3.3.1 Error Occurs When Using a Notebook Instance to Run Code, Indicating That No File Is Found in /tmp	 998
16.3.3.2 What Do I Do If a Notebook Instance Won't Run My Code?	999
16.3.3.3 Why Does the Instance Break Down When dead kernel Is Displayed During Training Code	
Running?	999
16.3.3.4 What Do I Do If cudaCheckError Occurs During Training?	1000
16.3.3.5 What Do I Do If Insufficient Space Is Displayed in DevEnviron?	1000
16.3.3.6 Why Does the Notebook Instance Break Down When opency.imshow Is Used?	1000
16.3.3.7 Why Cannot the Path of a Text File Generated in Windows OS Be Found In a Notebook Insta	nce?
	1001
16.3.3.8 What Do I Do If No Kernel Is Displayed After a Notebook File Is Created?	1001
16.3.4 JupyterLab Plug-in Faults	1002
16.3.4.1 What Do I Do If the Git Plug-in Password Is Invalid?	1002
16.3.5 Save an Image Failures	1003
16.3.5.1 What If the Error Message "there are processes in 'D' status, please check process status usin	g'ps
-aux' and kill all the 'D' status processes" or "Buildimge,False,Error response from daemon,Cannot pa	use
16.2.5.2 What Do I Do If Error "container size %dC is greater than threshold %dC" is Displayed When	1004
Save an Image?	1004
16.3.5.3 What Do I Do If Error "too many layers in your image" Is Displayed When I Save an Image?.	1005
16.3.5.4 What Do I Do If Error "The container size (xG) is greater than the threshold (25G)" Is Report	ed
When I Save an Image?	1005
16.3.6 Other Faults	1006
16.3.6.1 Failed to Open the checkpoints Folder in Notebook	1006
16.3.6.2 Failed to Use a Purchased Dedicated Resource Pool to Create New-Version Notebook Instance	es
	1007
16.3.6.3 Error Message "Permission denied" Is Displayed When the tensorboard Command Is Used to Open a Log File in a Notebook Instance	1008
16.4 Training Jobs	1000
16.4.1 ORS Operation Issues	1005
16.4.1 Error in Eile Peading	1009
16.4.1.2 Error Mossage Is Displayed Percentedly When a Tensor Elevy 1.8 Job Is Connected to OPS	1009
16.4.1.2 Error Message is Displayed Repeatedly When a Tensor tow-1.8 Job is Connected to Obs	1010
16.4.1.5 Tensor row stops writing rensor board to OBS when the size of written bata Reaches 5 GB.	1010
16.4.1.4 Error Mossage "ProkenDineError: Proken pipe" Displayed When OPS Data Is Conjed	1011
16.4.1.6 Error Message "VolueError Invalid endpoint, obsiver com" Displayed in Logs	1011
16.4.1.6 Error Message ValueError: Invalid endpoint: obs.xxxx.com Displayed II Logs	1013
16.4.2. In Cloud Migratian Adaptation Issues	1013
10.4.2 In-cloud Migration Adaptation Issues	1014
16.4.2.1 Failed to Import a Module	1014
16.4.2.2 Error Message "No module named .*" Displayed in Training Job Logs	1015
16.4.2.3 Failed to Install a Third-Party Package	1017

16.4.2.4 Failed to Download the Code Directory	1018
16.4.2.5 Error Message "No such file or directory" Displayed in Training Job Logs	.1018
16.4.2.6 Failed to Find the .so File During Training	1020
16.4.2.7 ModelArts Training Job Failed to Parse Parameters and an Error Is Displayed in the Log	.1021
16.4.2.8 Training Output Path Is Used by Another Job	1022
16.4.2.9 Error Message "RuntimeError: std::exception" Displayed for a PyTorch 1.0 Engine	1022
16.4.2.10 Error Message "retCode=0x91, [the model stream execute failed]" Displayed in MindSpore	Logs 1023
16.4.2.11 Error Occurred When Pandas Reads Data from an OBS File If MoXing Is Used to Adapt to a OBS Path	n 1023
16.4.2.12 Error Message "Please upgrade numpy to >= xxx to use this pandas version" Displayed in L	ogs 1024
16.4.2.13 Reinstalled CUDA Version Does Not Match the One in the Target Image	1024
16.4.2.14 Error ModelArts.2763 Occurred During Training Job Creation	1025
16.4.2.15 Error Message "AttributeError: module '***' has no attribute '***'' Displayed Training Job Log	js1025
16.4.2.16 System Container Exits Unexpectedly	1026
16.4.3 Hard Faults Due to Space Limit	1027
16.4.3.1 Downloading Files Timed Out or No Space Left for Reading Data	1027
16.4.3.2 Insufficient Container Space for Copying Data	1028
16.4.3.3 Error Message "No space left" Displayed When a TensorFlow Multi-node Job Downloads Date (cache	ta 1029
16.4.3.4 Size of the Log File Has Reached the Limit	1029
16.4.3.5 Error Message "write line error" Displayed in Logs	1030
16.4.3.6 Error Message "No space left on device" Displayed in Logs	1031
16.4.3.7 Training Job Failed Due to OOM	1032
16.4.3.8 Common Issues Related to Insufficient Disk Space and Solutions	1034
16.4.4 Internet Access Issues	1035
16.4.4.1 Error Message "Network is unreachable" Displayed in Logs	1035
16.4.4.2 URL Connection Timed Out in a Running Training Job	1036
16.4.5 Permission Issues	1036
16.4.5.1 What Should I Do If Error "stat:403 reason:Forbidden" Is Displayed in Logs When a Training Accesses OBS	Job 1036
16.4.5.2 Error Message "Permission denied" Displayed in Logs	1037
16.4.6 GPU Issues	1039
16.4.6.1 Error Message "No CUDA-capable device is detected" Displayed in Logs	1039
16.4.6.2 Error Message "RuntimeError: connect() timed out" Displayed in Logs	.1040
16.4.6.3 Error Message "cuda runtime error (10) : invalid device ordinal at xxx" Displayed in Logs	.1041
16.4.6.4 Error Message "RuntimeError: Cannot re-initialize CUDA in forked subprocess" Displayed in I	_ogs
16.4.6.5 No GPU Is Found for a Training Job	1042
16.4.7 Service Code Issues	1043
16.4.7.1 Error Message "pandas.errors.ParserError: Error tokenizing data. C error: Expected .* fields" Displayed in Logs	1043

16.4.7.2 Error Message "max_pool2d_with_indices_out_cuda_frame failed with error code 0" Display Logs	yed in 1043
16.4.7.3 Training Job Failed with Error Code 139	1044
16.4.7.4 Debugging Training Code in the Cloud Environment If a Training Job Failed	1045
16.4.7.5 Error Message "'(slice(0, 13184, None), slice(None, None, None))' is an invalid key" Display Logs	yed in 1045
16.4.7.6 Error Message "DataFrame.dtypes for data must be int, float or bool" Displayed in Logs	1045
16.4.7.7 Error Message "CUDNN_STATUS_NOT_SUPPORTED" Displayed in Logs	1046
16.4.7.8 Error Message "Out of bounds nanosecond timestamp" Displayed in Logs	1046
16.4.7.9 Error Message "Unexpected keyword argument passed to optimizer" Displayed in Logs	1047
16.4.7.10 Error Message "no socket interface found" Displayed in Logs	1047
16.4.7.11 Error Message "Runtimeerror: Dataloader worker (pid 46212) is killed by signal: Killed BP Displayed in Logs	" 1048
16.4.7.12 Error Message "AttributeError: 'NoneType' object has no attribute 'dtype'" Displayed in Lo	gs1048
16.4.7.13 Error Message "No module name 'unidecode'" Displayed in Logs	1049
16.4.7.14 Distributed Tensorflow Cannot Use tf.variable	1049
16.4.7.15 When MXNet Creates kvstore, the Program Is Blocked and No Error Is Reported	1050
16.4.7.16 ECC Error Occurs in the Log, Causing Training Job Failure	1051
16.4.7.17 Training Job Failed Because the Maximum Recursion Depth Is Exceeded	1051
16.4.7.18 Training Using a Built-in Algorithm Failed Due to a <b>bndbox</b> Error	1051
16.4.7.19 Training Job Status Is Reviewing Job Initialization	1052
16.4.7.20 Training Job Process Exits Unexpectedly	1052
16.4.7.21 Stopped Training Job Process	1053
16.4.8 Training Job Suspended	1054
16.4.8.1 Data Replication Suspension	1054
16.4.8.2 Suspension Before Training	1054
16.4.8.3 Suspension During Training	1055
16.4.8.4 Suspension in the Last Training Epoch	1056
16.4.9 Running a Training Job Failed	1057
16.4.9.1 Troubleshooting a Training Job Failure	1057
16.4.9.2 An NCCL Error Occurs When a Training Job Fails to Be Executed	1058
16.4.9.3 A Training Job Created Using a Custom Image Is Always in the Running State	1059
16.4.9.4 Running a Job Failed Due to Persistently Rising Memory Usage	1059
16.4.10 Training Jobs Created in a Dedicated Resource Pool	1060
16.4.10.1 No Cloud Storage Name or Mount Path Displayed on the Page for Creating a Training Job	o 1060
16.4.10.2 Storage Volume Failed to Be Mounted to the Pod During Training Job Creation	1061
16.4.11 Training Performance Issues	1062
16.4.11.1 Training Performance Deteriorated	1062
16.5 Inference Deployment	1063
16.5.1 AI Application Management	1063
16.5.1.1 Creating an AI Application Failed	1063
16.5.1.2 Failed to Build an Image or Import a File When an IAM user Creates an AI Application	1065

16.5.1.3 Obtaining the Directory Structure in the Target Image When Importing an AI Application Through OBS	1066
16.5.1.4 Failed to Obtain Certain Logs on the ModelArts Log Query Page	1067
16.5.1.5 Failed to Download a pip Package When an AI Application Is Created Using OBS	1067
16.5.1.6 Failed to Use a Custom Image to Create an AI application	1068
16.5.1.7 Insufficient Disk Space Is Displayed When a Service Is Deployed After an AI Application Is	
Imported	1069
16.5.1.8 Error Occurred When a Created AI Application Is Deployed as a Service	1070
16.5.1.9 Invalid Runtime Dependency Configured in an Imported Custom Image	1070
16.5.1.10 Garbled Characters Displayed in an Al Application Name Returned When Al Application Deb Are Obtained Through an API	tails 1071
16.5.1.11 The Model or Image Exceeded the Size Limit for AI Application Import	1071
16.5.1.12 A Single Model File Exceeded the Size Limit (5 GB) for AI Application Import	1072
16.5.1.13 Creating an AI Application Failed Due to Image Building Timeout	1072
16.5.2 Service Deployment	1073
16.5.2.1 Error Occurred When a Custom Image Model Is Deployed as a Real-Time Service	1073
16.5.2.2 Alarm Status of a Deployed Real-Time Service	1073
16.5.2.3 Failed to Start a Service	1074
16.5.2.4 What Do I Do If an Image Fails to Be Pulled When a Service Is Deployed, Started, Upgraded, Modified?	or 1076
16.5.2.5 What Do I Do If an Image Restarts Repeatedly When a Service Is Deployed, Started, Upgrade or Modified?	ed, 1077
16.5.2.6 What Do I Do If a Container Health Check Fails When a Service Is Deployed, Started, Upgrad or Modified?	ed, 1077
16.5.2.7 What Do I Do If Resources Are Insufficient When a Service Is Deployed, Started, Upgraded, o Modified?	r 1077
16.5.2.8 Error Occurred When a CV2 Model Package Is Used to Deploy a Real-Time Service	1078
16.5.2.9 Service Is Consistently Being Deployed	1079
16.5.2.10 A Started Service Is Intermittently in the Alarm State	1079
16.5.2.11 Failed to Deploy a Service and Error "No Module named XXX" Occurred	1080
16.5.2.12 Insufficient Permission to or Unavailable Input/Output OBS Path of a Batch Service	1080
16.5.2.13 What Can I Do if the Memory Is Insufficient?	1081
16.5.3 Service Prediction	1082
16.5.3.1 Service Prediction Failed	1082
16.5.3.2 Error "APIG.XXXX" Occurred in a Prediction Failure	1083
16.5.3.3 Error ModelArts.4206 Occurred in Real-Time Service Prediction	1085
16.5.3.4 Error ModelArts.4302 Occurred in Real-Time Service Prediction	1085
16.5.3.5 Error ModelArts.4503 Occurred in Real-Time Service Prediction	1085
16.5.3.6 Error MR.0105 Occurred in Real-Time Service Prediction	1087
16.5.3.7 Method Not Allowed	1088
16.5.3.8 Request Timed Out	1089
16.5.3.9 Error Occurred When an API Is Called for Deploying a Model Created Using a Custom Image	1089
16.6 MoXing	1089
16.6.1 Error Occurs When MoXing Is Used to Copy Data	1090

16.6.2 How Do I Disable the Warmup Function of the Mox?	091
16.6.3 Pytorch Mox Logs Are Repeatedly Generated10	091
16.6.4 Does moxing.tensorflow Contain the Entire TensorFlow? How Do I Perform Local Fine Tune on t Generated Checkpoint?	:he 092
16.6.5 Copying Data Using MoXing Is Slow and the Log Is Repeatedly Printed in a Training Job10	093
16.6.6 Failed to Access a Folder Using MoXing and Read the Folder Size Using get_size	094
16.7 APIs or SDKs	094
16.7.1 "ERROR: Could not install packages due to an OSError" Occurred During ModelArts SDK Installation	094
16.7.2 Error Occurred During Service Deployment After the Target Path to a File Downloaded Through ModelArts SDK Is Set to a File Name	a 095
16.7.3 A Training Job Created Using an API Is Abnormal	095
16.8 Change History10	096
17 Change History10	)98

# Service Overview

## **1.1 Infographics**

## 1.1.1 What Is ModelArts



## 1.2 What Is ModelArts?

ModelArts is a one-stop AI development platform geared toward developers and data scientists of all skill levels. It enables you to rapidly build, train, and deploy models anywhere, and manage full-lifecycle AI workflows. ModelArts accelerates AI development and fosters AI innovation with key capabilities, including data preprocessing and auto labeling, distributed training, automated model building, and one-click workflow execution.

ModelArts covers all stages of AI development, including data processing, algorithm development, and model training and deployment. The underlying technologies of ModelArts support various heterogeneous computing resources, allowing developers to flexibly select and use resources. In addition, ModelArts supports popular open-source AI development frameworks such as TensorFlow, PyTorch, and MindSpore. ModelArts also allows you to use customized algorithm frameworks tailored to your needs.

ModelArts aims to simplify AI development.

#### **Product Architecture**

ModelArts supports the entire development process, including data processing, and model training, management, and deployment. It also provides AI Gallery for sharing models.

ModelArts supports various AI application scenarios, such as image classification, object detection, video analysis, speech recognition, product recommendation, and exception detection.

#### Data optimization Model update Model Data Data processing Model training Deployment management Al apps Data labeling Version -Real-time services Model repository Online coding Batch services Common Al frameworks Built-in algorithms Precision tracking /ersion management Data import and export Dataset publishing Distributed clusters Model visualization ExeMI

#### Figure 1-1 ModelArts architecture

#### **Product Advantages**

• One-stop platform

The out-of-the-box and full-lifecycle AI development platform provides onestop data processing, and development, training, management, and deployment of models.

- Easy to use
  - Automatic optimization of hyperparameters
  - Code-free development and simplified operations

#### • High performance

- The self-developed MoXing deep learning framework accelerates algorithm development and training.
  - Models running on Ascend AI chips achieve more efficient inference.
- Flexible
  - Mainstream open-source frameworks such as TensorFlow, PyTorch, and MindSpore
  - Ascend chips
  - Exclusive use of dedicated resources
  - Custom images for custom frameworks and operators

## **1.3 Functions**

AI engineers face challenges in the installation and configuration of various AI tools, data preparation, and model training. To address these challenges, the onestop AI development platform ModelArts is provided. The platform integrates data preparation, algorithm development, model training, and model deployment into the production environment, allowing AI engineers to perform one-stop AI development.



#### Figure 1-2 Function overview

ModelArts has the following features:

#### • Data governance

Manages data preparation, such as data filtering and labeling, and dataset versions.

#### • Rapid and simplified model training

Enables high-performance distributed training and simplifies coding with the self-developed MoXing deep learning framework.

#### • Multi-scenario deployment

Deploys models in various production environments, and supports real-time and batch inference.

#### • Auto learning

Enables model building without coding and supports image classification, object detection, and predictive analytics.

## 1.4 Basic Knowledge

## **1.4.1 Introduction to the AI Development Lifecycle**

#### What Is Al

Artificial intelligence (AI) is a technology capable of simulating human cognition through machines. The core capability of AI is to make a judgment or prediction based on a given input.

#### What Is the Purpose of AI Development

Al development aims to centrally process and extract information from volumes of data to summarize internal patterns of the study objects.

Massive volumes of collected data are computed, analyzed, summarized, and organized by using appropriate statistics, machine learning, and deep learning methods to maximize data value.

#### **Basic Process of AI Development**

The basic process of AI development includes the following steps: determining an objective, preparing data, and training, evaluating, and deploying a model.

#### Figure 1-3 AI development process



#### Step 1 Determine an objective.

Before starting AI development, determine what to analyze. What problems do you want to solve? What is the business goal? Sort out the AI development framework and ideas based on the business understanding. For example, image classification and object detection. Different projects have different requirements for data and AI development methods.

#### Step 2 Prepare data.

Data preparation refers to data collection and preprocessing.

Data preparation is the basis of AI development. When you collect and integrate related data based on the determined objective, the most important thing is to ensure the authenticity and reliability of the obtained data. Typically, you cannot collect all the data at the same time. In the data labeling phase, you may find that some data sources are missing and then you may need to repeatedly adjust and optimize the data.

#### Step 3 Train a model.

Modeling involves analyzing the prepared data to find the causality, internal relationships, and regular patterns, thereby providing references for commercial decision making. After model training, usually one or more machine learning or

deep learning models are generated. These models can be applied to new data to obtain predictions and evaluation results.

#### Step 4 Evaluate the model.

A model generated by training needs to be evaluated. Typically, you cannot obtain a satisfactory model after the first evaluation, and may need to repeatedly adjust algorithm parameters and data to further optimize the model.

Some common metrics, such as the accuracy, recall, and area under the curve (AUC), help you effectively evaluate and obtain a satisfactory model.

#### Step 5 Deploy the model.

Model development and training are based on existing data (which may be test data). After a satisfactory model is obtained, the model needs to be formally applied to actual data or newly generated data for prediction, evaluation, and visualization. The findings can then be reported to decision makers in an intuitive way, helping them develop the right business strategies.

----End

#### **1.4.2 Basic Concepts of AI Development**

Machine learning is classified into supervised, unsupervised, and reinforcement learning.

- Supervised learning uses labeled samples to adjust the parameters of classifiers to achieve the required performance. It can be considered as learning with a teacher. Common supervised learning includes regression and classification.
- Unsupervised learning is used to find hidden structures in unlabeled data. Clustering is a form of unsupervised learning.
- Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

#### Regression

Regression reflects the time feature of data attributes and generates a function that maps one data attribute to an actual variable prediction to find the dependency between the variable and attribute. Regression mainly analyzes data and predicts data and data relationship. Regression can be used for customer development, retention, customer churn prevention, production lifecycle analysis, sales trend prediction, and targeted promotion.



#### Classification

Classification involves defining a set of categories based on the common features of objects and identifying which category an object belongs to. Classification can be used for customer classification, customer properties, feature analysis, customer satisfaction analysis, and customer purchase trend prediction.



#### Clustering

Clustering involves grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. Clustering can be used for customer segmentation, customer characteristic analysis, customer purchase trend prediction, and market segmentation.



Clustering analyzes data objects and produces class labels. Objects are grouped based on the maximized and minimized similarities to form clusters. In this way, objects in the same cluster are more similar to each other than to those in other clusters.

## 1.4.3 Common Concepts of ModelArts

#### **ExeML**

ExeML is the process of automating model design, parameter tuning, and model training, model compression, and model deployment with the labeled data. The process is code-free and does not require developers to have experience in model development. A model can be built in three steps: labeling data, training a model, and deploying the model.

#### Inference

Inference is the process of deriving a new judgment from a known judgment according to a certain strategy. In AI, machines simulate human intelligence, and complete inference based on neural networks.

#### **Real-Time Inference**

Real-time inference specifies a web service that provides an inference result for each inference request.

#### **Batch Inference**

Batch inference specifies a batch job that processes batch data for inference.

#### **Ascend Chip**

The Ascend chips are a series of AI chips with high computing performance and low power consumption.

#### **Resource Pool**

ModelArts provides large-scale computing clusters for model development, training, and deployment. There are two types of resource pools: public resource pool and dedicated resource pool. The public resource pool is provided by default. Dedicated resource pools are created separately and used exclusively.

### **1.4.4 Introduction to Development Tools**

#### **NOTE**

This document describes the DevEnviron notebook functions of the new version.

Software development is a process of reducing developer costs and improving development experience. In AI development, ModelArts is dedicated to improving AI development experience and simplifying the development process. ModelArts DevEnviron uses cloud native resources and integrates the development tool chain to provide better in-cloud AI development experience for AI development, exploration, and teaching.

ModelArts notebook for seamless in-cloud and on-premises collaboration

- In-cloud JupyterLab, local IDE, and ModelArts plug-ins for remote development and debugging, tailored to your needs
- In-cloud development environment with AI compute resources, cloud storage, and built-in AI engines
- Custom runtime environment saved as an image for training and inference

# Feature 1: Remote development, allowing remote access to notebook from a local IDE

The notebook of the new version provides remote development. After enabling remote SSH, you can remotely access the ModelArts notebook development environment to debug and run code from a local IDE.

Due to limited local resources, developers using a local IDE run and debug code typically on a CPU or GPU server shared between team members. Building and maintaining the CPU or GPU server are costly.

ModelArts notebook instances are out of the box with various built-in engines and flavors for you to select. You can use a dedicated container environment. Only after simple configurations, you can remotely access the environment to run and debug code from your local IDE.



Figure 1-4 Remotely accessing notebook from a local IDE

Uploaded through web or OBS Browser+

ModelArts notebook can be regarded as an extension of a local development environment. The operations such as data reading, training, and file saving are the same as those performed in a local environment.

ModelArts notebook allows you to use in-cloud resources while with local coding habits unchanged.

A local IDE supports Visual Studio (VS) Code, PyCharm, and SSH. In addition, the PyCharm Toolkit plug-ins allow you to easily use cloud resources.

# Feature 2: Preset images that are out-of-the-box with optimized configurations and supporting mainstream AI engines

The AI engines and versions preset in each image are fixed. When creating a notebook instance, specify an AI engine and version, including the chip type.

ModelArts DevEnviron provides a group of preset images, including PyTorch, TensorFlow, and MindSpore images. You can use a preset image to start your notebook instance. After the development in the instance, submit a training job without any adaptation.

The image versions preset in ModelArts are determined based on user feedback and version stability. If your development can be carried out using the versions preset in ModelArts, for example, MindSpore 1.5, use preset images. These images have been fully verified and have many commonly-used installation packages built in. They are out-of-the-box, relieving you from configuring the environment.

The images preset in ModelArts DevEnviron include:

 Common preset packages: common AI engines such as PyTorch and MindSpore based on standard Conda, common data analysis software packages such as Pandas and Numpy, and common tool software such as CUDA and CUDNN, meeting common AI development requirements. • Preset Conda environments: A Conda environment and basic Conda Python (excluding any AI engine) are created for each preset image. The following figure shows the Conda environment for the preset MindSpore.



Select a Conda environment based on whether the AI engine is used for debugging.

- Notebook: a web application that enables you to code on the GUI and combine the code, mathematical equations, and visualized content into a document.
- JupyterLab plug-ins: enable flavor changing and instance stopping to improving user experience.
- Remote SSH: allows you to remotely debug a notebook instance from a local PC.

#### D NOTE

- To simplify operations, ModelArts notebook of the new version does not support switchover between AI engines in a notebook instance.
- Al engines vary based on regions. For details about the Al engines available in a region, see the Al engines displayed on the management console.

# Feature 3: JupyterLab, an online interactive development and debugging tool

ModelArts integrates open-source JupyterLab for online interactive development and debugging. You can use the notebook on the ModelArts management console to compile and debug code and train models based on the code, without concerning environment installation or configuration.

JupyterLab is an interactive development environment. It is the next-generation product of Jupyter Notebook. JupyterLab enables you to compile notebooks, operate terminals, edit Markdown text, enable interaction, and view CSV files and images.

## **1.5 AI frameworks supported by ModelArts**

The AI frameworks and versions supported by ModelArts vary slightly based on the development environment notebook, training jobs, and model inference (AI application management and deployment). The following describes the AI frameworks supported by each module.

#### **Unified Image List**

ModelArts provides unified images of Arm+Ascend specifications, including MindSpore and PyTorch. You can use the images to develop environment, train models, and deploy services. For details, see **Unified Image List**.

Preset Image	Supported Processor	Applicable Scope
mindspore_2.2.0-cann_7.0.1-py_3.9- euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment
mindspore_2.1.0-cann_6.3.2-py_3.7- euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment
mindspore_2.2.10-cann_8.0.rc1- py_3.9-hce_2.0.2312-aarch64-snt9c	Ascend snt9c	Notebook, training, and inference deployment

#### Table 1-2 PyTorch

Preset Image	Supported Processor	Applicable Scope
pytorch_1.11.0-cann_6.3.2-py_3.7- euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment
pytorch_2.1.0-cann_8.0.rc1-py_3.9- hce_2.0.2312-aarch64-snt9c	Ascend snt9c	Notebook, training, and inference deployment
pytorch_1.11.0-cann_8.0.rc1-py_3.9- hce_2.0.2312-aarch64-snt9c	Ascend snt9c	Notebook, training, and inference deployment

#### **Development Environment Notebook**

The image and versions supported by development environment notebook instances vary based on runtime environments.
Image	Description	Suppor ted Chip	Remot e SSH	Online Jupyter Lab
pytorch_1.11.0-cann_7.0.1- py_3.9-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes
pytorch_2.1.0-cann_7.0.1- py_3.9-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes
mindspore_2.2.0-cann_7.0.1- py_3.9-euler_2.10.7-aarch64- snt9b	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
mindspore_2.1.0-cann_6.3.2- py_3.7-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine MindSpore	Ascend	Yes	Yes
pytorch_1.11.0-cann_6.3.2- py_3.7-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes

Table 1-3 Images supported by notebook of the new version

Image	Description	Suppor ted Chip	Remot e SSH	Online Jupyter Lab
mindspore1.7.0-cann5.1.0- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
mindstudio5.0.rc1-ascend- cann5.1.rc1-euler2.8.3- aarch64	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	No
mindspore1.8.0-cann5.1.2- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
tensorflow1.15-cann5.1.0- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. TensorFlow is preset in the Al engine.	Ascend	Yes	Yes
mindspore_2.0.0-cann_6.3.0- py_3.7-euler_2.8.3	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine MindSpore	Ascend	Yes	Yes
pytorch_1.11.0-cann_6.3.0- py_3.7-euler_2.8.3	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes

Image	Description	Suppor ted Chip	Remot e SSH	Online Jupyter Lab
tensorflow1.15- mindspore1.7.0-cann5.1.0- euler2.8-aarch64	Ascend+Arm algorithm development and training. TensorFlow and MindSpore are preset in the Al engine.	Ascend	Yes	Yes
tensorflow_1.15.0- cann_6.3.0-py_3.7- euler_2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
tensorflow1.15.0-cann5.1.2- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes

#### Table 1-4 Images supported by notebook of the old version

Runtime Environment	Built-in AI Engine and Version	Supported Chip
Ascend-Powered-Engine 1.0 (Python3)	MindSpore 1.2.0	Ascend
	MindSpore 1.1.1	Ascend
	TensorFlow 1.15.0	Ascend

# **Training Jobs**

The following table lists the AI engines.

The built-in training engines are named in the following format: <Training engine name\_version>-[cpu | <cuda\_version | cann\_version >]-<py\_version>-<OS name\_version>-< x86\_64 | aarch64>

Runtime Environmen t	System Archite cture	System Version	AI Engine and Version	Supported CUDA or Ascend Version
Ascend- Powered- Engine	aarch6 4	Euler2.8	mindspore_2.0.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0
PyTorch	aarch6 4	Euler2.8	pytorch_1.11.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0
TensorFlow	aarch6 4	Euler2.8	tensorflow_1.15.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0

Table 1-5 AI engines supported by training jobs

#### **NOTE**

Supported AI engines vary depending on regions.

#### Supported AI Engines for ModelArts Inference

If you import a model from a template or OBS to create an AI application, the following AI engines and versions are supported.

#### **NOTE**

- Runtime environments marked with **recommended** are unified runtime images, which will be used as mainstream base inference images.
- Images of the old version will be discontinued. Use unified images.
- The base images to be removed are no longer maintained.
- Naming a unified runtime image: <*AI engine name and version*> <*Hardware and version*: CPU, CUDA, or CANN> <*Python version*> <*OS version*> <*CPU architecture*>

Engine	Runtime
TensorFlow	tensorflow_1.15.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64
MindSpore	mindspore_2.0.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64
PyTorch	pytorch_1.11.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64

#### Table 1-6 Supported AI engines and their runtime

# **1.6 Related Services**

IAM	
	ModelArts uses Identity and Access Management (IAM) for authentication and authorization. For more information about IAM, see <i>Identity and Access Management User Guide</i> .
OBS	
	ModelArts uses Object Storage Service (OBS) to securely and reliably store data and models at low costs. For more details, see <i>Object Storage Service Console Operation Guide</i> .
EVS	
	ModelArts uses Elastic Volume Service (EVS) to store created notebook instances.
CCE	
	ModelArts uses Cloud Container Engine (CCE) to deploy models as real-time services. CCE enables high concurrency and provides elastic scaling. For more information about CCE, see <i>Cloud Container Engine User Guide</i> .
SWR	
	To use an AI framework that is not supported by ModelArts, use Software Repository for Container (SWR) to customize an image and import the image to ModelArts for training or inference. For details about SWR, see .
Cloud Eye	
	ModelArts uses Cloud Eye to monitor online services and model loads in real time and send alarms and notifications automatically. For details about Cloud Eye, see <i>Cloud Eye User Guide</i> .
стѕ	
	ModelArts uses Cloud Trace Service (CTS) to record operations for later query, audit, and backtrack operations. For details about CTS, see <i>Cloud Trace Service User Guide</i> .

# 1.7 How Do I Access ModelArts?

You can access ModelArts through the web-based management console or by using HTTPS-based application programming interfaces (APIs).

#### • Using the Management Console

ModelArts features a simple and easy-to-use management console, and provides a host of functions including ExeML, data management,

development environment, model training, AI application management, AI Hub, and service deployment. You can complete end-to-end AI development on the management console.

#### • Using APIs

If you want to integrate ModelArts into a third-party system for secondary development, use APIs to access ModelArts. For details about the APIs and operations, see *ModelArts API Reference*.

# **2** Preparations

# 2.1 Configuring Access Authorization (Global Configuration)

#### Scenarios

Exposed ModelArts functions are controlled through IAM permissions. For example, if you as an IAM user need to create a training job on ModelArts, you must have the **modelarts:trainJob:create** permission.

ModelArts must access other services for AI computing. For example, ModelArts must access OBS to read your data for training. For security purposes, ModelArts must be authorized to access other cloud services. This is agency authorization.

ModelArts provides one-click auto authorization. You can quickly configure agency authorization on the **Global Configuration** page of ModelArts. Then, ModelArts will automatically create an agency for you and configure it in ModelArts.

In this mode, the authorization scope is specified based on the preset system policies of dependent services to ensure sufficient permissions for using services. The created agency has almost all permissions of dependent services. If you want to precisely control the scope of permissions granted to an agency, use custom authorization. For more about permissions management, see **Permissions Management**.

This section introduces one-click auto authorization. This mode allows you to grant permissions to IAM users, federated users (virtual IAM users), agencies, and all users with one click.

#### **Adding Authorization**

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **Settings**. The **Global Configuration** page is displayed.
- 2. Click **Add Authorization**. On the **Add Authorization** page that is displayed, configure the parameters.

#### Table 2-1 Parameters

Parameter	Description
Authorized User	Options: IAM user, Federated user, Agency, and All users
	• <b>IAM user</b> : You can use a tenant account to create IAM users and assign permissions for specific resources. Each IAM user has their own identity credentials (password and access keys) and uses cloud resources based on assigned permissions.
	• Federated user: A federated user is also called a virtual enterprise user.
	Agency: You can create agencies in IAM.
	• All users: If you select this option, the agency permissions will be granted to all IAM users under the current account, including those created in the future. For individual users, choose All users.

Parameter	Description			
Authorized To	<ul> <li>This parameter is not displayed when Authorized User is set to All users.</li> <li>IAM user: Select an IAM user and configure an agency for the IAM user.</li> </ul>			
	Figure 2-1 Selecting an IAM user			
	Authorized User IAM user Federated user Agency All users			
	Authorized To			
	• Federated user: Enter the username or user ID of the target federated user.			
	Figure 2-2 Selecting a federated user			
	Authorized User IAM user Federated user Agency All users			
	Authorized To			
	<ul> <li>Agency: Select an agency name. You can create an agency under account A and grant the agency permissions to account B. When using account B, you can switch the role in the upper right corner of the console to account A and use the agency permissions of account A.</li> <li>Figure 2-3 Switch Role         English         Security Settings         My Credentials     </li> </ul>			
	Enterprise Management Switch Role Tag Management Log Out			
Agency	<ul> <li>Use existing: If there are agencies in the list, select an available one to authorize the selected user. Click the drop-down arrow next to an agency name to view its permission details.</li> <li>Add agency: If there is no available agency, create one. If you use ModelArts for the first time, select Add agency.</li> </ul>			

Parameter	Description
Add agency > Agency Name	The system automatically creates a changeable agency name.
Add agency > Authorization Method	<ul> <li>Role-based: A coarse-grained IAM authorization strategy to assign permissions based on user responsibilities. Only a limited number of service-level roles are available. When using roles to grant permissions, assign other roles on which the permissions depend to take effect. Roles are not ideal for fine-grained authorization and secure access control.</li> <li>Policy-based: A fine-grained authorization tool that defines permissions for operations on specific cloud resources under certain conditions. This type of authorization is more flexible and ideal for a secure access.</li> </ul>
Add agency > Permissions > Common User	<b>Common User</b> provides the permissions to use all basic ModelArts functions. For example, you can access data, and create and manage training jobs. Select this option generally. Click <b>View permissions</b> to view common user permissions.
Add agency > Permissions > Custom	If you need refined permissions management, select <b>Custom</b> to flexibly assign permissions to the created agency. You can select permissions from the permission list as required.

3. Click Create.

# **Viewing Authorized Permissions**

You can view the configured authorizations on the **Global Configuration** page. Click **View Permissions** in the **Authorization Content** column to view the permission details.

#### Figure 2-4 View Permissions

Authorized To 👙	Authorized User $\mbox{$\ddagger$}$	Authorization Type 💠	Authorization Content $\Rightarrow$	Creation Time 💠	Operation
	All users	Agency	modelarts_i	Jan 19, 2023 16:53:29 GMT+08:00	View Permissions Delete

#### Figure 2-5 Common user permissions

View Permissio	ns		
	Name	Туре	Description
	DLI FullAccess	System-defined policy	Full permissions for Data Lake Insight.
	VPC Administrator	System-defined role	VPC Administrator
	EPS FullAccess	System-defined policy	All operations on the Enterprise Project Management service.
	CTS Administrator	System-defined role	CTS Administrator
	ModelArts CommonOperations	System-defined policy	Common permissions of ModelArts service, except create, update, del
	SFS ReadOnlyAccess	System-defined policy	The read-only permissions to all SFS resources.
	OBS Administrator	System-defined policy	Object Storage Service Administrator
	DWS Administrator	System-defined role	Data Warehouse Service Administrator
	LTS FullAccess	System-defined policy	All permissions of Log Tank service.
	CES ReadOnlyAccess	System-defined policy	Read-only permissions for Cloud Eye.
	10 🔻 Total Records: 12 < 1	2 >	

#### Changing the Authorization Scope

1. To change the authorization scope, click **Modify permissions in IAM** in the **View Permissions** dialog box.

Figure 2-6 Modify permissions in IAM

Add Authorization Clear Authorization Enable strict authorization			
₽ Search or filter by keyword.			
Mary Remainstern	×	1 Time 💠	Operation
View Permissions		2023 16:12:04 GMT+08:00	View Permissions Delete
Authorized To		2023 10:48:34 GMT+08:00	View Permissions Delete
Agency Name		2023 17:13:38 GMT+08:00	View Permissions Delete
Agency Permission 26 permissions Modify permissions in IAM		2023 10:09:23 GMT+08:00	View Permissions   Delete

2. Modify the agency information. Select your required validity period.

Figure 2-7 Agency information

Age	ncies / modelarts_ag	jency-common	
	Basic Information	Permissions	
	Agency Name	modelarts_agency-common	
	* Agency Type	Cloud service	
	* Cloud Service	ModelArts •	
	★ Validity Period	Unlimited	
	Description	Created by ModelArts service.	
•			
		OK	

3. On the **Agencies** page, click **Authorize**, select policies or rules, and click **Next**. Select the scope for minimum authorization and click **OK**.

When setting the minimum authorization scope, you can select either **Global services** or **All resources**. If you select **All resources**, the selected permissions will be applied to all resources.

#### **Deleting Authorization**

To better manage your authorization, you can delete the authorization of an IAM user or delete the authorizations of all users in batches.

• Deleting the authorization of a user

On the **Global Configuration** page, locate the target user. Click **Delete** in the **Operation** column of the target user. Enter **DELETE** and click **OK**. After the deletion takes effect, the user cannot use ModelArts functions.

#### • Deleting authorizations in batches

On the **Global Configuration** page, click **Clear Authorization** above the authorization list. Enter **DELETE** and click **OK**. After the deletion, the account and all IAM users under the account cannot use ModelArts functions.

#### FAQs

- How do I configure authorization when I use ModelArts for the first time? On the Add Authorization page, set Agency to Add agency and select Common User, which provides the permissions to use all basic ModelArts functions. For example, you can access data, and create and manage training jobs. Select this option generally.
- Where is the entrance for authorization using an access key? Access key authorization on the global configuration page has been discontinued. If you used an access key for authorization before, switch to agency authorization. To do so, click Clear Authorization on the Global Configuration page and use an agency for authorization.
- 3. How do I obtain an access key (AK/SK)?

If you use AK/SK authentication to use certain functions, such as accessing real-time services and logging in using PyCharm Toolkit or VS Code, obtain an access key. For details, see **How Do I Obtain an Access Key?**.

4. How do I delete an existing agency from the agency list?

Agency	Use existing	Add agency	
	Ageno	cy Name	Validity Pe
	🔿 🗸 mode	larts_agency	days

Go to the IAM console, click **Agencies** in the navigation pane, and delete the target agency.

5. Why is a message indicating insufficient permission displayed when I access a page on the ModelArts management console?

Possible causes are insufficient user permissions or changes in module capabilities. To fix this issue, follow the prompts to update the authorization.

# 2.2 Creating an OBS Bucket

ModelArts uses OBS to store data and model backups and snapshots, achieving secure, reliable, and low-cost storage. Before using ModelArts, create an OBS bucket and folders for storing data.

#### OBS

OBS provides stable, secure, and efficient cloud storage service that lets you store virtually any volume of unstructured data in any format. Bucket and objects are basic concepts in OBS. A bucket is a container for storing objects in OBS. Each bucket is specific to a region and has specific storage class and access permissions. A bucket is accessible through its domain name over the Internet. An object is the basic unit of data storage in OBS.

ModelArts cannot store data and uses OBS as its data storage center. All the input data, output data, and cache data during AI development can be stored in OBS buckets for reading.

Before using ModelArts, create an OBS bucket and folders for storing data.

#### Procedure

 Log in to OBS Console and click Create Bucket in the upper right corner of the page to create an OBS bucket. For example, create an OBS bucket named c-flowers.

#### **NOTE**

The created OBS bucket and ModelArts are in the same region.

Do not enable **Default Encryption**. ModelArts cannot read the data from encrypted OBS buckets.

- 2. On the **Buckets** page, click the bucket name to view its details.
- 3. Click **Objects** in the navigation pane on the left. On the **Objects** page, click **Create Folder** to create an OBS folder. For example, create a folder named **flowers** in the created **c-flowers** OBS bucket. For details, see Creating a Folder.

#### Figure 2-8 Create Folder

< c-flower12 🗇				
Overview	Objects 🗇			
Objects				
Metrics NEW	Objects Deleted Objects	Fragments		
Permissions 🔹	Objects are basic units of data storage.	In OBS, files and folders are treated as o	bjects. Any file type can be uploaded and	managed in a bucket. Learn more
Basic Configurations 🔹	You can use OBS Browser+ to move an Upload Object Create Folde	object to any other folder in this bucket.		
Domain Name Mgmt				
Cross-Region	Name JΞ	Storage Class J≡	Size J≘	Encrypted J≡
Replication				
Back to Source				

# 3 Exeml

# 3.1 Introduction to ExeML

#### **ExeML Functions**

ModelArts ExeML is a customized code-free model development tool that helps you start codeless AI application development with high flexibility. ExeML automates model design, parameter tuning and training, and model compression and deployment based on the labeled data. With ExeML, you only need to upload data and perform simple operations as prompted on the ExeML GUI to train and deploy models.

You can use ExeML to quickly build models for image classification, and object detection. ExeML is widely used in industrial, retail, and security sectors.

- Image classification: identifies a class of objects in images.
- Object detection: identifies the position and class of each object in an image.

#### **ExeML Process**

With ModelArts ExeML, you can develop AI models without coding. You only need to upload data, create a project, label the data, train a model, and deploy the trained model. For details, see **Figure 3-1**. In the new-version ExeML, this process can be finished by a workflow. You can develop a DAG through a workflow. DAG execution is to use a task execution template to perform data labeling, dataset publishing, model training, model registration, and service deployment in sequence.



Figure 3-1 ExeML process

#### **ExeML Projects**

#### • Image Classification

An image classification project aims to classify images. You only need to add images and label them. Then, an image classification model can be quickly generated for automatically classifying offerings, vehicle types, and defective goods. For example, in the quality check scenario, you can upload a product image, label the image as qualified or unqualified, and train and deploy a model to inspect product quality.

#### • Object Detection

An object detection project aims to identify the class and location of objects in images. You only need to add images and label objects in the images with proper bounding boxes. The labeled images will be used as a training set for building a model to identify multiple objects or provide the number of objects in a single image. Object detection can also be used to inspect employees' dress code and perform unattended inspection of article placement.

# 3.2 Image Classification

# 3.2.1 Preparing Data

Before using ModelArts ExeML to build a model, upload data to an OBS bucket.

#### **Requirements on Datasets**

- Check that all images are undamaged and in a compatible format. The supported formats are JPG, JPEG, BMP, and PNG.
- Do not store data of different projects in the same dataset.
- Collect at least two classes of images with a similar number of images in each class. Make sure each class has a minimum of 20 images.
- To ensure the prediction accuracy of models, the training samples must be similar to the real-world use cases.
- To ensure the generalization capability of models, datasets should cover all possible scenarios.

#### Uploading Data to OBS

In this section, the OBS console is used to upload data.

#### Upload files to OBS according to the following specifications:

- The name of files cannot contain plus signs (+), spaces, or tabs.
- If you do not need to upload training data in advance, create an empty folder to store files generated in the future, for example, **/bucketName/data-cat**.
- If you need to upload images to be labeled in advance, create an empty folder and save the images in the folder. An example of the image directory structure is /bucketName/data-cat/cat.jpg.
- If you want to upload labeled images to the OBS bucket, upload them according to the following specifications:

 The dataset for image classification requires storing labeled objects and their label files (in one-to-one relationship with the labeled objects) in the same directory. For example, if the name of the labeled object is **10.jpg**, the name of the label file must be **10.txt**.

Example of data files:

ataset-import-path>
10.jpg
10.txt
11.jpg
11.txt
12.jpg
12.txt

- Only images in JPG, JPEG, PNG, and BMP formats are supported. When uploading images on the OBS console, ensure that the size of an image does not exceed 5 MB and the total size of images to be uploaded in one attempt does not exceed 8 MB. If the data volume is large, use OBS Browser+ to upload images.
- A label name can contain a maximum of 32 characters, including letters, digits, hyphens (-), and underscores (\_).
- The specifications of image classification label files (.txt) are as follows:
   Each row contains only one label.

```
flower
book
```

#### Procedure for uploading data to OBS:

Perform the following operations to upload data to OBS for model training and building.

- 1. Log in to OBS Console and create a bucket.
- 2. Upload the local data to the OBS bucket. If you have a large amount of data, use OBS Browser+ to upload data or folders. The uploaded data must meet the dataset requirements of the ExeML project.

#### D NOTE

Upload data from unencrypted buckets. Otherwise, training will fail because data cannot be decrypted.

#### **Creating a Dataset**

After the data preparation is completed, create a dataset of the type supported by the project. For details, see .

### 3.2.2 Creating a Project

ModelArts ExeML supports image classification and object detection projects. You can create any of them based on your needs. Perform the following operations to create an ExeML project.

#### Procedure

- 1. Log in to the ModelArts console. In the navigation pane, choose **ExeML**.
- 2. Click **Create Project** in the box of your desired project. The page for creating an ExeML project is displayed.

#### Figure 3-2 Creating a project

ExeML New Version	ew			
			((( ( ( ))))	
Image Classification	Object Detection	Predictive Analytics	Sound Classification New!	Text Classification New!
Identify classes of objects in Images.	Identify the position and class of each object in images.	Classify or predict structured data.	Identify classes of sounds in audio files.	Identify classes of text in text files.
Create Project	Create Project	Create Project	Create Project	Create Project

#### 3. On the project creation page, set parameters by referring to **Table 3-1**.

★ Billing Mode	Pay-per-use	
* Name	ExeMe_2	
Description		
	0/500	
* Datasets	Select a dataset.   C  C	ireate Dataset
★ Output Path	Se	lect
★ Training Flavor	Select a flavor.	

#### Table 3-1 Parameters

Parameter	Description
Name	Name of an ExeML project
	• Enter a maximum of 64 characters. Only digits, letters, underscores (_), and hyphens (-) are allowed. This parameter is mandatory.
	Start with a letter.
	The name must be unique.
Description	Brief description of a project
Dataset	You can select a dataset or click <b>Create Dataset</b> to create one.
	• Existing dataset: Select a dataset from the drop-down list box. Only datasets of the same type are displayed.
	• Creating a dataset: Click <b>Create Dataset</b> to create a dataset. For details, see .

Parameter	Description
Output Path	Select an OBS path for storing ExeML data. NOTE The output path stores all data generated in the ExeML project.
Training Flavor	Select a training flavor for this ExeML project. You will be billed based on different flavors.

- 4. Click Create Project. Then, the ExeML workflow is displayed.
- 5. Wait until the workflow of the image classification project executes the following phases in sequence:
  - a. Label Data: Check data labeling.
  - b. Publish Dataset Version: Publish a version for the labeled dataset.
  - c. Check Data: Check whether any exception occurs in your dataset.
  - d. **Classify Images**: Train the dataset of the published version to generate a model.
  - e. **Register Model**: Register the trained model with model management.
  - f. **Deploy Service**: Deploy the generated model as a real-time service.

#### **Quickly Searching for a Project**

On the ExeML overview page, you can use the search box to quickly search for and filter workflows based on the ExeML type (or project name).

- 1. Log in to the ModelArts console. In the navigation pane, choose **ExeML**.
- 2. In the search box above the ExeML project list, filter the desired workflows based on the required property, such as name, status, project type, current phase, and tag.

Figure 3-3 Property

Q Select a property or enter a keyword.			
Nam	Property	*	
	Name		
	Status		
	Project Type		
10	Phase		
	Tag		
	Executions		
	Created At	Ŧ	

×

3. To adjust the basic settings of ExeML and select the columns you want to see,

click 🙆 on the right of the search box.

**Table Text Wrapping**: This function is disabled by default. If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.

**Operation Column**: This function is enabled by default. If you enable this function, the **Operation** column is always fixed at the rightmost position of the table.

**Custom Columns**: By default, all items are selected. You can select columns you want to see.

#### Figure 3-4 Customizing table columns

Se	tti	n	g	s	

Basic settings		Custom Columns
Table Text Wrapping	Auto wrapping If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.	Search Q Name (default) Status
Operation Column	Fixed position If you enable this function, the Operation column is always fixed at the rightmost position of the table.	<ul> <li>Project Type</li> <li>Phase</li> <li>Tag</li> <li>Executions</li> <li>Created At</li> <li>Description</li> <li>Operation (default)</li> </ul>
	OK Cancel	

- 4. Click **OK**. Then, the columns will be displayed based on the settings.
- 5. To arrange ExeML projects by a specific property, click in the table header.

# 3.2.3 Labeling Data

Model training requires a large number of labeled images. Therefore, before model training, add labels to the images that are not labeled. ModelArts allows you to add labels in batches by one click. You can also modify or delete labels that have been added to images.

#### **NOTE**

The number of labeled images in the dataset must be no fewer than 100. Otherwise, checking the dataset will fail, affecting your model training.

After the project is created, you will be directed to the ExeML page and the project starts to run. Click the data labeling phase. After the status changes to **Awaiting operation**, confirm the data labeling status in the dataset. You can also modify labels, add data, or delete data in the dataset.

#### Figure 3-5 Data labeling status

Configurations	
🔒 Go to Instanc	e Details to label data.
Attribute	
Status	Awaiting operation
Started At	Jan 18, 2024 16:55:47 GMT+08:00
Duration	00:00:15
Updated At	Jan 18, 2024 16:56:02 GMT+08:00

#### Labeling Images

1. On the labeling phase of the new-version ExeML, click **Instance Details**. The data labeling page is displayed.

Figure 3-6 Clicking Instance Details

Configurations					
A Go to Instance Details to label data.					

2. Select the images to be labeled in sequence, or tick **Select Images on Current Page** to select all images on the page, and then add labels to the images in the right pane.

Figure 3-7 Labeling an image

Unlabeled 12 Labeled 0 I	o Be Confirmed 0 All statuses 12						Cabeling Description
Auto Labeling 🔻 🛛 Add data 🔻	Synchronize New Data	fard Examples 💌	₽ Filter ∨ Selected: 1 D	eselect 🗌 Select Images on Current Page	Ū Delete	Add label	
						Selected Images	Selected 1 image
						Label	yunbao
						OK Ca	ncel

- 3. After selecting an image, input a label in the **Label** text box, or select an existing label from the drop-down list. Click **OK**. The selected image is labeled. For example, you can select multiple images containing tulips and add label **tulips** to them. Then select other unlabeled images and label them as **sunflowers** and **roses**. After the labeling is complete, the images are saved on the **Labeled** tab page.
  - a. You can add multiple labels to an image.

- b. A label consists of letters, digits, hyphens (-), and underscores (\_).
- 4. After all the images are labeled, view them on the **Labeled** tab page or view **All Labels** in the right pane to check the name and quantity of the labels.

#### Synchronizing or Adding Images

On the labeling phase, click **Instance Details** to go to the data labeling page. Then, add images from your local PC or synchronize image data from OBS.

#### Figure 3-8 Adding local images

_	Unlabeled 40	Labeled 26	To Be Confirmed 0 A	ll statuses 66
	Auto Labeling 🔻	Add data 🔻	Synchronize New Data	Batch Process Hard
	Add	data		
	View	historical records		

#### Figure 3-9 Synchronizing OBS images

Auto Labeling 🔻	ļ	Add data 🔻	🖪 Synchronize New Data	Batch Process Hard Examples 💌

- Add data: You can click Add data to quickly add images on a local PC to ModelArts. These images will be automatically synchronized to the OBS path specified during project creation.
- **Synchronize New Data**: You can upload images to the OBS directory specified during project creation and click **Synchronize New Data** to quickly add the images in the OBS directory to ModelArts.
- **Delete Image**: You can delete images one by one, or tick **Select Current Page** to delete all images on the page.

#### **NOTE**

The deleted images cannot be recovered. Exercise caution when performing this operation.

#### Modifying Labeled Data

After labeling data, you can modify the labeled data on the **Labeled** tab page.

• Modifying based on images

On the data labeling page, click the **Labeled** tab, and select one or more images to be modified from the image list. Modify the image information in the label information area on the right.

- Adding a label: In the **Label** text box, select an existing label or enter a new label name, and then click
- Modifying a label: In the Labels of Selected Images area, click the editing icon in the Operation column, enter the correct label name in the text box, and click the check mark icon to complete the modification.

- Deleting a label: In the Labels of Selected Images area, click  $\overline{U}$  in the Operation column to delete the label.

#### • Modifying based on labels

On the labeling overview page, click **Label Management**. Information about all labels is displayed.

- Modifying a label: In the **Operation** column of the target label, click **Modify**, enter the new label, and click **OK**.
- Deleting a label: In the **Operation** column of the target label, click **Delete**, and click **OK**.

D NOTE

Deleted tags cannot be restored.

#### **Resuming Workflow Execution**

After confirming data labeling, return back to the new-version ExeML. Click **Next**. Then, the workflow continues to run in sequence until all phases are executed.

Figure 3-10 Resuming the workflow execution

Configurations	Next	Instance Details
A Go to instance Details to label data.		

# 3.2.4 Training a Model

After labeling the images, perform model training to obtain the required image classification model. Ensure that the labeled images meet the requirements specified in **Prerequisites**. Otherwise, checking the dataset will fail.

#### Prerequisites

- 1. The number of labeled images in your dataset is greater than or equal to 100.
- 2. At least two classes of samples are required for training, and each class with at least 5 samples.

#### Procedure

1. Ensure all your dataset has been labeled. For details, see Labeling Data.

Figure 3-11 Finding unlabeled data

exeML-f481 < Back to ExeML	
Labeled 27 Unlabeled 0	_
↔ Add Ū Delete	💽 Synchronize Data Source

- 2. In the data labeling phase of the new-version ExeML, click **Next** and wait until the workflow enters the training phase.
- 3. Wait until the training is complete. No manual operation is required. If you close or exit the page, the system continues training until it is complete.
- 4. On the image classification phase, wait until the training status changes from **Running** to **Completed**.
- 5. After the training, click on the image classification phase to view metric information. For details about the evaluation result parameters, see Table 3-2.

Parameter	Descriptio n	Description
Recall	Recall	Fraction of correctly predicted samples over all samples predicted as a class. It shows the ability of a model to distinguish positive samples.
Precision	Precision	Fraction of correctly predicted samples over all samples predicted as a class. It shows the ability of a model to distinguish negative samples.
Accuracy	Accuracy	Fraction of correctly predicted samples over all samples. It shows the general ability of a model to recognize samples.
F1 Score	F1 score	Harmonic average of the precision and recall of a model. It is used to evaluate the quality of a model. A high F1 score indicates a good model.

Table 3-2 Evaluation result parameters

#### D NOTE

An ExeML project supports multiple rounds of training, and each round generates an Al application version. For example, the first training version is **0.0.1**, and the next version is **0.0.2**. The trained models can be managed by training version. After the trained model meets your requirements, deploy the model as a service.

# 3.2.5 Deploying a Model as a Service

#### **Deploying a Service**

You can deploy a model as a real-time service that provides a real-time test UI and monitoring capabilities. After model training is complete, you can deploy a version with the ideal accuracy and in the **Successful** status as a service. The procedure is as follows:

- 1. On the phase execution page, after the service deployment status changes to **Awaiting input**, double-click **Deploy Service**. On the configuration details page, configure resource parameters.
- 2. On the service deployment page, select the resource specifications used for service deployment.

Configurations			Next
Attribute		Parameters	
Status	<ul> <li>Awaiting Input</li> </ul>	* model_name	ExeML_5d0e
Started At	Dec 07, 2023 09:32:06 GMT+08:00		
Duration	00:00:05		
Updated At	Dec 07, 2023 09:32:12 GMT+08:00	* Auto Stop (?)	
Input			
* Al Applicatio	n Source My Al application Workflow step		
AI Applicatio	n and Version ExeML_5d0e(Synchronou + 0.0.2(Nor + C		
* Resource Po	Public Resource Pool Dedicated Resource Pool		
Specification	s CPU: 2 core 8GB 🔹		
	Application scenario: Standard CPU specifications, meeting the running and prediction requirements of most models		
Price	¥ 0.80 /hour		
Traffic Ratio (%	- 100 +		
Compute Node:	- 1 +		
Environment Va	riable OAdd Environment Variable		

#### Figure 3-12 Resource specifications

- AI Application Source: defaults to the generated AI application.
- AI Application and Version: The current AI application version is automatically selected, which is changeable.
- **Resource Pool**: defaults to public resource pools.
- **Traffic Ratio**: defaults to **100** and supports a value range of 0 to 100.
- Specifications: Select available specifications based on the list displayed on the console. The specifications in gray cannot be used in the current environment. If there are no specifications after you select a public resource pool, no public resource pool is available in the current environment. In this case, use a dedicated resource pool or contact the administrator to create a public resource pool.

- **Compute Nodes**: an integer ranging from 1 to 5. The default value is **1**.
- Auto Stop: enables a service to automatically stop at a specified time. If this function is not enabled, the real-time service continuously runs. Auto stop is enabled by default and its default value is 1 hour later.

The auto stop options are **1 hour later**, **2 hours later**, **4 hours later**, **6 hours later**, and **Custom**. If you select **Custom**, enter any integer from 1 to 24 in the text box on the right.

#### D NOTE

You can choose the package that you have bought when you select specifications. On the configuration fee tag, you can view your remaining package quota and how much you will pay for any extra usage.

3. After configuring resources, click **Next**. Wait until the status changes to **Executed**. The AI application has been deployed as a real-time service.

#### **Testing a Service**

• After the service is deployed, click **Instance Details** to go to the real-time service details page. Click the **Prediction** tab to test the service.

Figure 3-13 Testing the service

< Back to Real-Time Service List

Basic Information	
Name	workflow_created_service_8f575a34-9919-462c-a0a9-a22c68d5b203
Status	Running(59 minutes until stop)
Source	My Deployment
Synchronize Data	Synchronize Data
Data Collection	
Traffic Limit	200
WebSocket	Disabled
Usage Guides	Prediction Configuration Updates Filter Monitoring Events Logs Authorizations

- You can also choose Service Deployment > Real-Time Services and click Predict in the Operation column of the target service for testing. The testing procedure is the same as that described in the following section. For details, see Testing a Service.
- You can also use code to test a service. For details, see "Accessing Real-Time Services".
- The following describes the procedure for performing a service test after the image classification model is deployed as a service on the ExeML page.
  - a. After the model is deployed, click **Instance Details** in the service deployment phase to go to the service page. On the **Prediction** tab page, click **Upload** and select a local image for test.

b. Click **Prediction** to conduct the test. After the prediction is complete, label **sunflowers** and its detection score are displayed in the prediction result area on the right. If the model accuracy does not meet your expectation, add images on the **Label Data** tab page, label the images, and train and deploy the model again. **Table 3-3** describes the parameters in the prediction result. If you are satisfied with the model prediction result, call the API to access the real-time service as prompted. For details, see "Accessing Real-Time Services".

Only JPG, JPEG, BMP, and PNG images are supported.

#### Figure 3-14 Prediction result



1	1
2	"predicted_label": "sunflowers",
3	"scores": [
- 4	0
5	"sunflowers",
6	"0.972"
7	],
8	Ę
9	"dandelion",
10	"0.028"
11	],
12	1
13	"daisy".
14	"0.000"
15	1.
16	E
17	"roses"
18	"0.000"
19	1,
20	r -
21	"tulips"
22	"0.000"
23	1
24	
25	1

Table 3-3 Parameters in the prediction result

Parameter	Description		
predicted_label	Image prediction label		
scores	Prediction confidence of top 5 labels		

#### D NOTE

• A running real-time service continuously consumes resources. If you do not need to use the real-time service, stop the service to stop billing. To do so, click **Stop** in the **More** drop-down list in the **Operation** column. If you want to use the service again, click **Start**.

# 3.3 Object Detection

### 3.3.1 Preparing Data

Before using ModelArts ExeML to build a model, upload data to an OBS bucket.

#### **Uploading Data to OBS**

This operation uses the OBS console to upload data.

Perform the following operations to import data to the dataset for model training and building.

- 1. Log in to OBS Console and create a bucket.
- 2. Upload the local data to the OBS bucket. If you have a large amount of data, use OBS Browser+ to upload data or folders. The uploaded data must meet the dataset requirements of the ExeML project.

#### **NOTE**

Upload data from unencrypted buckets. Otherwise, training will fail because data cannot be decrypted.

#### **Requirements on Datasets**

- The name of files in a dataset cannot contain Chinese characters, plus signs (+), spaces, or tabs.
- Ensure that no damaged image exists. The supported image formats include JPG, JPEG, BMP, and PNG.
- Do not store data of different projects in the same dataset.
- To ensure the prediction accuracy of models, the training samples must be similar to the actual application scenarios.
- To ensure the generalization capability of models, datasets should cover all possible scenarios.
- In an object detection dataset, if the coordinates of the bounding box exceed the boundaries of an image, the image cannot be identified as a labeled image.

#### **Requirements for Files Uploaded to OBS**

- If you do not need to upload training data in advance, create an empty folder to store files generated in the future, for example, **/bucketName/data-cat**.
- If you need to upload images to be labeled in advance, create an empty folder and save the images in the folder. An example of the image directory structure is /bucketName/data-cat/cat.jpg.
- If you want to upload labeled images to the OBS bucket, upload them according to the following specifications:
  - The dataset for object detection requires storing labeled objects and their label files (in one-to-one relationship with the labeled objects) in the same directory. For example, if the name of the labeled object is IMG\_20180919\_114745.jpg, the name of the label file must be IMG\_20180919\_114745.xml.

The label files for object detection must be in PASCAL VOC format. For details about the format, see **Table 3-4**.

Example of data files:

atuset import patit
IMG_20180919_114732.jpg
IMG_20180919_114732.xml
IMG_20180919_114745.jpg
IMG_20180919_114745.xml
IMG_20180919_114945.jpg
IMG_20180919_114945.xml

 Images in JPG, JPEG, PNG, and BMP formats are supported. When uploading images on the ModelArts console, ensure that the size of an image does not exceed 5 MB and the total size of images to be uploaded in one attempt does not exceed 8 MB. If the data volume is large, use OBS Browser+ to upload images.

 A label name can contain a maximum of 32 characters, including letters, digits, hyphens (-), and underscores (\_).

Field	Mandat ory	Description
folder	Yes	Directory where the data source is located
filename	Yes	Name of the file to be labeled
size	Yes	<ul> <li>Image pixel</li> <li>width: image width. This parameter is mandatory.</li> <li>height: image height. This parameter is mandatory.</li> <li>depth: number of image channels. This parameter is mandatory.</li> </ul>
segment ed	Yes	Segmented or not
object	Yes	<ul> <li>Object detection information. Multiple object{} functions are generated for multiple objects.</li> <li>name: class of the labeled object. This parameter is mandatory.</li> <li>pose: shooting angle of the labeled object. This parameter is mandatory.</li> <li>truncated: whether the labeled object is truncated (0 indicates that the object is not truncated). This parameter is mandatory.</li> <li>occluded: whether the labeled object is occluded (0 indicates that the object is not occluded). This parameter is mandatory.</li> <li>difficult: whether the labeled object is difficult to identify (0 indicates that the object is mandatory.</li> <li>confidence: confidence score of the labeled object. The value range is 0 to 1. This parameter is optional.</li> <li>bndbox: bounding box type. This parameter is mandatory. For details about the possible values, see Table 3-5.</li> </ul>

 Table 3-4 PASCAL VOC format description

	•	<b>C D</b>
type	Shape	Labeling Information
bndbox	Rectangl e	Coordinates of the upper left and lower right points
		<xmin>100<xmin></xmin></xmin>
		<ymin>100<ymin></ymin></ymin>
		<xmax>200<xmax></xmax></xmax>
		<ymax>200<ymax></ymax></ymax>

Table 3-5 Description of bounding box types

Example of the label file in KITTI format:

<annotation></annotation>
<folder>test_data</folder>
<pre><filename>260730932.jpg</filename></pre>
<size></size>
<width>767</width>
<height>959</height>
<depth>3</depth>
<segmented>0</segmented>
<object></object>
<name>bag</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<occluded>0</occluded>
<difficult>0</difficult>
 bndbox>
<xmin>108</xmin>
<ymin>101</ymin>
<xmax>251</xmax>
<ymax>238</ymax>

# 3.3.2 Creating a Project

ModelArts ExeML supports image classification and object detection projects. You can create any of them based on your needs. Perform the following operations to create an ExeML project.

#### Procedure

- 1. Log in to the ModelArts console. In the navigation pane, choose **ExeML**.
- 2. Click **Create Project** in the box of your desired project. The page for creating an ExeML project is displayed.

#### Figure 3-15 Create Project



#### 3. On the project creation page, set parameters by referring to **Table 3-6**.

★ Billing Mode	Pay-per-use	
* Name	ExeM	
Description		
		0/500
* Datasets	Select a dataset.	▼ C Create Dataset
★ Output Path		Select
★ Training Flavor	Select a flavor.	•

#### Table 3-6 Parameters

Parameter	Description	
Name	Name of a project	
	• Enter a maximum of 64 characters. Only digits, letters, underscores (_), and hyphens (-) are allowed. This parameter is mandatory.	
	• Start with a letter.	
	• The name must be unique.	
Description	Brief description of a project	
Dataset Source	You can select a dataset or click <b>Create Dataset</b> to create one.	
	• Existing dataset: Select a dataset from the drop-down list box. Only datasets of the same type are displayed.	
	• Creating a dataset: Click <b>Create Dataset</b> to create a dataset. For details, see .	
Output Path	An OBS path for storing ExeML data	
	<b>NOTE</b> The output path stores all data generated in the ExeML project.	
Training Flavor	Select a training flavor for this ExeML project. You will be billed based on different flavors.	

4. Click **Create Project**. Then, the ExeML workflow is displayed.

- 5. Wait until the workflow of the object detection project executes the following phases in sequence:
  - a. Label Data: Check data labeling.
  - b. **Publish Dataset Version**: Publish a version for the labeled dataset.
  - c. Check Data: Check whether any exception occurs in your dataset.
  - d. **Detect Objects**: Train the dataset of the published version to generate a model.
  - e. **Register Model**: Register the trained model with model management.
  - f. **Deploy Service**: Deploy the generated model as a real-time service.

#### **Quickly Searching for a Project**

On the ExeML overview page, you can use the search box to quickly search for and filter workflows based on the ExeML type (or project name).

- 1. Log in to the ModelArts console. In the navigation pane, choose **ExeML**.
- 2. In the search box above the ExeML project list, filter the desired workflows based on the required property, such as name, status, project type, current phase, and tag.

#### Figure 3-16 Property

Q Select a property or enter a keyword.				
Nam	Property	*		
	Name			
	Status			
	Project Type			
10	Phase			
10	Tag			
	Executions			
	Created At	-		

3. To adjust the basic settings of ExeML and select the columns you want to see,

click <sup>(2)</sup> on the right of the search box.

**Table Text Wrapping**: This function is disabled by default. If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.

**Operation Column**: This function is enabled by default. If you enable this function, the **Operation** column is always fixed at the rightmost position of the table.

**Custom Columns**: By default, all items are selected. You can select columns you want to see.

Settings			×
Basic settings		Custom Columns	
Table Text Wrapping	Auto wrapping If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.	Search          Image: Search	Q
Operation Column	Fixed position If you enable this function, the Operation column is always fixed at the rightmost position of the table.	<ul> <li>Project Type</li> <li>Phase</li> <li>Tag</li> <li>Executions</li> <li>Created At</li> <li>Description</li> <li>Operation (default)</li> </ul>	
	OK Cancel		

Figure 3-17 Customizing table columns

- 4. Click **OK**. Then, the columns will be displayed based on the settings.
- 5. To arrange ExeML projects by a specific property, click in the table header.

# 3.3.3 Labeling Data

Before data labeling, consider how to design labels. The labels must correspond to the distinct characteristics of the detected images and are easy to identify (the detected object in an image is highly distinguished from the background). Each label specifies the expected recognition result of the detected images. After the label design is complete, prepare images based on the designed labels. It is recommended that the number of all images to be detected be greater than 100. If the labels of some images are similar, prepare more images. At least two classes of samples are required for training, and each class with at least 50 samples.

- During labeling, the variance of a class should be as small as possible. That is, the labeled objects of the same class should be as similar as possible. The labeled objects of different classes should be as different as possible.
- The contrast between the labeled objects and the image background should be as stark as possible.
- In object detection labeling, a target object must be entirely contained within a labeling box. If there are multiple objects in an image, do not relabel or miss any objects.

After a project is created, you will be redirected to the new-version ExeML and the project starts to run. When the data labeling phase changes to **Awaiting operation**, manually confirm data labeling in the dataset. You can also add or delete data in the dataset and modify labels.

#### Figure 3-18 Data labeling status

Configurations	
🛕 Go to Instan	ce Details to label data.
Attribute	
Status	Awaiting operation
Started At	Jan 18, 2024 16:55:47 GMT+08:00
Duration	00:00:15
Updated At	Jan 18, 2024 16:56:02 GMT+08:00

#### Labeling Images

1. On the labeling phase of the new-version ExeML, click **Instance Details**. The data labeling page is displayed. Click an image to go to the labeling page.



2. Left-click and drag the mouse to select the area where the target object is located. In the dialog box that is displayed, select the label color, enter the label name, for example, **yunbao**, and press **Enter**. After the labeling is complete, the status of the images changes to **Labeled**.

More descriptions of data labeling are as follows:

- You can click the arrow keys in the upper and lower parts of the image, or press the left and right arrow keys on the keyboard to select another image. Then, repeat the preceding operations to label the image. If an image contains more than one object, you can label all the objects.
- You can add multiple labels with different colors for an object detection ExeML project for easy identification. After selecting an object, select a new color and enter a new label name in the dialog box that is displayed to add a new label.
- In an ExeML project, object detection supports only rectangular labeling boxes. In the **Data Management** function, more types of labeling boxes are supported for object detection datasets.
- In the Label Data window, you can scroll the mouse to zoom in or zoom out on the image to quickly locate the object.

#### D NOTE

For an object detection dataset, you can add multiple labeling boxes and labels to an image during labeling. The labeling boxes cannot extend beyond the image boundary.

3. After all images in the image directory are labeled, return to the ExeML workflow page and click **Next**. The workflow automatically publishes a data labeling version and performs training.

#### Synchronizing or Adding Images

In the labeling phase, click **Instance Details** to go to the data labeling page. Then, add images from your local PC or synchronize images from OBS.

#### Figure 3-19 Adding local images

	Unlabeled 40	Labeled 26	To Be Confirmed 0 All	statuses 66
(	Auto Labeling 🔻	Add data 🔻	Synchronize New Data	Batch Process Hard
	Add o	data historical records		

#### Figure 3-20 Synchronizing images from OBS

		C	
Auto Labeling 🔻	Add data 🔻	🕞 Synchronize New Data	Batch Process Hard Examples 🔻

- Add data: You can quickly add images on a local PC to ModelArts. These images will be automatically synchronized to the OBS path specified during project creation. Click Add data and import data.
- **Synchronize New Data**: You can upload images to the OBS directory specified during project creation and click **Synchronize New Data** to quickly add the new images in the OBS directory to ModelArts.
- **Delete**: You can delete images one by one, or select **Select Images on Current Page** to delete all images on the page.

**NOTE** 

Deleted images cannot be recovered.

#### Modifying Labeled Data

After labeling data, you can modify the labeled data on the **Labeled** tab page.

• Modifying based on images

On the dataset details page, click the **Labeled** tab, and then select the image to be modified. Modify the image information in the label information area on the right.

 Modifying a label: In the Labeling area, click the editing icon, enter the correct label name in the text box, and click the check mark to complete the modification. The label color cannot be modified.  Deleting a label: In the Labeling area, click the deletion button to delete a label for the image.

After the label is deleted, click the project name in the upper left corner of the page to exit the labeling page. The image will be returned to the **Unlabeled** tab page.

#### • Modifying based on labels

On the labeling job overview page, click **Label Management** on the right. You will see the label management page, which shows information about all labels.

Figure 3-21 Label management page

Add Label     Delete Label			
Label Name	Attribute	Label Color	Operation
UNBAO YUNBAO		*	Modify Delete

- Modifying a label: Click **Modify** in the **Operation** column. In the dialog box that appears, enter a new label and click **OK**. After the modification, the images that have been added with the label use the new label name.
- Deleting a label: Click the deletion button in the **Operation** column. In the dialog box that appears, confirm the operation and click **OK**.

D NOTE

Deleted tags cannot be restored.

#### **Resuming Workflow Execution**

After confirming data labeling, return back to the new-version ExeML. Click **Next**. Then, the workflow continues to run in sequence until all phases are executed.

Figure 3-22 Resuming the workflow execution

Configurations	Next	Instance Details
A Go to instance Details to label data.		

# 3.3.4 Training a Model

After labeling the images, perform auto training to obtain an appropriate model version.

#### Procedure

1. On the ExeML page of the new version, click the name of the target project. Then, click **Instance Details** on the labeling phase to label data.

#### Figure 3-23 Finding unlabeled data

exeML-f481 < Back to ExeML	
Labeled 27 Unlabeled	0
🕀 Add 🗵 Delete	Synchronize Data Source

- 2. Return to the labeling phase of the new-version ExeML, click **Next** and wait until the workflow enters the training phase.
- 3. Wait until the training is complete. No manual operation is required. If you close or exit the page, the system continues training until it is complete.
- 4. On the object detection phase, wait until the training status changes from **Running** to **Completed**.
- 5. After the training, click on the object detection phase to view metric information. For details about the evaluation result parameters, see Table 3-7.

Parameter	Description	
Recall	Fraction of correctly predicted samples over all samples predicted as a class. It shows the ability of a model to distinguish positive samples.	
Precision	Fraction of correctly predicted samples over all samples predicted as a class. It shows the ability of a model to distinguish negative samples.	
Accuracy	Fraction of correctly predicted samples over all samples. It shows the general ability of a model to recognize samples.	
F1 Score	Harmonic average of the precision and recall of a model. It is used to evaluate the quality of a model. A high F1 score indicates a good model.	

Table 3-7	Evaluation	result	parameters
-----------	------------	--------	------------

#### **NOTE**

An ExeML project supports multiple rounds of training, and each round generates an AI application version. For example, the first training version is **0.0.1**, and the next version is **0.0.2**. The trained models can be managed by training version. After the trained model meets your requirements, deploy the model as a service.
# 3.3.5 Deploying a Model as a Service

# **Deploying a Service**

You can deploy a model as a real-time service that provides a real-time test UI and monitoring capabilities. After the model is trained, you can deploy a **Successful** version with ideal accuracy as a service. The procedure is as follows:

- 1. On the phase execution page, after the service deployment status changes to **Awaiting input**, double-click **Deploy Service**. On the configuration details page, configure resource parameters.
- 2. On the service deployment page, select the resource specifications used for service deployment.

#### Figure 3-24 Resource specifications

Configurations			NEX
Attribute		Parameters	
Status	Awaiting Input	* model_name	ExeML_5d0e
Started At	Dec 07, 2023 09:32:06 GMT+08:00		
Duration	00:00:05		_
Updated At	Dec 07, 2023 09:32:12 GMT+08:00	* Auto Stop (?)	
Input			
* Al Application	Source My AI application Workflow step		
AI Application	and Version ExeML_5d0e(Synchronou + 0.0.2(Nor + C		
* Resource Poo	Public Resource Pool Dedicated Resource Pool		
Specifications	CPU: 2 core 8GB +		
	Application scenario: Standard CPU specifications, meeting the running and prediction requirements of most models		
Price	¥ 0.80 /hour		
Traffic Ratio (%)	- 100 +		
Compute Nodes	- t +		
Environment Vari	iable OAdd Environment Variable		

- **AI Application Source**: defaults to the generated AI application.
- **AI Application and Version**: The current AI application version is automatically selected, which is changeable.
- **Resource Pool**: defaults to public resource pools.
- **Traffic Ratio**: defaults to **100** and supports a value range of 0 to 100.
- Specifications: Select available specifications based on the list displayed on the console. The specifications in gray cannot be used in the current environment. If there are no specifications after you select a public resource pool, no public resource pool is available in the current environment. In this case, use a dedicated resource pool or contact the administrator to create a public resource pool.
- **Compute Nodes**: an integer ranging from 1 to 5. The default value is **1**.
- Auto Stop: enables a service to automatically stop at a specified time. If this function is not enabled, the real-time service continuously runs. Auto stop is enabled by default and its default value is 1 hour later.

The auto stop options are **1 hour later**, **2 hours later**, **4 hours later**, **6 hours later**, and **Custom**. If you select **Custom**, enter any integer from 1 to 24 in the text box on the right.

#### 

You can choose the package that you have bought when you select specifications. On the configuration fee tag, you can view your remaining package quota and how much you will pay for any extra usage.

3. After configuring resources, click **Next**. Wait until the status changes to **Executed**. The AI application has been deployed as a real-time service.

# **Testing a Service**

• After the service is deployed, click **Instance Details** to go to the real-time service details page. Click the **Prediction** tab to test the service.

Figure 3-25 Testing the service

Back to Real-Time Service List		
Basic Information		
Name	workflow_created_service_8f575a34-9919-462c-a0a9-a22c68d5b203	
Status	Running(59 minutes until stop) 3	
Source	My Deployment	
Synchronize Data	Synchronize Data	
Data Collection   ?		
Traffic Limit	200	
WahSockat	Dicabled	
Usage Guides	Prediction Configuration Updates Filter Monitoring Events Logs Authorizations	Та

- You can also choose Service Deployment > Real-Time Services and click Predict in the Operation column of the target service for testing. The testing procedure is the same as that described in the following section. For details, see Testing a Service.
- You can also use code to test a service. For details, see "Accessing Real-Time Services".
- The following describes the procedure for performing a service test after the object detection model is deployed as a service on the ExeML page.
  - a. After the model is deployed, click **Instance Details** in the service deployment phase to go to the service page. On the **Prediction** tab page, click **Upload** and select a local image for test.
  - b. Click Predict to perform the test. After the prediction is complete, the result is displayed in the Test Result pane on the right. If the model accuracy does not meet your expectation, add images on the Label Data tab page, label the images, and train and deploy the model again. Table 3-8 describes the parameters in the prediction result. If you are satisfied with the model prediction result, call the API to access the real-time service as prompted. For details, see "Accessing Real-Time Services".

Currently, only JPG, JPEG, BMP, and PNG images are supported.

Parameter	Description
detection_class es	Label of each detection box
detection_boxe s	Coordinates of four points (y_min, x_min, y_max, and x_max) of each detection box, as shown in Figure <b>3-26</b>
detection_score s	Confidence of each detection box

 Table 3-8 Parameters in the prediction result





D NOTE

• A running real-time service keeps consuming resources. If you do not need to use the real-time service, click **Stop** in the **Version Manager** pane to stop the service. If you want to use the service again, click **Start**.

# **3.4 Predictive Analytics**

# 3.4.1 Preparing Data

Before using ModelArts to build a predictive analytics model, upload data to OBS.

# **Requirements on Datasets**

The data set used in the predictive analytics project must be a table dataset in .csv format. For details about the table dataset, see .

# **NOTE**

To convert the data from .xlsx to .csv, perform the following operations:

Save the original table data in .xlsx. Choose **File** > **Save As**, select a local address, set **Save as type:** to **CSV (Comma delimited)**, and click **Save**. Then, click **OK** in the displayed dialog box.

#### Requirements on the training data:

- The number of columns in the training data must be the same, and there has to be at least 100 data records (a feature with different values is considered as different data records).
- The training columns cannot contain timestamp data (such as yy-mm-dd or yyyy-mm-dd).
- If a column has only one value, the column is considered invalid. Ensure that there are at least two values in the label column and no data is missing.

#### 

The label column is the training target specified in a training task. It is the output (prediction item) for the model trained using the dataset.

- In addition to the label column, the dataset must contain at least two valid feature columns. Ensure that there are at least two values in each feature column and that the percentage of missing data must be lower than 10%.
- Due to the limitation of the feature filtering algorithm, place the predictive data column at the last. Otherwise, the training may fail.

# Example of a table dataset:

The following table takes the bank deposit predictive dataset as an example. Data sources include age, occupation, marital status, cultural level, and whether there is a personal mortgage or personal loan.

Field	Meaning	Туре	Description
attr_1	Age	Int	Age of the customer
attr_2	Occupation	String	Occupation of the customer
attr_3	Marital status	String	Marital status of the customer
attr_4	Education status	String	Education status of the customer
attr_5	Real estate	String	Housing situation of the customer
attr_6	Loan	String	Loan of the customer
attr_7	Deposit	String	Deposit of the customer

Table 3-9 Fields and meanings of data sources

attr_1	attr_2	attr_3	attr_4	attr_5	attr_6	attr_7
31	blue- collar	married	secondar y	yes	no	no
41	manage ment	married	tertiary	yes	yes	no
38	technicia n	single	secondar y	yes	no	no
39	technicia n	single	secondar y	yes	no	yes
39	blue- collar	married	secondar y	yes	no	no
39	services	single	unknown	yes	no	no

 Table 3-10
 Sample data of the dataset

# **Uploading Data to OBS**

In this section, the OBS console is used to upload data.

#### Upload files to OBS according to the following specifications:

The OBS path of the predictive analytics projects must comply with the following rules:

- The OBS path of the input data must redirect to the data files. The data files must be stored in a folder in an OBS bucket rather than the root directory of the OBS bucket, for example, **/obs-xxx/data/input.csv**.
- There must be at least 100 lines of valid data in .csv. There cannot be more than 200 columns of data and the total data size must be smaller than 100 MB.

#### Procedure for uploading data to OBS:

Perform the following operations to import data to the dataset for model training and building.

- 1. Log in to the OBS console and create a bucket.
- 2. Upload the local data to the OBS bucket. If you have a large amount of data, use OBS Browser+ to upload data or folders. The uploaded data must meet the dataset requirements of the ExeML project.

#### **NOTE**

Upload data from unencrypted buckets. Otherwise, training will fail because data cannot be decrypted.

# **Creating a Dataset**

After the data is prepared, create a proper dataset. For details, see .

# FAQs

How do I process Schema information when creating a table dataset using data selected from OBS?

Schema information includes the names and types of table columns, which must be the same as those of the imported data.

- If the original table contains a table header, enable **Contain Table Header**. The first row of the file will be used as column names. You do not need to modify the Schema information.
- If the original table does not contain a table header, disable Contain Table Header. After data is selected from OBS, the column names will be used as the first row of the table by default. Change the column names to attr\_1, attr\_2, ..., attr\_n. attr\_n is the prediction column placed at last.

# 3.4.2 Creating a Project

ModelArts ExeML supports image classification, and object detection projects. You can create any of them based on your needs. Perform the following operations to create an ExeML project.

# Procedure

- 1. Log in to the ModelArts console. In the navigation pane, choose **ExeML**.
- 2. Click **Create Project** in the box of your desired project.

# Figure 3-27 Creating a project (1)



3. On the displayed page, set the parameters by referring to Table 3-11.

# Figure 3-28 Creating a project (2)

* Billing Mode	Pay-per-use		
* Name	ExeML_1210		
Description			
		0/500	
* Datasets	Select a dataset.	*	Create Dataset
* Label Column	Please Select Label Column.	•	
* Output Path			Select
* Training Flavor	Select a flavor.	*	

# Table 3-11 Parameters

Parameter	Description
Name	Name of a project
	• Enter a maximum of 64 characters. Only digits, letters, underscores (_), and hyphens (-) are allowed. This parameter is mandatory.
	Start with a letter.
	The name must be unique.
Description	Brief description of a project
Datasets	You can select a dataset or click <b>Create Dataset</b> to create one.
	• Existing dataset: Select a dataset from the drop-down list box. Only datasets of the same type are displayed.
	• Creating a dataset: Click <b>Create Dataset</b> to create a dataset. For details, see .

Parameter	Description
Label Column	Select the column you want to predict. The label column is the output of an inference model. During model training, all information is used to train an inference model. The model uses the data of other columns as the input and outputs the inference result in the label column. You can publish the model as a real-time inference service.
Output Path	Select an OBS path for storing ExeML data. <b>NOTE</b> The output path stores all data generated in the ExeML project.
Training Flavor	Select a training flavor for this ExeML project. You will be billed based on different flavors.

- 4. Click **Create Project**. Then, the ExeML workflow is displayed.
- 5. Wait until the workflow of the predictive analytics project executes the following phases in sequence:
  - a. **Publish Dataset Version**: Publish a version for the labeled dataset.
  - b. Check Data: Check whether any exception occurs in your dataset.
  - c. **Predict**: Train the dataset of the published version to generate a model.
  - d. **Register Model**: Register the trained model with model management.
  - e. **Deploy Service**: Deploy the generated model as a real-time service.

# **Quickly Searching for a Project**

On the ExeML overview page, you can use the search box to quickly search for and filter workflows based on the ExeML type (or project name).

- 1. Log in to the ModelArts console. In the navigation pane, choose **ExeML**.
- 2. In the search box above the ExeML project list, filter the desired workflows based on the required property, such as name, status, project type, current phase, and tag.

# Figure 3-29 Property

Q Select a property or enter a keyword.		
Nam	Property	•
	Name	
	Status	
	Project Type	
10	Phase	
10	Tag	
	Executions	
	Created At	-

3. To adjust the basic settings of ExeML and select the columns you want to see,

click  $^{\textcircled{0}}$  on the right of the search box.

**Table Text Wrapping**: This function is disabled by default. If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.

**Operation Column**: This function is enabled by default. If you enable this function, the **Operation** column is always fixed at the rightmost position of the table.

**Custom Columns**: By default, all items are selected. You can select columns you want to see.

#### Figure 3-30 Customizing table columns

Settings			
Basic settings		Custom Columns	
Table Text Wrapping	Auto wrapping If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.	Search          Image: Search         Image: Status	Q
Operation Column	Fixed position If you enable this function, the Operation column is always fixed at the rightmost position of the table.	<ul> <li>Project Type</li> <li>Phase</li> <li>Tag</li> <li>Executions</li> <li>Created At</li> <li>Description</li> <li>Operation (default)</li> </ul>	
	OK Cancel		

4. Click **OK**. Then, the columns will be displayed based on the settings.

 $\times$ 

5. To arrange ExeML projects by a specific property, click in the table header.

# 3.4.3 Training a Model

3.

After the ExeML task is created, a model is trained for predictive analytics. You can publish the model as a real-time inference service.

# Procedure

- 1. On the ExeML page of the new version, click the name of the target project to view the execution status of the current workflow.
- 2. On the predictive analytics phase, wait until the phase status changes from **Running** to **Executed**.
  - Click to view the training details, such as the label column, data type, accuracy, and evaluation result.

The example is a discrete value of binary classification. For details about the evaluation result parameters, see **Table 3-12**.

For details about the evaluation results generated for different data types of label columns, see **Evaluation Results**.

# **NOTE**

An ExeML project supports multiple rounds of training, and each round generates an Al application version. For example, the first training version is **0.0.1**, and the next version is **0.0.2**. The trained models can be managed by training version. After the trained model meets your requirements, deploy the model as a service.

# **Evaluation Results**

The parameters in evaluation results vary depending on the training data type.

Discrete values

The evaluation parameters include recall, precision, accuracy, and F1 score, which are described in the following table.

Param eter	Description
Recall	Fraction of correctly predicted samples over all samples predicted as a class. It shows the ability of a model to distinguish positive samples.
Precisi on	Fraction of correctly predicted samples over all samples predicted as a class. It shows the ability of a model to distinguish negative samples.
Accura cy	Fraction of correctly predicted samples over all samples. It shows the general ability of a model to recognize samples.

Table 3-12 Parameters in discrete value evaluation results

Param eter	Description
F1 Score	Harmonic average of the precision and recall of a model. It is used to evaluate the quality of a model. A high F1 score indicates a good model.

## • Continuous values

The evaluation parameters include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The three error values represent a difference between a real value and a predicted value. During multiple rounds of modeling, a group of error values is generated for each round of modeling. Use these error values to determine the quality of a model. A smaller error value indicates a better model.

# 3.4.4 Deploying a Model as a Service

# **Deploying a Service**

You can deploy a model as a real-time service that provides a real-time test UI and monitoring capabilities. After the model is trained, you can deploy a **Successful** version with ideal accuracy as a service. The procedure is as follows:

- 1. On the phase execution page, after the service deployment status changes to **Awaiting input**, double-click **Deploy Service**. On the configuration details page, configure resource parameters.
- 2. On the service deployment page, select the resource specifications used for service deployment.

#### Figure 3-31 Resource specifications

Configurations			Next
Attribute		Parameters	
Status	Awaiting Input	* model_name	ExeML_5d0e
Started At	Dec 07, 2023 09:32:06 GMT+08:00		
Duration	00:00:05		-
Updated At	Dec 07, 2023 09:32:12 GMT+08:00	* Auto Stop 🧿	
Input			
* Al Application	n Source My Al application Workflow step		
AI Application	n and Version ExeML_5dDe(Synchronou v 0.0.2(Nor v C		
* Resource Po	Public Resource Pool Dedicated Resource Pool		
Specification	IS CPU: 2 core 8GB +		
	Application scenario: Standard CPU specifications, meeting the running and prediction requirements of most models		
Price	¥ 0.80 /hour		
Traffic Ratio (%)	) - 100 +		
Compute Nodes	s - 1 +		
Environment Va	eriable  OAdd Environment Variable		

- **AI Application Source**: defaults to the generated AI application.
- **AI Application and Version**: The current AI application version is automatically selected, which is changeable.

- **Resource Pool**: defaults to public resource pools.
- Traffic Ratio: defaults to 100 and supports a value range of 0 to 100.
- Specifications: Select available specifications based on the list displayed on the console. The specifications in gray cannot be used in the current environment. If there are no specifications after you select a public resource pool, no public resource pool is available in the current environment. In this case, use a dedicated resource pool or contact the administrator to create a public resource pool.
- **Compute Nodes**: an integer ranging from 1 to 5. The default value is **1**.
- Auto Stop: enables a service to automatically stop at a specified time. If this function is disabled, a real-time service will continue to run. The auto stop function is enabled by default. The default value is 1 hour later.

The options are **1 hour later**, **2 hours later**, **4 hours later**, **6 hours later**, and **Custom**. If you select **Custom**, enter any integer from 1 to 24 in the text box on the right.

D NOTE

You can choose the package that you have bought when you select specifications. On the configuration fee tag, you can view your remaining package quota and how much you will pay for any extra usage.

3. After configuring resources, click **Next** and confirm the operation. Wait until the status changes to **Executed**, which means the AI application has been deployed as a real-time service.

# **Testing the Service**

• After the service is deployed, click **Instance Details** to go to the real-time service details page. Click the **Prediction** tab to test the service.

#### Figure 3-32 Testing the service

< Back to Real-Time Service List

Basic Information	I							
Name	workflov	v_created_service_8f575a34-991	9-462c-a0a9-	-a22c68d5b203				
Status	🗿 Run	ning(59 minutes until stop) Ö						
Source	My Dep	loyment						
Synchronize Data	Syn	chronize Data						
Data Collection								
Traffic Limit	200							
WebSocket	Disable	1						
Usage Guides	Prediction	Configuration Updates	Filter	Monitoring	Events	Logs	Authorizations	Тад

• You can also choose **Service Deployment** > **Real-Time Services** and click **Predict** in the **Operation** column of the target service for testing. The testing procedure is the same as that described in the following section. For details, see "Testing the Deployed Service".

- You can also use code to test a service. For details, see "Accessing Real-Time Services".
- The following describes the procedure for performing a service test after the predictive analytics model is deployed as a service on the ExeML page.
  - a. After the model is deployed, you can test the model using code. In ExeML, click **Instance Details** on the **Deploy Service** page to go to the real-time service page. On the **Prediction** tab page, enter the debugging code in the **Inference Code** area.
  - b. Click **Predict** to perform the test. After the prediction is complete, the result is displayed in the **Test Result** pane on the right. If the model accuracy does not meet your expectation, train and deploy the model again on the **Label Data** tab page. If you are satisfied with the model prediction result, call the API to access the real-time service as prompted. For details, see "Accessing Real-Time Services".
    - In the input code, the label column of a predictive analytics database must be named class. Otherwise, the prediction will fail.

```
"data": {
    "req_data": [{
        "attr_1": "34",
        "attr_2": "blue-collar",
        "attr_3": "single",
        "attr_4": "tertiary",
        "attr_5": "no",
        "attr_6": "tertiary"
    }]
}
```

In the preceding code snippet, predict is the inference result of the label column.

#### Figure 3-33 Prediction result





}

- A running real-time service continuously consumes resources. If you do not need to use the real-time service, stop the service to stop billing. To do so, click **Stop** in the **More** drop-down list in the **Operation** column. If you want to use the service again, click **Start**.
- If you enable auto stop, the service automatically stops at the specified time and no fees will be generated then.

# 3.5 Tips

# 3.5.1 How Do I Quickly Create an OBS Bucket and a Folder When Creating a Project?

When creating a project, select a training data path. This section describes how to quickly create an OBS bucket and folder when you select the training data path.

- 1. On the page for creating an ExeML project, click and on the right of **Input Dataset Path**. The **Input Dataset Path** dialog box is displayed.
- 2. Click **Create Bucket**. The **Create Bucket** page is displayed. For details, see "Creating a Bucket" in *Object Storage Service Console Operation Guide*.

Input Dataset Path			×
Select a folder.			
Oobs			С
Create Bucket >>		Enter a name.	Q
Name	Last Modified $\exists \equiv$	Scenario ↓Ξ Size ↓Ξ	
lobs-IsIsIs		Bucket	

# Figure 3-34 Creating an OBS bucket

- 3. Select the bucket, and click **Create Folder**. In the dialog box that is displayed, enter the folder name and click **OK**.
  - The name cannot contain the following special characters: \/:\*?"<>|
  - The name cannot start or end with a period (.) or slash (/).
  - The absolute path of a folder cannot exceed 1,023 characters.
  - Any single slash (/) separates and creates multiple levels of folders at once.

# **3.5.2 Where Are Models Generated by ExeML Stored? What Other Operations Are Supported?**

# **Unified Model Management**

For an ExeML project, after the model training is complete, the generated model is automatically displayed on the **AI Application Management > AI Applications** page. The model name is automatically generated by the system. Its prefix is the same as the name of the ExeML project for easy identification.

# 

Models generated by ExeML cannot be downloaded.

# What Other Operations Are Supported for Models Generated by ExeML?

• Deploying models as real-time and batch services

On the **ExeML** page, models can only be deployed as real-time services. You can deploy models as batch services on the **AI Application Management > AI Applications** page.

# • Creating a version

When creating a new version, you can select a meta model only from a ModelArts training job, OBS, model template, or custom image. You cannot create a version from the original ExeML project.

# • Deleting a model or its version

# **4** Workflow

# 4.1 MLOps Overview

# What Is MLOps?

Machine Learning Operations (MLOps) are a set of practices with machine learning (ML) and DevOps combined. With the development of ML, it is expected not only to make breakthroughs in academic research, but also to systematically implement these technologies in various scenarios. However, there is a significant gap between academic research and the implementation of ML technologies. In academic research, an AI algorithm is developed for a certain dataset (a public dataset or a scenario-specific dataset). The algorithm is continuously iterated and optimized for this specific dataset. Scenario-oriented systematical AI development involves the development of both models and the entire system. Then, the successful experience in software system development "DevOps" is naturally introduced to AI development. However, in the AI era, traditional DevOps cannot cover the entire development process of an AI system.

# DevOps

Development and Operations (DevOps) are a set of processes, approaches, and systems that facilitate communication, collaboration, and integration between software development, O&M, and quality assurance (QA) departments. DevOps is a proven approach in large-scale software system development. DevOps not only accelerates the interaction and iteration between services and development, but also resolves the conflicts between development and O&M. Development pursues speed, while O&M requires stability. This is the inherent and root conflict between development and O&M. Similar conflicts occur during the implementation of AI applications. The development of AI applications requires basic algorithm knowledge as well as fast, efficient algorithm iteration. Professional O&M personnel pursue stability, security, and reliability. Their professional knowledge is quite different from that of AI algorithm personnel. O&M personnel have to understand the design and ideas of algorithm personnel for service assurance, which are difficult for them to achieve. In this case, the algorithm personnel are required to take end-to-end responsibilities, leading to high labor cost. This method is feasible if a small number of models are used. However, when AI

applications are implemented on a large scale, manpower will become a bottleneck.

# **MLOps Functions**

The ML development process consists of project design, data engineering, model building, and model deployment. AI development is not a unidirectional pipeline job. During development, multiple iterations of experiments are performed based on the data and model results. To achieve better model results, algorithm engineers perform diverse data processing and model optimization based on the data features and labels of existing datasets. Traditional AI development ends with a one-off delivery of the final model output by iterative experimentation. As time passes after an application is released however, model drift occurs, leading to worsening effects when applying new data and features to the existing model. Iterative experimentation of MLOps forms a fixed pipeline which contains data engineering, model algorithms, and training configurations. You can use the pipeline to continuously perform iterative training on data that is being continuously generated. This ensures that the AI application of the model, built using the pipeline, is always in an optimum state.



## Figure 4-1 MLOps

An entire MLOps link, which covers everything from algorithm development to service delivery and O&M, requires an implementation tool. Originally, the development and delivery processes were conducted separately. The models developed by algorithm engineers were delivered to downstream system engineers. In this process, algorithm engineers are highly involved, which is different from MLOps. There are general delivery cooperation rules in each enterprise. When it comes to project management, working process management needs to be added to AI projects as the system does not simply build and manage pipelines, but acts as a job management system.

The tool for the MLOps link must support the following features:

- Process analysis: Accumulated industry sample pipelines help you quickly design AI projects and processes.
- Process definition and redefinition: You can use pipelines to quickly define AI projects and design workflows for model training and release for inference.

- Resource allocation: You can use account management to allocate resource quotas and permissions to participants (including developers and O&M personnel) in the pipeline and view resource usage.
- Task arrangement: Sub-tasks can be arranged based on sub-pipelines. Additionally, notifications can be enabled for efficient management and collaboration.
- Process quality and efficiency evaluation: Pipeline execution views are provided, and checkpoints for different phases such as data evaluation, model evaluation, and performance evaluation are added so that AI project managers can easily view the quality and efficiency of the pipeline execution.
- Process optimization: In each iteration of the pipeline, you can customize core metrics and obtain affected data and causes. In this way, you can quickly determine the next iteration based on these metrics.

# 4.2 What Is Workflow?

A workflow is a pipeline tool developed based on service scenarios for deploying models or applications. In ML, a pipeline may involve data labeling, data processing, model development and training, model evaluation, application development, and application evaluation.

# Figure 4-2 Workflow



Different from traditional ML-based model building, workflows can be used to develop production pipelines. Based on MLOps, workflows enable runtime recording, monitoring, and continuous running. The development and continuous iteration of a workflow are separated in products based on roles and concepts.

A pipeline consists of multiple phases. The functions required by the pipeline and the function parameters are called through workflow SDKs. When developing a pipeline, you can use SDKs to describe phases and the relationships between phases. Developing a pipeline is the development state of the workflow. After a pipeline is determined, you can consolidate and provide it for others to use. You do not need to pay attention to what algorithms are used in the pipeline or how the pipeline is implemented. Instead, you only need to check whether the models or applications produced by the pipeline meet the release requirements. If not, you need to check whether the data and parameters need to be adjusted for iteration. Using such a consolidated pipeline is the running state of the workflow.

The development and running states of a workflow are as follows:

- Development state: Workflow Python SDKs are used to develop and debug a pipeline.
- Running state: You can configure and run a produced pipeline in visualized mode.

Leveraging DevOps principles and practices, workflows orchestrate ModelArts capabilities to help you efficiently train, develop, and deploy AI models.

Different functions are implemented in the development and running states of a workflow.

# Workflow Development State

Based on service requirements, you can use Python SDKs provided by ModelArts workflows to offer each ModelArts capability as a step in a pipeline. This is a familiar and flexible development mode for AI developers. Python SDKs support:

- Debugging: partially execution, fully execution, and debugging.
- Release: Release a workflow from the development state to the running state.
- Experiment record: for persistence and the management of experiments.

# Workflow Running State

Workflows are executed in visualized mode. You only need to pay attention to some simple parameter settings, whether the model needs to be retrained, and model deployment.

Running workflows are released from the development state or subscribed to from AI Hub.

A running workflow supports:

- Unified configuration management: The parameters and resources required for a workflow are centrally managed.
- Workflow operations: include starting, stopping, copying, and deleting workflows.
- Running record: records historical running parameters and statuses of the workflow.

# Workflow Components

A workflow is the description of a directed acyclic graph (DAG). You can develop a DAG through a workflow. A DAG consists of phases and the relationships between phases. To define a DAG, specify the execution content and sequence on phases. A green rectangle indicates a phase, and the link between phases shows the phase relationship. A DAG is actually an ordered job execution template.

# Sample Workflows

ModelArts provides abundant scenario-oriented sample workflows. You can subscribe to them in .

# Subscribing to and Using an AI Hub Workflow

- 1. On the workflow asset page of AI Hub, select and subscribe to a workflow.
- 2. After the subscription, click **Run**. You will be automatically redirected to the ModelArts console. Select an asset version, workflow name, service region, and workspace, and click **Import**. The workflow details page is displayed.
- 3. Click **Configure** in the upper right corner. On the configuration page that appears, set parameters and click **Save** in the upper right corner to save the configuration.
- 4. Click **Start** in the top right corner to start the workflow.
- 5. On the workflow execution page, wait for the workflow to start running.
- 6. On the dashboard, check the status of each phase. The workflow runs automatically from one phase to the next until it finishes all the phases.

# 4.3 How to Use a Workflow?

# 4.3.1 Using a Workflow Subscribed to From AI Hub

- 1. On the workflow asset page of AI Hub, select and subscribe to a workflow.
- 2. After the subscription, click **Run**. You will be automatically redirected to the ModelArts console. Select an asset version, workflow name, service region, and workspace, and click **Import**. The workflow details page is displayed.
- 3. Click **Configure** in the upper right corner. On the configuration page that appears, set parameters and click **Save** in the upper right corner to save the configuration.
- 4. Click **Start** in the upper right corner to start the workflow.
- 5. On the workflow execution page, wait for the workflow to start running.

On the dashboard, check the status of each phase. The workflow runs automatically from one phase to the next until it finishes all the phases.

# **4.3.2 Configuring a Workflow**

# 4.3.2.1 Configuration Entries

Before or during the execution of a workflow, configure the parameters and resources required by the workflow. After obtaining the workflow, modify the configuration as required so that the produced model or application is better suited for your needs.

Workflow configurations include the configurations before and during the workflow execution.

# **Configurations Before Workflow Execution**

Log in to the ModelArts management console and choose **Workflow** to go to the workflow list page. There are two entries to configure a workflow before it runs.

• Click **Configure** in the **Operation** column of the target workflow to go to the workflow configuration page.

#### Figure 4-3 Configure



• On the workflow list page, click the name of the target workflow. On the workflow details page that is displayed, click **Configure** in the upper right corner.

## Figure 4-4 Configure

-	-		
Start	Delete Workflow	Configure	

# **Configurations During Workflow Execution**

Certain phases require parameter configuration during the execution of a workflow. When the workflow runs to such a phase, it pauses and waits for your input.

On the workflow overview page, view the to-dos on the right. Click the workflow name to go to the phase in the awaiting input status. Set the parameters for the phase and click **Next**.

#### Figure 4-5 Workflow to-dos



# 4.3.2.2 Runtime Configurations

Work directories can be centrally managed in ModelArts workflow. The root directory is configured in **Runtime Configurations**.

- 1. On the workflow list page, click the name of the target workflow.
- 2. Click **Configure** in the upper right corner.
- 3. On the **Workflow Configurations** tab page, configure the **Runtime Configurations** settings.

#### Figure 4-6 Runtime Configurations

## Runtime Configurations

output_storage	Select

# 4.3.2.3 Resource Configurations

You can configure resources for multiple phases within a workflow, with the ability to specify different configurations for each phase.

Figure 4-7 Resource Configurations

Resources			
Step	Dedicated Resource Pool	Flavor	Price
		0.PU: 11NVIDIA-V100(3208) *	

If you need to use a dedicated resource pool, enable **Dedicated Resource Pool**.

Specify the required inference resource specifications when the workflow runs on the service deployment phase.

- 1. Wait until the workflow runs on the service deployment phase, and the phase enters the **Awaiting input** status.
- 2. In the Input area, select the required inference resource specifications.
- 3. Click Next.

# 4.3.2.4 Tab Configuration

You can filter workflows by tag for easy classification, which saves a lot of time.

# **Configuring Tags**

- 1. On the ModelArts console, choose **Workflow** from the navigation pane. The workflow list page is displayed.
- 2. Locate the workflow you want to tag and click its name. The workflow details page is displayed.
- 3. Click  $\overset{@}{=}$  in the upper left corner.
- 4. In the **Edit Workflow** dialog box that appears, enter a tag in the **Tag** text box and click **Add Tag**. The new tag is displayed below. You can add multiple tags at a time. After the tags are added, click **Yes**.

# Searching for a Workflow by Tag

Workflows with tags can be filtered by tag in the search box.

1. In the search box above the workflow list, set **Property** to **Tag**.



2. On the tag list that appears, click the target tag. The workflow list displays workflows with that tag.

# 4.3.2.5 Input and Output Configurations

You can set input and output parameters on the configuration page, or when the workflow is running.

When a workflow is running, you can configure parameters for the phase in the **Awaiting input** state.

# **Input Configurations**

The following table describes the parameters you need to specify.

Input Parameter	Description
dataset	Select an existing dataset or create a new one.
obs	Select your OBS path.
label task	Select a labeling job under your dataset.
service	Select a deployed real-time service.
swr image	Select the image storage path required for registering the model.

Table 4-1 Input parameters

# **Output Configurations**

Click **Select** to select the OBS path to store the output data.

# 4.3.2.6 Phase Parameters

You can configure different parameters for each phase.

# 4.3.2.7 Saving Configurations

On the workflow configuration page, click **Save** in the upper right corner after you complete the configuration.



Details	Save	Start

After the workflow is saved, click **Start** in the upper right corner of the page. In the dialog box that is displayed, click **OK**. The workflow is started and the runtime page is displayed.

# 4.3.3 Starting, Stopping, Searching for, Copying, or Deleting a Workflow

# Starting a Workflow

When a workflow is not running, you can start it in any of the following ways:

- On the workflow list page, click **Start** in the **Operation** column. In the displayed dialog box, click **OK**.
- On the runtime configuration page, click **Start** in the upper right corner. In the displayed dialog box, click **OK**.
- On the workflow configuration page, click **Start** in the upper right corner. In the displayed dialog box, click **OK**.

# Searching for a Workflow

On the workflow list page, you can use the search box to quickly search for workflows based on workflow properties.

- 1. Log in to the ModelArts console. In the navigation pane, choose **Workflow**.
- 2. In the search box above the workflow list, filter workflows based on the required property, such as the name, status, current phase, start time, running duration, or tag.

# Figure 4-9 Property

Q	Q Select a property or enter a keyword		
Nam	Property		
	Name		
	Status		
	Phase		
	Started At		
	Duration		
	Тад		
	Executions		

- 3. Click on the right of the search box to set the content you want to display on the workflow list page and modify other display settings.
  - **Table Text Wrapping**: This function is disabled by default. If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.
  - Operation Column: This function is enabled by default. If you enable this function, the Operation column is always fixed at the rightmost position of the table.
  - **Custom Columns**: By default, all items are selected. You can select columns you want to see.

# Figure 4-10 Settings Settings

Basic settings		Custom Columns	
Table Text Wrapping	Auto wrapping	Search	Q
	If you enable this function, excess text will move down to the next line; otherwise, the text will be truncated.	Name (default) Status	^
Operation Column	Fixed position	Phase	- 1
	If you enable this function, the Operation	Started At	
	column is always fixed at the rightmost position	Duration	- 1
	of the table.	🔽 Tag	
		Executions	
		✓ Created At	- 1
		Modified At	
		Created By	- 1
		Source	- 1
		Description	-
	OK Cancel		

- 4. Click OK.
- 5. To arrange workflows by a specific property, click 💼 in the table header.

# Stopping a Workflow

You can stop a running workflow in either of the following ways:

Workflow list page

When a workflow is running, the **Stop** button is available in the **Operation** column. Click **Stop**. In the displayed dialog box, click **OK**.

• Click the name of a running workflow and click **Stop** in the upper right corner of the displayed page. In the displayed dialog box, click **OK**.

# **NOTE**

The **Stop** button is available only for a workflow that is running.

After a workflow is stopped, the associated training jobs and real-time services are also stopped.

# **Copying a Workflow**

A workflow can have only one running instance. If you want to concurrently run a workflow, copy the workflow. To do so, click **More** in the **Operation** column and select **Copy**. In the displayed dialog box, a new name is automatically generated in the format of "Original workflow name\_copy".

You can rename the new workflow. Ensure that the name complies with naming specifications.

# 

A workflow name is 1 to 64 characters long, starting with a letter and containing only letters, digits, underscores (\_), and hyphens (-).

# **Deleting a Workflow**

You can delete a workflow in either of the following ways:

- Workflow list page
  - a. Click More in the Operation column and select Delete.
  - b. In the displayed dialog box, enter **delete** and click **OK**.

Figure 4-11 Confirming the deletion

The associated instances and files generated in the workflow will be retained.
retaineu.
Enter DELETE to confirm
DELETE

• Runtime configuration page

Click **Delete** in the upper right corner of the page. In the displayed dialog box, enter **DELETE** and click **OK**.

# **NOTE**

- Deleted workflows cannot be recovered.
- After a workflow is deleted, the corresponding training jobs and real-time services are not deleted accordingly. To delete them, go to the Training Management > Training Jobs and Service Deployment > Real-Time Services pages.

# 4.3.4 Viewing Workflow Execution Records

All runtime statuses of a workflow are recorded.

- 1. On the workflow list page, click the name of the target workflow.
- 2. On the workflow details page, view all runtime records of the workflow in the left pane.

#### Figure 4-12 Viewing execution records



- 3. Delete or edit the runtime records, or rerun the workflow.
  - To delete a runtime record that is no longer needed, click **Delete**. In the displayed dialog box, click **Yes**.
  - To distinguish a runtime record from others, click Edit Tag to add a tag to it.
  - To rerun the workflow, click **Rerun** on a runtime record.
- 4. Filter and compare all runtime records of the workflow.
  - Filter: You can filter all runtime records by status or tag.

# Figure 4-13 Filtering

Execution	†‡† Filter	Compare
Status Filter		
All		•
Tag Filter		
		•
Reset		

Compare: You can compare all runtime records by status, execution record, start time, duration, and metrics.

#### Figure 4-14 Comparison

Executi	eculur companion										
Drify Store Salected 🕥											
		Status 🖓	Execution 77 P	Phase	Started At ↓⊟	Duration ↓⊟	Metrics				
	Name						Precison	Recall	MAP	MAP50	
	execution-001	Executed	execution-001		Feb 8, 2023 16:53:51	00:17:56	0.8118549853751054	0.8917458266683074	0.6325202216772374	0.9110373234649735	
	execution-001	Executed	execution-001		Feb 8, 2023 16:52:38	00:01:13					

After you click **Start** to run a workflow, the execution record list is refreshed. In addition, the data is updated on both the DAG and dashboard. An execution record is added after each startup.

You can click any phase on the workflow details page to obtain the phase status, including attributes (status, start time, and duration).

	e			-				
<b>火</b> 数据标注	● ● ● 数据集版本发	→@	图像分类训练	•		·→@	服务部署	
数据标注								
运行状况								详情
展性				参数				
状态	● 還行成功			task_name	dataset-9334			
启动时间	2022/06/02 10:32:00 GMT+08:00				寄输入一个只包含大小写字母、要 直接使用该标注任务;填写新标识	效字、下划线、中划线或客中文字符6 E任务名称,则自动创建新的标注任3	的名称。 填写已有标注任9 号	招称,则
运行时长	00:00:15							
输入								
labeling_input	dataset-9334 V002							
输出								
labeling_out	dataset-9334							

Figure 4-15 Viewing the status of a node

# 4.3.5 Retrying, Stopping, or Proceeding a Phase

• Retrying a phase

If executing a single phase failed, you can click **Retry** to re-execute the current phase without restarting the workflow. Before the retry, you can modify configurations on the **Global Configuration** page. The modification takes effect after the affected phase is retried.

• Stopping a phase

Click a phase to view its details. On this page, you can stop the running phase.

• Proceeding a phase

If parameters need to be configured during the runtime of a single phase, the phase is awaiting operation. After the parameters are configured, you can click **Proceed** to proceed to the execution of the current phase.

# 4.3.6 Partial Execution

To reduce the time consumed by repeated execution in large-scale and complex workflows, you can choose specific phases to execute in sequence.

• Creation

Predefine the phases to be executed when you use the SDK to create a workflow.

• Configuration

When configuring a workflow, enable **Execute Certain Phases**, select phases to be executed, and configure parameters for these phases.

#### Figure 4-16 Execute Certain Phases

Workflow Configurations	Step Configurations	Service Configurations
Execute Certain Phases		
Execute Certain P ?		

# • Start

After saving the configuration, click **Start** to execute certain phases.

# **5** Data Management

# 5.1 Introduction to Data Preparation

# D NOTE

Data management is being upgraded and is invisible to users who have not used data management.

The driving forces behind AI are computing power, algorithms, and data. Data quality affects model precision. Generally, a large amount of high-quality data is more likely to train a high-precision AI model. Models trained using normal data achieves 85% to 90% accuracy, while commercial applications have higher requirements. If you want to improve the model accuracy to 96% or even 99%, a large amount of high-quality data is required. In this case, the data must be more refined, scenario-based, and professional. The preparation of a large amount of high-quality data has become a challenging issue in AI development.

ModelArts is a one-stop AI development platform that supports AI lifecycle development, including data processing, algorithm development, model training, and model deployment. In addition, ModelArts provides AI Hub that can be used to share data, algorithms, and models. ModelArts data management provides end-to-end data preparation, processing, and labeling.

ModelArts data management provides the following functions for you to obtain high-quality AI data:

- Data acquisition
  - Allows you to import data from OBS, MRS, DLI, and GaussDB(DWS).
  - Provides 18+ data augmentation operators to increase data volume for training.
- Improved data quality
  - Allows you to preview various formats of data including images, text, audios, and videos, helping you identify data quality.
  - Allows you to filter data by multiple search criteria, such as sample attributes and labeling information.
  - Provides 12+ labeling tools for refined, scenario-based, and professional data labeling.

- Performs feature analysis based on samples and labeling results, helping you understand data quality.
- More efficient data preparation
  - Allows you to manage data by version for more efficient data management.
  - Provides data processing operators for data validation, data selection, and data cleansing to help you quickly process data.
  - Provides capabilities such as interactive labeling and auto labeling for more efficient data labeling.
  - Enables team labeling and team labeling management for labeling a large amount of data.

# 5.2 Getting Started

This section uses preparing data for training an object detection model as an example to describe how to analyze and label sample data. During actual service development, you can select one or more data management functions to prepare data based on service requirements. The operation process is as follows:

- Making Preparations
- Creating a Dataset
- Analyzing Data
- Labeling Data
- Publishing Data
- Exporting Data

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

# Preparations

Before using data management of ModelArts, complete the following preparations:

When using data management, ModelArts needs to access dependent services such as OBS. Therefore, grant permissions on the **Global Configuration** page. For details, see **Configuring Access Authorization (Global Configuration)**.

# **Creating a Dataset**

In this example, an OBS path is used as the input path to create a dataset. Perform the following operations to create an object detection dataset and import the data to the dataset:

- **Step 1** Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- **Step 2** Click **Create**. On the **Create Dataset** page, create a dataset based on the data type and data labeling requirements.

1. Set the basic information, the name and description of the dataset.

#### Figure 5-1 Basic information of a dataset

* Name	dataset-1445	•
Description		]
		4

2. Set labeling scene and type. In this example, choose **Images** and **Object detection**.

Figure 5-2 Dataset labeling scene and type



3. Select an OBS path as **Input Dataset Path**, and select another OBS path as **Output Dataset Path**.

Figure 5-3 Input and output dataset path

* Input Dataset Path 🕜	Select an OBS path.	Ð
* Output Dataset Path	Select an OBS path.	Ð
Label Set	Enter a label name. +	
Team Labeling ⑦		

4. After setting the parameters, click **Create** in the lower right corner to create a dataset.

----End

# **Analyzing Data**

After a dataset is created, you can perform data analysis based on image features, such as blurs and brightness, to better understand the data quality and determine whether the dataset meets your algorithm and model requirements.

- 1. Create a feature analysis task.
  - a. Before performing feature analysis, publish a dataset version. On the dataset **Dashboard** page, click **Publish** in the upper right corner to publish a new version of the dataset.

b. After the version is published, go to the **Dashboard** page. Click **View Data Feature** and **Feature Analysis**. In the displayed dialog box, select the newly published dataset version and click **OK** to start feature analysis.

## Figure 5-4 Starting feature analysis



c. View the task progress.

You can click **View Task History** to view the task progress. When the task status changes to **Successful**, the task is complete.

Figure 5-5 Feature analysis progress

Dataset Version	Task ID	Created	Duration(hh:	Status
V002	ZJrO7W8735	Mar 16, 2022	00:01:23	Successful

2. View feature analysis results.

After feature analysis is complete, you can select **Version**, **Type**, and **Data Feature Metric** on the **View Data Feature** tab page. Then, the selected versions and metrics are displayed on the page. The displayed chart helps you understand data distribution for a better understanding of your data.

- Version: Select one or more versions for comparison.
- Type: Select types to be analyzed. The values all, train, eval, and inference are available for you to select. They indicate all, training, evaluation, and inference, respectively.
- Data Feature Metric: Select the metrics to be displayed. For details about the metrics, see Data feature metrics.

Figure 5-6 Viewing feature analysis results



In feature analysis results, for example, image brightness distribution is uneven, which means images of a certain brightness are lacking. This greatly affects model training. In this case, increase images of that brightness to make data more even for subsequent model building.

# Labeling Data

• Manual labeling

- a. On the **Unlabeled** tab page, click an image. The system automatically directs you to the page for labeling the image.
- b. On the toolbar of the labeling page, select a proper labeling tool. In this example, a rectangle is used for labeling.

Figure 5-7 Labeling tools

- c. Drag the mouse to select an object, enter a new label name in the displayed text box. If labels already exist, select one from the drop-down list box. Click **Add**.
- d. Click **Back to Data Labeling Preview** in the upper left part of the page to view the labeling information. In the dialog box that is displayed, click **Yes** to save the labeling settings. The selected image is automatically moved to the **Labeled** tab page. On the **Unlabeled** and **All** tab pages, the labeling information is updated along with the labeling process, including the added label names and the number of images for each label.
- Auto labeling

Auto labeling allows you to automatically label remaining data after a small amount of data is manually labeled.

- a. On the dataset details page, click Auto Label in the upper right corner.
- b. In the **Enable Auto Labeling** dialog box, set the following parameters and click **Submit**.
  - Auto Labeling Type: Active learning
  - Algorithm Type: Fast

Retain the default values of other parameters.

Figure 5-8 Starting auto labeling

Enable Auto Labelir	ng
Auto Labeling Type	Active learning Pre-labeling The system uses semi-supervised learning and hard example filtering to perform auto labeling, reducing manual labeling workload and helping you find hard examples.
Algorithm Type (	Fast     Precise
* Specifications	GPU: 1*NVIDIA-V100(32GB)   CPU: 8 vCPUs •
A To enable Auto Lai number of sample: rectangle labeling	peling, add at least one class of label to the data and ensure that the s with the label is not less than 5. Auto Labeling can identify and add only boxes.
Limited-time fr	ee Submit

Auto Labeling is billed based on the training duration. Pricing details

c. View auto labeling progress.

After auto labeling is started, you can view the task progress on the **To Be Confirmed** tab page. After a task is complete, you can view the automatically labeled data on the **To Be Confirmed** tab page.



All 60 Unlabeled 54	Labeled 6 To Be	Confirmed 0			
Auto Labeling Progress			0%		
	Starting	Initialize	Label	Process Result	Ending
		Auto labeling is b	peing performed. Confirm the la Time: 00:00:01	beling result later.	
			Stop		

d. Confirm auto labeling results.

After auto labeling is complete, click the image on the **To be confirmed** tab page. On the labeling details page, you can view or modify the auto labeling result.

For correct labeling, click **Labeled** on the right. For wrong labeling, correct wrong labels. For auto labeling of object detection datasets, confirm images one by one. Ensure that all images are confirmed and go to the next step.

# **Publishing Data**

ModelArts training management allows you to create training jobs using ModelArts datasets or files in an OBS directory. If a dataset is used as the data source of a training job, specify a dataset and version. Therefore, you must have published a dataset version. For details, see **Publishing a Data Version**.

# **NOTE**

Data that is from the same source and labeled in different batches are differentiated by version. This facilitates subsequent model building and development. You can select specified versions.

Figure 5-10 Data source for creating a training job



# **Exporting Data**

ModelArts training management allows you to create training jobs using ModelArts datasets or files in an OBS directory. If you create a training job using an OBS directory, export the prepared data to OBS.

- 1. Export data to OBS.
  - a. On the dataset details page, select or filter the data to be exported, and click **Export** in the upper right corner.
  - b. Set Type to OBS, enter related information, and click OK.
×

**Storage Path**: path where the data to be exported is stored. You are advised not to save data to the input or output path of the current dataset.

#### Figure 5-11 Exporting to OBS

# Export To

Туре	New Dataset	OBS	
Storage Path   ?	Select an OBS path.		Đ
	ОК	Cancel	

- c. After the data is exported, view it in the specified path.
- 2. View task history.

After exporting data, you can view the export task details in **Export History**.

- a. On the dataset details page, click **Export History** in the upper right corner.
- b. In the **View Task History** dialog box, view the export task history of the current dataset. You can view the task ID, creation time, export type, export path, total number of exported samples, and export status.

Figure 5-12 Export history



# **5.3 Introduction to Data Preparation**

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

The driving forces behind AI are computing power, algorithms, and data. Data quality affects model precision. Generally, a large amount of high-quality data is more likely to train a high-precision AI model. Models trained using normal data achieves 85% to 90% accuracy, while commercial applications have higher requirements. If you want to improve the model accuracy to 96% or even 99%, a large amount of high-quality data is required. In this case, the data must be more refined, scenario-based, and professional. The preparation of a large amount of high-quality data has become a challenging issue in AI development.

ModelArts is a one-stop AI development platform that supports AI lifecycle development, including data processing, algorithm development, model training, and model deployment. In addition, ModelArts provides AI Hub that can be used

to share data, algorithms, and models. ModelArts data management provides end-to-end data preparation, processing, and labeling.

ModelArts data management provides the following functions for you to obtain high-quality AI data:

- Data acquisition
  - Allows you to import data from OBS, MRS, DLI, and GaussDB(DWS).
  - Provides 18+ data augmentation operators to increase data volume for training.
- Improved data quality
  - Allows you to preview various formats of data including images, text, audios, and videos, helping you identify data quality.
  - Allows you to filter data by multiple search criteria, such as sample attributes and labeling information.
  - Provides 12+ labeling tools for refined, scenario-based, and professional data labeling.
  - Performs feature analysis based on samples and labeling results, helping you understand data quality.
- More efficient data preparation
  - Allows you to manage data by version for more efficient data management.
  - Provides data processing operators for data validation, data selection, and data cleansing to help you quickly process data.
  - Provides capabilities such as interactive labeling and auto labeling for more efficient data labeling.
  - Enables team labeling and team labeling management for labeling a large amount of data.

# 5.4 Creating a Dataset

Before using ModelArts to prepare data, create a dataset. Then, you can perform operations on the dataset, such as importing data, analyzing data, and labeling data.

# 5.4.1 Dataset Overview

### D NOTE

Data management is being upgraded and is invisible to users who have not used data management.

### **Dataset Types**

ModelArts supports the following types of datasets:

- Images: in .jpg, .png, .jpeg, or .bmp format for image classification, image segmentation, and object detection
- Audio: in .wav format for sound classification, speech labeling, and speech paragraph labeling

- Text: in .txt or .csv format for text classification, named entity recognition, and text triplet labeling
- Video: in .mp4 format for video labeling
- Free format: allows data in any format. Labeling is not available for free format data. The free format applies if labeling is not required or needs to be customized. Select this format if your data is in multiple formats or your data is not in any of the preceding formats.

### Figure 5-13 Example of a dataset in free format

Dashboard Versions				
Delete				
File	Size	Format	Uploaded	File Path
snake_env.py	6.61 KB	Free format	Aug 02, 2021 02:13:14 GMT+0	/0802-swx1037871/modelarts/custom_env/snake_env/
Single_Step_Data_For_Inference (1).txt		Free format	Aug 02, 2021 04:04:24 GMT+0	/0802-swx1037871/modelarts/
Config.json	1.00 KB	Free format	Aug 02, 2021 04:36:09 GMT+0	/0802-swx1037871/modelarts/output/model/
variables.index	2.44 KB	Free format	Aug 02, 2021 06:50:05 GMT+0	/0802-swx1037871/modelarts/out3-3/model/variables/
customize_service.py	4.75 KB	Free format	Aug 02, 2021 06:50:06 GMT+0	/0802-swx1037871/modelarts/out3-3/model/

### **Dataset Functions**

Different types of datasets support different functions, such as auto labeling and team labeling. For details, see **Table 5-1**.

Data set Type	Label ing Type	Creat ing a Datas et	lmpo rting Data	Expo rting Data	Publi shing a Datas et	Modi fying a Data set	Mana ging Datas et Versi ons	Auto Grou ping	Data Featu res
lmag e	lmag e classif icatio n	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	Supp orted
	Objec t detec tion	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	Supp orted
	lmag e segm entati on	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	N/A

Table 5-1 Functions supported by different types of datasets

Data set Type	Label ing Type	Creat ing a Datas et	lmpo rting Data	Expo rting Data	Publi shing a Datas et	Modi fying a Data set	Mana ging Datas et Versi ons	Auto Grou ping	Data Featu res
Audio	Soun d classif icatio n	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Speec h labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Speec h parag raph labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
Text	Text classif icatio n	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Name d entity recog nition	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Text triplet	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
Video	Video labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
Free form at	Free forma t	Supp orted	N/A	_	Suppo rted	Supp orted	Supp orted	N/A	N/A
Table	Table	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A

# **Specifications Restrictions**

• The maximum numbers of samples and labels in a single text, video, or audio database other than a table dataset are 1,000,000 and 10,000, respectively.

- The maximum size of a sample in a single text, video, or audio database other than an image dataset is 5 GB.
- The maximum size of an image for object detection, image segmentation, or image classification is 25 MB.
- The maximum size of a manifest file is 5 GB.
- The maximum size of a text file in a line is 100 KB.
- The maximum size of a labeling result file is 100 MB.

# 5.4.2 Creating a Dataset

Before using ModelArts to manage data, create a dataset. Then, you can perform operations on the dataset, such as labeling data, importing data, and publishing the dataset. This section describes how to create a dataset of the non-table type (image, audio, text, video, and free format) and table type.

### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

### Prerequisites

- You have been authorized to access OBS. To do so, click the **Settings** page in the navigation pane of the ModelArts management console and add access authorization using an agency.
- OBS buckets and folders for storing data are available. In addition, the OBS buckets and ModelArts are in the same region. OBS parallel file systems are not supported. Select object storage.
- OBS buckets are not encrypted. ModelArts does not support encrypted OBS buckets. When creating an OBS bucket, do not enable bucket encryption.

### Image, Audio, Text, Video, and Free Format

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.

#### Figure 5-14 Dataset management page

Cn	Maximum datasets: S00 , Available for creatio	n: 140			All types	▼ Enter a name. Q C 🛞
	Name	Version	Labeling Progress	Created ↓ Ξ	Description	Operation
~	🗟 lch-test2 🥥 ISmUxWNQZVt2Ccoxao	V005	72.73% (16/22)	Apr 03, 2020 11:41:24 GMT+08:00	create from dataset xubo_cla	Import   Publish   Labeling   Export   Delete   More 👻
~	El Ich-test1 O 0ZQJXoobOBoPXBLFB8O	V002	100.00% (3/3)	Apr 03, 2020 11:32:00 GMT+08:00	create from dataset xianao-te	Import   Publish   Labeling   Export   Delete   More 👻

### D NOTE

The number of datasets that can be created under an account in a region is limited. For details, see the number displayed on the **Dataset** page.

2. Click **Create**. On the **Create Dataset** page, create a dataset based on the data type and data labeling requirements.

#### Figure 5-15 Parameter settings

★ Data Type	Images         Audio         Text         Video           Supported formats: .jpg, .png, .jpeg, .bmp	Free format Table
* Data Source	OBS AI Gallery Local file	
* Import Mode	Path	
	You can save the dataset file to be imported to the OBS path	n that you have permission to access. Labeling file format
* Import Path	Select an OBS path.	Ð
	You can import up to 1000000 samples and 10000 labels.	
* Labeling Status	Unlabeled Labeled	
* Output Dataset Path	Select an OBS path.	8

Path for storing output files such as labeled files. The path cannot be the same as the import path or subdirectory of the import path

- Name: name of the dataset, which is customizable
- **Description**: details about the dataset
- Data Type: Select a data type based on your needs.
- Data Source
  - i. Importing data from OBS

If data is available in OBS, select **OBS** for **Data Source**, and configure other mandatory parameters. The labeling formats of the input data vary depending on the dataset type. For details about the labeling formats supported by ModelArts, see **Introduction to Data Importing**.

#### Figure 5-16 Importing data from OBS

* Data Source	OBS	AI Gallery	Local file			
* Import Mode	Path					
	You can save the	dataset file to be in	nported to the OB	5 path that you have p	ermission to access. La	abeling file forma
* Import Path	Select an OBS	oath.		Đ		
	You can import u	p to 1000000 samp	les and 10000 labe	ls.		
* Labeling Status	Unlabeled	Labeled				

ii. Importing data from a local path

If data is not stored in OBS and the required data cannot be downloaded from AI Gallery, ModelArts enables you to upload the data from a local path. Before uploading data, configure **Storage Path** and **Labeling Status**. Click **Upload data** to select the local file for uploading. Select a labeling format when the labeling status is **Labeled**. The labeling formats of the input data vary depending on the dataset type. For details about the labeling formats supported by ModelArts, see **Introduction to Data Importing**.

* Data Source	OBS	AI Gallery	Local file			
★ Storage Path	Select an OBS	Select an OBS path.				
	You can import u	ip to 1000000 samp	oles and 10000 labels.			
Uploading Data	⊕ Upload da	ta				
* Labeling Status	Unlabeled	Labeled				

- For more details about parameters, see **Table 5-2**.

Table 5-2 Dataset parameters

Parameter	Description		
Import Path	OBS path from which your data is to be imported. This path is used as the data storage path of the dataset.		
	OBS parallel file systems are not supported. Select an OBS bucket.		
	When you create a dataset, data in the OBS path will be imported to the dataset. If you modify data in OBS, the data in the dataset will be inconsistent with that in OBS. As a result, certain data may be unavailable. To modify data in a dataset, follow the operations provided in <b>Import Mode</b> or <b>Importing</b> <b>Data from an OBS Path</b> .		
	If the numbers of samples and labels of the dataset exceed quotas, importing the samples and labels will fail.		
Labeling Status	Labeling status of the selected data, which can be <b>Unlabeled</b> or <b>Labeled</b> .		
	If you select <b>Labeled</b> , specify a labeling format and ensure the data file complies with format specifications. Otherwise, the import may fail.		
	Only image (object detection, image classification, and image segmentation), audio (sound classification), and text (text classification) labeling tasks support the import of labeled data.		
Output	OBS path where your labeled data is stored.		
Dataset	NOTE		
	<ul> <li>Ensure that your OBS path name contains letters, digits, and underscores (_) and does not contain special characters, such as ~'@#\$%^&amp;*{}[];;+=&lt;&gt;/ and spaces.</li> </ul>		
	<ul> <li>The dataset output path cannot be the same as the data input path or subdirectory of the data input path.</li> </ul>		
	<ul> <li>It is a good practice to select an empty directory as the dataset output path.</li> </ul>		
	<ul> <li>OBS parallel file systems are not supported. Select an OBS bucket.</li> </ul>		

3. After setting the parameters, click **Submit**.

# Table

1. Log in to the . In the navigation pane, choose **Data Management** > **Datasets**.

### Figure 5-17 Dataset management page

Create Maximum datasets: 500 , Available for creation	an: 140			All types	Enter a name.     Q     C
Name	Version	Labeling Progress	Created ↓≡	Description	Operation
V Ish-test2 O ISmlxxWNQZVt2Ccoxao	V005	72.73% (16/22)	Apr 03, 2020 11:41:24 GMT+08:00	create from dataset xubo_cla	Import   Publish   Labeling   Export   Delete   More 💌
V Relich-test1 O 0ZQJXoobOBoPXBLFB8O	V002	100.00% (3/3)	Apr 03, 2020 11:32:00 GMT+08:00	create from dataset xianao-te	Import   Publish   Labeling   Export   Delete   More 👻

### **NOTE**

The number of datasets that can be created under an account in a region is limited. For details, see the number displayed on the **Dataset** page.

2. Click **Create**. On the **Create Dataset** page, create a table dataset based on the data type and data labeling requirements.

### Figure 5-18 Parameters of a table dataset

* Name	dataset-a108
Description	
	0/256
★ Data Type	Images Audio Text Video Free format Table
Data Source	OBS DWS DLI MRS Local file
	<ul> <li>★ File Path Select an OBS path.</li> <li>Contain Table Header</li> </ul>
* Schema 🕜	Column Name Type String
★ Output Dataset Path	Select an OBS path.

- **Name**: name of the dataset, which is customizable
- **Description**: details about the dataset
- Data Type: Select a data type based on your needs.
- For more details about parameters, see **Table 5-3**.

#### Table 5-3 Dataset parameters

Parameter	Description
Local file	Storage Path: Select an OBS path.

Parameter	Description
Schema	Names and types of table columns, which must be the same as those of the imported data. Set the column name based on the imported data and select the column type. For details about the supported types, see <b>Table 5-4</b> .
	Click <b>Add Schema</b> to add a new record. When creating a dataset, you must specify a schema. Once created, the schema cannot be modified.
	When data is imported from OBS, the schema of the CSV file in the file path is automatically obtained. If the schemas of multiple CSV files are inconsistent, an error will be reported.
	<b>NOTE</b> After you select data from OBS, column names in <b>Schema</b> are automatically displayed, which is the first-row data of the table by default. To ensure the correct prediction code, you need to change column names in <b>Schema</b> to <b>attr_1</b> , <b>attr_2</b> ,, and <b>attr_n</b> . <b>attr_n</b> is the last column, indicating the prediction column.
Output Dataset Path	OBS path for storing table data. The data imported from the data source is stored in this path. The path cannot be the same as the file path in the OBS data source or subdirectories of the file path.
	After a table dataset is created, the following four directories are automatically generated in the storage path:
	• <b>annotation</b> : version publishing directory. Each time a version is published, a subdirectory with the same name as the version is generated in this directory.
	• <b>data</b> : data storage directory. Imported data is stored in this directory.
	logs: directory for storing logs.
	• <b>temp</b> : temporary working directory.

Table 5-4 Schema data types

Туре	Description	Stora ge Space	Range
String	String type	N/A	N/A
Short	Signed integer	2 bytes	-32768 to 32767
Int	Signed integer	4 bytes	-2147483648 to 2147483647

Туре	Description	Stora ge Space	Range
Long	Signed integer	8 bytes	-9223372036854775808 to 9223372036854775807
Double	Double-precision floating point	8 bytes	N/A
Float	Single-precision floating point	4 bytes	N/A
Byte	Signed integer	1 byte	-128 to 127
Date	Date type in the format of "yyyy-MM-dd", for example, 2014-05-29	N/A	N/A
Timesta mp	Timestamp that represents date and time in the format of "yyyy-MM-dd HH:mm:ss"	N/A	N/A
Boolean	Boolean type	1 byte	TRUE/FALSE

### D NOTE

When using a CSV file, pay attention to the following:

- When the data type is set to **String**, the data in the double quotation marks is regarded as one record by default. Ensure the double quotation marks in the same row are closed. Otherwise, the data will be too large to display.
- If the number of columns in a row of the CSV file is different from that defined in the schema, the row will be ignored.
- 3. After setting the parameters, click Submit.

# 5.4.3 Modifying a Dataset

The basic information of a created dataset can be modified to keep pace with service changes.

## Prerequisites

A created dataset is available.

## Modifying the Basic Information of a Dataset

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- 2. In the dataset list, choose **More** > **Modify** in the **Operation** column of the target dataset.

×

3. Modify the basic information by referring to **Table 5-5** and click **OK**.

### Figure 5-19 Modify Dataset

# 

### Table 5-5 Parameters

Parameter	Description
Name	Name of a dataset, which must be 1 to 64 characters long and start with a letter. Only letters, digits, underscores (_), and hyphens (-) are allowed. The name must start with a letter.
Description	Brief description of the dataset.

# 5.5 Importing Data

# 5.5.1 Introduction to Data Importing

After a dataset is created, you can import more data. ModelArts allows you to import data from different data sources.

- Importing Data from OBS
- Importing Data from Local Files

ModelArts AI Gallery provides a large number of built-in datasets, including file and table datasets. You can download and use the built-in datasets from AI Gallery. You can also import your data to ModelArts.

### **File Data Sources**

You can import data by downloading built-in datasets from AI Gallery, or from OBS or a local file. After the import, the data from the import path is automatically synchronized to the data source path of the dataset.

• **OBS**: Import data from an OBS path or a manifest file.

• Local file: Import local data that has been uploaded to an OBS path.

### **Table Data Sources**

You can import data by downloading built-in datasets from AI Gallery, or from OBS, DWS, DLI, MRS, and local files.

### **Import Mode**

There are five modes for importing data to a dataset.

• When you create a dataset, select an import path. The data is automatically synchronized from the import path.

Figure 5-20 Importing data when creating a dataset

* Data Source	OBS	AI Gallery	Local file		
★ Import Mode	Path				
	You can save the	dataset file to be ir	mported to the OBS	5 path that you have p	ermission to access. Labeling file forma
* Import Path	Select an OBS	path.		ð	
	You can import u	p to 1000000 samp	les and 10000 labe	ls.	
the Laboling Ctatur	Uplabeled	Labalad			
× Labeling status	Unlabeled	Labeled			

• After a dataset is created, click **Import** in the **Operation** column on the dataset list page.

Figure 5-21 Importing data on the dataset list page



• On the dataset list page, click a dataset. On the dataset details page, choose **Import** > **Import**.

Figure 5-22 Importing data on the dataset details page

Publ	ish 🔻	Import 🔺	Export 🔻	С
	Impo	rt		
	View	Task History		

• On the dataset list page, click a dataset. On the dataset details page, click **Synchronize Data Source** to synchronize data from OBS.



• Add data on the labeling job details page.

### Figure 5-24 Adding data on the labeling job details page



# 5.5.2 Importing Data from OBS

# 5.5.2.1 Introduction to Importing Data from OBS

## **Import Modes**

You can import data from OBS through an OBS path or a manifest file.

- OBS path: indicates that the dataset to be imported has been stored in an OBS path. In this case, select an OBS path that you can access. In addition, the directory structure in the OBS path must comply with the specifications. For details, see Specifications for Importing Data from an OBS Directory. This import mode is available only for the following types of datasets: Image classification, Object detection, Text classification, Table, and Sound classification. For other types of datasets, data can be imported only through a manifest file.
- Manifest file: indicates that the dataset file is in the manifest format and the manifest file has been uploaded to OBS. The manifest file defines the mapping between labeling objects and content. For details about the specifications of manifest files, see Specifications for Importing a Manifest File.

### D NOTE

Before importing an object detection dataset, ensure that the labeling range of the labeling file does not exceed the size of the original image. Otherwise, the import may fail.

Table 5-6	Import	modes	supported	by	datasets
-----------	--------	-------	-----------	----	----------

Dat aset Typ e	Labeling Type	From an OBS Path	From a Manifest File
lma ges	lmage classificati on	Supported You can import unlabeled or labeled data. Format specifications of labeled data: Image Classification	Supported You can import unlabeled or labeled data. Format specifications of labeled data: Image Classification
	Object detection	Supported You can import unlabeled or labeled data. Format specifications of labeled data: <b>Object</b> <b>Detection</b>	Supported You can import unlabeled or labeled data. Format specifications of labeled data: <b>Object</b> <b>Detection</b>
	lmage segmenta tion	Supported You can import unlabeled or labeled data. Format specifications of labeled data: Image Segmentation	Supported You can import unlabeled or labeled data. Format specifications of labeled data: Image Segmentation
Aud io	Sound classificati on	Supported You can import unlabeled or labeled data. Follow the format specifications described in Sound Classification.	Supported You can import unlabeled or labeled data. Format specifications of labeled data: Sound Classification
	Speech labeling	Supported You can import unlabeled data.	Supported You can import unlabeled or labeled data. Format specifications of labeled data: <b>Speech</b> <b>Labeling</b>
	Speech paragrap h labeling	Supported You can import unlabeled data.	Supported You can import unlabeled or labeled data. Format specifications of labeled data: <b>Speech</b> <b>Paragraph Labeling</b>

Dat aset Typ e	Labeling Type	From an OBS Path	From a Manifest File
Text	Text	Supported	Supported
	classificati on	You can import unlabeled or labeled data.	You can import unlabeled or labeled data.
		Format specifications of labeled data: <b>Text Classification</b>	Format specifications of labeled data: <b>Text</b> Classification
	Named	Supported	Supported
	entity recognitio	You can import unlabeled data.	You can import unlabeled or labeled data.
	11		Format specifications of labeled data: <b>Named Entity</b> <b>Recognition</b>
	Text	Supported	Supported
	triplet	You can import unlabeled data.	You can import unlabeled or labeled data.
			Format specifications of labeled data: <b>Text Triplet</b>
Vide	Video	Supported	Supported
0	labeling	You can import unlabeled data.	You can import unlabeled or labeled data.
			Format specifications of labeled data: Video Labeling
Oth	Free	Supported	N/A
er	format	You can import unlabeled data.	
Tabl	Table	Supported	N/A
e		Follow the format specifications described in <b>Tables</b> .	

# 5.5.2.2 Importing Data from an OBS Path

# Prerequisites

- A dataset is available.
- The data to be imported is stored in OBS. The manifest file is stored in OBS.
- The OBS bucket and ModelArts are in the same region and you can operate the bucket.

# Importing File Data from an OBS Path

The parameters on the GUI for data import vary according to the dataset type. The following uses a dataset of the image classification type as an example.

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- 2. Locate the row that contains the desired dataset and click **Import** in the **Operation** column. Alternatively, click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Import** in the upper right corner.
- 3. In the Import dialog box, configure parameters as follows and click OK.
  - Data Source: OBS

Import

- Import Mode: Path
- Import Path: OBS path for storing data
- Labeling Status: Labeled
- Advanced Feature Settings: disabled by default

**Import by Tag** enables the system to automatically obtain the labels of the current dataset. Click **Add Label** to add a label. This parameter is optional. If **Import by Tag** is disabled, you can add or delete labels for imported data when labeling data.

Figure 5-25 Importing data from an OBS path

k Import Mode	Path manifest	
	You can save the dataset file to be im permission to access. Labeling file for	ported to the OBS path that you hav mat
k Import Path	Select an OBS path.	i
k Labeling Status	Unlabeled Labeled	
* Labeling Format	Object detection 🔹	ModelArts PASCAL VOC 1.0
	The labeled object and labeled file m their names must be the same.Labele	ust be stored in the same directory, a ed files must be in the PASCAL_VOC fo
	A ALLES A PROPERTY AND A PROPERTY AN	
	Ataset-import-example	
	dataset-import-example	\$732.jpg
	dataset-import-example     IMG_20180919_114     IMG_20180919_114	4732.jpg 1732.xml
	dataset-import-example     IMG_20180919_114     MG_20180919_114     IMG_20180919_114     IMG_20180919_114	4732.jpg 1732.xml 4745.jpg
	dataset-import-example     IMG_20180919_114     MG_20180919_114     IMG_20180919_114     IMG_20180919_114	1732.jpg 1732.xml 1745.jpg

After the data is imported, it will be automatically synchronized to the dataset. On the **Datasets** page, click the dataset name to view its details and create a labeling job to label the data.

## Labeling Status of File Data

The labeling status can be Unlabeled or Labeled.

- **Unlabeled**: Only the labeling object (such as unlabeled images or texts) is imported.
- **Labeled**: Both the labeling object and content are imported. Labeling content importing is not supported for datasets in free format.

To ensure that the labeling content can be correctly read, you must store data in strict accordance with the specifications.

If **Import Mode** is set to **Path**, store the data to be imported according to the labeling file specifications. For details, see **Specifications for Importing Data from an OBS Directory**.

If **Import Mode** is set to **manifest**, the manifest file specifications must be met.

**NOTE** 

- If the labeling status is set to **Labeled**, ensure that the folder or manifest file complies with the format specifications. Otherwise, the import may fail.
- After the import of labeled data, check whether the imported data is in the labeled state.

### Importing a Table Dataset from OBS

ModelArts allows you to import table data (CSV files) from OBS.

Import description:

- The prerequisite for successful import is that the schema of the data source must be the same as that specified during dataset creation. The schema indicates column names and types of a table. Once specified during dataset creation, the values cannot be changed.
- When a CSV file is imported from OBS, the data type is not validated, but the number of columns must be the same as that in the schema of the dataset. If the data format is invalid, the data is set to null. For details, see Table 5-4.
- You must select the directory where the CSV file is stored. The number of columns in the CSV file must be the same as that in the dataset schema. The schema of the CSV file can be automatically obtained.

Data Source ⑦ OBS DWS DLI MRS  * File Path Contain Table Header  * Schema ⑦ Column Name Type String  Add Schema	Create Dataset (1) Basic Information	Back to Dataset List     Onput Data     (3) Finish	
<ul> <li>* File Path Select an OBS path.</li> <li>* Schema</li> <li>Column Name Type String •</li> <li>Add Schema</li> </ul>	Data Source 🕥	OBS DWS DLI MRS	
* Schema (2) Column Name Type String   Add Schema		* File Path Contain Table Header	ð
	* Schema 🕐	Column Name Type String	

# 5.5.2.3 Specifications for Importing Data from an OBS Directory

When importing data from OBS, the data storage directory and file name must comply with the ModelArts specifications.

Only the following labeling types of data can be imported by **Labeling Format**: image classification, object detection, image segmentation, text classification, and sound classification.

### **NOTE**

- To import data from an OBS directory, you must have the read permission on the OBS directory.
- The OBS buckets and ModelArts must be in the same region.

### **Image Classification**

Data for image classification can be stored in two formats:

Format 1: ModelArts imageNet 1.0

 Images with the same label must be stored in the same directory, with the label name as the directory name. If there are multiple levels of directories, the last level is used as the label name.

In the following example, **Cat** and **Dog** are label names.

```
dataset-import-example
Cat
10.jpg
11.jpg
12.jpg
1.jpg
2.jpg
3.jpg
```

Format 2: ModelArts image classification 1.0

• The image and labeled file must be stored in the same directory, with the content in the labeled file used as label names.

In the following example, **import-dir-1** and **import-dir-2** are the imported subdirectories:

```
dataset-import-example
```

The following shows a label file for a single label, for example, the **1.txt** file: Cat

The following shows a label file for multiple labels, for example, the **2.txt** file: Cat Dog

• Only images in JPG, JPEG, PNG, and BMP formats are supported. The size of a single image cannot exceed 5 MB, and the total size of all images uploaded at a time cannot exceed 8 MB.

### **Object Detection**

Data for object detection can be stored in two formats:

Format 1: ModelArts PASCAL VOC 1.0

 The simple mode of object detection requires you to store labeled objects and your label files (in one-to-one relationship with the labeled objects) in the same directory. For example, if the name of the labeled object file is IMG\_20180919\_114745.jpg, the name of the label file must be IMG\_20180919\_114745.xml.

The label files must be in PASCAL VOC format. For details about the format, see **Table 5-14**.

Example:

```
-dataset-import-example
IMG_20180919_114732.jpg
IMG_20180919_114732.xml
IMG_20180919_114745.jpg
IMG_20180919_114745.xml
IMG_20180919_114945.jpg
IMG_20180919_114945.xml
```

A label file example is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
  <folder>NA</folder>
  <filename>bike_1_1593531469339.png</filename>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>554</width>
    <height>606</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
     <name>Dog</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
```

```
<occluded>0</occluded>
    <br/>
<br/>
hdbox>
       <xmin>279</xmin>
       <ymin>52</ymin>
       <xmax>474</xmax>
       <ymax>278</ymax>
    </bndbox>
  </object>
  <object>
    <name>Cat</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
       <xmin>279</xmin>
       <ymin>198</ymin>
       <xmax>456</xmax>
       <ymax>421</ymax>
    </bndbox>
  </object>
</annotation>
```

• Only images in JPG, JPEG, PNG, and BMP formats are supported. A single image cannot exceed 5 MB, and the total size of all images uploaded at a time cannot exceed 8 MB.

Format 2: YOLO

• A YOLO dataset must comply with the following structure:

	1,5,5,5,2,5,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7
	<ul> <li>obj.names # Label set file</li> <li>obj.data # Files and relative paths for recording dataset information</li> <li>train.txt # Relative path of images in the training set</li> <li>valid.txt # Relative path of images in the validation set</li> </ul>
	bi train data/ # Directory where the images in the training set and the corresponding label
file	es are stored
	image1.txt # BBox label list for image 1
	image1.jpg
	image2.txt
	image2.jpg
	│
	bj_valid_data/ # Directory where the images in the validation set and the corresponding
lat	pel files are stored
	image101.txt
	image101.jpg
	image102.txt
	image102.jpg

A YOLO dataset supports only training sets and validation sets. If other sets are imported, they will be invalid in the YOLO dataset.

obj.data contains the following content and at least one of the train and valid subsets must be contained. The file paths are relative paths.
 classes = 5 # Optional
 names = <path/to/obj.names># For example, obj.names
 train = <path/to/train.txt># For example, train.txt

```
valid = <path/to/valid.txt># Optional, for example, valid.txt
```

backup = backup/ # Optional

| <u>|</u> ...

• The **obj.names** file records the label list. Each row label is used as the file index.

```
label1 # index of label 1: 0
label2 # index of label 2: 1
label3
```

- The file paths in train.txt and valid.txt are relative paths, and the file list must be in one-to-one relationship with the files in the directories. The file structures of the two files are as follows:
   <path/to/image1.jpg># For example, obj\_train\_data/image.jpg
   <path/to/image2.jpg># For example, obj\_train\_data/image.jpg
- The .txt files in the obj\_train\_data/ and obj\_valid\_data/ directories contain the BBox label information of the corresponding images. Each line indicates a BBox label.
   # image1.txt:
   # <label\_index> <x\_center> <y\_center> <width> <height>

```
# <label_index> <x_center> <y_center> <width> <height>
0 0.250000 0.400000 0.300000 0.400000
3 0.600000 0.400000 0.266667
```

**x\_center**, **y\_center**, **width**, and **height** indicate the normalized parameters for the target bounding box: the x-coordinate and y-coordinate of the center point, width, and height.

 Only images in JPG, JPEG, PNG, and BMP formats are supported. A single image cannot exceed 5 MB, and the total size of all images uploaded at a time cannot exceed 8 MB.

### Image Segmentation

ModelArts image segmentation 1.0:

• Labeled objects and their label files (in one-to-one relationship with the labeled objects) must be in the same directory. For example, if the name of the labeled object file is IMG\_20180919\_114746.jpg, the name of the label file must be IMG\_20180919\_114746.xml.

Fields **mask\_source** and **mask\_color** are added to the label file in PASCAL VOC format. For details about the format, see **Table 5-10**.

Example:

```
-dataset-import-example
IMG_20180919_114732.jpg
IMG_20180919_114732.xml
IMG_20180919_114745.jpg
IMG_20180919_114745.xml
IMG_20180919_114945.jpg
IMG_20180919_114945.xml
```

A label file example is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
  <folder>NA</folder>
  <filename>image_0006.jpg</filename>
  <source>
    <database>Unknown</database>
  </source>
  <size>
     <width>230</width>
     <height>300</height>
    <depth>3</depth>
  </size>
  <segmented>1</segmented>
  <mask_source>obs://xianao/out/dataset-8153-Jmf5ylLjRmSacj9KevS/annotation/V001/
segmentationClassRaw/image_0006.png</mask_source>
  <object>
     <name>bike</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
```

```
<mask_color>193,243,53</mask_color>
    <occluded>0</occluded>
     <polygon>
       <x1>71</x1>
       <y1>48</y1>
       <x2>75</x2>
       <y2>73</y2>
       <x3>49</x3>
       <y3>69</y3>
       <x4>68</x4>
       <y4>92</y4>
       <x5>90</x5>
       <y5>101</y5>
       <x6>45</x6>
       <y6>110</y6>
       <x7>71</x7>
       <y7>48</y7>
     </polygon>
  </object>
</annotation>
```

## **Text Classification**

txt and csv files can be imported for text classification, with the text encoding format of UTF-8 or GBK.

Labeled objects and labels for text classification can be stored in two formats:

 ModelArts text classification combine 1.0: The labeled objects and labels for text classification are in the same text file. You can specify a separator to separate the labeled objects and labels, as well as multiple labels.

For example, the following shows an example text file. The **Tab** key is used to separate the labeled objects from the labels.

It touches good and responds quickly. I don't know how it performs in the future. positive Three months ago, I bought a very good phone and replaced my old one with it. It can operate longer between charges. positive

Why does my phone heat up if I charge it for a while? The volume button stuck after being pressed down. negative

It's a gift for Father's Day. The delivery is fast and I received it in 24 hours. I like the earphones because the bass sounds feel good and they would not fall off. positive

 ModelArts text classification 1.0: The labeled objects and labels for text classification are text files, and correspond to each other based on the rows. For example, the first row in a label file indicates the label of the first row in the file of the labeled object.

For example, the content of the labeled object **COMMENTS\_20180919\_114745.txt** is as follows:

It touches good and responds quickly. I don't know how it performs in the future. Three months ago, I bought a very good phone and replaced my old one with it. It can operate longer between charges.

Why does my phone heat up if I charge it for a while? The volume button stuck after being pressed down.

It's a gift for Father's Day. The delivery is fast and I received it in 24 hours. I like the earphones because the bass sounds feel good and they would not fall off.

The content of the label file **COMMENTS\_20180919\_114745\_result.txt** is as follows:

positive negative negative positive

This data format requires you to store labeled objects and your label files (in one-to-one relationship with the labeled objects) in the same directory. For

#### example, if the name of the labeled object file is COMMENTS\_20180919\_114745.txt, the name of the label file must be COMMENTS \_20180919\_114745\_result.txt.

Example of data files:

-dataset-import-example
COMMENTS_20180919_114732.txt
COMMENTS _20180919_114732_result.txt
COMMENTS _20180919_114745.txt
COMMENTS _20180919_114745_result.txt
COMMENTS 20180919 114945.txt
COMMENTS _20180919_114945_result.txt

# **Sound Classification**

ModelArts audio classification dir 1.0: Sound files with the same label must be stored in the same directory, and the label name is the directory name.

#### Example:



### Tables

CSV files can be imported from OBS. Select the directory where the files are stored. The number of columns in the CSV file must be the same as that in the dataset schema. The schema of the CSV file can be automatically obtained.

-dataset-import-example table\_import\_1.csv table\_import\_2.csv table\_import\_3.csv table\_import\_4.csv

# 5.5.2.4 Importing a Manifest File

### Prerequisites

- You have created a dataset.
- You have stored the data to be imported in OBS. You have stored the manifest file in OBS.
- The OBS bucket and ModelArts are in the same region and you can operate the bucket.

## Importing File Data from a Manifest File

The parameters for data import vary according to the dataset type. The following uses an image dataset as an example.

1. Log in to the . In the navigation pane, choose **Data Management** > **Datasets**.

- 2. Locate the row that contains the desired dataset and click **Import** in the **Operation** column. Alternatively, you can click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Import** in the upper right corner.
- 3. In the Import dialog box, set the parameters as follows and click OK.
  - Data Source: OBS
  - Import Mode: manifest
  - Manifest File: OBS path for storing the manifest file
  - Labeling Status: Labeled
  - Advanced Feature Settings: disabled by default

**Import by Tag** The system automatically obtains the labels of the dataset. You can click **Add Label** to add a label. This parameter is optional. If **Import by Tag** is disabled, you can add or delete labels for imported data when labeling data.

**Import Only Hard Examples**: If this parameter is selected, only the **hard** attribute data of the manifest file is imported.

Figure 5-26 Importing a manifest file

Import



After the data is imported, it will be automatically synchronized to the dataset. On the **Datasets** page, click the dataset name to view its details and create a labeling job to label the data.

### Labeling Status of File Data

The labeling status can be Unlabeled or Labeled.

• **Unlabeled**: Only the labeling object (such as unlabeled images or texts) is imported.

• **Labeled**: Both the labeling object and content are imported. Labeling content importing is not supported for datasets in free format.

To ensure that the labeling content can be correctly read, you must store data in strict accordance with the specifications.

If **Import Mode** is set to **Path**, store the data to be imported according to the labeling file specifications.

If **Import Mode** is set to **manifest**, the manifest file specifications must be met. For details, see **Specifications for Importing a Manifest File**.

**NOTE** 

If the labeling status is set to **Labeled**, ensure that the folder or manifest file complies with the format specifications. Otherwise, the import may fail.

### 5.5.2.5 Specifications for Importing a Manifest File

The manifest file defines the mapping between labeled objects and content. The manifest file import mode means that the manifest file is used for dataset import. The manifest file can be imported from OBS. When importing a manifest file from OBS, ensure that you have the permissions to access the directory where the manifest file is stored.

#### **NOTE**

There are many requirements on the manifest file compilation. Import new data from OBS. Generally, manifest file import is used for data migration of ModelArts in different regions or using different accounts. If you have labeled data in a region using ModelArts, you can obtain the manifest file of the published dataset from the output path. Then you can import the dataset using the manifest file to ModelArts of other regions or accounts. The imported data carries the labeling information and does not need to be labeled again, improving development efficiency.

The manifest file that contains information about the original file and labeling can be used in labeling, training, and inference scenarios. The manifest file that contains only information about the original file can be used in inference scenarios or used to generate an unlabeled dataset. The manifest file must meet the following requirements:

- The manifest file uses the UTF-8 encoding format.
- The manifest file uses the JSON Lines format (jsonlines.org). A line contains one JSON object.

```
{"source": "/path/to/image1.jpg", "annotation": ... }
{"source": "/path/to/image2.jpg", "annotation": ... }
{"source": "/path/to/image3.jpg", "annotation": ... }
```

In the preceding example, the manifest file contains multiple lines of JSON object.

 The manifest file can be generated by you, third-party tools, or ModelArts Data Labeling. The file name can be any valid file name. To facilitate the internal use of the ModelArts system, the file name generated by the ModelArts data labeling function consists of the following strings: DatasetName-VersionName.manifest. For example, animalv201901231130304123.manifest.

# **Image Classification**

{

}

```
"source":"s3://path/to/image1.jpg",
"usage":"TRAIN",
"hard":"true",
"hard-coefficient":0.8,
"id":"0162005993f8065ef47eefb59d1e4970",
"annotation": [
   {
       "type": "modelarts/image_classification",
"name": "cat",
"property": {
"color":"white",
"kind":"Persian cat"
       },
"hard":"true",
       "hard-coefficient":0.8,
       "annotated-by":"human",
"creation-time":"2019-01-23 11:30:30"
   },
{
       "type": "modelarts/image_classification",
"name":"animal",
       "annotated-by":"modelarts/active-learning",
       "confidence": 0.8,
       "creation-time":"2019-01-23 11:30:30"
   }],
"inference-loc":"/path/to/inference-output"
```

### Table 5-7 Parameters

Parameter	Manda tory	Description	
source	Yes	URI of an object to be labeled. For details about data source types and examples, see <b>Table 5-8</b> .	
usage	No	<ul> <li>By default, the parameter value is left blank. Possible values are as follows:</li> <li>TRAIN: The object is used for training.</li> <li>EVAL: The object is used for evaluation.</li> <li>TEST: The object is used for testing.</li> <li>INFERENCE: The object is used for inference.</li> <li>If the parameter value is left blank, you decide how to use the object.</li> </ul>	
id	No	Sample ID exported from the system. You do not need to set this parameter when importing the sample.	
annotation	No	If the parameter value is left blank, the object is not labeled. The value of <b>annotation</b> consists of an object list. For details about the parameters, see <b>Table 5-9</b> .	
inference-loc	No	This parameter is available when the file is generated by the inference service, indicating the location of the inference result file.	

## Table 5-8 Data source types

Туре	Example
OBS	"source":"s3://path-to-jpg"
Content	"source":"content://I love machine learning"

## Table 5-9 annotation objects

Parameter	Mandat ory	Description	
type	Yes	Label type. Possible values are as follows: • image_classification: image classification • text_classification: text classification • text_entity: named entity recognition • object_detection: object detection • audio_classification: sound classification • audio_content: speech labeling • audio_segmentation: speech paragraph labeling	
name	Yes/No	This parameter is mandatory for the classification type but optional for other types. This example uses the image classification type.	
id	Yes/No	Label ID. This parameter is mandatory for triplets but optional for other types. The entity label ID of a triplet is in <b>E+number</b> format, for example, <b>E1</b> and <b>E2</b> . The relationship label ID of a triplet is in <b>R</b> <b>+number</b> format, for example, <b>R1</b> and <b>R2</b> .	
property	No	Labeling property. In this example, the cat has two properties: color and kind.	
hard	No	Indicates whether the example is a hard example. <b>True</b> indicates that the labeling example is a hard example, and <b>False</b> indicates that the labeling example is not a hard example.	
annotated-by	No	The default value is <b>human</b> , indicating manual labeling. • human	
creation-time	No	Time when the labeling job was created. It is the time when labeling information was written, not the time when the manifest file was generated.	

Parameter	Mandat ory	Description
confidence	No	Confidence score of machine labeling. The value ranges from 0 to 1.

# Image Segmentation

```
{
    "annotation": [{
        "annotation-format": "PASCAL VOC",
        "type": "modelarts/image_segmentation",
        "annotation-loc": "s3://path/to/annotation/image1.xml",
        "creation-time": "2020-12-16 21:36:27",
        "annotated-by": "human"
    ]],
    "usage": "train",
    "source": "s3://path/to/image1.jpg",
    "id": "16d196c19bf61994d7deccafa435398c",
    "sample-type": 0
}
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in **Image Classification**. For details, see **Table 5-7**.
- **annotation-loc** indicates the path for saving the label file. This parameter is mandatory for image segmentation and object detection but optional for other labeling types.
- **annotation-format** indicates the format of the label file. This parameter is optional. The default value is **PASCAL VOC**. Only **PASCAL VOC** is supported.
- **sample-type** indicates a sample format. Value **0** indicates image, **1** text, **2** audio, **4** table, and **6** video.

Parameter	Mand atory	Description	
folder	Yes	Directory where the data source is located	
filename	Yes	Name of the file to be labeled	
size	Yes	<ul> <li>Image pixel</li> <li>width: image width. This parameter is mandatory.</li> <li>height: image height. This parameter is mandatory.</li> <li>depth: number of image channels. This parameter is mandatory.</li> </ul>	
segmented	Yes	Segmented or not	
mask_source	No	Segmentation mask path	

Table 5-10 PASCAL VOC format parameters

Parameter	Mand atory	Description	
object	Yes	Object detection information. Multiple <b>object{}</b> functions are generated for multiple objects.	
		• <b>name</b> : type of the labeled content. This parameter is mandatory.	
		<ul> <li>pose: shooting angle of the labeled content. This parameter is mandatory.</li> </ul>	
		<ul> <li>truncated: whether the labeled content is truncated (0 indicates that the content is not truncated). This parameter is mandatory.</li> </ul>	
		• <b>occluded</b> : whether the labeled content is occluded ( <b>0</b> indicates that the content is not occluded). This parameter is mandatory.	
		• <b>difficult</b> : whether the labeled object is difficult to identify ( <b>0</b> indicates that the object is easy to identify). This parameter is mandatory.	
		• <b>confidence</b> : confidence score of the labeled object. The value ranges from 0 to 1. This parameter is optional.	
		<ul> <li>bndbox: bounding box type. This parameter is mandatory. For details about the possible values, see Table 5-11.</li> </ul>	
		<ul> <li>mask_color: label color, which is represented by the RGB value. This parameter is mandatory.</li> </ul>	

Parameter	Shape	Labeling information
polygon	Polygon	Coordinates of points
		<x1>100<x1></x1></x1>
		<y1>100<y1></y1></y1>
		<x2>200<x2></x2></x2>
		<y2>100<y2></y2></y2>
		<x3>250<x3></x3></x3>
		<y3>150<y3></y3></y3>
		<x4>200<x4></x4></x4>
		<y4>200<y4></y4></y4>
		<x5>100<x5></x5></x5>
		<y5>200<y5></y5></y5>
		<x6>50<x6></x6></x6>
		<y6>150<y6></y6></y6>
		<x7>100<x7></x7></x7>
		<y7>100<y7></y7></y7>

**Table 5-11** Bounding box types

#### Example:

<?xml version="1.0" encoding="UTF-8" standalone="no"?> <annotation> <folder>NA</folder> <filename>image\_0006.jpg</filename> <source> <database>Unknown</database> </source> <size> <width>230</width> <height>300</height> <depth>3</depth> </size> <segmented>1</segmented> <mask\_source>obs://xianao/out/dataset-8153-Jmf5ylLjRmSacj9KevS/annotation/V001/
segmentationClassRaw/image\_0006.png</mask\_source> <object> <name>bike</name> <pose>Unspecified</pose> <truncated>0</truncated> <difficult>0</difficult> <mask\_color>193,243,53</mask\_color> <occluded>0</occluded> <polygon> <x1>71</x1> <y1>48</y1> <x2>75</x2> <y2>73</y2> <x3>49</x3> <y3>69</y3> <x4>68</x4> <y4>92</y4> <x5>90</x5> <y5>101</y5> <x6>45</x6> <y6>110</y6>

```
<x7>71</x7>
<y7>48</y7>
</polygon>
</object>
</annotation>
```

# **Text Classification**

```
{
    "source": "content://I like this product ",
    "id":"XGDVGS",
    "annotation": [
    {
        "type": "modelarts/text_classification",
        "name": " positive",
        "annotated-by": "human",
        "creation-time": "2019-01-23 11:30:30"
    }]
}
```

The **content** parameter indicates the text to be labeled. The other parameters are the same as those described in **Image Classification**. For details, see **Table 5-7**.

# Named Entity Recognition

```
"source":"content://Michael Jordan is the most famous basketball player in the world.",
  "usage":"TRAIN",
  "annotation":[
     {
        "type":"modelarts/text_entity",
        "name":"Person",
        "property":{
           "@modelarts:start_index":0,
           "@modelarts:end_index":14
       },
        "annotated-by":"human",
        "creation-time":"2019-01-23 11:30:30"
     },
{
        "type":"modelarts/text_entity",
        "name":"Category",
        "property":{
          "@modelarts:start_index":34,
           "@modelarts:end_index":44
        },
        "annotated-by":"human",
        "creation-time":"2019-01-23 11:30:30"
     }
  ]
}
```

The parameters such as **source**, **usage**, and **annotation** are the same as those described in **Image Classification**. For details, see **Table 5-7**.

Table 5-12 describes the property parameters. For example, if you want to extract Michael from "source":"content://Michael Jordan", the value of start\_index is 0 and that of end\_index is 7.

Table 5-12	property	parameters
------------	----------	------------

Parameter	Data type	Description
@modelarts:start_in dex	Integer	Start position of the text. The value starts from 0, including the characters specified by <b>start_index</b> .
@modelarts:end_ind ex	Integer	End position of the text, excluding the characters specified by <b>end_index</b> .

# Text Triplet

{ "sou	urce":"content://"Three Body" is a series of long science fiction novels created by Liu Cix."
"anr	notation":[
{	
	"type":"modelarts/text_entity", "name":"Person", "id":"E1"
	"property":{ "@modelarts:start_index":67, "@modelarts:ord_index":74
	<pre>winddetaits.end_index ./4 },</pre>
	"annotated-by":"human", "creation-time":"2019-01-23 11:30:30"
},	
{	
	"type":"modelarts/text_entity", "name":"Book", "' universe
	"Id":"E2", "property":{
	"@modelarts:start_index":0, "@modelarts:end_index":12
	},
	"creation-time":"2019-01-23 11:30:30"
},	
ť	"type":"modelarts/text_triplet".
	"name":"Author",
	"id":"R1", "property":{
	"@modelarts:from":"E1", "@modelarts:to":"E2"
	},
	"annotated-by":"human", "creation-time":"2019-01-23 11:30:30"
}, ſ	
ı	"type":"modelarts/text_triplet", "name":"Works",
	"id":"R2",
	roperty :{ "@modelarts:from":"E2", "@modelarts:to":"E1"
	},
	"annotated-by":"human", "creation-time":"2019-01-23 11:30:30"
}	
}	

The parameters such as **source**, **usage**, and **annotation** are the same as those described in **Image Classification**. For details, see **Table 5-7**.

Table 5 property parameters describes the property parameters. @modelarts:start\_index and @modelarts:end\_index are the same as those of named entity recognition. For example, when source is set to content://"Three Body" is a series of long science fiction novels created by Liu Cix., Liu Cix is an entity person, Three Body is an entity book, the person is the author of the book, and the book is works of the person.

Parameter	Data type	Description
@modelarts:start_in dex	Integer	Start position of the triplet entities. The value starts from 0, including the characters specified by <b>start_index</b> .
@modelarts:end_ind ex	Integer	End position of the triplet entities, excluding the characters specified by <b>end_index</b> .
@modelarts:from	String	Start entity ID of the triplet relationship
@modelarts:to	String	Entity ID pointed to in the triplet relationship

Table 5-13	property	parameters
------------	----------	------------

## **Object Detection**

```
{
    "source":"s3://path/to/image1.jpg",
    "usage":"TRAIN",
    "hard":"true",
    "hard-coefficient":0.8,
    "annotation": [
        {
            "type":"modelarts/object_detection",
            "annotation-loc": "s3://path/to/annotation1.xml",
            "annotation-format":"PASCAL VOC",
            "annotated-by":"human",
            "creation-time":"2019-01-23 11:30:30"
        }]
}
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in **Image Classification**. For details, see **Table 5-7**.
- annotation-loc indicates the path for saving the label file. This parameter is mandatory for object detection and image segmentation but optional for other labeling types.
- annotation-format indicates the format of the label file. This parameter is optional. The default value is PASCAL VOC. Only PASCAL VOC is supported.

Parameter	Mand atory	Description	
folder	Yes	Directory where the data source is located	
filename	Yes	Name of the file to be labeled	
size	Yes	<ul> <li>Image pixel</li> <li>width: image width. This parameter is mandatory.</li> <li>height: image height. This parameter is mandatory.</li> <li>depth: number of image channels. This parameter is mandatory.</li> </ul>	
segmented	Yes	Segmented or not	
object	Yes	<ul> <li>Object detection information. Multiple object{} functions are generated for multiple objects.</li> <li>name: type of the labeled content. This parameter is mandatory.</li> <li>pose: shooting angle of the labeled content. This parameter is mandatory.</li> <li>truncated: whether the labeled content is truncated (0 indicates that the content is not truncated). This parameter is mandatory.</li> <li>occluded: whether the labeled content is occluded (0 indicates that the content is not occluded). This parameter is mandatory.</li> <li>difficult: whether the labeled object is difficult to identify (0 indicates that the object is easy to identify). This parameter is mandatory.</li> <li>confidence: confidence score of the labeled object. The value ranges from 0 to 1. This parameter is optional.</li> <li>bndbox: bounding box type. This parameter is mandatory. For details about the possible values, see Table 5-15.</li> </ul>	

### Table 5-14 PASCAL VOC format parameters

### Table 5-15 Bounding box types

Parameter	Shape	Labeling information
point	Point	Coordinates of a point
		<x>100<x></x></x>
		<y>100<y></y></y>

Parameter	Shape	Labeling information
line	Line	Coordinates of points <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2></y2></y2></x2></x2></y1></y1></x1></x1>
bndbox	Rectangle	Coordinates of the upper left and lower right points <xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymax>200<ymax></ymax></ymax></xmax></xmax></ymin></ymin></xmin></xmin>
polygon	Polygon	Coordinates of points <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<x4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6></y6></y6></x6></x6></y5></y5></x5></x5></x4></y4></x4></x4></y3></y3></x3></x3></y2></y2></x2></x2></y1></y1></x1></x1>
circle	Circle	Center coordinates and radius <cx>100<cx> <cy>100<cy> <r>50<r></r></r></cy></cy></cx></cx>

## Example:

<annotation>
<folder>test\_data</folder>
<filename>260730932.jpg</filename>
<size>
<width>767</width>
<height>959</height>
<depth>3</depth>
</size>
<segmented>0</segmented>
<object>
<name>point</name>
<pose>Unspecified</pose>
<truncated>0</truncated>

<occluded>0</occluded> <difficult>0</difficult> <point> <x1>456</x1> <y1>596</y1> </point> </object> <object> <name>line</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <line> <x1>133</x1> <y1>651</y1> <x2>229</x2> <y2>561</y2> </line> </object> <object> <name>bag</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <bndbox> <xmin>108</xmin> <ymin>101</ymin> <xmax>251</xmax> <ymax>238</ymax> </bndbox> </object> <object> <name>boots</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <hard-coefficient>0.8</hard-coefficient> <polygon> <x1>373</x1> <y1>264</y1> <x2>500</x2> <y2>198</y2> <x3>437</x3> <y3>76</y3> <x4>310</x4> <y4>142</y4> </polygon> </object> <object> <name>circle</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <circle> <cx>405</cx> <cy>170</cy> <r>100<r> </circle> </object> </annotation>

# **Sound Classification**

{ "source":
```
"s3://path/to/pets.wav",
  "annotation": [
     {
        "type": "modelarts/audio_classification",
"name":"cat",
        "annotated-by":"human",
        "creation-time":"2019-01-23 11:30:30"
     }
  ]
```

The parameters such as source, usage, and annotation are the same as those described in Image Classification. For details, see Table 5-7.

### Speech Labeling

}



- The parameters such as source, usage, and annotation are the same as those described in Image Classification. For details, see Table 5-7.
- The **@modelarts:content** parameter in **property** indicates speech content. The data type is **String**.

### Speech Paragraph Labeling

```
"source":"s3://path/to/audio1.wav",
  "usage":"TRAIN",
  "annotation":[
     {
"type":"modelarts/audio_segmentation",
        "property":{
           "@modelarts:start_time":"00:01:10.123",
           "@modelarts:end_time":"00:01:15.456",
           "@modelarts:source":"Tom",
           "@modelarts:content":"How are you?"
        },
       "annotated-by":"human",
       "creation-time":"2019-01-23 11:30:30"
     },
     {
       "type":"modelarts/audio_segmentation",
        "property":{
           "@modelarts:start_time":"00:01:22.754",
           "@modelarts:end_time":"00:01:24.145",
           "@modelarts:source":"Jerry",
"@modelarts:content":"I'm fine, thank you."
        },
        "annotated-by":"human",
        "creation-time":"2019-01-23 11:30:30"
     }
  ]
}
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in **Image Classification**. For details, see **Table 5-7**.
- Table 5-16 describes the property parameters.

Table 5-16	property	parameters
------------	----------	------------

Parameter	Data type	Description	
@modelarts:start_ time	String	Start time of the sound. The format is <b>hh:mm:ss.SSS</b> .	
		<b>hh</b> indicates the hour, <b>mm</b> indicates the minute, <b>ss</b> indicates the second, and <b>SSS</b> indicates the millisecond.	
@modelarts:end_t ime	String	End time of the sound. The format is <b>hh:mm:ss.SSS</b> .	
		<b>hh</b> indicates the hour, <b>mm</b> indicates the minute, <b>ss</b> indicates the second, and <b>SSS</b> indicates the millisecond.	
@modelarts:sourc e	String	Sound source	
@modelarts:conte nt	String	Sound content	

# Video Labeling

```
{
    "annotation": [{
        "annotation-format": "PASCAL VOC",
        "type": "modelarts/object_detection",
        "annotation-loc": "s3://path/to/annotation1_t1.473722.xml",
        "creation-time": "2020-10-09 14:08:24",
        "annotated-by": "human"
    }],
    "usage": "train",
    "property": {
        "@modelarts:parent_duration": 8,
        "@modelarts:parent_duration": 8,
        "@modelarts:parent_source": "s3://path/to/annotation1.mp4",
        "@modelarts:time_in_video": 1.473722
    },
    "source": "s3://input/path/to/annotation1_t1.473722.jpg",
    "id": "43d88677c1e9a971eeb692a80534b5d5",
        "sample-type": 0
}
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in **Image Classification**. For details, see **Table 5-7**.
- **annotation-loc** indicates the path for saving the label file. This parameter is mandatory for object detection but optional for other labeling types.
- **annotation-format** indicates the format of the label file. This parameter is optional. The default value is **PASCAL VOC**. Only **PASCAL VOC** is supported.
- **sample-type** indicates a sample format. Value **0** indicates image, **1** text, **2** audio, **4** table, and **6** video.

Parameter	Data type	Description
@modelarts:parent_ duration	Double	Duration of the labeled video, in seconds
@modelarts:time_in _video	Double	Timestamp of the labeled video frame, in seconds
@modelarts:parent_ source	String	OBS path of the labeled video

### Table 5-17 property parameters

Table 5-18 PASCAL VOC format parameters

Parameter	Mand atory	Description
folder	Yes	Directory where the data source is located
filename	Yes	Name of the file to be labeled
size	Yes	<ul> <li>Image pixel</li> <li>width: image width. This parameter is mandatory.</li> <li>height: image height. This parameter is mandatory.</li> <li>depth: number of image channels. This parameter is mandatory.</li> </ul>
segmented	Yes	Segmented or not

Parameter	Mand atory	Description	
object	Yes	Object detection information. Multiple <b>object{}</b> functions are generated for multiple objects.	
		• <b>name</b> : type of the labeled content. This parameter is mandatory.	
		• <b>pose</b> : shooting angle of the labeled content. This parameter is mandatory.	
		<ul> <li>truncated: whether the labeled content is truncated (0 indicates that the content is not truncated). This parameter is mandatory.</li> </ul>	
		<ul> <li>occluded: whether the labeled content is occluded (0 indicates that the content is not occluded). This parameter is mandatory.</li> </ul>	
		• <b>difficult</b> : whether the labeled object is difficult to identify ( <b>0</b> indicates that the object is easy to identify). This parameter is mandatory.	
		• <b>confidence</b> : confidence score of the labeled object. The value ranges from 0 to 1. This parameter is optional.	
		<ul> <li>bndbox: bounding box type. This parameter is mandatory. For details about the possible values, see Table 5-19.</li> </ul>	

Table 5-19	Bounding	box types
------------	----------	-----------

Parameter	Shape	Labeling information
point	Point	Coordinates of a point <x>100<x> <y>100<y></y></y></x></x>
line	Line	Coordinates of points <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2></y2></y2></x2></x2></y1></y1></x1></x1>
bndbox	Rectangle	Coordinates of the upper left and lower right points <xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymax>200<ymax></ymax></ymax></xmax></xmax></ymin></ymin></xmin></xmin>

Parameter	Shape	Labeling information
polygon	Polygon	Coordinates of points
		<x1>100<x1></x1></x1>
		<y1>100<y1></y1></y1>
		<x2>200<x2></x2></x2>
		<y2>100<y2></y2></y2>
		<x3>250<x3></x3></x3>
		<y3>150<y3></y3></y3>
		<x4>200<x4></x4></x4>
		<y4>200<y4></y4></y4>
		<x5>100<x5></x5></x5>
		<y5>200<y5></y5></y5>
		<x6>50<x6></x6></x6>
		<y6>150<y6></y6></y6>
circle	Circle	Center coordinates and radius
		<cx>100<cx></cx></cx>
		<cy>100<cy></cy></cy>
		<r>50<r></r></r>

#### Example:

```
<annotation>
  <folder>test_data</folder>
  <filename>260730932_t1.473722.jpg.jpg</filename>
  <size>
    <width>767</width>
<height>959</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>point</name>
    <pose>Unspecified</pose>
<truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <point>
       <x1>456</x1>
       <y1>596</y1>
    </point>
  </object>
  <object>
    <name>line</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <line>
       <x1>133</x1>
       <y1>651</y1>
       <x2>229</x2>
       <y2>561</y2>
    </line>
  </object>
```

<object> <name>bag</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <bndbox> <xmin>108</xmin> <ymin>101</ymin> <xmax>251</xmax> <ymax>238</ymax> </bndbox> </object> <object> <name>boots</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <hard-coefficient>0.8</hard-coefficient> <polygon> <x1>373</x1> <y1>264</y1> <x2>500</x2> <y2>198</y2> <x3>437</x3> <y3>76</y3> <x4>310</x4> <y4>142</y4> </polygon> </object> <object> <name>circle</name> <pose>Unspecified</pose> <truncated>0</truncated> <occluded>0</occluded> <difficult>0</difficult> <circle> <cx>405</cx> <cy>170</cy> <r>100<r> </circle> </object> </annotation>

# 5.5.3 Importing Data from Local Files

# Prerequisites

- You have created a dataset.
- You have created an OBS bucket. The OBS bucket and ModelArts are in the same region and you can operate the bucket.

### **Import Operation**

Both file and table data can be uploaded from local files. The data uploaded from local files should be stored in an OBS directory. You must have created an OBS bucket.

In a single batch upload, a maximum of 100 files can be uploaded at a time, and the total size of the files cannot exceed 5 GB.

The parameters on the GUI for data import vary according to the dataset type. The following uses a dataset of the image classification type as an example.

Х

- 1. Log in to the . In the navigation pane, choose **Data Management** > **Datasets**.
- 2. Locate the row that contains the desired dataset and click **Import** in the **Operation** column.

#### Figure 5-27 Importing data

	Name	Version	Labeling Progress	Created ↓Ξ	Description	Operation
~	🔁 dxf-dataset-image-20200519-001 ytf14r2DMbj256RCKEB	V003	75.00% (12/16)	May 19, 2020 15:56:12 GMT+08:00		Import   Publish   Labeling   Export   Delete   More 🔻

Alternatively, you can click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Import** in the upper right corner.

- 3. In the Import dialog box, set the parameters as follows and click OK.
  - Data Source: Local file
  - **Storage Path**: Select an OBS path.
  - Uploading Data: Click Upload data, upload local data, and click OK.

#### Figure 5-28 Importing data from local files

Import		
* Data Source	OBS Local file	
* Storage Path	Select an OBS path.	Ð
Uploading Data	🕀 Upload data	
* Labeling Status	Unlabeled Labeled	
	<b>OK</b> Cancel	

# 5.6 Data Analysis and Preview

Generally, the quality of raw data cannot meet training requirements, for example, invalid or duplicate data exists. To help you improve data quality, ModelArts provides the following capabilities:

- Auto Grouping: pre-classifies data through clustering to allow you to label data based on clustering results, which ensures that different labels have the same or the almost same number of samples.
- **Data Filtering**: enables you to filter data based on sample attributes and auto grouping results.
- **Data Feature Analysis**: analyzes data features or labeling results, such as the brightness and bounding box distribution, helping you analyze data balance and improve the model training effect.

# 5.6.1 Processing Data

After data is collected and imported, the data cannot directly meet the training requirements. Process data during R&D to ensure data quality and prevent negative impact on subsequent operations (such as data labeling and model training). ModelArts provides data processing to extract valuable and meaningful data from a large amount of disordered and difficult-to-understand data.

ModelArts provides four basic data processing functions:

- Data validation: helps AI developers identify invalid data, such as damaged data and unqualified data, and effectively prevent algorithm precision deterioration or training failures caused by noisy data.
- Data cleansing: checks data consistency based on data validation and correct some invalid values.
- Data selection: During AI development, a large amount of duplicate data may exist in the collected data. The duplicate data does not improve the model precision. Moreover, it takes a long time to label the data. In this case, use data selection to preprocess data and deduplicate collected data.
- Data augmentation: increases the data volume.

# 5.6.2 Auto Grouping

To improve the precision of auto labeling algorithms, you can evenly label multiple classes. ModelArts provides built-in grouping algorithms. You can enable auto grouping to improve data labeling efficiency.

Auto grouping can be understood as data labeling preprocessing. Clustering algorithms are used to cluster unlabeled images, and images are labeled or cleaned by group based on the clustering result.

For example, a user searches for XX through a search engine, downloads and uploads related images to the dataset, and then uses the auto grouping function to classify XX images, such as papers, posters, images confirmed as XX, and others. The user can quickly remove unwanted images from a group or select all images of a type and add labels to the images.

### **NOTE**

Only datasets of image classification, object detection, and image segmentation types support the auto grouping function.

### **Starting Auto Grouping Tasks**

- Log in to the . In the navigation pane, choose Data Management > Label Data.
- 2. In the labeling job list, select a labeling job of the object detection or image classification type and click the labeling job name to go to the labeling job details page.
- 3. On the **All statuses** tab page of the dataset details page, choose **Auto Grouping** > **Start Task**.

#### **NOTE**

You can start auto group tasks or view task history only on the All tab page.

- 4. In the displayed Auto Grouping dialog box, set parameters and click OK.
  - **Groups**: Enter an integer from 2 to 200. The parameter value indicates the number of groups into which images are divided.
  - Result Processing Method: Select Update attribute or Save to OBS.
  - Attribute Name: If you select Update attribute, you need to enter an attribute name.
  - Result Storage Path: If you select Save to OBS, specify an OBS path.
  - Advanced Feature Settings: After this function is enabled, you can select Clarity, Brightness, and Color dimensions for the auto grouping function so that the grouping is based on the image brightness, color, and clarity. You can select multiple options.
- 5. After the task is submitted, the task progress is displayed in the upper right corner of the page. After the task is complete, you can view the history of the auto grouping tasks to learn task status.

### Viewing the Auto Grouping Result

On the **All** tab page of the dataset details page, expand **Filter Criteria**, set **Sample Attribute** to the attribute name of the auto grouping task, and set the sample attribute value to filter the grouping result.

Figure 5-29 Viewing the auto grouping result

Filter Criteria	Label   roses × + Add Filter Criterion Clear All
Example Type Label	Hard example Non-hard example All dandelion daisy roses sunflowers tulips
Sample Creation Time	Within 1 month Within 1 day Custom
File N 🔻	Enter a keyword and press Enter to create a filter criterion.
Labeled By	Select an annotator.
Sample Attribute	-Select- v No attributes available. Click Auto Grouping and select Start Task to create data management attribute

### Viewing Auto Grouping Task History

On the **All** tab page of the dataset details page, choose **Auto Grouping** > **View Task History**. In the **View Task History** dialog box, basic information about the auto grouping tasks of the current dataset is displayed.

#### Figure 5-30 Auto grouping task history

#### View Task History

If the Result Processing Method is Update attribute, you can select attribute values based on the sample attribute as a filter criterion to obtain the result. If the Result Processing Method is Save to OBS, you can view or download the grouping result in the storage path.

Created	Groups	Result Processin	Storage Path/Att	Status	Opera
2020-03-13 09:20	2	Update attribute	sunflowers	🔅 Running[The j	Stop

# 5.6.3 Data Filtering

On the **Dashboard** tab page of the dataset, the summary of the dataset is displayed by default. In the upper right corner of the page, click **Label**. The dataset details page is displayed, showing all data in the dataset by default. On the **All**, **Unlabeled**, or **Labeled** tab page, you can add filter criteria in the filter criteria area to quickly filter the data you want to view.

The following filter criteria are supported. You can set one or more filter criteria.

- Example Type: Select Hard example or Non-hard example.
- Label: Select All or one or more labels you specified.
- Sample Creation Time: Select Within 1 month, Within 1 day, or Custom to customize a time range.
- File Name or Path: Filter files by file name or file storage path.
- Labeled By: Select the name of the user who labeled the image.

# 5.6.4 Data Feature Analysis

Images or target bounding boxes are analyzed based on image features, such as blurs and brightness to draw visualized curves to help process datasets.

You can also select multiple versions of a dataset to view their curves for comparison and analysis.

### Background

- Data feature analysis is only available for image datasets of the image classification and object detection types.
- Data feature analysis is only available for the published datasets. The published dataset versions in **Default** format support data feature analysis.
- A data scope for feature analysis varies depending on the dataset type.
  - In a dataset of the object detection type, if the number of labeled samples is 0, the View Data Feature tab page is unavailable and data features are not displayed after a version is published. After the images are labeled and the version is published, the data features of the labeled images are displayed.
  - In a dataset of the image classification type, if the number of labeled samples is 0, the View Data Feature tab page is unavailable and data features are not displayed after a version is published. After the images are labeled and the version is published, the data features of all images are displayed.

- The analysis result is valid only when the number of images in a dataset reaches a certain level. Generally, more than 1,000 images are required.
- Image classification supports the following data feature metrics: **Resolution**, **Aspect Ratio**, **Brightness**, **Saturation**, **Blur Score**, and **Colorfulness** Object detection supports all data feature metrics. **Supported Data Feature Metrics** provides all data feature metrics supported by ModelArts.

### Data Feature Analysis

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- 2. Select a dataset and click **Data Features** in the **Operation** column. The **Data Features** tab page of the dataset page is displayed.

You can also click a dataset name to go to the dataset page and click the **Data Features** tab.

3. By default, feature analysis is not started for published datasets. You need to manually start feature analysis tasks for each dataset version. On the **Data Features** tab page, click **Feature Analysis**.

#### Figure 5-31 Feature Analysis

Dashboa	rd Versions	Data Feature	s Labeling Progress					
Version	-Select-	• Туре	-Select-	Data Feature Metric	-Select-	•	Feature Analysis	Analysis History

4. In the dialog box that is displayed, configure the dataset version for feature analysis and click **OK** to start analysis.

Version: Select a published version of the dataset.

Figure 5-32 Starting a data feature analysis task

Version	Select-	-		•
		ОК	Cancel	

5. After a data feature analysis task is started, it takes a certain period of time to complete, depending on the data volume. If the selected version is displayed in the **Version** drop-down list and can be selected, the analysis is complete.

Figure 5-33 Selecting a version for which feature analysis has been performed

Version	-Select-	•	
	V002		
	V001		

6. View the data feature analysis result.

**Version**: Select the version to be compared from the drop-down list You can also select only one version.

**Type**: Select the type to be analyzed. The value can be **all**, **train**, **eval**, or **inference**.

**Data Feature Metric**: Select metrics to be displayed from the drop-down list. For details, see **Supported Data Feature Metrics**.

Then, the selected version and metrics are displayed on the page, as shown in **Figure 5-34**. The displayed chart helps you understand data distribution for better data processing.

Figure 5-34 Data feature analysis



7. View historical records of the analysis task.

After data feature analysis is complete, you can click **Task History** on the right of the **Data Features** tab page to view historical analysis tasks and their statuses in the dialog box that is displayed.

Figure 5-35 Viewing the task history

# View Task History

Dataset Vers	Task ID	Created	Duration(hh:	Status
V002	Rdnjwum33T	Apr 03, 2020	00:01:50	Successful
V001	gpw2hG6D6	Apr 02, 2020	00:02:23	Successful

# **Supported Data Feature Metrics**

### Table 5-20 Data feature metrics

Metric	Description	Explanation
Resolution	Image resolution. An area value is used as a statistical value.	Metric analysis results are used to check whether there is an offset point. If an offset point exists, you can resize or delete the offset point.
Aspect Ratio	An aspect ratio is a proportional relationship between an image's width and height.	The chart of the metric is in normal distribution, which is generally used to compare the difference between the training set and the dataset used in the real scenario.
Brightness	Brightness is the perception elicited by the luminance of a visual target. A larger value indicates better image brightness.	The chart of the metric is in normal distribution. You can determine whether the brightness of the entire dataset is high or low based on the distribution center. You can adjust the brightness based on your application scenario. For example, if the application scenario is night, the brightness should be lower.
Saturation	Color saturation of an image. A larger value indicates that the entire image color is easier to distinguish.	The chart of the metric is in normal distribution, which is generally used to compare the difference between the training set and the dataset used in the real scenario.
Blur Score Clarity	Image clarity, which is calculated using the Laplace operator. A larger value indicates clearer edges and higher clarity.	You can determine whether the clarity meets the requirements based on the application scenario. For example, if data is collected from HD cameras, the clarity must be higher. You can sharpen or blur the dataset and add noises to adjust the clarity.

Metric	Description	Explanation
Colorfulness	Horizontal coordinate: Colorfulness of an image. A larger value indicates richer colors. Vertical coordinate: Number of images	Colorfulness on the visual sense, which is generally used to compare the difference between the training set and the dataset used in the real scenario.
Bounding Box Number	Horizontal coordinate: Number of bounding boxes in an image Vertical coordinate: Number of images	It is difficult for a model to detect a large number of bounding boxes in an image. Therefore, more images containing many bounding boxes are required for training.
Std of Bounding Boxes Area Per Image Standard Deviation of Bounding Boxes Per Image	Horizontal coordinate: Standard deviation of bounding boxes in an image. If an image has only one bounding box, the standard deviation is 0. A larger standard deviation indicates higher bounding box size variation in an image. Vertical coordinate: Number of images	It is difficult for a model to detect a large number of bounding boxes with different sizes in an image. You can add data for training based on scenarios or delete data if such scenarios do not exist.
Aspect Ratio of Bounding Boxes	Horizontal coordinate: Aspect ratio of the target bounding boxes Vertical coordinate: Number of bounding boxes in all images	The chart of the metric is generally in Poisson distribution, which is closely related to application scenarios. This metric is mainly used to compare the differences between the training set and the validation set. For example, if the training set is a rectangle, the result will be significantly affected if the validation set is close to a square.

Metric	Description	Explanation
Area Ratio of Bounding Boxes	Horizontal coordinate: Area ratio of the target bounding boxes, that is, the ratio of the bounding box area to the entire image area. A larger value indicates a higher ratio of the object in the image. Vertical coordinate: Number of bounding boxes in all images	The metric is used to determine the distribution of anchors used in the model. If the target bounding box is large, set the anchor to a large value.
Marginalization Value of Bounding Boxes	Horizontal coordinate: Marginalization degree, that is, the ratio of the distance between the center point of the target bounding box and the center point of the image to the total distance of the image. A larger value indicates that the object is closer to the edge. (The total distance of an image is the distance from the intersection point of a ray (that starts from the center point of the image and passes through the center point of the image border to the center point of the image.) Vertical coordinate: Number of bounding boxes in all images	Generally, the chart of the metric is in normal distribution. The metric is used to determine whether an object is at the edge of an image. If a part of an object is at the edge of an image, you can add a dataset or do not label the object.

Metric	Description	Explanation
Overlap Score of Bounding Boxes Overlap Score of Bounding Boxes	Horizontal coordinate: Overlap degree, that is, the part of a single bounding box overlapped by other bounding boxes. The value ranges from 0 to 1. A larger value indicates that more parts are overlapped by other bounding boxes. Vertical coordinate: Number of bounding boxes in all images	The metric is used to determine the overlapping degree of objects to be detected. Overlapped objects are difficult to detect. You can add a dataset or do not label some objects based on your needs.
Brightness of Bounding Boxes Brightness of Bounding Boxes	Horizontal coordinate: Brightness of the image in the target bounding box. A larger value indicates brighter image. Vertical coordinate: Number of bounding boxes in all images	Generally, the chart of the metric is in normal distribution. The metric is used to determine the brightness of an object to be detected. In some special scenarios, the brightness of an object is low and may not meet the requirements.
Blur Score of Bounding Boxes Clarity of Bounding Boxes	Horizontal coordinate: Clarity of the image in the target bounding box. A larger value indicates higher image clarity. Vertical coordinate: Number of bounding boxes in all images	The metric is used to determine whether the object to be detected is blurred. For example, a moving object may become blurred during collection and its data needs to be collected again.

# 5.7 Labeling Data

Model training requires a large amount of labeled data. Therefore, before training a model, label data. You can create a manual labeling job labeled by one person or by a group of persons (team labeling), or enable auto labeling to quickly label images. You can also modify existing labels, or delete them and re-label.

- Manual labeling: allows you to manually label data.
- Auto labeling: allows you to automatically label remaining data after a small amount of data is manually labeled.
- Team labeling: allows you to perform collaborative labeling for a large amount of data.

For details about data labeling, see .

# 5.8 Publishing Data

# 5.8.1 Introduction to Data Publishing

ModelArts distinguishes data of the same source according to versions processed or labeled at different time, which facilitates the selection of dataset versions for subsequent model building and development.

### **About Dataset Versions**

- For a newly created dataset (before publishing), there is no dataset version information. The dataset must be published before being used for model development or training.
- The default naming rules of dataset versions are V001 and V002 in ascending order. You can customize the version number during publishing.
- You can set any version to the current version. Then the details of the version are displayed on the dataset details page.
- You can obtain the dataset in the manifest file format corresponding to each dataset version based on the value of **Storage Path**. The dataset can be used when you import data or filter hard examples.
- The version of a table dataset cannot be changed.

# 5.8.2 Publishing a Data Version

- 1. Log in to the . In the navigation pane, choose **Data Management** > **Datasets**.
- 2. Locate the row containing the target dataset and click **Publish** in the **Operation** column. Alternatively, click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Publish** in the upper right corner.
- 3. In the displayed dialog box, set the parameters and click **OK**.

### Figure 5-36 Publishing a dataset version

### Publish New Version

* Version	V003
A Before you enab least two multi-	ole splitting, ensure each label has at least five labeled samples. Ensure there are at label samples, if any. For details, go to the Dashboard tab page.
* Labeling Type	Image cl Object d Image se Free for
Description	
	0/256
	<b>OK</b> Cancel

### Table 5-21 Parameters for publishing a dataset

Parameter	Description
Version	The naming rules of V001 and V002 in ascending order are used by default. A version name can be customized. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Format	Only table datasets support version format setting. Available values are <b>CSV</b> and <b>CarbonData</b> .
	<b>NOTE</b> If the exported CSV file contains any command starting with =, +, -, or @, ModelArts automatically adds the Tab setting and escapes the double quotation marks (") for security purposes.

Parameter	Description
Splitting	Only image classification, object detection, text classification, and sound classification datasets support data splitting.
	By default, this function is disabled. After this function is enabled, set the training and validation ratios.
	Enter a value ranging from 0 to 1 for <b>Training Set Ratio</b> . After the training set ratio is set, the validation set ratio is determined. The sum of the training set ratio and the validation set ratio is 1.
	<b>NOTE</b> To ensure the model accuracy, you are advised to set the training set ratio to 0.8 or 0.9.
	The training set ratio is the ratio of sample data used for model training. The validation set ratio is the ratio of the sample data used for model validation. The training and validation ratios affect the performance of training templates.
Description	Description of the current dataset version.
Hard Example	Only image classification and object detection datasets support hard example attributes.
	By default, this function is disabled. After this function is enabled, information such as the hard example attributes of the dataset are written to the corresponding manifest file.

### **Directory Structure of Dataset Versions**

Datasets are managed based on OBS directories. After a new version is published, the directory is generated based on the new version in the output dataset path.

Take an image classification dataset as an example. After the dataset is published, the directory structure of related files generated in OBS is as follows:

|-- user-specified-output-path |-- DatasetName-datasetId |-- annotation |-- VersionMame1 |-- VersionMame1.manifest |-- VersionMame2 ... |-- ...

The following uses object detection as an example. If a manifest file is imported to the dataset, the following provides the directory structure of related files after the dataset is published:

```
|-- user-specified-output-path

|-- DatasetName-datasetId

|-- annotation

|-- VersionMame1

|-- VersionMame1.manifest

|-- annotation

|-- file1.xml
```

|-- VersionMame2 ... |-- ...

Take video labeling as an example. After the dataset is published, the labeling result file (XML) is stored in the dataset output directory.

 user-specified-output-path
DatasetName-datasetId
annotation
VersionMame1
VersionMame1.manifest
annotations
images
videoName1
videoName1.timestamp.xml
videoName2
videoName2.timestamp.xml
VersionMame2

The key frames for video labeling are stored in the dataset input directory.

```
|-- user-specified-input-path

|-- images

|-- videoName1

|-- videoName1.timestamp.jpg

|-- videoName2

|-- videoName2.timestamp.jpg
```

# 5.8.3 Managing Data Versions

During data preparation, you can publish data into multiple versions for dataset management. You can view version updates, set the current version, and delete versions.

### **Viewing Dataset Version Updates**

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- In the dataset list, choose More > Manage Version in the Operation column. The Manage Version tab page is displayed.

You can view basic information about the dataset, and view the versions and publish time on the left.



shboard Versions View Data Feature Labeling Progress	
Toole ) (III 4 <sup>a</sup> ) Total Venion: 2	V002
	ID 4jPeCR0FqnvnL4R2vrzD
Current wersion	Saved Mar 16, 2022 17:33:14 GMT+08:00
V002 Mar 16, 2022 17:33:10	Format Default
	Status 📀 Normal
	Verification Disabled
V001	Split Ratio 1.00
Mar 14, 2022 17:23:50	Description
	El~ 40
	1163 40
	Labeled 0% (0/40)
	Storage Path /1037871/model/work 1647249819141/dataset-bd59-COFKQLx1FbrLWx7k1xl/annotation/V002/V002.marifest
	Set to Current Version Delete

### **Setting to Current Version**

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- 2. In the dataset list, choose **More > Manage Version** in the **Operation** column. The **Manage Version** tab page is displayed.
- 3. On the **Manage Version** tab page, select the desired dataset version, and click **Set to Current Version** in the basic information area on the right. After the setting is complete, **Current version** is displayed to the right of the version name.

Only the version in Normal status can be set to the current version.

### **Deleting a Dataset Version**

- 1. Log in to the . In the navigation pane, choose **Data Management** > **Datasets**.
- In the dataset list, choose More > Manage Version in the Operation column. The Manage Version tab page is displayed.
- 3. Locate the row that contains the target version, and click **Delete** in the **Operation** column. In the dialog box that is displayed, click **OK**.

### **NOTE**

Deleting a dataset version does not remove the original data. Data and its labeling information are still stored in the OBS directory. However, this affects version management. Exercise caution when performing this operation.

# 5.9 Exporting Data

# 5.9.1 Introduction to Exporting Data

You can select data or filter data based on the filter criteria in a dataset and export to a new dataset or the specified OBS path. The historical export records can be viewed in task history.

Only datasets of image classification, object detection, and image segmentation types can be exported.

- For image classification datasets, only the label files in TXT format can be exported.
- For object detection datasets, only XML label files in Pascal VOC format can be exported.
- For image segmentation datasets, only XML label files in Pascal VOC format and mask images can be exported.

# 5.9.2 Exporting Data to a New Dataset

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- 2. In the dataset list, select an image dataset and click the dataset name to go to the **Dashboard** tab page of the dataset.

X

3. Click **Export** in the upper right corner. In the displayed **Export To** dialog box, enter the related information and click **OK**.

#### Type: New Dataset.

Name: name of the new dataset

**Storage Path**: input path of the new dataset, that is, the OBS path where the data to be exported is stored

**Output Path**: output path of the new dataset, that is, the output path after labeling is complete The output path cannot be the same as the storage path, and the output path cannot be a subdirectory of the storage path.

- 4. After the data is exported, view it in the specified path. After the data is exported, you can view the new dataset in the dataset list.
- 5. On the **Dashboard** tab page, click **Export History** in the upper right corner. In the displayed dialog box, view the task history of the dataset.

# 5.9.3 Exporting Data to OBS

- 1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- 2. In the dataset list, select an image dataset and click the dataset name to go to the **Dashboard** tab page of the dataset.
- 3. Click **Export** in the upper right corner. In the displayed **Export To** dialog box, enter the related information and click **OK**.

Type: OBS.

**Storage Path**: path where the data to be exported is stored. You are advised not to save data to the input or output path of the current dataset.

Figure 5-38 Exporting data to OBS

Export To			
Туре	New Dataset	OBS	
Storage Path	Select an OBS path.		Ð
	ОК	Cancel	

- 4. After the data is exported, view it in the specified path.
- 5. On the **Dashboard** tab page, click **Export History** in the upper right corner. In the displayed dialog box, view the task history of the dataset.

# 5.10 Introduction to Data Labeling

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

Model training requires a large amount of labeled data. Therefore, before training a model, label data. ModelArts provides you with the following labeling functions:

- Manual Labeling: allows you to manually label data.
- **Auto Labeling**: allows you to automatically label remaining data after a small amount of data is manually labeled.
- **Team Labeling**: allows you to perform collaborative labeling for a large amount of data.

### Manual Labeling

Create a labeling job based on the dataset type. ModelArts supports the following types of labeling jobs:

- Images
  - Image classification: identifies a class of objects in images.
  - Object detection: identifies the position and class of each object in an image.
  - Image segmentation: segments an image into different areas based on objects in the image.
- Audio
  - Sound classification: classifies and identifies different sounds.
  - Speech labeling: labels speech content.
  - Speech paragraph labeling: segments and labels speech content.
- Text
  - Text classification: assigns labels to text according to its content.
  - Named entity recognition: assigns labels to named entities in text, such as time and locations.
  - Text triplet: assigns labels to entity segments and entity relationships in the text.
- Video

Video labeling: identifies the position and class of each object in a video. Only the MP4 format is supported.

### **Auto Labeling**

In addition to manual labeling, ModelArts also provides the auto labeling function to quickly label data, reducing the labeling time by more than 70%. Auto labeling means learning and training are performed based on the labeled images and an existing model is used to quickly label the remaining images.

Only datasets of image classification and object detection types support the auto labeling function.

### **Team Labeling**

Generally, a small data labeling task can be completed by an individual. However, team work is required to label a large dataset. ModelArts provides the team labeling function. A labeling team can be formed to manage labeling for the same dataset.

The team labeling function supports only datasets for image classification, object detection, text classification, named entity recognition, text triplet, and speech paragraph labeling.

### **Dataset Functions**

Dataset functions vary depending on dataset types. For details, see **Table 5-22**.

Datas et Type	Labeling Type	Manual Labeling	Auto Labeling	Team Labeling
lmage s	Image classification	Supported	Supported	Supported
	Object detection	Supported	Supported	Supported
	Image segmentation	Supported	N/A	N/A
Audio	Sound classification	Supported	N/A	N/A
	Speech labeling	Supported	N/A	N/A
	Speech paragraph labeling	Supported	N/A	Supported
Text	Text classification	Supported	N/A	Supported
	Named entity recognition	Supported	N/A	Supported
	Text triplet	Supported	N/A	Supported
Video	Video labeling	Supported	N/A	N/A
Free format	N/A	N/A	N/A	N/A
Table	N/A	N/A	N/A	N/A

 Table 5-22
 Functions supported by different types of datasets

# 5.11 Manual Labeling

# 5.11.1 Creating a Labeling Job

Model training requires a large amount of labeled data. Therefore, before training a model, label data. You can create a manual labeling job labeled by one person or by a group of persons (team labeling), or enable auto labeling to quickly label images. You can also modify existing labels, or delete them and re-label.

## Labeling Job Types

Create a labeling job based on the dataset type. ModelArts supports the following types of labeling jobs:

- Images
  - Image classification: identifies a class of objects in images.
  - Object detection: identifies the position and class of each object in an image.
  - Image segmentation: segments an image into different areas based on objects in the image.
- Audio
  - Sound classification: classifies and identifies different sounds.
  - Speech labeling: labels speech content.
  - Speech paragraph labeling: segments and labels speech content.
- Text
  - Text classification: assigns labels to text according to its content.
  - Named entity recognition: assigns labels to named entities in text, such as time and locations.
  - Text triplet: assigns labels to entity segments and entity relationships in the text.
- Videos

Video labeling: identifies the position and class of each object in a video. Only the MP4 format is supported.

### Prerequisites

Before labeling data, create a dataset.

### D NOTE

Data management is being upgraded and is invisible to users who have not used data management.

### Procedure

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. On the **Data Labeling** page, click **Create Labeling Job** in the upper right corner. On the page that is displayed, create a labeling job.
  - a. Enter basic information about the labeling job, including **Name** and **Description**.

### Figure 5-39 Basic information about a labeling job

* Name	dataset-test	
Description		
		0/256

b. Select a labeling scene and type as required.

Figure 5-40 Selecting a labeling scene and type



- c. Set the parameters based on the labeling job type. For details, see the parameters of the following labeling job types:
  - Images (Image Classification, Image Segmentation, and Object Detection)
  - Audio (Sound Classification, Speech Labeling, and Speech Paragraph Labeling)
  - Text (Text Classification, Named Entity Recognition, and Text Triplet)
  - Videos
- d. Click **Create** in the lower right corner of the page.

After the labeling job is created, the data labeling management page is displayed. You can perform the following operations on the labeling job: start auto labeling, publish new versions, modify the labeling job, and delete the labeling job.

# Images (Image Classification, Image Segmentation, and Object Detection)



**Figure 5-41** Parameters of labeling jobs for image classification and object detection

### Table 5-23 Parameters of an image labeling job

Parameter	Description
Dataset Name	Select a dataset that supports the labeling type.
Label Set	<ul> <li>Label name: Enter a label name with 1 to 1024 characters.</li> <li>Add Label: Click Add Label to add one or more labels.</li> <li>Label color: Set label colors for object detection and image segmentation labeling jobs. Select a color from the color palette on the right of a label, or enter the hexadecimal color code to set the color.</li> <li>Add Label Attribute: For an object detection labeling job, you can click the plus sign (+) on the right to add label attributes after setting a label color. Label attributes are used to distinguish different attributes of the objects with the same label. For example, yellow kittens and black kittens have the same label cat and their label attribute is color.</li> </ul>

Parameter	Description
Team Labeling	Enable or disable team labeling. Image segmentation does not support team labeling. Therefore, this parameter is unavailable when you use image segmentation.
	After enabling team labeling, enter the type of the team labeling job, and select the labeling team and team members. For details about the parameter settings, see <b>Creating a Team Labeling Job</b> .
	Before enabling team labeling, ensure that you have added a team and members on the <b>Labeling Teams</b> page. If no labeling team is available, click the link on the page to go to the <b>Labeling Teams</b> page, and add your team and members. For details, see Adding a Team.
	After a dataset is created with team labeling enabled, you can view the <b>Team Labeling</b> mark in <b>Labeling Type</b> .

# Audio (Sound Classification, Speech Labeling, and Speech Paragraph Labeling)

**Figure 5-42** Parameters of labeling jobs for sound classification, speech labeling, and speech paragraph labeling

* Name	task-73f8	0	
Description		0/256	
* Labeling Scene	🖂 Images 🔮 Audio [ 🕅 Text	▶ Video	
* Labeling Type	Sound classification Identify if a sound appears in a piece of audio. 	Speech tabeling Label speech content.	Speech paragraph labeling Segment and label speech content.
* Dataset Name	dataset-d56c    Create Dataset		
Label Set	Enter a label name.		

### Table 5-24 Parameters of an audio labeling job

Parameter	Description
Dataset Name	Select a dataset that supports the labeling type.

Parameter	Description
Label Set (for sound classification)	<ul> <li>You can add a label set for labeling jobs of sound classification.</li> <li>Label name: Enter 1 to 1024 characters in the Label Set text box.</li> </ul>
	• Add Label: Click Add Label to add one or more labels.
Label Management (for speech paragraph labeling)	<ul> <li>Label management is available for speech paragraph labeling.</li> <li>Single Label <ul> <li>A single label is used to label a piece of audio that has only one class.</li> <li>Label: Enter a label name, with 1 to 1024 characters.</li> <li>Label Color: Set the label color in the Label Color column. You can select a color from the color palette or enter a hexadecimal color code to set the color.</li> </ul> </li> </ul>
	• Multiple Labels Multiple labels are suitable for multi-dimensional labeling. For example, you can label a piece of audio as both noise and speech. For speech, you can label the audio with different speakers. You can click Add Label Class to add multiple label classes. A label class can contain multiple labels. The label class or name contains 1 to 256 characters. Only letters, digits, periods (.), underscores (_), and hyphens (-) are allowed.
	<ul> <li>Add Label Class: Enter a label class.</li> </ul>
	<ul> <li>Label: Enter a label name.</li> <li>Add Label: Click Add Label to add one or more labels</li> </ul>
Speech Labeling (for speech paragraph labeling)	Only datasets for speech paragraph labeling support speech labeling. By default, speech labeling is disabled. If this function is enabled, you can label speech content.
Team Labeling (for speech paragraph labeling)	<ul> <li>Only datasets of speech paragraph labeling support team labeling.</li> <li>After enabling team labeling, enter the type of the team labeling job, and select the labeling team and team members. For details about the parameter settings, see Creating a Team Labeling Job.</li> <li>Before enabling team labeling, ensure that you have added a team and members on the Labeling Teams page. If no labeling team is available, click the link on the page to go to the Labeling Teams page, and add your team and members. For details, see Adding a Team.</li> <li>After a dataset is created with team labeling enabled, you can</li> </ul>
	view the <b>Team Labeling</b> mark in <b>Labeling Type</b> .

# Text (Text Classification, Named Entity Recognition, and Text Triplet)

* Name	task-73f8
Description	
	0/256
* Labeling Scene	🖂 Images 🔱 Audio 🚺 Text 💌 Video
* Labeling Type	Text classification       Named entity recognition       Text triplet         Assign labels to text according to its content.       Assign labels to named entities in text, such as time       Text triplet         Goott       ///       Image: Content in the image: Content in th
* Dataset Name Label Set	Select a dataset.   C Create Dataset  Enter a label name.  Add Label ⑦ You can create 9999 more labels.
Team Labeling	

**Figure 5-43** Parameters of labeling jobs for text classification, named entity recognition, and text triplet

### Table 5-25 Parameters of a text labeling job

Parameter	Description	
Dataset Name	Select a dataset that supports the labeling type.	
Label Set (for text classification and named entity recognition)	<ul> <li>Label name: Enter a label name, with 1 to 1024 characters.</li> <li>Add Label: Click Add Label to add one or more labels.</li> <li>Label color: Select a color from the color palette or enter the hexadecimal color code to set the color.</li> </ul>	
Label Set (for text triplet)	<ul> <li>For datasets of the text triplet type, set entity labels and relationship labels.</li> <li>Entity Label: Set the label name and label color. You can click the plus sign (+) on the right of the color area to add multiple labels.</li> <li>Relationship Label: a relationship between two entities. Set the source entity and target entity. Therefore, add at least two entity labels before adding a relationship label.</li> </ul>	

Parameter	Description
Team Labeling	Enable or disable team labeling. After enabling team labeling, enter the type of the team labeling job, and select the labeling team and team members. For details about the parameter settings, see <b>Creating a Team</b> <b>Labeling Job</b> .
	Before enabling team labeling, ensure that you have added a team and members on the <b>Labeling Teams</b> page. If no labeling team is available, click the link on the page to go to the <b>Labeling Teams</b> page, and add your team and members. For details, see <b>Adding a Team</b> .
	After a dataset is created with team labeling enabled, you can view the <b>Team Labeling</b> mark in <b>Labeling Type</b> .

# Videos

* Name	task-73f8				
Description					
					0/256
* Labeling Scene	M Images	🔮 Audio	A Text	► Video	
k Labeling Type	Vi	deo labeling	~		
	Only MP4	files are supported.	Man		
	Women	-			
* Dataset Name	dataset-64ff_v2	• C	Create Dataset		
Label Set	Enter a label name	8		-	

Figure 5-44 Parameters of a video labeling job

Parameter	Description
Dataset Name	Select a dataset that supports the labeling type.
Label Set	• Label name: Enter a label name, with 1 to 1024 characters.
	Add Label: Click Add Label to add one or more labels.
	• Label color: Select a color from the color palette or enter the hexadecimal color code to set the color.

Table 5-26 Parameters of a video labeling job

# 5.11.2 Image Labeling

### 5.11.2.1 Image Classification

Training a model uses a large number of labeled images. Therefore, label images before the model training. You can add labels to images by manual labeling or auto labeling. In addition, you can modify the labels of images, or remove their labels and label the images again.

Before labeling an image in image classification scenarios, pay attention to the following:

- You can add multiple labels to an image.
- A label name can contain a maximum of 1024 characters, including letters, digits, hyphens (-), and underscores (\_).

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

### **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. On the right of the labeling job list, select a labeling type from the job type drop-down list. Click the job to be performed based on the labeling type. The details page of the job is displayed.

### Figure 5-45 Selecting a labeling type



3. The job details page displays all data of the labeling job.

### Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to labeling jobs.

To quickly obtain the latest data in a dataset, on the **All statuses**, **Unlabeled**, or **Labeled** tab page of the labeling job details page, click **Synchronize New Data**.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

### **Filtering Data**

On the **All statuses**, **Unlabeled**, or tab page, click in the filter criteria area and add filter criteria to quickly filter the data you want to view.

The following filter criteria are available. You can set one or more filter criteria.

- Example Type: Select Hard example or Non-hard example.
- Label: Select All or one or more labels you specified.
- File Name or Path: Filter files by file name or file storage path.
- Labeled By: Select the name of the user who labeled the image.

## **Manually Labeling Images**

The labeling job details page displays the **All statuses**, **Unlabeled**, and **Labeled** tab pages. The **Unlabeled** tab page is displayed by default. Click an image to preview it. For the images that have been labeled, the label information is displayed at the bottom of the preview page.

- 1. On the **Unlabeled** tab page, select the images to be labeled.
  - Manual selection: In the image list, click the selection box in the upper left corner of an image to enter the selection mode, indicating that the image is selected. You can select multiple images of the same type and add labels to them together.
  - Batch selection: If all the images on the current page of the image list belong to the same type, you can click **Select Images on Current Page** in the upper right corner to select all the images on the current page.
- 2. Add labels to the selected images.
  - a. In the label adding area on the right, set a label in the **Label** text box.

Click the **Label** text box and select an existing label from the drop-down list. If the existing labels cannot meet the requirements, input a label in the text box.

b. Click **OK**. The selected images are automatically moved to the **Labeled** tab page. On the **Unlabeled** and **All statuses** tab pages, the labeling information is updated along with the labeling process, including the added label names and the number of images for each label.

#### **NOTE**

For details about how to label data, see **Labeling Description** on the dataset details page.

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management > Label Data**. The **Data Labeling** page is displayed.
- 2. On the **My Creations** or **My Participations** tab page, find the dataset to be labeled.
- 3. Click the dataset name. The labeling details page is displayed. (By default, the **Unlabeled** tab page is displayed.)
- 4. In the upper right corner of the labeling details page, click **Labeling Description**.

Figure 5-46 Labeling Description

**O** Labeling Description

### Viewing Labeled Images

On the labeling job details page, click the **Labeled** tab to view the list of labeled images. By default, the corresponding labels are displayed under the image thumbnails. You can also select an image and view the label information of the image in the **Labels of Selected Images** area on the right.

## **Modifying Labeled Data**

After labeling data, you can modify labeled data on the **Labeled** tab page.

• Modifying based on images

On the labeling job details page, click the **Labeled** tab, and select one or more images to be modified from the image list. Modify the image information in the label information area on the right.

Modifying a label: In the **Labels of Selected Images** area, click the edit icon in the **Operation** column, enter the correct label name in the text box, and click the check mark to complete the modification.

Deleting a label: In the **Labels of Selected Images** area, click the delete icon in the **Operation** column to delete the label. This operation deletes only the labels added to the selected image.

### Figure 5-47 Modifying a label

#### Labels of Selected Images

Name	Labels	Operation	
dog_image	1	2 ū	

### • Modifying based on labels

- On the labeling job details page, click Label Management. All labels are displayed on the list.
  - Modifying a label: Click Modify in the Operation column. In the dialog box that is displayed, enter a new label name and click OK. After the modification, the images that have been added with the label use the new label name.
  - Deleting a label: Click **Delete** in the **Operation** column to delete the label from all images that have been added with the label.

#### Figure 5-48 Label Management

All Labels 12	Label Management $\mathbf C$
Name	Labels 🚛
animal	168
cake	50
cat_image	19
data	293
Data	6

#### Figure 5-49 All labels

Add Label     Delete Label		
Label Name	Attribute	Operation
🗍 animal		Modify Delete
C cale		Modify Delete
Cat, image	5. C	Modify Delete

- Click **Label** in the **Operation** column of the target labeling job to go to the label management page.
  - Click Modify in the Operation column of the target label to modify it.
  - Click **Delete** in the **Operation** column of the target label to delete it.

### **Adding Data**

In addition to the data automatically synchronized from datasets, you can directly add images to labeling jobs for labeling. The added data is first imported to the dataset associated with the labeling job. Then, the labeling job automatically synchronizes the latest data from the dataset.

1. On the labeling job details page, click **All statuses**, **Labeled**, or **Unlabeled** tab, click **Add data** in the upper left corner.

#### Figure 5-50 Adding data



2. Configure the data source, import mode, import path, and labeling status.

×

Import			
* Data Source	OBS	Local file	
* Storage Path	Select an OBS p	path.	Ð
Uploading Data	🕀 Upload da	ata	
* Labeling Status	Unlabeled	Labeled	
		OK Cancel	

3. Click OK.

The images you have added will be automatically displayed in the image list on the **All statuses** tab page. You can choose **Add data** > **View historical records** to view task history.
#### Figure 5-51 Viewing historical data

View Task Histor	у					,
Created	Import Mode	Import Path	Samples	Imported Samples	Labeled Samples	Import Stat
Mar,24,2022 10:21:	OBS path	obs://ei-modelarts	19	19	0	Succeed

## **Deleting Images**

You can quickly delete the images you want to discard.

On the **All statuses**, **Unlabeled**, or **Labeled** tab page, select the images to be deleted or click **Select Images on Current Page**, and click **Delete**. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **Yes** to delete the images.

Figure 5-52 Deleting Images

<b>▽</b> Filter ∨	Selected: 1	Deselect	Select Images on Current Page	Ū 0	elete	
						1

If a tick is displayed in the upper left corner of an image, the image is selected. If no image is selected on the page, the **Delete** button is unavailable.

#### **NOTE**

If you select **Delete the source files from OBS**, images stored in the OBS directory will be deleted accordingly. This operation may affect other dataset versions or datasets using those files, for example, leading to an error in page display, training, or inference. Deleted data cannot be recovered. Exercise caution when performing this operation.

#### **Managing Annotators**

If team labeling is enabled for a labeling job, view its labeling details on the **Annotator Management** tab page. Additionally, you can add, modify, or delete annotators.

- 1. Choose **Data Management** > **Label Data**. On the **My Creations** tab page, view the list of all labeling jobs.
- 2. Locate the row that contains the target team labeling job. (The name of a team labeling iob is followed by  $\mathcal{P}_{a}$ .)
- Choose More > Annotator Management in the Operation column. Alternatively, click the job name to go to the job details page, and choose Team Labeling > Annotator Management in the upper right corner.

#### Figure 5-53 Annotator Management (1)

	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created ↓≣	Description	Operation
~	El task-f98d A	dataset-e8df	0% (0/2)		Mar 29, 2023 14:52:46 GMT+08:00		Auto Labeling   Label   Publish   More -
~	🖾 task-2a81	dataset-1c68			Nov 13, 2022 15:58:21 GMT+08:00	🖉	Auto Labeling   Label   Pu Task Statistics
~	task-c010	dataset-1c68			Nov 01, 2022 15:58:08 GMT+08:00	🖉	Auto Labeling   Label   PL Annotator Management
~	⊠ dataset-63e0 🛆	dataset-63e0	79% (854/1083)		Oct 17, 2022 09:59:07 GMT+08:00	🖉	Auto Labeling   Label   Pu Modify

Figure 5-54 Annotator Management (2)

Publish	Task Statistics	Team Labeling 🔻
		Accept
		Stop Acceptance
		Continue Acceptance
All Labels 1		Acceptance Report
Name		Annotator Management
	Publish All Labels 1 Name	Publish Task Statistics All Labels 1 Name

• Adding an annotator

Click Add Member, select a member name, and click OK.

Click **Send Email** in the **Operation** column to send the labeling job to the annotator by email.

- Modifying annotator information
   Click Modify in the Operation column to modify the role of the annotator.
- Deleting an annotator
   Click **Delete** in the **Operation** column to delete the annotator.

## 5.11.2.2 Object Detection

Training a model uses a large number of labeled images. Therefore, label images before the model training. You can add labels to images by manual labeling or auto labeling. In addition, you can modify the labels of images, or remove their labels and label the images again.

Before labeling an image in object detection scenarios, pay attention to the following:

- All target objects in the image must be labeled.
- Target objects are clear without any blocking and contained within bounding boxes.
- Only the entire object must be contained within a bounding box. The bounding box contains the entire object. The edge of the bounding box cannot intersect the edge outline of the object to be labeled. Ensure that there is no gap between the edge and the object to be labeled to prevent the background from interfering with the model training.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

## **Starting Labeling**

1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.

2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

#### Figure 5-55 Selecting a labeling type

All types	*
Search	Q
All types	
Image classification	
Object detection	
Text classification	
Named entity recognition	
Text triplet	
Sound classification	

3. The job details page displays all data of the labeling job.

# Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in a dataset, on the **All statuses**, **Unlabeled**, or **Labeled** tab page of the labeling job details page, click **Synchronize New Data**.

**NOTE** 

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

## **Filtering Data**

On the **All statuses**, **Unlabeled**, or tab page, click  $\checkmark$  in the filter criteria area and add filter criteria to quickly filter the data you want to view.

The following filter criteria are available. You can set one or more filter criteria.

• Example Type: Select Hard example or Non-hard example.

- Label: Select All or one or more labels you specified.
- File Name or Path: Filter files by file name or file storage path.
- Labeled By: Select the name of the user who labeled the image.

## **Manually Labeling Images**

The labeling job details page displays the **All statuses**, **Unlabeled**, and **Labeled** tab pages. The **Unlabeled** tab page is displayed by default.

- 1. On the **Unlabeled** tab page, click an image. The system automatically directs you to the page for labeling the image. For details about how to use common buttons on this page, see **Table 5-28**.
- 2. In the tool bar, select a proper labeling shape. The default labeling shape is a rectangle. In this example, the rectangle is used for labeling.

**NOTE** 

In the tool bar, multiple tools are provided for you to label images. After you select a shape to label the first image, the shape automatically applies to subsequent images. You can switch the shape as required.

#### Table 5-27 Supported bounding box

lcon	Description
	Rectangle. You can also press <b>1</b> . Click the edge of the upper left corner of the object to be labeled. A rectangle will be displayed. Drag the rectangle to cover the object and click to label the object.
Q	Polygon. You can also press <b>2</b> . In the area where the object to be labeled is located, click to label a point, move the mouse and click multiple points along the edge of the object, and then click the first point again. All the points form a polygon. In this way, the object to be labeled is within the bounding box.
0	Round. You can also press <b>3</b> . Click the center point of an object, and move the mouse to draw a circle to cover the object and click to label the object.
/	Straight. You can also press <b>4</b> . Click to specify the start and end points of an object, and move the mouse to draw a straight line to cover the object and click to label the object.
	Dashed line. You can also press <b>5</b> . Click to specify the start and end points of an object, and move the mouse to draw a dashed line to cover the object and click to label the object.
۲	Dot. You can also press <b>6</b> . Click the object in an image to label a point.

3. In the **Add Label** text box, enter a new label name, select the label color, and click **Add**. Alternatively, select an existing label from the drop-down list.

Label all objects in an image. Multiple labels can be added to an image. After labeling an image, click the right arrow (or press D) in the upper right corner of the image to switch to the next image and label the image.

4. Click **Back to Data Labeling Preview** in the upper left part of the page to view the labeling information. In the dialog box that is displayed, click **Yes** to save the labeling settings.

The selected images are automatically moved to the **Labeled** tab page. On the **Unlabeled** and **All statuses** tab pages, the labeling information is updated along with the labeling process, including the added label names and the number of images for each label.

Button	Features
<b>~</b>	Cancel the previous operation. You can also press <b>Ctrl+Z</b> .
$\rightarrow$	Redo the previous operation. You can also press <b>Ctrl+Shift+Z</b> .
⊙	Zoom in an image. You can also use the mouse wheel to zoom in.
Q	Zoom out an image. You can also use the mouse wheel to zoom out.
ΰ	Delete all bounding boxes on the current image. You can also press <b>Shift+Delete</b> .
Ø	Show or hide a bounding box. This operation can be performed only on a labeled image. You can also press <b>Shift +H</b> .
÷÷	Drag a bounding box to another position or drag the edge of the bounding box to resize it. You can also use $X$ + left mouse button.
đ	Reset a bounding box. After dragging a bounding box, you can click this button to quickly restore the bounding box to its original shape and position. You can also press <b>Esc</b> .

Table 5-28 Common	icons	on the	labeling	page
-------------------	-------	--------	----------	------

## Viewing Labeled Images

On the labeling job details page, click the **Labeled** tab to view the list of labeled images. The labels of each image are displayed below the image.

#### Figure 5-56 Labels



## **Quick Review**

To simplify operations, ModelArts provides quick review so that you can batch review and modify labeled data.

- Log in to the ModelArts management console. In the navigation pane, choose Data Management > Label Data. On the My Creations tab page, select the target labeling job type from the All types drop-down list in the upper right corner. (Only object detection and image segmentation support quick review.)
- 2. In the labeling job list, click the target labeling job. The labeling details page is displayed.
- 3. Click **Quick Review** on the **Labeled** tab. On the displayed page, confirm the labeling results.

#### Figure 5-57 Quick Review



- 4. Batch review images of the same label.
  - a. On the review page, select the label type from the drop-down list next to **Filter by Label**.
  - b. Sort images of the selected label type by bounding box area or aspect ratio.
  - c. Click an incorrectly labeled image, and then drag the labeling box to relabel the image. (**Modified** is displayed on the modified images.)
  - d. You can select the incorrectly labeled images, and then click  $\overline{U}$  in the upper right corner to delete the label. (**Deleted** is displayed on the images whose label has been deleted.)

#### Figure 5-58 Modified



- e. You can also modify the label of a labeled image.
  - i. Select the target images and click  $\textcircled{\oplus}$  in the **All Labels** area on the right.
  - ii. Type a new label and click **OK**.

#### Figure 5-60 All Labels

Select the incorrectly labeled lick the correct target label on correct the images' labels in a b	l images, then the right to atch.
black Cat	
Cat	
person	
werewr	
white Cat	

Figure 5-61 Adding a label

#### Add Label

t Jet	Enter a label name.		-	+ 🗓
	() Address			
	( Add Label			

5. After the modification, click **Apply Modifications**. In the displayed dialog box, click **OK**. The system automatically returns to the labeling overview page and overwrites the original labeling data.

6.

×

#### Figure 5-62 Apply Modifications

Are ye	ou sure you	want to app	oly all mo	difications to	the labeled data?
After y	you click OK, I	the review resu	ılt will ove	rwrite the origin	al data.
			ОК	Cancel	

Figure 5-63 Cancel Modifications

# **Cancel Modifications**

Are you sure you want to cancel all modifications in this review?

After you click OK, all the modifications in this review will be cancelled.



Table 5-29 Buttons on the quick review page

Button	Features
ច	Delete the label.
Ð	Undo all operations on the current page.
<b>~</b>	Undo the previous operation.
$\rightarrow$	Redo the previous operation.

## Modifying Labeled Data

After labeling data, you can modify labeled data on the **Labeled** tab page.

• Modifying based on images

On the labeling job details page, click the **Labeled** tab and then the image to be modified. The labeling page is displayed. Modify the image information in the label information area on the right.

- Modifying a label: In the Labeling area, click the edit icon, enter the correct label name in the text box, and click the check mark to complete the modification. Alternatively, click a label. In the image labeling area, adjust the position and size of the labeling box. After the adjustment, right-click the labeling box and choose Modify from the shortcut menu. Enter the new label and click Modify to save the modification.
- Deleting a label: In the **Labeling** area, click the deletion icon to delete a label from the image.

After deleting the label, click **Back to Data Labeling Preview** in the upper left corner of the page to exit the labeling page. In the dialog box that is displayed, save the modification. After all labels of an image are deleted, the image is displayed on the **Unlabeled** tab page.

#### Figure 5-64 Editing an object detection label

Labeling	Operation
O Cat	⊡ ∠ ⊽

#### • Modifying based on labels

- On the labeling job details page, click **Label Management** on the right. All label information is displayed.
  - Modifying a label: Click Modify in the Operation column. In the dialog box that is displayed, enter a new label name, select a new label color, and click OK. After the modification, the images that have been added with the label use the new label name.
  - Deleting a label: Click **Delete** in the **Operation** column, or select the label to be deleted and click **Delete Label** above the label list.

#### Figure 5-65 Label Management

All Labels 5	Label Management ${f C}$
Name	Labels 🚛
black Cat	5
Cat	7
person	0
werewr	1
white Cat	5

#### Figure 5-66 All labels

Add Label     Delete Label			
Label Name	Attribute	Label Color	Operation
apple	-		Modify Delete
banana		•	Modify Delete
dog		<ul> <li>• • • • • • • • • • • • • • • • • • •</li></ul>	Modify Delete
object	Rectangle	•	Modify Delete
orange		<ul> <li>•</li> </ul>	Modify Delete

- Alternatively, click **Label** in the **Operation** column of the target labeling job to go to the label management page.

Figure 5-67 Accessing the label management page from the labeling job list

Operation				
Auto Labeling	Label	Publish	More 🕶	

- Click Modify in the Operation column of the target label to modify it.
- Click **Delete** in the **Operation** column of the target label to delete it.

## Adding Data

In addition to the data automatically synchronized from datasets, you can directly add images to labeling jobs for labeling. The added data is first imported to the dataset associated with the labeling job. Then, the labeling job automatically synchronizes the latest data from the dataset.

1. On the labeling job details page, click **All statuses**, **Labeled**, or **Unlabeled** tab, click **Add data** in the upper left corner.

Figure 5-68 Adding data

10000 0000 000		
Add data 🔻	Synchronize New Data	Batch Process Hard Examples

2. Configure the data source, import mode, import path, and labeling status.

A Note: The da latest data in	ta will be imported to the dataset associated with the labeling task, which will then automatically synchronize the the dataset.
Data Source	CHIS Local file
Import Mode	Path manifest You can save the dataset file to be imported to the OBS path that you have permission to access. Labeling file format
Import Path	Select an OBS path.
Labeling Status	Unlabeled Labeled
	OK Carrel
	OK Cancel
	OK Cancel
	<b>OK</b> Cancel
	<b>OK</b> Cancel
mport	OK Cancel
mport	OK Cancel
mport	Cancel
mport	Cancel ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the in the dataset.
Mport	Cancel ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the in the dataset.
Mport Note: The d Latest data	OK     Cancel       ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the nthe dataset.       OB5     Local file
Mport Note: The d latest data t Data Source	OK     Cancel       ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the in the dataset.       OB5     Local file
Mport Note: The d latest data	OK     Cancel       ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the n the dataset.       OB5     Local file       Select an OBS path.
Mport  Note: The d Latest data  * Data Source  * Storage Path Linipading Data	ok       Cancel         ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the nthe dataset.         O85       Local file         Select an O85 path.         Image: Unlocal data
mport Note: The d Latest data * Data Source * Storage Path Uploading Data	ata will be imported to the dataset associated with the labeling task, which will then automatically synchronize the in the dataset.

Figure 5-69 Adding images

3. Click OK.

The images you have added will be automatically displayed in the image list on the **All statuses** tab page. You can choose **Add data** > **View historical records** to view task history.

Figure 5-70 Viewing historical data

View Task Histor	у					×
Created	Import Mode	Import Path	Samples	Imported Samples	Labeled Samples	Import Stat
		N	o data available.			

## **Deleting Images**

You can quickly delete the images you want to discard.

On the **All statuses**, **Unlabeled**, or **Labeled** tab page, select the images to be deleted or click **Select Images on Current Page**, and click **Delete**. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **Yes** to delete the images.

#### Figure 5-71 Deleting images

▼ Filter ~     Selected: 1     Deselect     □     Select Images on Current Page     1	可 Delete
---	----------

If a tick is displayed in the upper left corner of an image, the image is selected. If no image is selected on the page, the **Delete** button is unavailable.

#### **NOTE**

If you select **Delete the source files from OBS**, images stored in the OBS directory will be deleted accordingly. This operation may affect other dataset versions or datasets using those files, for example, leading to an error in page display, training, or inference. Deleted data cannot be recovered. Exercise caution when performing this operation.

#### **Managing Annotators**

If team labeling is enabled for a labeling job, view its labeling details on the **Annotator Management** tab page. Additionally, you can add, modify, or delete annotators.

- Choose Data Management > Label Data. On the My Creations tab page, view the list of all labeling jobs.
- 2. Locate the row that contains the target team labeling job. (The name of a team labeling job is followed by  $\stackrel{\text{P}}{\rightarrow}$  .)
- Choose More > Annotator Management in the Operation column. Alternatively, click the job name to go to the job details page, and choose Team Labeling > Annotator Management in the upper right corner.

Figure 5-72 Annotator Management (1)

	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created J≣	Description	Operation
~	task-aa55 A	dataset-c42d	33% (20/60)	0	May 13, 2021 18:33:15 GMT+08:00	🖉	Auto Labeling   Label   Publish   More 👻
~	🖾 dataset-c42d 🛆	dataset-c42d	33% (20/60)	0	May 13, 2021 16:23:04 GMT+08:00	L	Auto Labeling   Label   Pu Task Statistics
~	🖾 dataset-car-al-test-135	dataset-car-al-test-13	67% (8/12)	0	Apr 26, 2021 09:49:32 GMT+08:00	🖉	Auto Labeling   Label   Pu Annotator Management
~	🖾 dataset-4c9c 🛆	dataset-4c9c	0% (0/200)		Apr 23, 2021 20:34:17 GMT+08:00	🖉	Auto Labeling   Label   Pg Modify

Figure 5-73 Annotator Management (2)

Label	Publish	Task Statistics	Team Labeling 🔻
			Accept
			Stop Acceptance
			Continue Acceptance
7 Delete	All Labels 1		Acceptance Report
	Name		Annotator Management

• Adding an annotator

Click Add Member, select a member name, and click OK.

Click **Send Email** in the **Operation** column to send the labeling job to the annotator by email.

Modifying annotator information

Click **Modify** in the **Operation** column to modify the role of the annotator.

• Deleting an annotator

Click **Delete** in the **Operation** column to delete the annotator.

## 5.11.2.3 Image Segmentation

Training a model uses a large number of labeled images. Therefore, label images before the model training. You can label images on the ModelArts management console. Alternatively, modify labels, or delete them and label them again.

Before labeling an image in image segmentation scenarios, pay attention to the following:

- All objects whose contours need to be extracted from the image must be labeled.
- Polygons can be used for labeling.
  - In polygon labeling, draw a polygon based on the outline of the target object.
- When labeling an image, ensure that the polygons are within the image. Otherwise, an error will occur in subsequent operations.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

## **Starting Labeling**

1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.

2. On the right of the labeling job list, select a labeling type from the job type drop-down list. Click the job to be performed based on the labeling type. The details page of the job is displayed.

# All typesbearchQAll typesQAll typesImage classificationObject detectionImage classificationText classificationImage classificationNamed entity recognitionImage classificationText tripletImage classificationSound classificationImage classification

#### Figure 5-74 Selecting a labeling type

3. The job details page displays all data of the labeling job.

## Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in a dataset, on the **All statuses**, **Unlabeled**, or **Labeled** tab page of the labeling job details page, click **Synchronize New Data**.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

## **Filtering Data**

On the **All statuses**, **Unlabeled**, or tab page, click  $\checkmark$  in the filter criteria area and add filter criteria to quickly filter the data you want to view.

The following filter criteria are available. You can set one or more filter criteria.

- **Example Type**: Select **Hard example** or **Non-hard example**.
- Label: Select All or one or more labels you specified.
- File Name or Path: Filter files by file name or file storage path.
- **Labeled By**: Select the name of the user who labeled the image.

## **Manually Labeling Images**

The labeling job details page displays the **All statuses**, **Unlabeled**, and **Labeled** tab pages. The **Unlabeled** tab page is displayed by default.

- 1. On the **Unlabeled** tab page, click an image. The system automatically directs you to the page for labeling the image. For details about how to use common buttons on this page, see **Table 5-31**.
- 2. Select a labeling method.

On the labeling page, common **labeling methods** and **buttons** are provided in the toolbar. By default, polygon labeling is selected.

#### **NOTE**

After you select a method to label the first image, the labeling method automatically applies to subsequent images.

#### Figure 5-75 Toolbar



#### Table 5-30 Labeling methods

lcon	Description
Ø	Polygon. In the area where the object to be labeled is located, click to label a point, move the mouse and click multiple points along the edge of the object, and then click the first point again. All the points form a polygon. In this way, the object to be labeled is within the bounding box.

#### Table 5-31 Toolbar buttons

Button	Features
<b></b>	Cancel the previous operation.
$\rightarrow$	Redo the previous operation.
Q	Zoom in an image.
Q	Zoom out an image.

Button	Features
Ū	Delete all bounding boxes on the current image.
Ø	Show or hide a bounding box. This operation can be performed only on a labeled image.
÷	Drag a bounding box to another position or drag the edge of the bounding box to resize it.
đ	Reset a bounding box. After dragging a bounding box, you can click this button to quickly restore the bounding box to its original shape and position.
	Display the labeled image in full screen.

3. Label an object.

After labeling an image, click <sup>A</sup> below the image to view in the image list and click an unlabeled image to label the new image.

4. Click **Back to Data Labeling Preview** in the upper left part of the page to view the labeling information. In the dialog box that is displayed, click **Yes** to save the labeling settings.

The selected images are automatically moved to the **Labeled** tab page. On the **Unlabeled** and **All statuses** tab pages, the labeling information is updated along with the labeling process, including the added label names and the number of images for each label.

## Viewing Labeled Images

On the labeling job details page, click the **Labeled** tab to view the list of labeled images. Click an image to view its labeling information in the **File Labels** area on the right.

#### **Quick Review**

To simplify operations, ModelArts provides quick review so that you can batch review and modify labeled data.

- Log in to the ModelArts management console. In the navigation pane, choose Data Management > Label Data. On the My Creations tab page, select the target labeling job type from the All types drop-down list in the upper right corner. (Only object detection and image segmentation support quick review.)
- 2. In the labeling job list, click the target labeling job. The labeling details page is displayed.
- 3. Click **Quick Review** on the **Labeled** tab. On the displayed page, confirm the labeling results.

#### Figure 5-76 Quick Review



- 4. Batch review images of the same label.
  - a. On the review page, select the label type from the drop-down list next to **Filter by Label**.
  - b. Sort images of the selected label type by bounding box area or aspect ratio.
  - c. Click an incorrectly labeled image, and then drag the labeling box to relabel the image. (**Modified** is displayed on the modified images.)
  - d. You can select the incorrectly labeled images, and then click  $\overline{U}$  in the upper right corner to delete the label. (**Deleted** is displayed on the images whose label has been deleted.)

#### Figure 5-77 Modified

Filter by Label person	▼ Sort by Bounding t	ox area ↓Ξ Aspect ratio ↓Ξ	Only images labeled using rectangular boxes can be quickly reviewed. Select Images on Current Pag	e Ū 🗉	<b>€</b> ∂	Apply Modifications	Cancel Modifications
Madilier							
Figure 5-78	Deleted						
Filter by Label person	<ul> <li>Sort by Bounding box area</li> </ul>	■ Aspect ratio 4 Only Im	ages labeled using rectangular boxes can be quickly reviewed.	Apply	Modifications	Cancel Modifications	

Canada

- e. You can also modify the label of a labeled image.
  - i. Select the target images and click  $^{\textcircled{}}$  in the **All Labels** area on the right.
  - ii. Type a new label and click **OK**.

#### Figure 5-79 All Labels



#### Figure 5-80 Adding a label

#### Add Label

5. After the modification, click **Apply Modifications**. In the displayed dialog box, click **OK**. The system automatically returns to the labeling overview page and overwrites the original labeling data.

Figure 5-81 Apply Modifications



6. If you are not satisfied with the modified data, you can click **Cancel Modifications** to retain the original labeling data.

## Figure 5-82 Cancel Modifications

# Cancel Modifications

Are you sure you want to cancel all modifications in this review?

After you click OK, all the modifications in this review will be cancelled.



Table 5-32 Buttons on the quick review page

Button	Features
Ū	Delete the label.
đ	Undo all operations on the current page.
4	Undo the previous operation.
À	Redo the previous operation.

## Modifying a Label

After labeling data, you can modify labeled data on the **Labeled** tab page.

On the labeling details page, click the **Labeled** tab and then the image to be modified. On the labeling page that is displayed, modify the labeling information in the **File Labels** area on the right.

• Modifying a label: In the Labeling area, click the edit icon, set the target label

name or color in the displayed dialog box, and click to save the modification. Alternatively, click a label to be modified. In the image labeling area, adjust the position and size of the bounding box. After the adjustment is complete, click another label to save the modification.

• Deleting a label: In the **Labeling** area, click the deletion icon to delete a label from the image. After all labels of an image are deleted, the image is displayed on the **Unlabeled** tab page.

After the labeling information is modified, click **Back to Data Labeling Preview** in the upper left part of the page to exit the labeling page. In the dialog box that is displayed, click **Yes** to save the modification.

## **Adding Data**

In addition to the data automatically synchronized from datasets, you can directly add images to labeling jobs for labeling. The added data is first imported to the dataset associated with the labeling job. Then, the labeling job automatically synchronizes the latest data from the dataset.

1. On the labeling job details page, click **All statuses**, **Labeled**, or **Unlabeled** tab, click **Add data** in the upper left corner.

#### Figure 5-83 Adding Data

🖪 Synchronize New Data	Batch Process Hard Examples 👘
	Synchronize New Data

×

2. Configure the data source, import mode, import path, and labeling status.

#### Figure 5-84 Adding images

latest data in	a will be imported to the dataset.	the dataset associated	with the labeling ta	sk, which will then	automatically synchro	onize the
Data Source	OBS	Local file				
import Mode	Path You can save the d format	manifest ataset file to be impor	ted to the OBS path	that you have perr	nission to access. Labe	eling file
mport Path	Select an OBS pa	ath.			Ð	
Labeling Status	Unlabeled	Labeled				
		ОК	Cancel			
aport		ок	Cancel			
nport		OK	Cancel			
Note: The da latest data in	ta will be imported to the dataset.	OK the dataset associate	Cancel	ask, which will the	n automatically synch	ronize the
Note: The di latest data in Data Source	ta will be imported to the dataset. OBS	OK D the dataset associate Local file	Cancel	ask, which will the	n automatically synchs	ronize the
Note: The di latest data in Data Source Storage Path	ta will be imported to the dataset. OBS Select an OBS p	OK O the dataset associate Local file	Cancel	ask, which will the	n automatically synch	ronize the
Note: The dr Latest data in Data Source Storage Path Uploading Data	ta will be imported to the dataset. OBS Select an OB5 p O Upload da	or the dataset associated	Cancel	ask, which will the	n automatically synch	ronize the

3. Click OK.

The images you have added will be automatically displayed in the image list on the **All statuses** tab page. You can choose **Add data** > **View historical records** to view task history.

#### Figure 5-85 Viewing historical data



## **Deleting Images**

You can quickly delete the images you want to discard.

On the **All statuses**, **Unlabeled**, or **Labeled** tab page, select the images to be deleted or click **Select Images on Current Page**, and click **Delete** in the upper left corner to delete them. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **Yes** to delete the images.

If a tick is displayed in the upper left corner of an image, the image is selected. If no image is selected on the page, the **Delete** button is unavailable.

#### **NOTE**

If you select **Delete the source files from OBS**, images stored in the OBS directory will be deleted accordingly. This operation may affect other dataset versions or datasets using those files, for example, leading to an error in page display, training, or inference. Deleted data cannot be recovered. Exercise caution when performing this operation.

## 5.11.3 Text Labeling

## 5.11.3.1 Text Classification

Model training requires a large amount of labeled data. Therefore, before the model training, add labels to the files that are not labeled. In addition, you can modify, delete, and re-label the labeled text.

Text classification classifies text content based on labels. Before labeling text content, pay attention to the following:

- Text labeling supports multiple labels. That is, you can add multiple labels to a labeling object.
- A label name can contain a maximum of 1024 characters, including letters, digits, hyphens (-), underscores (\_), and special characters.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

## **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

#### Figure 5-86 Selecting a labeling type

All types	*
5earch	Q
All types	
Image classification	
Object detection	
Text classification	
Named entity recognition	n
Text triplet	
Sound classification	

3. The job details page displays all data of the labeling job.

## Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in the datasets, on the **Unlabeled** tab page of the labeling job details page, click **Synchronize New Data**.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

## Labeling Text Files

The labeling job details page displays the **Unlabeled** and **Labeled** tabs. The **Unlabeled** tab page is displayed by default.

1. On the **Unlabeled** tab page, the objects to be labeled are listed in the left pane. In the list, click the text object to be labeled, and select a label in the **Label Set** area in the right pane. Multiple labels can be added to a labeling object.

You can repeat this operation to select objects and add labels to the objects.

Figure 5-87 Labeling for text classification

Label Set 🕀	label2	label3	label4		
Name	label1				

-			-	
Contont	of the	Laboling	Obi	
Content	u ule	Lavenny	UU	eur

It looks beautiful at a glance. It's probably good to start playing.

2. After all objects are labeled, click **Save Current Page** at the bottom of the page.

## Adding a Label

• Adding labels on the **Unlabeled** tab page: Click the plus sign (+) next to **Label Set**. On the **Add Label** page that is displayed, add a label name, select a label color, and click **OK**.

Figure 5-88 Adding a label (1)

el Set (+)			
label1	label2	label3	label4

• Adding labels on the **Labeled** tab page: Click the plus sign (+) next to **Label Set**. On the **Add Label** page that is displayed, add a label name, select a label color, and click **OK**.

Figure 5-89 Adding a label (2)

Unlabeled 51	Labeled 21120					
Delete		Select Current Page	Label Set (+)	label2	label3	label4

## Viewing the Labeled Text

On the labeling job details page, click the **Labeled** tab to view the list of labeled texts. You can also view all labels supported by the labeling job in the **All Labels** area on the right.

## **Modifying Labeled Data**

After labeling data, you can modify labeled data on the **Labeled** tab page.

• Modifying based on texts

On the labeling job details page, click the **Labeled** tab and select the text to be modified from the text list.

In the text list, click the text. When the text background turns blue, the text is selected. If a text file has multiple labels, you can click above a label to delete the label.

### • Modifying based on labels

On the labeling job details page, click the **Labeled** tab. The information about all labels is displayed on the right.

- Batch modification: In the All Labels area, click the edit icon in the Operation column, modify the label name in the text box, select a label color, and click OK.
- Batch deletion: In the All Labels area, click the deletion icon in the Operation column to delete the label. In the dialog box that is displayed, select Delete the label or Delete the label and objects with only the label, and click OK.

## Adding a File

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** tab, click **Add data** in the upper left corner.
- 2. Configure input data and click **OK**.

For details about how to import data, see section "Importing Data".

#### Figure 5-90 Importing data

Unlabeled 11 Labeled 42	Import	
Add data 🔻 🖳 Synchronize New Data	A Note: The data latest data in t	will be imported to the dataset associated with the labeling task, which will then automatically synchronize the he dataset.
Labeling Objects (Rows in the Uploaded File)	* Data Source	OBS Local file
🖉 Unlab	* Import Mode	Path manifest You can save the dataset file to be imported to the OBS path that you have permission to access. Labeling file format
🕲 Unlabs	* Import Path	Select an OBS path.
🖉 Unlabi	* Labeling Status	Unlabeled Labeled
🖉 Unlabi		OK Cancel

## **Deleting a File**

You can quickly delete the files you want to discard.

- On the **Unlabeled** tab page, select the text to be deleted, and click **Delete** in the upper left corner to delete the text.
- On the **Labeled** tab page, select the text to be deleted and click **Delete**. Alternatively, tick **Select Current Page** to select all text objects on the current page and click **Delete** in the upper left corner.

The background of the selected text is blue.

#### **Managing Annotators**

If team labeling is enabled for a labeling job, view its labeling details on the **Annotator Management** tab page. Additionally, you can add, modify, or delete annotators.

- 1. Choose **Data Management** > **Label Data**. On the **My Creations** tab page, view the list of all labeling jobs.
- 2. Locate the row that contains the target team labeling job. (The name of a team labeling job is followed by  $\mathcal{P}_{\cdot}$ .)
- Choose More > Annotator Management in the Operation column. Alternatively, click the job name to go to the job details page, and choose Team Labeling > Annotator Management in the upper right corner.

#### Figure 5-91 Annotator Management (1)

	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created 4≣	Description	Operation
~	Ed task-f98d ,A	dataset-e8df	0% (0/2)		Mar 29, 2023 14:52:46 GMT+08:00		Auto Labeling   Label   Publish   More -
~	🖼 task-2a81	dataset-1c68	0% (1/1618)		Nov 13, 2022 15:58:21 GMT+08:00	@	Auto Labeling   Label   Pu Task Statistics
~	El task-c010	dataset-1c68	1% (10/1663)		Nov 01, 2022 15:58:08 GMT+08:00	@	Auto Labeling   Label   PL Annotator Management
~	🖬 dataset-63e0 🛆	dataset-63e0	79% (854/1083)		Oct 17, 2022 09:59:07 GMT+08:00	@	Auto Labeling   Label   Pu Modify

Figure 5-92 Annotator Management (2)

Label	Publish	Task Statistics	Team Labeling 🔻
			Accept
			Stop Acceptance
			Continue Acceptance
J Delete	All Labels 1		Acceptance Report
	Name		Annotator Management

• Adding an annotator

Click Add Member, select a member name, and click OK.

Click **Send Email** in the **Operation** column to send the labeling job to the annotator by email.

- Modifying annotator information
   Click Modify in the Operation column to modify the role of the annotator.
- Deleting an annotator
   Click **Delete** in the **Operation** column to delete the annotator.

## 5.11.3.2 Named Entity Recognition

Named entity recognition assigns labels to named entities in text, such as time and locations. Before labeling, pay attention to the following:

A label name of a named entity can contain a maximum of 1024 characters, including letters, digits, hyphens (-), underscores (\_), and special characters.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

## **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

#### Figure 5-93 Selecting a labeling type



3. The job details page displays all data of the labeling job.

## Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in the datasets, on the **Unlabeled** tab page of the labeling job details page, click **Synchronize New Data**.

**NOTE** 

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

#### Labeling Text Files

The labeling job details page displays the **Unlabeled** and **Labeled** tabs. The **Unlabeled** tab page is displayed by default.

1. On the **Unlabeled** tab page, the objects to be labeled are listed in the left pane. In the list, click the text object to be labeled, select a part of text displayed under **Label Set** for labeling, and select a label in the **Label Set** area in the right pane.

You can repeat this operation to select objects and add labels to the objects.

2. Click **Save Current Page** in the lower part of the page to complete the labeling.

## Adding a Label

 Adding labels on the Unlabeled tab page: Click the plus sign (+) next to Label Set. On the Add Label page that is displayed, add a label name, select a label color, and click OK.

Figure 5-94 Adding a named entity label (1)

Label Set	$\oplus$		
Person		Place	Time

• Adding labels on the **Labeled** tab page: Click the plus sign (+) next to **Label Set**. On the **Add Label** page that is displayed, add a label name, select a label color, and click **OK**.

Figure 5-95 Adding a named entity laber (2	igure 5-9	ة Adding ،	a named	entity	label	(2)
--	-----------	------------	---------	--------	-------	-----

Label Set 🕀
Person Place Time

## Viewing the Labeled Text

On the dataset details page, click the **Labeled** tab to view the list of the labeled text. You can also view all labels supported by the dataset in the **All Labels** area on the right.

## **Modifying Labeled Data**

After labeling data, you can modify labeled data on the **Labeled** tab page.

On the labeling job details page, click the **Labeled** tab, and modify the text information in the label information area on the right.

#### • Modifying based on texts

On the labeling job details page, click the **Labeled** tab, and select the text to be modified from the text list.

Manual deletion: In the text list, click the text. When the text background turns blue, the text is selected. On the right of the page, click  $\times$  above a text label to delete the label.

#### • Modifying based on labels

On the labeling job details page, click the **Labeled** tab. The information about all labels is displayed on the right.

- Batch modification: In the All Labels area, click the edit icon in the Operation column, add a label name in the text box, select a label color, and click OK.
- Batch deletion: In the All Labels area, click the deletion icon in the Operation column to delete the label. In the dialog box that is displayed, select Delete the label or Delete the label and objects with only the label, and click OK.

## Adding a File

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** tab, click **Add data** in the upper left corner.
- 2. Configure input data and click **OK**.

For details about how to import data, see section "Importing Data".

#### Figure 5-96 Importing data

Unlabeled 1993 Labeled 7	Import	
id data 🔻	A Note: The data will be imported to the dataset associated w latest data in the dataset.	th the labeling task, which will then automatically synchronize the
abeling Objects (Rows in the Uploaded File)	* Data Source 085 Local file	
	* Import Mode Path manifest You can save the dataset file to be importe format	I to the OBS path that you have permission to access. Labeling file
	* Import Path Select an OBS path.	Đ
	* Labeling Status Unlabeled Labeled	
	_	
	ОК	Cancel

## Deleting a File

You can quickly delete the files you want to discard.

- On the **Unlabeled** tab page, select the text to be deleted, and click **Delete** in the upper left corner to delete the text.
- On the **Labeled** tab page, select the text to be deleted and click **Delete**. Alternatively, tick **Select Current Page** to select all text objects on the current page and click **Delete** in the upper left corner.

The background of the selected text is blue.

#### **Managing Annotators**

If team labeling is enabled for a labeling job, view its labeling details on the **Annotator Management** tab page. Additionally, you can add, modify, or delete annotators.

- 1. Choose **Data Management** > **Label Data**. On the **My Creations** tab page, view the list of all labeling jobs.
- 2. Locate the row that contains the target team labeling job. (The name of a team labeling job is followed by  $\mathcal{P}_{\cdot}$ .)
- Choose More > Annotator Management in the Operation column. Alternatively, click the job name to go to the job details page, and choose Team Labeling > Annotator Management in the upper right corner.

#### Figure 5-97 Annotator Management (1)

	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created ↓≣	Description	Operation
~	El task-f98d A	dataset-e8df	0% (0/2)		Mar 29, 2023 14:52:46 GMT+08:00		Auto Labeling   Label   Publish   More -
~	🖾 task-2a81	dataset-1c68			Nov 13, 2022 15:58:21 GMT+08:00	🖉	Auto Labeling   Label   Pu Task Statistics
~	task-c010	dataset-1c68			Nov 01, 2022 15:58:08 GMT+08:00	🖉	Auto Labeling   Label   PL Annotator Management
~	⊠ dataset-63e0 🛆	dataset-63e0	79% (854/1083)		Oct 17, 2022 09:59:07 GMT+08:00	🖉	Auto Labeling   Label   Pu Modify

Figure 5-98 Annotator Management (2)

Publish	Task Statistics	Team Labeling 🔻
		Accept
		Stop Acceptance
		Continue Acceptance
All Labels 1		Acceptance Report
Name		Annotator Management
	Publish All Labels 1 Name	Publish Task Statistics All Labels 1 Name

• Adding an annotator

Click Add Member, select a member name, and click OK.

Click **Send Email** in the **Operation** column to send the labeling job to the annotator by email.

Modifying annotator information

Click **Modify** in the **Operation** column to modify the role of the annotator.

• Deleting an annotator

Click **Delete** in the **Operation** column to delete the annotator.

## 5.11.3.3 Text Triplet

Triplet labeling is suitable for scenarios where structured information, such as subjects, predicates, and objects, needs to be labeled in statements. With this function, not only entities in statements, but also relationships between entities can be labeled. Triplet labeling is often used in natural language processing tasks such as dependency syntax analysis and information extraction.

Text triplet labeling involves two classes of important labels: **Entity Label** and **Relationship Label**. For **Relationship Label**, set its **Source entity** and **Target entity**.

- You can define multiple entity and relationship labels for a text object.
- The **Entity Label** defined during dataset creation cannot be deleted.

## Precautions

Before labeling, ensure that the **Entity Label** and **Relationship Label** of a labeling job have been defined. For **Relationship Label**, set its **Source entity** and **Target entity**. **Relationship Label** must be between the defined **Source entity** and **Target entity**.

For example, if two entities are labeled as **Place**, you cannot add any relationship label between them. If a relationship label cannot be added, a red cross is displayed.

#### D NOTE

Data management is being upgraded and is invisible to users who have not used data management.

## Starting Labeling

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

All types	*
Search	Q
All types	
Image classification	
Object detection	
Text classification	1
Named entity recognition	n
Text triplet	
Sound classification	

Figure 5-99 Selecting a labeling type

3. The job details page displays all data of the labeling job.

## Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in the datasets, on the **Unlabeled** tab page of the labeling job details page, click **Synchronize New Data**.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

## **Labeling Text Files**

The labeling job details page displays the **Unlabeled** and **Labeled** tabs. The **Unlabeled** tab page is displayed by default.

1. On the **Unlabeled** tab page, the objects to be labeled are listed in the left pane. In the list, click a text object, select the corresponding text content on the right pane, and select an entity name from the displayed entity list to label the content.

Figure 5-100 Labeling an entity



2. After labeling multiple entities, click the source entity and target entity in sequence and select a relationship type from the displayed relationship list to label the relationship.

Figure 5-101 Labeling a relationship

/	k	oirthplace	e	·¥					
Per				Place					
Eric	was	born	in	BeiJing,	but	he	lives	in	HangZhou.

3. After all objects are labeled, click **Save Current Page** at the bottom of the page.

#### **NOTE**

You cannot modify the labels of a dataset in the text triplet type on the labeling page. Instead, click **Label Management** and modify the **Entity Label** and **Relationship Label**.

## Modifying Labeled Data

After labeling data, you can modify labeled data on the **Labeled** tab page.

On the labeling job details page, click the **Labeled** tab. Select a text object in the left pane and the right pane displays the detailed label information. You can move your cursor to the entity or relationship label, and right-click to delete it. You can also click the source entity and target entity in sequence to add a relationship label.

You can click **Delete Labels on Current Item** at the bottom of the page to delete all labels in the selected text object.

## Adding a File

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** tab, click **Add data** in the upper left corner.
- 2. Configure input data and click **OK**.

For details about how to import data, see section "Importing Data".

#### Figure 5-102 Importing data

Labeling Task Statistics Label Management Unlabeled 52 Labeled 1	Import	Х	
Add data 🔻	A Note: The data will be imported to the dataset associated with the labeling task, which will then automatically synchronize the latest data in the dataset.		
Labeling Objects (Rows in the Uploaded File)	* Data Source OBS Local file		
🖉 Unla	* Import Mode Path manifest You can save the dataset file to be imported to the OBS path that you have permission to access. Labeling file		
🖉 Unla	format		
🖉 Unia	* Import Path Select an OBS path.		
🖉 Unla	* Labeling Status Unlabeled Labeled		
🖉 Unla	OK Cancel		

## Deleting a File

You can quickly delete the files you want to discard.

- On the **Unlabeled** tab page, select the text to be deleted, and click **Delete** in the upper left corner to delete the text.
- On the **Labeled** tab page, select the text to be deleted and click **Delete**. Alternatively, tick **Select Current Page** to select all text objects on the current page and click **Delete** in the upper left corner.

The background of the selected text is blue. If no text is selected on the page, the **Delete** button is unavailable.

## **Managing Annotators**

If team labeling is enabled for a labeling job, view its labeling details on the **Annotator Management** tab page. Additionally, you can add, modify, or delete annotators.

1. Choose **Data Management** > **Label Data**. On the **My Creations** tab page, view the list of all labeling jobs.

- 2. Locate the row that contains the target team labeling job. (The name of a team labeling job is followed by  $\mathcal{P}_{\cdot}$ .)
- 3. Choose More > Annotator Management in the Operation column. Alternatively, click the job name to go to the job details page, and choose Team Labeling > Annotator Management in the upper right corner.

Figure	5-103	Annotator	Management	(1	)
--------	-------	-----------	------------	----	---

	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created ↓Ⅲ	Description	Operation
~	Ed task-f98d A	dataset-e8df	0% (0/2)		Mar 29, 2023 14:52:46 GMT+08:00		Auto Labeling   Label   Publish   More -
~	🖾 task-2a81	dataset-1c68			Nov 13, 2022 15:58:21 GMT+08:00	2	Auto Labeling   Label   Pu Task Statistics
~	🖾 task-c010	dataset-1c68			Nov 01, 2022 15:58:08 GMT+08:00	2	Auto Labeling   Label   PL Annotator Management
~	🖬 dataset-63e0 🛆	dataset-63e0	79% (854/1083)		Oct 17, 2022 09:59:07 GMT+08:00	@	Auto Labeling   Label   Pu Modify

Figure 5-104 Annotator Management (2)

Label	Publish	Task Statistics	Team Labeling 🔻
			Accept
			Stop Acceptance
			Continue Acceptance
T Delete	All Labels 1		Acceptance Report
	Name		Annotator Management

• Adding an annotator

Click Add Member, select a member name, and click OK.

Click **Send Email** in the **Operation** column to send the labeling job to the annotator by email.

- Modifying annotator information
   Click Modify in the Operation column to modify the role of the annotator.
- Deleting an annotator
   Click **Delete** in the **Operation** column to delete the annotator.

# 5.11.4 Audio Labeling

## 5.11.4.1 Sound Classification

Model training requires a large amount of labeled data. Therefore, before the model training, label the unlabeled audio files. ModelArts enables you to label audio files in batches by one click. In addition, you can modify the labels of audio files, or remove their labels and label the audio files again.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

## **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

Figure 5-105 Selecting a labeling type

All types	•
þearch	Q
Named entity recognition	1
Text triplet	
Sound classification	
Speech labeling	_
Speech paragraph labelin	g
Video labeling	
Image segmentation	

3. The job details page displays all data of the labeling job.

## Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in the datasets, on the **Unlabeled** or **Labeled** tab page of the labeling job details page, click **Synchronize New Data**.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

## **Labeling Audio Files**

The labeling job details page displays the Unlabeled and Labeled tabs. The

**Unlabeled** tab page is displayed by default. Click  $\bigcirc$  on the left of the audio to preview the audio.

- 1. On the **Unlabeled** tab page, select the audio files to be labeled.
  - Manual selection: In the audio list, click the target audio. If the blue check box is displayed in the upper right corner, the audio is selected. You can select multiple audio files of the same type and label them together.
  - Batch selection: If all audio files of the current page belong to one type, you can click **Select Current Page** in the upper right corner of the list to select all the audio files on the page.
- 2. Add labels.
  - a. In the label adding area on the right, set a label in the **Label** text box.

Method 1 (the required label already exists): In the right pane, select a shortcut from the **Shortcut** drop-down list, select an existing label name from the **Label** text box, and click **OK**.

Method 2 (adding a label): In the right pane, select a shortcut from the **Shortcut** drop-down list, and enter a new label name in the **Label** text box.

b. The selected audio files are automatically moved to the **Labeled** tab page. On the **Unlabeled** tab page, the labeling information is updated along with the labeling process, including the added label names and the number of audio files corresponding to each label.

#### **NOTE**

**Shortcut key description**: After specifying a shortcut key for a label, you can select an audio file and press the shortcut key to add a label for the audio file. Example: Specify 1 as the shortcut key for the **aa** label. Select one or more files and press 1. A message is displayed, asking you whether to label the files with **aa**. Click **OK**.

Each label has a shortcut key. A shortcut key cannot be specified for different labels. Shortcut keys can greatly improve the labeling efficiency.

#### Figure 5-106 Adding an audio label

Selected Images	Selected	1 sound files.			
Label	test				
Shortcut (?)	1 .	•			
1	ОК	Cancel			
#### Viewing the Labeled Audio Files

On the labeling job details page, click the **Labeled** tab to view the list of labeled audio files. Click an audio file. You can view the label information about the audio file in the **File Labels** area on the right.

#### Modifying Labeled Data

After labeling data, you can modify labeled data on the **Labeled** tab page.

• Modifying based on audio

On the labeling job details page, click the **Labeled** tab. Select one or more audio files to be modified from the audio list. Modify the label in the label details area on the right.

- Modifying a label: In the File Labels area, click the edit icon in the Operation column, enter the correct label name in the text box, and click the check mark to complete the modification.
- Deleting a label: In the **File Labels** area, click the delete icon in the **Operation** column to delete the label.
- Modifying based on labels

On the labeling job details page, click the **Labeled** tab. The information about all labels is displayed on the right.

All Labels 4		Label Man	agement C
Label	Count 🚛	Shortcut	Operation
8	1	6	2 Ū
8888	7	1	LŪ
qqqq	1	2	₽Ū
test	1		20

#### Figure 5-107 Information about all labels

- Modifying a label: Click the edit icon in the **Operation** column. In the dialog box that is displayed, enter the new label name and click **OK**. After the modification, the new label applies to the audio files that contain the original label.
- Deleting a label: Click the deletion icon in the **Operation** column. In the displayed dialog box, select the object to be deleted as prompted and click **OK**.

#### **Adding Audio Files**

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** or **Labeled** tab, click **Add data** in the upper left corner.
- Configure input data and click **OK**.
   For details about how to import data, see section "Importing Data".

#### Figure 5-108 Importing data

Labeling Task Statistics Label			Х	
Unlabeled 2 Labeled 1	Import			n
Add data 🔹	A Note: The data latest data in f	a will be imported to the dataset associated with the labeling task, which will then automatically synchronize the the dataset.		ļe
	* Data Source	COBS Local file		
1_1606477469828.wav	* Import Mode	Path manifest You can save the dataset file to be imported to the OBS path that you have permission to access. Labeling file format		
	★ Import Path	Select an OBS path.		l
	* Labeling Status	Unlabeled Labeled		
		OK Cancel		

#### **Deleting Audio Files**

You can quickly delete the audio files you want to discard.

On the **Unlabeled** or **Labeled** tab page, select the audio files to be deleted one by one or tick **Select Current Page** to select all audio files on the page, and then click **Delete File** in the upper left corner. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **OK** to delete the audio files.

If a tick is displayed in the upper right corner of an audio file, the audio file is selected. If no audio file is selected on the page, the **Delete File** button is unavailable.

#### **NOTE**

If you select **Delete the source files from OBS**, audio files stored in the corresponding OBS directory will be deleted when you delete the selected audio files. Deleting source files may affect other dataset versions or datasets using those files. As a result, the page display, training, or inference is abnormal. Deleted data cannot be recovered. Exercise caution when performing this operation.

#### 5.11.4.2 Speech Labeling

Model training requires a large amount of labeled data. Therefore, before the model training, label the unlabeled audio files. ModelArts enables you to label

audio files in batches by one click. In addition, you can modify the labels of audio files, or remove their labels and label the audio files again.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

#### **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

All types	*
Search	Q
Named entity recognition	on
Text triplet	
Sound classification	
Speech labeling	
Speech paragraph label	ing
Video labeling	
Image segmentation	

Figure 5-109 Selecting a labeling type

3. The job details page displays all data of the labeling job.

#### Synchronizing New Data

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in the datasets, on the **Unlabeled** tab page of the labeling job details page, click **Synchronize New Data**.

#### D NOTE

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

#### **Labeling Audio Files**

The labeling job details page displays the labeled and unlabeled audio files. The **Unlabeled** tab page is displayed by default.

1. In the audio file list on the **Unlabeled** tab page, click the target audio file. In

the area on the right, the audio file is displayed. Click 🕑 below the audio file to play the audio.

- 2. In Speech Content, enter the speech content.
- 3. After entering the content, click **Label** to complete the labeling. The audio file is automatically moved to the **Labeled** tab page.

#### Figure 5-110 Labeling speech content

1-11687.4-47_1558680418773 wav	(1) (1)
D Play	00:00:00/00:00:05
Speech Content:	
Enter the speech content.	
	0/256

#### Viewing the Labeled Audio Files

On the labeling job details page, click the **Labeled** tab to view the list of labeled audio files. Click the audio file to view the audio content in the **Speech Content** text box on the right.

#### Modifying Labeled Data

After labeling data, you can modify labeled data on the Labeled tab page.

On the labeling job details page, click the **Labeled** tab and select the audio file to be modified from the audio file list. In the label information area on the right, modify the content of the **Speech Content** text box, and click **Label** to complete the modification.

#### Adding an Audio File

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** tab, click **Add data** in the upper left corner.
- 2. Configure input data and click **OK**.

For details about how to import data, see section "Importing Data".

#### Figure 5-111 Importing data

Lucing	
Unlabeled 2 Labeled 1	
Add data 🔻 🗓 Delete File 🕅 🕅 Import	×
1_1606477181283.wav Duration:00:00:21  Note: The data will be imported to the dataset associated with the labeling task, which will then automatically synchronize the	he
0_1606477181500.wav latest data in the dataset. Duration:00:00:19	
* Data Source OBS Local file	
* Import Mode Path manifest You can save the dataset file to be imported to the OBS path that you have permission to access. Labeling fi format	le
★ Import Path Select an OBS path.	
* Labeling Status Unlabeled Labeled	
OK Cancel	

#### **Deleting Audio Files**

You can quickly delete the audio files you want to discard.

On the **Unlabeled** or **Labeled** tab page, select the audio files to be deleted, and then click **Delete File** in the upper left corner. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **OK** to delete the audio files.

#### **NOTE**

If you select **Delete the source files from OBS**, audio files stored in the corresponding OBS directory will be deleted when you delete the selected audio files. Deleting source files may affect other dataset versions or datasets using those files. As a result, the page display, training, or inference is abnormal. Deleted data cannot be recovered. Exercise caution when performing this operation.

#### 5.11.4.3 Speech Paragraph Labeling

Model training requires a large amount of labeled data. Therefore, before the model training, label the unlabeled audio files. ModelArts enables you to label audio files. In addition, you can modify the labels of audio files, or remove their labels and label the audio files again.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

#### **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

All types	•
Search	Q
Sound classification	
Speech labeling	
Speech paragraph labeling	
Video labeling	-1
Image segmentation	
Head Posture Estimation	
Facial Keypoint Labeling	

#### Figure 5-112 Selecting a labeling type

3. The job details page displays all data of the labeling job.

#### Synchronizing Data Sources

ModelArts automatically synchronizes data and labeling information from datasets to the labeling job.

To quickly obtain the latest data in the OBS bucket, click **Synchronize Data Source** on the **Unlabeled** tab page of the labeling job details page to add the data uploaded using OBS to the dataset.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

#### Labeling Audio Files

The labeling job details page displays the **Unlabeled** and **Labeled** tabs. The **Unlabeled** tab page is displayed by default.

1. In the audio file list on the **Unlabeled** tab page, click the target audio file. In

the area on the right, the audio file is displayed. Click 🕑 below the audio file to play the audio.

2. Select an audio segment based on the content being played, and enter the audio file label and content in the **Speech Content** text box.

Figure 5-113 Speech paragraph labeling

· · · · · · · · · · · · · · · · · · ·	 	-least all arms	 and a final sector of the	an a distance of	for some for some so	1
Play					00:00	00/00:00:
peech Content:						
Enter the speech content.						
						0/2
	-	Label				

3. After entering the content, click **Label** to complete the labeling. The audio file is automatically moved to the **Labeled** tab page.

#### Viewing the Labeled Audio Files

On the labeling job details page, click the **Labeled** tab to view the list of labeled audio files. Click the audio file to view the labeling information on the right.

#### Modifying Labeled Data

After labeling data, you can modify labeled data on the **Labeled** tab page.

- Modifying a label: On the labeling details page, click the Labeled tab, and select the audio file to be modified from the audio file list. In the right area, modify labeling information and click Label to complete the modification.
- Deleting a label: Click **Delete** in the **Operation** column of the target number to delete the label of the audio segment. Alternatively, you can click above the labeled audio file to delete the label. Then click **Label**.

#### Adding an Audio File

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** tab, click **Add data** in the upper left corner.
- 2. Configure input data and click **OK**.

#### Figure 5-114 Importing data

Labeling     Task Statistics     Label       Unlabeled 2     Labeled 1       Add data •	Import  Note: The data will be imported to the dataset associated with the labeling task, which will then automatically synchronize the latest data in the dataset.	×	
00:0021 0_1606477181500.wav 00:00:19	* Data Source     OBS     Local file       * Import Mode     Path     manifest       You can save the dataset file to be imported to the OBS path that you have permission to access. Labeling file format       * Import Path     Select an OBS path.		 
	* Labeling Status Unlabeled Labeled OK Cancel		

#### **Deleting Audio Files**

You can quickly delete the audio files you want to discard.

On the **Unlabeled** or **Labeled** tab page, select the audio files to be deleted, and then click **Delete File** in the upper left corner. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **OK** to delete the audio files.

#### D NOTE

If you select **Delete the source files from OBS**, audio files stored in the corresponding OBS directory will be deleted when you delete the selected audio files. Deleting source files may affect other dataset versions or datasets using those files. As a result, the page display, training, or inference is abnormal. Deleted data cannot be recovered. Exercise caution when performing this operation.

#### **Managing Annotators**

If team labeling is enabled for a labeling job, view its labeling details on the **Annotator Management** tab page. Additionally, you can add, modify, or delete annotators.

- 1. Choose **Data Management** > **Label Data**. On the **My Creations** tab page, view the list of all labeling jobs.
- 2. Locate the row that contains the target team labeling job. (The name of a team labeling job is followed by  $\mathcal{P}_{\cdot}$ .)
- Choose More > Annotator Management in the Operation column. Alternatively, click the job name to go to the job details page, and choose Team Labeling > Annotator Management in the upper right corner.

#### Figure 5-115 Annotator Management (1)

	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created ↓≣	Description	Operation
~	Eil task-f98d A	dataset-e8df	0% (0/2)		Mar 29, 2023 14:52:46 GMT+08:00		Auto Labeling   Label   Publish   More +
~	🖼 task-2a81	dataset-1c68	0% (1/1618)		Nov 13, 2022 15:58:21 GMT+08:00	🖉	Auto Labeling   Label   Pu Task Statistics
~	El task-c010	dataset-1c68			Nov 01, 2022 15:58:08 GMT+08:00	🖉	Auto Labeling   Label   PL Annotator Management
~	🖬 dataset-63e0 🛆	dataset-63e0	79% (854/1083)		Oct 17, 2022 09:59:07 GMT+08:00	🖉	Auto Labeling   Label   Pu Modify

Figure 5-116 Annotator Management (2)

Label	Publish	Task Statistics	Team Labeling 🔻
			Accept
			Stop Acceptance
			Continue Acceptance
Delete	All Labels 1		Acceptance Report
	Name		Annotator Management

• Adding an annotator

Click Add Member, select a member name, and click OK.

Click **Send Email** in the **Operation** column to send the labeling job to the annotator by email.

- Modifying annotator information
   Click Modify in the Operation column to modify the role of the annotator.
- Deleting an annotator

Click **Delete** in the **Operation** column to delete the annotator.

# 5.11.5 Video Labeling

Model training requires a large amount of labeled video data. Therefore, before the model training, label the unlabeled video files. ModelArts enables you to label video files. In addition, you can modify the labels of video files, or remove their labels and label the video files again.

#### **NOTE**

- Video labeling applies only to video frames.
- Data management is being upgraded and is invisible to users who have not used data management.

#### **Starting Labeling**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- 2. In the labeling job list, select a labeling type from the **All type** drop-down list, click the job to be performed based on the labeling type. The details page of the job is displayed.

#### Figure 5-117 Selecting a labeling type

All types	*
Search	Q
Sound classification	
Speech labeling	
Speech paragraph labeling	
Video labeling	
Image segmentation	
Head Posture Estimation	
Facial Keypoint Labeling	

3. The job details page displays all data of the labeling job.

#### Synchronizing Data Sources

ModelArts automatically synchronizes data and labeling information from **Input Dataset Path** to the dataset details page.

To quickly obtain the latest data in the OBS bucket, on the **Labeled** or **Unlabeled** tab page of the labeling job details page, click **Synchronize Data Source**.

#### **NOTE**

Symptom:

After the labeled data is uploaded to OBS and synchronized, the data is displayed as unlabeled.

Possible causes:

Automatic encryption is enabled in the OBS bucket.

Solution:

Create an OBS bucket and upload data again, or disable bucket encryption and upload data again.

#### Video Labeling

The labeling job details page displays the **Unlabeled**, **Labeled**, and **All statuses** tabs.

- 1. On the **Unlabeled** tab page, click the target video file in the video list on the left. The labeling page is displayed.
- 2. Play the video. When the video is played to the time point to be labeled, click the pause button in the progress bar to pause the video to a specific image.
- 3. In the upper pane, select a bounding box. By default, a rectangular box is selected. Drag the mouse to select an object in the video image, enter a new label name in the displayed **Add Label** text box, select a label color, and click **Add** to label the object. Alternatively, select an existing label from the drop-

down list and click **Add** to label the object. Label all objects in the image. Multiple labels can be added to an image.

The supported labeling boxes are the same as those for object detection. For details, see **Common icons on the labeling page**.

Figure 5-118 Video labeling



4. After the previous image is labeled, click the play button on the progress bar to resume the playback. Then, repeat **3** to complete labeling on the entire video.

Click **Label List** in the upper right corner of the page. The time points when the video is labeled are displayed.

#### Figure 5-119 File labels

File Labels



5. Click **Back to Data Labeling Preview** in the upper left corner of the page. The labeling job details page is displayed, and the labeled video file is displayed on the **Labeled** tab page.

#### FAQs

Q: What can I do if the video dataset cannot be displayed or videos cannot be played?

A: If this issue occurs, check the video format. Only MP4 videos can be displayed and played.

#### **Modifying Labeled Data**

After labeling data, you can modify labeled data on the **Labeled** tab page.

- On the Labeled tab page, click the target video file. In the upper right corner of the labeling page, click Label List to go to the File Labels page. You can click the triangle icon on the right of the time point to view details, modify labels, and delete labels.
- Modifying a label: On the **File Labels** area, click the edit button on the right of a label to modify it.
- Deleting a label: On the **File Labels** area, click the delete button on the right of a label to delete it. If you click the delete icon on the right of the image time, all labels on the image are deleted.

#### Figure 5-120 Modifying Labeled Data

File Labels	
00:00:07	Ū ^
dog	🗆 🖉 ប៊
00:00:02	Ū ^
dog	🗆 🖉 ប៊
00:00:09	Ū ^
dog	🗆 🖉 ប៊

#### Adding Video Files

In addition to the data synchronized, you can directly add data on labeling job details page for labeling.

- 1. On the labeling job details page, click the **Unlabeled** or **Labeled** tab, click **Add data** in the upper left corner.
- 2. Configure the data source, import mode, and other parameters, and click **OK**.

#### Figure 5-121 Importing data

ibening i ius		Label Management	
	K Statistics	Laber Management	
All 6 Unlab	eled 6 Labe	led 0	
can import 999994 mor	re samples and 10000	more labels	
Add data 👻 🗍	j Delete	Synchronize New Data Bate	ch Process Hard Examples 💌
Import			
A Neter The de	the second s	the desces consisted with the labor	ling and which will also a standard when the
A Note: The da	ata will be imported to	the dataset associated with the labe	ung task, which will then automatically synchronize the
tatest uata ii	i ule uduset.		
* Data Source	OBS	Local file	
* Data Source	OBS	Local file	
* Data Source	OBS Path	Local file manifest	
* Data Source * Import Mode	OBS Path You can save the	Local file manifest dataset file to be imported to the OBS	S path that you have permission to access. Labeling file
* Data Source * Import Mode	OBS Path You can save the format	Local file manifest dataset file to be imported to the OB	S path that you have permission to access. Labeling file
* Data Source * Import Mode * Import Path	OBS Path You can save the format	Local file manifest dataset file to be imported to the OB ath.	S path that you have permission to access. Labeling file
* Data Source * Import Mode * Import Path	OBS Path You can save the format Select an OBS p You can import	Local file manifest dataset file to be imported to the OBS ath.	S path that you have permission to access. Labeling file
* Data Source * Import Mode * Import Path	OBS Path You can save the format Select an OBS p You can import	Local file manifest dataset file to be imported to the OBS ath. 999994 more samples and 10000 mo	S path that you have permission to access. Labeling file           Image: Constraint of the second se
* Data Source * Import Mode * Import Path * Labeling Status	OBS Path You can save the format Select an OBS p You can import Unlabeled	Local file manifest dataset file to be imported to the OBS ath. 999994 more samples and 10000 mo	S path that you have permission to access. Labeling file
* Data Source * Import Mode * Import Path * Labeling Status	OBS Path You can save the format Select an OBS p You can import Unlabeled	Local file manifest dataset file to be imported to the OBS ath. 9999994 more samples and 10000 mo Labeled	S path that you have permission to access. Labeling file
* Data Source * Import Mode * Import Path * Labeling Status	OBS Path You can save the format Select an OBS p You can import Unlabeled	Local file manifest dataset file to be imported to the OBS ath. 999994 more samples and 10000 mo Labeled	S path that you have permission to access. Labeling file
* Data Source * Import Mode * Import Path * Labeling Status	OBS Path You can save the format Select an OBS p You can import Unlabeled	Local file manifest dataset file to be imported to the OBS ath. 999994 more samples and 10000 mo Labeled OK Cancel	S path that you have permission to access. Labeling file re labels.

#### Deleting a Video File

You can quickly delete the video files you want to discard.

On the **All statuses**, **Unlabeled**, or **Labeled** tab page, select the video files to be deleted or click **Select Images on Current Page** to select all video files on the page, and click **Delete** in the upper part to delete the video files. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **OK** to delete the videos.

If a tick is displayed in the upper left corner of a video file, the video file is selected. If no video file is selected on the page, the **Delete** button is unavailable.

#### **NOTE**

If you select **Delete the source files from OBS**, video files stored in the corresponding OBS directory will be deleted when you delete the selected video files. Deleting source files may affect other dataset versions or datasets using those files. As a result, the page display, training, or inference is abnormal. Deleted data cannot be recovered. Exercise caution when performing this operation.

# 5.11.6 Viewing Labeling Jobs

#### 5.11.6.1 Viewing My Created Labeling Jobs

On the ModelArts **Data Labeling** page, view your created labeling jobs on the **My Creations** tab page.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

#### Procedure

- Log in to the ModelArts management console. In the left navigation pane, choose Data Management > Label Data. The Data Labeling page is displayed.
- 2. On the **My Creations** tab, view all labeling jobs created by you. You can view information about these labeling jobs.

#### Figure 5-122 My Creations

My Cre	ations My Participati	ons						
							All types 💌	Enter a name.
	Name	Dataset	Labeling Progress (Labeled/Total)	Pending Confir	Created ↓Ξ	Description	Operation	
~	a task-1fcd	dataset-c7ce	100% (291/291)	-	Jul 24, 2022 17:32:57 GMT+08:00	@	Auto Labeling   Label	Publish   More 🔻
~	🖸 dataset-f606 🔒	dataset-f606	0% (0/311)		Jul 23, 2022 15:20:49 GMT+08:00	🖉	Auto Labeling   Label	Publish   More 🔻
~	111 dataset-d56c 🔒	dataset-d56c	<ul> <li>4% (1/23)</li> </ul>	-	Jul 23, 2022 14:31:31 GMT+08:00	@	Label   Publish   Task St	atistics Modify

#### Copying a Labeling Job

- Log in to the ModelArts management console. In the left navigation pane, choose Data Management > Label Data. The Data Labeling page is displayed.
- 2. On the **My Creations** tab, locate the labeling job you want to copy.
- 3. Choose **More** > **Copy** in the **Operation** column of the job.
- 4. In the **Copy Task** dialog box, enter the job description and job name *Task name*-**copy**-*xxxx*, where *xxxx* is a randomly generated code to distinguish the new job from the copied job. You can also change the name of the new job. Click **Yes**.

#### Copy Task

You are copying the labeling task **dataset-a4d9**. After you click OK, a new labeling task is created based on the task.

Name	dataset-a4d9-copy-c78f
Description	
	0/2

O The copy operation rebuilds the job based on the dataset associated with the job. If the dataset sample changes, the new job will change accordingly. In addition, the labeling result cannot be copied.

Yes	No
-----	----

5. After the labeling job is copied, you can obtain the new labeling job on the labeling job list page. The new labeling job information includes the samples, labels, and team labeling information.

#### 5.11.6.2 Viewing My Participated Labeling Jobs

On the ModelArts **Data Labeling** page, view your participated labeling jobs on the **My Participations** tab page.

#### Prerequisites

Team labeling is enabled when a labeling job is created.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

#### Procedure

- Log in to the ModelArts management console. In the left navigation pane, choose Data Management > Label Data. The Data Labeling page is displayed.
- 2. Click the **My Participations** tab page to view the labeling jobs you have participated in, including the members in the labeling team and the labeling progress.

# 5.12 Auto Labeling

# 5.12.1 Creating an Auto Labeling Job

In addition to manual labeling, ModelArts also provides the auto labeling function to quickly label data, reducing the labeling time by more than 70%. Auto labeling means learning and training are performed based on the labeled images and an existing model is used to quickly label the remaining images.

#### Context

- Only labeling jobs of image classification and object detection types support auto labeling.
- There are at least two types of labels in the labeling job for auto labeling, and each label has been added to at least five images.
- At least one unlabeled image must exist when you enable auto labeling.
- Before starting an auto labeling job, ensure that no auto labeling job is in progress.
- Before starting an auto labeling job, ensure that the image data does not contain any RGBA four-channel images. These images will cause the job to fail. Delete them from the dataset if you find any.

#### **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

#### Starting an Auto Labeling Job

- Log in to the ModelArts management console. In the left navigation pane, choose Data Management > Label Data. The Data Labeling page is displayed.
- 2. In the labeling job list, locate the row containing a labeling job of the object detection or image classification type and click **Auto Labeling** in the **Operation** column.
- 3. On the **Enable Auto Labeling** page, select **Active learning** or **Pre-labeling**. For details, see **Table 5-33** and **Table 5-34**.

Paramet er	Description
Auto Labeling Type	Active learning: The system uses semi-supervised learning and hard example filtering to perform auto labeling, reducing manual labeling workload and helping you find hard examples.
Algorith m Type	For a dataset of the image classification type, set the following parameters:
	Fast: Use the labeled samples for training.
Specificat ions	Resource specifications used by an auto labeling job. Only GPU specifications are supported.

Table 5-33 Active learning

Paramet er	Description
Compute	The default value is <b>1</b> , indicating the single-node system mode.
Nodes	Only this parameter value is supported.

#### Table 5-34 Pre-labeling

Paramet er	Description
Auto Labeling Type	<b>Pre-labeling</b> : Select a model on the <b>My AI Applications</b> tab page. Ensure that the model type matches the dataset labeling type. After the pre-labeling is complete, if the labeling result complies with the standard labeling format defined by the platform, the system filters hard examples. This step does not affect the pre-labeling result.
Model and Version	• <b>My AI Applications</b> : Select a model as required. Click the drop-down arrow on the left of the target AI application and select a proper version. For details about how to import a model, see <b>Creating an AI Application</b>

#### **NOTE**

For labeling jobs of the object detection type, only rectangular boxes can be recognized and labeled when **Active learning** is selected.

#### Figure 5-123 Enabling auto labeling (pre-labeling)

Auto Labeling Type	Active learning Pre-labeling Select a model created on the Model Management page. Ensure 1 type matches the dataset labeling type. After the pre-labeling is of labeling result complies with the standard labeling format defined platform, the system filters hard examples. This step does not affer labeling result.	that the mode complete, if th d by the ect the pre-
Model and Version	Sel	ect
Specifications	▼	
Compute Nodes	- 1 +	

- 4. After setting the parameters, click **Submit** to enable auto labeling.
- 5. In the labeling job list, click a labeling job name to go to the labeling job details page.

6. Click the **To Be Confirmed** tab page to view the auto labeling progress.

You can also enable auto labeling or view the auto labeling history on this tab page.

#### Figure 5-124 Labeling progress

Labeling	Task Statistics	Label Ma	inagement
All 40	Unlabeled 0	Labeled 40	To Be Confirmed 0
Filter Criteria	No filter cr	iteria selected.	

#### **NOTE**

If there are too many auto labeling jobs, they may have to wait in a queue due to limited free resources. This means that they will stay in the labeling state until their turn comes. To ensure that your labeling job can run properly, you are advised to avoid peak hours.

- 7. After auto labeling is complete, all the labeled images are displayed on the **To Be Confirmed** page.
  - Image classification labeling job

On the **To Be Confirmed** page, check whether labels are correct, select the correctly labeled images, and click **OK** to confirm the auto labeling results. The confirmed image will be categorized to the **Labeled** page.

Object detection labeling job

On the **To Be Confirmed** page, click images to view their labeling details and check whether labels and target bounding boxes are correct. For the correctly labeled images, click **Labeled** to confirm the auto labeling results. The confirmed image will be categorized to the **Labeled** page.

#### FAQs

#### • What can I do if auto labeling fails?

Auto labeling is free of charge. If there are too many auto labeling jobs, they may have to wait in a queue due to limited free resources. Create an auto labeling job again or avoid peak hours.

#### • What can I do if auto labeling takes a long time?

Auto labeling is free of charge. If there are too many auto labeling jobs, they may have to wait in a queue due to limited free resources. You are advised to avoid peak hours.

# 5.13 Team Labeling

# 5.13.1 Team Labeling Overview

Generally, a small data labeling job can be completed by an individual. However, team work is required to label a large dataset. ModelArts provides the team labeling function. A labeling team can be formed to manage labeling for the same dataset.

#### **NOTE**

Team labeling is available only to datasets for image classification, object detection, text classification, named entity recognition, text triplet, and speech paragraph labeling.

Generally, a small data labeling job can be completed by an individual. However, team work is required to label a large dataset. ModelArts provides the team labeling function. A labeling team can be formed to manage labeling for the same dataset.

The team labeling function supports only datasets for image classification, object detection, text classification, named entity recognition, text triplet, and speech paragraph labeling.

For labeling jobs with team labeling enabled, you can create team labeling jobs and assign them to different teams so that team members can complete the labeling jobs together. During data labeling, members can initiate acceptance, continue acceptance, and view acceptance reports.

Team labeling is managed in a unit of teams. To enable team labeling for a dataset, a team must be specified. Multiple members can be added to a team.

- An account can have a maximum of 10 teams.
- An account must have at least one team to enable team labeling for datasets. If the account has no team, add a team by referring to Adding a Team.

# 5.13.2 Creating and Managing Teams

#### 5.13.2.1 Managing Teams

Team labeling is managed in a unit of teams. To enable team labeling for a dataset, a team must be specified. Multiple members can be added to a team.

#### Background

- An account can have a maximum of 10 teams.
- An account must have at least one team to enable team labeling for datasets. If the account has no team, add a team by referring to Adding a Team.

#### Adding a Team

- In the left navigation pane of the ModelArts management console, choose Data Management > Labeling Teams. The Labeling Teams page is displayed.
- 2. On the Labeling Teams page, click Add Team.
- 3. In the displayed **Add Team** dialog box, enter a team name and description and click **OK**. The labeling team is added.

The new team is displayed on the **Labeling Teams** page. You can view team details in the right pane. There is no member in the new team. Add members to the new team by referring to **Adding a Member**.

#### **Deleting a Team**

You can delete a team that is no longer used.

On the **Labeling Teams** page, select the target team and click **Delete**. In the dialog box that is displayed, click **OK**.

#### 5.13.2.2 Managing Team Members

There is no member in a new team. You need to add members who will participate in a team labeling job.

A maximum of 100 members can be added to a team. If there are more than 100 members, add them to different teams for better management.

#### Adding a Member

- In the left navigation pane of the ModelArts management console, choose Data Management > Labeling Teams. The Labeling Teams page is displayed.
- 2. On the **Labeling Teams** page, select a team from the team list on the left and click a team name. The team details are displayed in the right pane.
- 3. In the Team Details area, click Add Member.
- 4. An email address uniquely identifies a team member. Different members cannot use the same email address. The email address you enter will be recorded and saved in ModelArts. It is used only for ModelArts team labeling. After a member is deleted, the email address will also be deleted.

Possible values of **Role** are **Labeler**, **Reviewer**, and **Team Manager**. Only one **Team Manager** can be set.

No annotator cannot be deleted from a labeling team with labeling tasks assigned. The labeling result of an annotator can be synchronized to the overall labeling result only after the annotator's labeling is approved, and the labeling result cannot be filtered.

Information about the added member is displayed in the **Team Details** area.

#### **Modifying Member Information**

You can modify member information if it is changed.

- 1. In the **Team Details** area, select the desired member.
- 2. In the row containing the desired member, click **Modify** in the **Operation** column. In the displayed dialog box, modify the description or role.

The email address of a member cannot be changed. To change the email address of a member, delete the member, and set a new email address when adding a member.

Possible values of **Role** are **Labeler**, **Reviewer**, and **Team Manager**. Only one **Team Manager** can be set.

#### **Deleting Members**

• Deleting a single member

In the **Team Details** area, select the desired member, and click **Delete** in the **Operation** column. In the dialog box that is displayed, click **OK**.

Batch Deletion

In the **Team Details** area, select members to be deleted and click **Delete**. In the dialog box that is displayed, click **OK**.

# 5.13.3 Creating a Team Labeling Job

If you enable team labeling when creating a labeling job and assign a team to label the dataset, the system creates a labeling job based on the team by default. After creating the labeling job, you can view the job on the **My Creations** tab page of the dataset.

You can also create a team labeling job and assign it to different members in the same team or to other labeling teams.

#### Methods

• Choose **Data Management** > **Label Data** on the console. When creating a labeling job, enable **Team Labeling** and select a team or task manager.

Figure 5-125 En	abling team labeling		
Team Labeling			
Туре	● Team    ○ Task Manager		
Select Team	team-info 🔹 🌖 Selec	t at least one labeler	
	Immediately assign the unlabeled files in the da reviewing. After receiving an email from the sys instructed.	ataset to a specified tear stem, the team member	m member for labeling and labels and reviews the files as
	Member Name	Role	Created
	I	No data available.	
	Automatically synchronize new files to the	team labeling task.	
	New files in the dataset will be automatically sy	ynchronized to the label	ing task that has been started.
	Automatically load the intelligent labeling	results to files that nee	d to be labeled.
	Files are automatically labeled. Labelers can the	en accept or modify the	labels.

• Choose **Data Management** > **Datasets** on the console. In the **Operation** column of the target dataset, click **Labeling**. On the **Create Labeling Job** page that is displayed, enable **Team Labeling**. You can create multiple team labeling jobs for the same dataset.

#### Figure 5-126 Enabling team labeling

Team Labeling			
Туре			
Select Team	team-info 🔹 🌖 Selec	ct at least one labeler	
	Immediately assign the unlabeled files in the da reviewing. After receiving an email from the sys instructed.	ataset to a specified tea stem, the team membe	am member for labeling and r labels and reviews the files as
	Member Name	Role	Created
		No data available.	
	Automatically synchronize new files to the	e team labeling task.	
	New files in the dataset will be automatically s	ynchronized to the labe	eling task that has been started.
	Automatically load the intelligent labeling     Files are automatically labeled. Labelers can the	results to files that ne en accept or modify th	ed to be labeled. e labels
	,		

#### D NOTE

- Team members receive emails for team labeling jobs. No email will be sent when you create a labeling team or add members to a labeling team. Additionally, after all samples are labeled, no email will be sent when you create a team labeling job.
- After a team labeling job is created, all unlabeled samples are assigned to annotators randomly and evenly.

#### Procedure

You can create multiple team labeling jobs for the same dataset and assign them to different members in the same team or to other labeling teams.

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Datasets**.
- 2. In the dataset list, select a dataset that supports team labeling, and click the dataset name to go to the **Dashboard** tab page of the dataset.
- 3. In the **Labeling Job** area, view existing labeling jobs of the dataset. Click **Create** to create a job.

Alternatively, you can choose **Data Management** > **Label Data** and click **Create Labeling Job**.

- 4. In the displayed **Create Labeling Job** page, set parameters and click **Create**.
  - **Name**: Enter a job name.
  - Labeling Scene: Select the type of the labeling job.
  - Label Set: All existing labels and label attributes of the dataset are displayed.
  - Team Labeling: Click the button on the right and set the following parameters:
    - **Type**: Select a job type, **Team** or **Task Manager**.
    - Select Team: If Type is set to Team, select a team and members for labeling. The drop-down list displays the labeling teams and their members created by the current account.
    - Select Task Manager: If Type is set to Task Manager, select one Team Manager member from all teams as the task manager.
    - Automatically synchronize new files to the team labeling task: New files in the dataset will be automatically synchronized to the labeling job that has been started.
    - Automatically load the intelligent labeling results to files that need to be labeled: Files are automatically labeled. Annotators can then accept or modify the labels.

#### **NOTE**

The process of loading auto labeling results to a team labeling job is as follows:

- If you set **Type** to **Team**, you are required to create a team labeling task before executing the job.
- If you set **Type** to **Task Manager**, select a team labeling job on the **My Participations** tab page and click **Assign Task**.

After the job is created, you can view the new job on the **My Creations** tab page.

### 5.13.4 Logging In to ModelArts

Typically, users label data in **Data Management** of the ModelArts console. **Data Management** provides data management capabilities such as dataset management, data labeling, data import and export, auto labeling, and team labeling and management. After a team labeling job is created, team members can log in to the ModelArts console to view the job.

1. After a labeling job is created, receive a labeling notification email as a team member to which the job is assigned.

#### Figure 5-127 Task email

Lend Up the Long K William
2023/1/5 (周四) 17:37
HUAWEI CLOUD <noreplyxfz02@mail01.huawei.com></noreplyxfz02@mail01.huawei.com>
您有新的标注任务待查收
收件人等效率的必要效率
❶ 如果显示此邮件的方式有问题, 请单击此处以在 Web 浏览器中查看该邮件。
单击此处可下载图片。为了帮助保护您的隐私,Outlook 禁止自动下载该邮件中的某些图片。
区 您被邀请参与一个新的标注任务:task-5d9e,请及时确认。 请点击如下钱捷雪/gonsole/huaweicloud.com/modelarts/?region=cn-north-4#/dataLabel?tabActive=labelConsole 如有任何疑问,请联系您的管理员。
本邮件由系统自动发送,请勿直接回复
本邮件由系统自动发送,请勿直接回复 官方网站: <u>https://www.huaweicloud.com</u> (中国大陆) <u>https://intl.huaweicloud.com</u> (国际站)

 Click the labeling job link in the email. The Data Management > Data Labeling > My Participations tab page on the ModelArts console is displayed.

Ienant name or HUAWEI	CLOUD account name
IAM user name or email a	uddress
	Ø
L	og In

3. On the My Participations tab page, you can view your labeling jobs.

#### Figure 5-129 My Participations

Figure 5-128 Logging in to ModelArts

Data	Labeling Maximum Labelin	g task: 500,Available for creation: 48	8					G Feedback	Create Labeling Job
My	Treations My Particip	ations							
								Enter a name.	Q C
	Task Name	Labeling Type	Samples	Unlabeled/To Be Confirmed/Rejected/P	Created ↓Ξ	Created By	Role	Operation	n
~	task-5d9e	Image classification	4	4/0/0/0/0/0	Jan 05, 2023 17:36:27 GMT+08:00	sWX1037871	Labeler	Labeling	

If a team member has bound an email address, the team member can receive a job notification email and access the data labeling console using the address provided in the email.

Upon your login, only the team labeling jobs and related data of the current user (the mailbox user) are displayed.

# 5.13.5 Starting a Team Labeling Job

After logging in to the data labeling page on the management console, you can click the **My Participations** tab page to view the assigned labeling job and click the job name to go to the labeling page. The labeling method varies depending on the labeling job type. For details, see the following:

- Image Classification
- Object Detection
- Text Classification
- Named Entity Recognition
- Text Triplet
- Speech Paragraph Labeling

On the labeling page, each member can view the images that are not labeled, to be confirmed, rejected, to be reviewed, approved, and accepted. Pay attention to the images rejected by the administrator and the images to be corrected.

If the Reviewer role is assigned for a team labeling job, the labeling result needs to be reviewed. After the labeling result is reviewed, it is submitted to the administrator for acceptance.

#### Figure 5-130 Labeling platform



## 5.13.6 Reviewing Team Labeling Results

After team labeling is complete, the reviewer can review the labeling result.

 Log in to the ModelArts management console. In the navigation pane, choose Data Management > Label Data. On the Data Labeling page, click My Participations. Locate the row containing the target labeling job and click Review in the Operation column to initiate the review.

#### Figure 5-131 Initiating review

Data La	Data Labeling Maximum Labeling task: 500 Available for creation: 407								
My	Creations   My Partici	ipations							
							Enter a name. Q C		
	Task Name	Samples	Unlabeled/To Be Confirmed/Rejected/Pendi	Created ↓Ξ	Created By	Role	Operation		
~	☑ task-d196	35	35/0/0/0/0/0	Feb 01, 2023 11:21:54 GM	El_modelarts_00382652_02	Team Manager	Assign Task Review Labeling		

2. On the review page, check the samples that are not reviewed, reviewed, approved, or rejected.

#### Figure 5-132 Labeling result review



3. Choose **Confirm** or **Reject** on the right of the review page.

If you choose **Confirm**, set **Rating** to **A**, **B**, **C**, or **D**. Option **A** indicates the highest score. If you choose **Reject**, enter the rejection reason in the text box.

Figure 5-133 Pass						
Accepta	ince Result	t				
Rating: 💿	А () В (	○ c ○ D				
	Confirm	Reject	<u>Skip</u>			
Figure 5-134 Reject						
Accepta	nce Result					
not good						

not good				
				8/256
Pas	s	Confirm	<u>Skip</u>	

# 5.13.7 Accepting Team Labeling Results

#### Task Acceptance (Administrator)

#### • Initiating acceptance

After team members complete data labeling, the labeling job creator can initiate acceptance to check labeling results. The acceptance can be initiated only when a labeling member has labeled data. Otherwise, the acceptance initiation button is unavailable.

- a. Log in to the ModelArts management console. In the left navigation pane, choose **Data Management** > **Label Data**.
- b. On the **My Participations** tab page, click a team labeling job to go to its details page. Choose **Team Labeling** > **Accept** in the upper right corner.

#### Figure 5-135 Initiating acceptance

Publish	Task Statistics	Team Labeling 🔻
		Accept Stop Acceptance
		Continue Acceptance
All Labels 1		Acceptance Report
Name		Annotator Management

c. In the displayed dialog box, set **Sample Policy** to **By percentage** or **By quantity**. Click **OK** to start the acceptance.

**By percentage**: Sampling is performed based on a percentage for acceptance.

By quantity: Sampling is performed based on quantity for acceptance.

d. After the acceptance is initiated, an acceptance report is displayed on the console. In the **Acceptance Result** area on the right, click **Pass** or **Reject**.

If you click **Pass**, set **Rating** to **A**, **B**, **C**, or **D**. Option **A** indicates the highest score. If you click **Reject**, enter your rejection reasons in the text box.

#### • Continuing acceptance

You can continue accepting tasks whose acceptance is not completed. For tasks for which an acceptance process is not initiated, the **Continue Acceptance** button is unavailable.

In the **Labeling Progress** pane on the **Task Statistics** tab page, click **Continue Acceptance** to continue accepting jobs. The **Real-Time Acceptance Report** page is displayed. You can continue to accept the images that are not accepted.

kdb42 ( Back to Data Labeling List Data Labeling List Continue Acceptance Report								
Labeling Task Statistics Label Management Annotator Manag	ement							
Labeling Type Object detection Team Labeling Created Jun 16, 2022 16:3754 GMT-08:00	Datavets datavet-1602 Updated Jun 16, 2022 16:37:54 GMT-48:00	Description -						
Labeling Progress All 35 Unlabeled 1 Labeled 34	Label Statistics	Annotators' Progress						
40	Label	Labeled By						
30 -								

#### • Finishing acceptance

After the continue acceptance is complete, click **Stop Acceptance** in the upper right corner. On the page that is displayed, view the acceptance status of the labeling job, such as the number of sampled files, configure parameters, and perform the acceptance. The labeling information is synchronized to the **Labeled** tab page of the labeling job only after the acceptance is complete.

Once the labeled data is accepted, team members cannot modify the labeling information. Only the dataset creator can modify the labeling information.

task-db42 / Accept		Finish
Sampling Policy Samples for Acceptance 1 Sampling Policy By percentage 100 %, Count 1	Real-time acceptance reports Acceptance Rest State 0% File Samples 1	
Pending Acceptance(1)         Accepted(0)         Passed(0)         Rejected(0)	Presidua(P) 1/1 Nex(N)	

 Table 5-35 Parameters for finishing acceptance

Parameter	Description
Modifying Labeled Data	• <b>Not overwrite</b> : For the same data, do not overwrite the existing data with the labeling result of the current team.
	• <b>Overlays</b> : For the same data, overwrite the existing data with the labeling result of the current team. Overwritten data cannot be recovered.
Acceptance Scope	• <b>All passed</b> : All items, including the rejected ones will pass the review.
	• All rejects: All items, including the ones that have passed the review will be rejected. In this case, the passed items must be labeled and reviewed again in the next acceptance.
	• All remaining items pass: The rejected items are still rejected, and the remaining items will automatically pass the review.
	• All remaining items rejects: The selected items that have passed the review do not need to be labeled. All the selected items that have been rejected and the items that have not been selected must be labeled again for acceptance.

#### **Viewing an Acceptance Report**

You can view the acceptance report of an ongoing or finished labeling job. Log in to the management console and choose **Data Management** > **Label Data**. On the **Data Labeling** page, select **My Creations** and click the name of a team labeling job. The job details page is displayed. In the upper right corner of the page, click **Acceptance Report**. In the displayed dialog box, view report details.

#### Deleting a Labeling Job

After a job is accepted, delete it if the labeling job is no longer used. After a job is deleted, the labeling details that are not accepted will be lost. However, the original data in the dataset and the labeled data that has been accepted are still stored in the corresponding OBS bucket.

# 6 Devenviron

# 6.1 Introduction to DevEnviron

#### D NOTE

This document describes the DevEnviron notebook functions of the new version.

Software development is a process of reducing developer costs and improving development experience. In AI development, ModelArts is dedicated to improving AI development experience and simplifying the development process. ModelArts DevEnviron uses cloud native resources and integrates the development tool chain to provide better in-cloud AI development experience for AI development, exploration, and teaching.

ModelArts notebook for seamless in-cloud and on-premises collaboration

- In-cloud JupyterLab, local IDE, and ModelArts plug-ins for remote development and debugging, tailored to your needs
- In-cloud development environment with AI compute resources, cloud storage, and built-in AI engines
- Custom runtime environment saved as an image for training and inference

# Feature 1: Remote development, allowing remote access to notebook from a local IDE

The notebook of the new version provides remote development. After enabling remote SSH, you can remotely access the ModelArts notebook development environment to debug and run code from a local IDE.

Due to limited local resources, developers using a local IDE run and debug code typically on a CPU or GPU server shared between team members. Building and maintaining the CPU or GPU server are costly.

ModelArts notebook instances are out of the box with various built-in engines and flavors for you to select. You can use a dedicated container environment. Only after simple configurations, you can remotely access the environment to run and debug code from your local IDE.



Figure 6-1 Remotely accessing notebook from a local IDE

Uploaded through web or OBS Browser+

ModelArts notebook can be regarded as an extension of a local development environment. The operations such as data reading, training, and file saving are the same as those performed in a local environment.

ModelArts notebook allows you to use in-cloud resources while with local coding habits unchanged.

A local IDE supports Visual Studio (VS) Code, PyCharm, and SSH. In addition, the PyCharm Toolkit and VS Code Toolkit plug-ins allow you to easily use cloud resources.

#### Feature 2: One-click image saving to save a development environment

ModelArts notebook of the new version allows you to save a running notebook instance as a custom image with one click.

When an image is saved, the installed pip dependency package is retained. In remote development through VS Code, the plug-ins installed on the server are retained.

# Feature 3: Preset images that are out-of-the-box with optimized configurations and supporting mainstream AI engines

The AI engines and versions preset in each image are fixed. When creating a notebook instance, specify an AI engine and version, including the chip type.

ModelArts DevEnviron provides a group of preset images, including PyTorch, TensorFlow, and MindSpore images. You can use a preset image to start your notebook instance. After the development in the instance, submit a training job without any adaptation.

The image versions preset in ModelArts are determined based on user feedback and version stability. If your development can be carried out using the versions preset in ModelArts, for example, MindSpore 1.5, use preset images. These images have been fully verified and have many commonly-used installation packages built in. They are out-of-the-box, relieving you from configuring the environment.

The images preset in ModelArts DevEnviron include:

- Common preset packages: common AI engines such as PyTorch and MindSpore based on standard Conda, common data analysis software packages such as Pandas and Numpy, and common tool software such as CUDA and CUDNN, meeting common AI development requirements.
- Preset Conda environments: A Conda environment and basic Conda Python (excluding any AI engine) are created for each preset image. The following figure shows the Conda environment for a preset MindSpore image.



Select a Conda environment based on whether the AI engine is used for debugging.

- Notebook: a web application that enables you to code on the GUI and combine the code, mathematical equations, and visualized content into a document.
- JupyterLab plug-ins: enable flavor changing and instance stopping to improving user experience.
- Remote SSH: allows you to remotely debug a notebook instance from a local PC.

#### **NOTE**

- To simplify operations, ModelArts notebook of the new version supports switchover between AI engines in a notebook instance.
- Al engines vary based on regions. For details about the Al engines available in a region, see the Al engines displayed on the management console.

# Feature 4: JupyterLab, an online interactive development and debugging tool

ModelArts integrates open-source JupyterLab for online interactive development and debugging. You can use the notebook on the ModelArts management console to compile and debug code and train models based on the code, without concerning environment installation or configuration.

JupyterLab is an interactive development environment. It is the next-generation product of Jupyter Notebook. JupyterLab enables you to compile notebooks, operate terminals, edit Markdown text, enable interaction, and view CSV files and images.

# **6.2 Application Scenarios**

ModelArts provides flexible, open development environments. Select a development environment based on site requirements.

- In-cloud notebook, which is out of the box, relieving you from concerning environment installation or configuration. For details, see JupyterLab Overview and Common Operations.
- Local IDE for model development. After enabling remote SSH, you can remotely access the ModelArts notebook development environment to debug and run code from a local IDE. The local IDE allows you to use the in-cloud notebook development environment while with local coding habits unchanged.

A local IDE supports Visual Studio (VS) Code, PyCharm, and SSH. Additionally, PyCharm Toolkit and VS Code Toolkit are provided for convenient remote access. For details, see and .

# 6.3 Managing Notebook Instances

# 6.3.1 Creating a Notebook Instance

Before developing a model, create a notebook instance and access it for coding.

#### **Constraints and Limitations**

- Only running notebook instances can be accessed or stopped.
- A maximum of 10 notebook instances can be created under one account.

#### Procedure

1. Log in to the ModelArts management console. In the navigation pane, choose **Settings** and check whether the access authorization has been configured. If not, configure access authorization. For details, see "Configuring Access Authorization".

#### Figure 6-2 Configuring authorization

Global Configuration 💿								
ModelArts strict authorization is now available. In this mode, all operations require explicit authorization, facilitating precise authorization control. Switch to this new mode and give it a try. Learn more about ModelArts permissions.								
Add Authorization Clear Authorization	Enable strict mode							
V Search or filter by keyword.	V Search or filter by keyword.							
Authorized To 0         Authorization Type 0         Authorization Content 0         Creation Time 0         Operation								
	All users	Agency	modelarts_i	Jan 19, 2023 16:53:29 GMT+08:00	View Permissions Delete			

- 2. Log in to the ModelArts management console. In the navigation pane on the left, choose **DevEnviron** > **Notebook**.
- 3. Click **Create** in the upper right corner. On the **Create Notebook** page, configure parameters.
  - a. Configure basic information of the notebook instance, including its name, description, and auto stop status. For details, see **Table 6-1**.

#### Table 6-1 Basic parameters

Paramete r	Description
Name	Name of the notebook instance, which is automatically generated by the system. You can rename it based on service requirements. A name consists of a maximum of 128 characters and cannot be empty. It can contain only digits, letters, underscores (_), and hyphens (-).
Descriptio n	Brief description of the notebook instance
Auto Stop	Automatically stops the notebook instance at a specified time. This function is enabled by default. The default value is <b>1 hour</b> , indicating that the notebook instance automatically stops after running for 1 hour. The options are <b>1 hour</b> , <b>2 hours</b> , <b>4 hours</b> , <b>6 hours</b> , and <b>Custom</b> . You can select <b>Custom</b> to specify any integer from 1 to 24 hours.
	<ul> <li>Stop as scheduled: If this option is enabled, the notebook instance automatically stops when the running duration exceeds the specified duration.</li> <li>NOTE         To protect in-progress jobs, a notebook instance does not automatically stop immediately at the auto stop time. Instead     </li> </ul>
	there is a period of 2 to 5 minutes provided for you to renew the auto stop time.

b. Configure notebook parameters, such as the image and instance flavor. For details, see **Table 6-2**.

Table 6-2	Notebook	instance	parameters
-----------	----------	----------	------------

Paramete r	Description
lmage	<ul> <li>Public and private images are supported.</li> <li>Public images are the AI engines built in ModelArts.</li> <li>Private images can be created using an instance that is created using a public image.</li> <li>An image corresponds to an AI engine. When you select an image during instance creation, the AI engine is specified accordingly. Select an image as required. Enter a keyword of the image name in the search box on the right to quickly search for the image.</li> <li>You can change an image on a stopped notebook instance.</li> </ul>
Resource Type	Select a resource pool as required.

Paramete r	Description
Туре	Processor type, which can be <b>CPU</b> , <b>ASCEND</b> , or <b>GPU</b> . The chips vary depending on the selected image.
Flavor	The flavor of your notebook instance. Select a flavor based on your needs.
Storage	The storage configuration varies based on resource types and specifications. Configure this parameter based on your needs.
	All storage paths of <b>EVS</b> and <b>SFS</b> are mounted to the / home/ma-user/work directory. All read and write operations on files in the notebook instance are stored in this directory, not in OBS.
	The data is retained in <b>/home/ma-user/work</b> , even if the notebook instance is stopped or restarted.
	When a notebook instance is deleted, the EVS storage is released and the stored data is not retained. SFS can be mounted to a new notebook instance and data can be retained.
Remote SSH	• After you enable this function, you can remotely access the development environment of the notebook instance from your local development environment.
	• When a notebook instance is stopped, you can update the SSH configuration on the instance details page.
	<b>NOTE</b> The notebook instances with remote SSH enabled have VS Code plug-ins (such as Python and Jupyter) and the VS Code server package pre-installed, which occupy about 1 GB persistent storage space.
Key Pair	Set a key pair after remote SSH is enabled.
	Select an existing key pair.
	Alternatively, click <b>Create</b> on the right of the text box to create one on the DEW console. To do so, choose <b>Key Pair Service</b> > <b>Private Key Pairs</b> and click <b>Create Key Pair</b> .
	After a notebook instance is created, you can change the key pair on the instance details page.
	<b>CAUTION</b> Download the created key pair and properly keep it. When you use a local IDE to remotely access the notebook development environment, the key pair is required for authentication.

Paramete r	Description
Whitelist	Set a whitelist after remote SSH is enabled. This parameter is optional.
	Add the IP addresses for remotely accessing the notebook instance to the whitelist, for example, the IP address of your local PC or the public IP address of the source device. A maximum of five IP addresses can be added and separated by commas (,). If the parameter is left blank, all IP addresses will be allowed for remote SSH access.
	If your source device and ModelArts are isolated from each other in network, obtain the public IP address of your source device using a mainstream search engine, for example, by entering "IP address lookup", but not by running <b>ipconfig</b> or <b>ifconfig/ip</b> locally.
	Figure 6-3 IP address lookup
	<u>IP地址查询</u>
	本机IP: 12000000000000000000000000000000000000
	请输入ip地址 查询
	4/11P世有力法 IP地址设置力法
	After a notebook instance is created, you can change the whitelist IP addresses on the instance details page.

- 4. Click Next.
- 5. After confirming the parameter settings, click **Submit**.

Switch to the notebook instance list. The notebook instance is being created. It will take several minutes when its status changes to **Running**. Then, the notebook instance is created.

6. In the notebook instance list, click the instance name. On the instance details page that is displayed, view the instance configuration.

If **Remote SSH** is enabled, you can click the modification icon on the right of the whitelist to modify it. You can click the modification icon on the right of **Authentication** to update the key pair of a stopped notebook instance.

If an EVS disk is used, click **Expansion** on the right of **Storage Capacity** to dynamically expand the EVS disk capacity. For details, see **Dynamically Expanding EVS Disk Capacity**.

## 6.3.2 Accessing a Notebook Instance

Access a notebook instance in the **Running** state for coding.

The methods of accessing notebook instances vary depending on the AI engine based on which the instance was created.
- Remote access: Use PyCharm, VS Code, or SSH in the local IDE. For details, see and Connecting to a Notebook Instance Through VS Code with One Click.
- Online access: Use JupyterLab. For details, see JupyterLab Overview and Common Operations.

Create an instance and mount the persistent storage to **/home/ma-user/work**.

sh-4.4\$pwd	
/home/ma-user	
sh-4.4\$cd work/	
ah-4.4\$pwd	
home/ma-user/work	
sh-4.4\$	

The data stored in only the **work** directory is retained after the instance is stopped or restarted. When you use a development environment, store the data for persistence in **/home/ma-user/work**.

# 6.3.3 Searching for, Starting, Stopping, or Deleting a Notebook Instance

### Searching for an Instance

All created instances are displayed on the notebook page. To display a specific instance, search for it based on filter criteria. Click the search box and select one or more search criteria.

- Enable **View all** to check all notebook instances created by all sub-users in the IAM project.
- Select search criteria, such as name, ID, status, image, flavor, description, and creation time.

### **Customizing Table Columns**

Click to customize the columns to be displayed in the table.

### Starting or Stopping an Instance

Stop the notebook instances that are not needed. You can also restart a stopped instance.

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **DevEnviron** > **Notebook**.
- 2. Start or stop the target notebook instance.
  - To start a notebook instance, click **Start** in the **Operation** column of the target notebook instance. Only stopped notebook instances can be started.
  - To stop a notebook instance, click **Stop** in the **Operation** column of the target notebook instance. Only running notebook instances can be stopped.



After a notebook instance is stopped:

• The data stored only in **/home/ma-user/work** is retained. For example, the external dependency packages installed in other directories in the development environment will be deleted.

### **Deleting an Instance**

Delete the notebook instances that are not needed.

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **DevEnviron** > **Notebook**.
- 2. In the notebook list, locate the target notebook instance, and click **Delete** in the **Operation** column. In the displayed dialog box, confirm the information, enter **DELETE** in the text box, and click **OK**.

### 

Deleted notebook instances cannot be recovered. After a notebook instance is deleted, the data stored in the mounted directory will be deleted.

# 6.3.4 Changing a Notebook Instance Image

ModelArts allows you to change images on a notebook instance to flexibly adjust its AI engine.

### Constraints

The target notebook instance is stopped.

### Procedure

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **DevEnviron** > **Notebook**.
- 2. In the notebook list, click **More** in the **Operation** column of the target notebook instance and select **Change Image**.
- 3. In the **Change Image** dialog box, select a new image and click **OK**. After the modification, you can view the new image on the notebook list page.

# 6.3.5 Changing the Flavor of a Notebook Instance

ModelArts allows you to change the node flavor for a notebook instance.

# Constraints

Specifications of a notebook instance can be modified only when the notebook instance is in the **Stopped**, **Running**, or **Startup failed** state.

# Procedure

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **DevEnviron** > **Notebook**.
- In the notebook instance list, locate the row that contains the target notebook instance and choose More > Modify Specifications in the Operation column. In the Modify Specifications dialog box that appears, select the required flavor.

# 6.3.6 Selecting Storage in DevEnviron

Storage varies depending on performance, usability, and cost. No storage media can cover all scenarios. Learning about in-cloud storage application scenarios for better usage.

### D NOTE

Only OBS parallel file systems (PFS) and object storage in the same region can be mounted.

Storag e	Application Scenario	Advantage	Disadvantage
EVS	Data and algorithm exploration only in the development environment.	Block storage SSDs feature better overall I/O performance than NFS. The storage capacity can be dynamically expanded to up to 4096 GB.	This type of storage can only be used in a single development environment.
		As persistent storage, EVS disks are mounted to / home/ma-user/work. The data in this directory is retained after the instance is stopped. The storage capacity can be expanded online based on demand.	

Table 6-3 In-cloud storage	application	scenarios
----------------------------	-------------	-----------

Storag e	Application Scenario	Advantage Disadvan		
PFS	NOTE PFS is a whitelist function. To use this function, contact technical support. PFS buckets mounted as persistent storage for AI development and exploration. - Storage for datasets. Datasets are directly mounted to notebooks for browsing and data processing and can be directly used during training. For details, see How Do I Upload Data to OBS? 2. Storage for code. After debugging on a notebook instance, specify the OBS path as the code path for starting training, facilitating temporary modification. - Storage for checking training. Mount storage to the training output path such as the path to training logs. In this way, view and check training on the notebook instance in real time. This is especially suitable for analyzing the output of jobs trained using TensorBoard or notebook.	PFS is an optimized high- performance object storage file system with low storage costs and large throughput. It can quickly process high-performance computing (HPC) workloads. PFS mounting is recommended if OBS is used. <b>NOTE</b> Package or split the data to be uploaded by 128 MB or 64 MB. Download and decompress the data in local storage for better I/O and throughput performance.	Due to average performance in frequent read and write of small files, PFS storage is not suitable for large model training or file decompression. <b>NOTE</b> Before mounting PFS storage to a notebook instance, grant ModelArts with full read and write permissions on the PFS bucket. The policy will be retained even after the notebook instance is deleted.	

Storag e	Application Scenario	Advantage	Disadvantage
OBS	NOTE OBS is a whitelist function. To use this function, contact technical support. When uploading or downloading a large amount of data in the development environment, you can use OBS buckets to transfer data.	Low storage cost and high throughput, but average performance in reading and writing small files. It is a good practice to package or split the file by 128 MB or 64 MB. In this way, you can download the packages, decompress them, and use them locally.	The object storage semantics is different from the Posix semantics and needs to be further understood.
SFS	Available only in dedicated resource pools. Use SFS storage in informal production scenarios such as exploration and experiments. One SFS device can be mounted to both a development environment and a training environment. In this way, you do not need to download data each time your training job starts. This type of storage is not suitable for heavy I/O training on more than 32 cards.	SFS is implemented as NFS and can be shared between multiple development environments and between development and training environments. This type of storage is preferred for non-heavy-duty distributed training jobs, especially for the ones not requiring to download data additionally when the training jobs start.	The performance of the SFS storage is not as good as that of the EVS storage.

Storag e	Application Scenario	Advantage	Disadvantage
Local storage	First choice for heavy- duty training jobs.	High-performance SSDs for the target VM or BMS, featuring high file I/O throughput. For heavy-duty training jobs, store data in the target directory and then start training. By default, the storage is mounted to the <b>/cache</b> directory. For details about the available space of the <b>/</b> <b>cache</b> directory, see What Are Sizes of the /cache Directories for Different Notebook Specifications in DevEnviron?.	The storage lifecycle is associated with the container lifecycle. Data needs to be downloaded each time the training job starts.

# Using the Storage

How do I use EVS in a development environment?

When creating a notebook instance, select a small-capacity EVS disk. You can scale out the disk as needed. For details, see Dynamically Expanding EVS Disk Capacity.

# 6.3.7 Dynamically Expanding EVS Disk Capacity

### Overview

If a notebook instance uses an EVS disk for storage, the disk is mounted to / home/ma-user/work/ of the notebook container and the disk capacity can be expanded by up to 100 GB at a time when the instance is running.

### **Application Scenarios**

During notebook development, select a small EVS disk capacity, for example, 5 GB, when creating a notebook instance because the storage requirements are low at the initial stage. After the development, a large volume of data must be trained. Then, expand the disk capacity to cost-effectively meet your service needs.

# Restrictions

- The target notebook instance must use EVS for storage.
- Up to 100 GB can be expanded at a time. Additionally, the total capacity after expansion cannot exceed 4096 GB.
- If the original capacity of an EVS disk is 4096 GB, the disk capacity cannot be expanded.

• After the instance is stopped, the expanded capacity still takes effect.

# Procedure

- 1. Log in to the ModelArts management console. In the left navigation pane, choose **DevEnviron** > **Notebook**.
- 2. Click the name of a running notebook instance. On the instance details page, click **Expansion**.

### Figure 6-4 Instance details page

< notebook-3d1b		
Name	notebook	
Status	🧿 Running(54 minutes left) ♂	
ID		fe3de56 🗖
Storage Mount	/home/ma-user/work/	
Storage Capacity	5 GB (EVS) Expansion	

3. Set the capacity to be expanded and click **OK**. **Expanding** shows that the capacity expansion is in progress. After the expansion, the displayed storage capacity is the expanded capacity.

# 6.3.8 Modifying the SSH Configuration for a Notebook Instance

ModelArts allows you to modify the SSH configuration for notebook instances.

If a notebook instance is created with remote SSH disabled, you can enable remote SSH on the notebook details page.

During the creation of a notebook instance, if you set a whitelist for remotely accessing it, you can change the IP addresses in the whitelist on the notebook instance details page. You can also change the key pair.

### Constraints

The target notebook instance must be stopped.

### Changing the Key Pair and Remote Connection IP Address

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **DevEnviron** > **Notebook**.
- 2. Click the target notebook instance. Enable remote SSH and change the key pair and whitelist.

 $\times$ 

### D NOTE

/ notobook 0c0d

For manually enabled remote SSH, see **Figure 6-5**. After the SSH configuration is updated, the remote SSH function cannot be disabled.

For remote SSH enabled by default in the selected image, see Figure 6-6.

#### Figure 6-5 Update SSH Configuration

### Update SSH Configuration

★ Key Pair			•	С	Create
Whitelist					
	ОК	Canel			

### Figure 6-6 Changing the whitelist and key pair

10tebook-st	54		
Name	noteboc 🖉 🖉	Flavor	CPU: 2vCPUs 8GB 👻
Status	Stopped	Image	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04
ID	f993 🗇	Created At	Aug 05, 2022 09:03:57 GMT+08:00
Storage Path	/home/n	Updated At	Aug 05, 2022 09:07:00 GMT+08:00
Storage Capacity	50 GB (Default)		
Remote SSH		Address	
Whitelist	🖉	Authentication	KeyPair-1174 🖉

- Click and choose an existing key pair, or click Create to create a new key pair.
- For details about how to configure a whitelist, see Setting an IP Address for Remotely Accessing a Notebook Instance. After you change the IP addresses, the existing links are still valid. After the links are released, the new links only from the changed IP addresses can be set up.

### Setting an IP Address for Remotely Accessing a Notebook Instance

< notebook-9c	9d			
Name	noteboc 🖉		Flavor	CPU: 2vCPUs 8GB 👻
Status	Stopped		Image	tensorflow2.1-cuda10.1-cudnn7-ubuntu18.04
ID	f993	0	Created At	Aug 05, 2022 09:03:57 GMT+08:00
Storage Path	/home/n		Updated At	Aug 05, 2022 09:07:00 GMT+08:00
Storage Capacity	50 GB (Default)			
Remote SSH			Address	
Whitelist	🖉		Authentication	KeyPair-1174 🖉

Figure 6-7 Setting an IP address for remotely accessing a notebook instance

Ensure that public IP addresses are set. If your source device and the ModelArts are isolated from each other in network, obtain the public IP address of your source device using a mainstream search engine, for example, by entering "IP address lookup", but not by running **ipconfig** or **ifconfig/ip** locally.

### Figure 6-8 IP address lookup

<u>IP地址查询</u>				
中机IP: 12 <b>%%和你%%%%</b> 》	广东省广州市 <b>彩线</b> 云			
请输入ip地址	查询			
本机IP查看方法 IP地址设置方法				

# 6.3.9 Viewing the Notebook Instances of All IAM Users Under One Tenant Account

Any IAM user granted with the **listAllNotebooks** and **listUsers** permissions can click **View all** on the notebook page to view the instances of all IAM users in the current IAM project.

### **NOTE**

Users granted with these permissions can also access OBS and SWR of all users in the current IAM project.

# **Assigning the Required Permissions**

- 1. Log in to the ModelArts management console as a tenant user, hover the cursor over your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
- On the IAM console, choose Permissions > Policies/Roles from the navigation pane, click Create Custom Policy in the upper right corner, and create two policies.

Policy 1: Create a policy that allows users to view all notebook instances of an IAM project, as shown in **Figure 6-9**.

- Policy Name: Enter a custom policy name, for example, Viewing all notebook instances.
- Policy View: Select Visual editor.
- Policy Content: Select Allow, ModelArts Service, modelarts:notebook:listAllNotebooks, and default resources.

Policies/Roles / Create Cu	ustom Policy					
1 You can use custon	n policies to supplement system-define	d policies for fine-grained permissions management. Learn m	ore			
* Policy Name 🚺	policyM3rw					
Policy View 2	Visual editor JSO	a				
* Policy Content	∧ 引 🖸 Allow	4 ModelArts Service	3 C Actions: 1	0 AI	(Optional) Add request condition	ÐŪ
	Select all modelarts:n	olebook:listAlNotebooks			X   Q	
	n 🔽 Leidniy					
	O I I modelats notebook isAllviebooks Ourry the ist of all development environment instances					
		) Add Permissions				
Description	Enter a brief description.					
			0/256			
Scope	Project-level services					
8	OK Cancel					

Policy 2: Create a policy that allows users to view all users of an IAM project.

- Policy Name: Enter a custom policy name, for example, Viewing all users of the current IAM project.
- Policy View: Select Visual editor.
- Policy Content: Select Allow, Identity and Access Management, iam:users:listUsers, and default resources.
- 3. In the navigation pane, choose **User Groups**. Then, click **Authorize** in the **Operation** column of the target user group. On the **Authorize User Group** page, select the custom policies created in **2**, and click **Next**. Then, select the scope and click **OK**.

After the configuration, all users in the user group have the permission to view all notebook instances created by users in the user group.

If no user group is available, create a user group, add users using the user group management function, and configure authorization. If the target user is not in a user group, you can add the user to a user group through the user group management function.

### Starting Notebook Instances of Other IAM Users

If an IAM user wants to access another IAM user's notebook instance through remote SSH, they need to update the SSH key pair to their own. Otherwise, error **ModelArts.6789** will be reported. For details about how to update a key pair, see **Modifying the SSH Configuration for a Notebook Instance**.

Erro message: ModelArts.6789: Failed to use SSH key pair KeyPair-xxx. Update the key pair and try again later.

# 6.4 JupyterLab

# 6.4.1 Operation Process in JupyterLab

ModelArts allows you to access notebook instances online using JupyterLab and develop AI models based on the PyTorch, TensorFlow, or MindSpore engines. The following figure shows the operation process.

Figure 6-10 Using JupyterLab to develop and debug code online



1. Create a notebook instance.

On the ModelArts management console, create a notebook instance with a proper AI engine. For details, see **Creating a Notebook Instance**.

- 2. Use JupyterLab to access the notebook instance. For details, see Accessing JupyterLab.
- 3. Upload training data and code files to JupyterLab. For details, see **Uploading Files from a Local Path to JupyterLab**.
- 4. Compile and debug code in JupyterLab. For details, see JupyterLab Overview and Common Operations.
- 5. In JupyterLab, call the ModelArts SDK to create a training job for in-cloud training.

For details, see Using ModelArts SDK.

# 6.4.2 JupyterLab Overview and Common Operations

JupyterLab is the next-generation web-based interactive development environment of Jupyter Notebook, enabling you to compile notebooks, operate terminals, edit Markdown text, enable interaction, and view CSV files and images.

JupyterLab is the future mainstream development environment for developers. It has the same components as Jupyter Notebook, but offering more flexible and powerful functions.

# Accessing JupyterLab

To access JupyterLab from a running notebook instance, perform the following operations:

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **DevEnviron** > **Notebook**.
- 2. Click **Open** in the **Operation** column of a running notebook instance to access JupyterLab.

pytorch1.8-cuda10.2-cudnn7-ubunt. CPU: 2vCPUs 8GB \*

Figure 6-11 Accessing a notebook instance

Sep 12, 2023 10:41:52 GMT+08:00 Open Start | Stop | More 🕶

- 3. The **Launcher** page is automatically displayed. Perform required operations. For details, see **JupyterLab Documentation**.
  - **Notebook**: Select a kernel for running notebook, for example, TensorFlow or Python.
  - **Console**: Call the terminal for command control.
  - Other: Edit other files.

### Creating an IPYNB File in JupyterLab

On the JupyterLab homepage, click a proper AI engine in the **Notebook** area to create an IPYNB file.

The AI engines supported by each notebook instance vary depending on the runtime environment. The following figure is only an example. Select an AI engine based on site requirements.

The created IPYNB file is displayed in the navigation pane on the left.

### Creating a Notebook File and Accessing the Console

A console is a Python terminal, which is similar to the native IDE of Python, displaying the output after a statement is entered.

On the JupyterLab homepage, click a proper AI engine in the **Console** area to create a notebook file.

The AI engines supported by each notebook instance vary depending on the runtime environment. The following figure is only an example. Select an AI engine based on site requirements.

After the file is created, the console page is displayed.

$\bowtie$	File	Edit	View	Run	Kernel	Git	Tabs	Settings	Help	
		+			1	:	C	\$\$⁺		🗏 Untitled.ipynb 🔹 Console 2 🛛 🗙
0	Fi	lter file	s by na	me					Q	Ŭ.
U		/								Python 3.7.10 (default, Jun 4 2021, 14:48:32) Type 'convright', 'credits' or 'license' for more information
$\mathbf{O}$	Nan	ne					•	Last Mo	odified	IPython 7.31.1 An enhanced Interactive Python. Type '?' for help.
	•	Untitle	d.ipynb					a minu	ite ago	
≣										

Figure 6-12 Creating a notebook file (console)

# Editing a File in JupyterLab

JupyterLab allows you to open multiple notebook instances or files (such as HTML, TXT, and Markdown files) in one window and displays them on different tab pages.

In JupyterLab, you can customize the display of multiple files. In the file display area on the right, you can drag a file to adjust its position. Multiple files can be concurrently displayed.



Figure 6-13 Customized display of multiple files

When writing code in a notebook instance, you can create multiple views of a file to synchronously edit the file and view execution results in real time.

To open multiple views, open an IPYNB file and choose **File** > **New View for Notebook**.

Figure 6-14 Multiple views of a file



Before coding in the code area of an IPYNB file in JupyterLab, add an exclamation mark (!) before the code.

For example, install an external library Shapely.

pip install Shapely!

For example, obtain PythonPath.

echo \$PYTHONPATH!

#### Figure 6-15 Running code



### Renewing or Automatically Stopping a Notebook Instance

If you enable auto stop when you created or started a notebook instance, the remaining duration for stopping the instance is displayed in the upper right corner of JupyterLab. You can click the time for renewal.

#### Figure 6-16 Remaining duration



# **Common JupyterLab Buttons and Plug-ins**

	+ 🗈	±	C	\$ <sup>*</sup>
0	Filter files by name			Q
<u>۲</u>	<b>I</b> /			
�	Name		•	Last Modified
Ť	• 🖪 Untitled.ipynb			seconds ago
≣				

### Figure 6-17 Common JupyterLab buttons and plug-ins

### Table 6-4 JupyterLab buttons

Button	Description
+	Quickly open notebook instances and terminals. Open the <b>Launcher</b> page, on which you can quickly create notebook instances, consoles, or other files.
	Create a folder.

Button	Description
<u>*</u>	Upload files.
G	Refresh the file directory.
${\bf O}^{+}$	Git plug-in, which can be used to access the GitHub code library associated with the notebook instance.

 Table 6-5
 JupyterLab
 plug-ins

Plug-in	Description
	List files. Click this button to show all files in the notebook instance.
0	Display the terminals and kernels that are running in the current instance.
•>	Git plug-in, which can be used to quickly access the GitHub code library.
° <b>¢</b>	Property inspector.
≡	Show the document organization.

# Figure 6-18 Buttons in the navigation bar

A File Edit View Run Kernel Git Tabs Settings Help

Table 6-	6 Buttons	in th	e navigation	bar
----------	-----------	-------	--------------	-----

Button	Description
File	Actions related to files and directories, such as creating, closing, or saving notebooks.
Edit	Actions related to editing documents and other activities in the IPYNB file, such as undoing, redoing, or cutting cells.
View	Actions that alter the appearance of JupyterLab, such as showing the bar or expanding code.
Run	Actions for running code in different activities such as notebooks and code consoles.

Button	Description
Kernel	Actions for managing kernels, such as interrupting, restarting, or shutting down a kernel.
Git	Actions on the Git plug-in, which can be used to quickly access the GitHub code library.
Tabs	A list of the open documents and activities in the dock panel.
Settings	Common settings and an advanced settings editor.
Help	A list of JupyterLab and kernel help links.

# Figure 6-19 Buttons in the menu bar of an IPYNB file

<ul> <li>Untitled1.ipynb</li> </ul>		
🖻 + % 🗇 🗂 ▶ ■ C	▶         Code         ∨         ()         git         2 vCPU + 8 GiB         ≇         PyTorch-1.	8 O <
🗷 Untitled.ipynb	×	
🖻 + 🛠 🗇 🗳 🕨	■ C Markdown ~ ③ git 2 vCPU + 4 GiB PyTorch-1.4	0 <

### Table 6-7 Buttons in the menu bar of an IPYNB file

Button	Description
8	Save a file.
+	Add a new cell.
ж	Cut the selected cell.
	Copy the selected cell.
Ċ	Paste the selected cell.
•	Execute the selected cell.
	Terminate a kernel.
C	Restart a kernel.
**	Restart a kernel and run all code of the current notebook again.
Code ~	There are four options in the drop-down list:
	<b>Code</b> (Python code), <b>Markdown</b> (Markdown code, typically used for comments), <b>Raw</b> (a conversion tool), and - (not modified)

Button	Description
0	View historical code versions.
git	Git plug-in. The gray button indicates that the plug-in is unavailable in the current region.
2 vCPU + 4 GiB	Instance flavor.
PyTorch-1.4	Kernel for you to select.
0	Code running status.   indicates the code is being executed.

# Monitoring Resources

To obtain resource usage, select **Resource Monitor** in the right pane. The CPU usage and memory usage can be viewed.

### Figure 6-20 Resource usage

	Untitled1.ip	ynb		×								Resource	Monitor		
8	+ %		►	-	C	Code	~	٩	git	2 vCPU +	8 GiB	CRUUICO	0		0%
1.1	E 1 : [											20.95			0.0.04
1.7											_				
											- 11	15 96 -			
											- 11	10 %			
											- 11	5 %			
											- 11	0.96			
															and the second sec
											- 11				usage within ous
											- 11	Memory	Jsage		6% <b>^</b>
												Memory 20 %	Jsage		6% <b>^</b>
												Memory 20 % 15 %	Jsage		6% <b>^</b>
												Memory 20 %	Jsage		6% A
												Memory 20 % 15 % 10 %	Jsage		6% A
												Memory 20 %	Jsage		6% A
												Memory 20 % 15 % 10 % 5 % 0 %	Jsage		usage within 60s

# 6.4.3 Code Parametrization Plug-in

The code parametrization plug-in simplifies notebook cases. You can quickly adjust parameters and train models based on notebook cases without complex code. This plug-in can be used to customize notebook cases for competitions and learning.

### **Use Guide**

• The **Add Form** and **Edit Form** buttons are available only to the shortcut menu of code cells.

### Figure 6-21 Viewing a code cell



• After opening new code, add a form before editing it.

### Figure 6-22 Shortcut menu of code cells

8	+	Ж	Ċ	•	G	**	Code	~	🕓 gi	it					
	0							Add Fo Edit Fo	rm rm		_	_	_	_	•
								Cut Ce	lls						Х
								Сору С	Cells						С
								Paste C	ells Bel	ow					V
								Delete	Cells					D,	D
								Split Ce	ell				C	trl+Shift	+-
								Merge	Selecte	d Cell	s			Shift+	М
								Merge	Cell Ab	ove			Ctrl+	Backspa	ce
								Merge	Cell Bel	low			Ctr	l+Shift+	М

### Add Form

If you click **Add Form**, a code cell will be split into the code and form edit area. Click **Edit** on the right of the form to change the default title.

#### Figure 6-23 Two edit areas

#@title Default title text	Default title text	Ľ
----------------------------	--------------------	---

### **Edit Form**

If you click **Edit Form**, four sub-options will be displayed: **Add new form field**, **Hide code**, **Hide form**, and **Show All**.

• You can set the form field type to **dropdown**, **input**, and **slider**. See **Figure 6-24**. Each time a field is added, the corresponding variable is added to the code and form areas. If a value in the form area is changed, the corresponding variable in the code area is also changed.

#### **NOTE**

When creating a dropdown form, click **ADD Item** and add at least two items. See **Figure 6-25**.

Figure 6-24 Form style of dropdown, input, and slider

# Default title text

variable_name: 1 v	dropdown
variable_name: " please input string here	input "
variable_name: 🗧 🛛 0	slider

### Figure 6-25 Creating a dropdown form

Form field type	Variable type
dropdown	✓ string ✓
	Add Item
	option1 input a value
	option2 input a value
Variable name	
variable, name	

### Figure 6-26 Deleting a form

Default title text	+ ^ ¥ Ш
variable_name1: bb 🗸	<b> </b>
variable_name2: [1, 2, 3]	<u> </u>
variable_name3: {'1': 'a'; 'b':2}	<u> </u>
variable_name4: (1, 2, 3)	<u> </u>

- If the form field type is set to dropdown, the supported variable types are raw and string.
- If the form field type is set to input, the supported variable types are boolean, date, integer, number, raw, and string.
- If the form field type is set to **slider**, the minimum value, maximum value, and step can be set.
- If you click **Hide code**, the code area will be hidden.
- If you click **Hide form**, the form area will be hidden.
- If you click **Show All**, both the code and form areas will be displayed.

# 6.4.4 Using ModelArts SDK

Notebook instances allow you to use ModelArts SDK to manage OBS, training jobs, models, and real-time services.

Your notebook instances have automatically obtained your AK/SK for authentication and the region. Therefore, SDK sessions are automatically authenticated.

### **Example Code**

Create a training job.
 from modelarts.session import Session
 from modelarts.estimator import Estimator
 session = Session()
 estimator = Estimator(

	modelarts_session=session,	
	framework type='PyTorch'.	# Al engine name
	framework version='PvTorch-1.0.0-pvthon3.6'	# Al engine version
	code dir-'/obs-bucket-name/src/'	# Training script directory
	boot file='/obs bucket name/src/putersh contin	mont py # Training boot script
	boot_nie= /obs-bucket-name/sic/pytoicit_sentin	nent.py, # naining boot script
directory		
	log_url='/obs-bucket-name/log/',	# Training log directory
	hyperparameters=[	
	{"label":"classes",	
	"value": "10"},	
	{"label":"lr".	
	"value": "0.001"}	
	1	
	output nath-'/obs bucket name/output/'	# Training output directory
	turin instance turne land delaute une anu n100	# Training Output directory
	train_instance_type='modelarts.vm.gpu.p100',	# Training environment
specifications		
	train_instance_count=1,	# Number of training nodes
	job_description='pytorch-sentiment with Model	Arts SDK') # Training job description
job_instance =	estimator.fit(inputs='/obs-bucket-name/data/tra	ain/', wait=False,
iob_name='m	v training job')	
	,	
Obtain a n	nodel list.	
from modelar	ts.session import Session	
from modelar	ts model import Model	
	ing ()	

session = Session()
model\_list\_resp = Model.get\_model\_list(session, model\_status="published", model\_name="digit",
order="desc")
Obtain comvice dataile

Obtain service details. from modelarts.session import Session from modelarts.model import Predictor session = Session() predictor\_instance = Predictor(session, service\_id="input your service\_id") predictor\_info\_resp = predictor\_instance.get\_service\_info()

# 6.4.5 Using the Git Plug-in

In JupyterLab, you can use the Git plug-in to clone the GitHub open-source code repository, quickly view and edit data, and submit the modified data.

# Prerequisites

The notebook instance is running.

# Starting the Git Plug-in of JupyterLab

In the notebook instance list, locate the target instance and click **Open** in the **Operation** column to go to the JupyterLab page.

Figure 6-27 shows the Git plug-in of JupyterLab.

### Figure 6-27 Git plug-in

A File Edit View Run Ken	nel Git Tabs Settings Hel	p
+ to	± (	ž ♦ <sup>+</sup>
■ /		
Name		<ul> <li>Last Modified</li> </ul>
0		
•		
v		
<b>B</b>		

### Cloning a GitHub Open-Source Code Repository

Access a GitHub open-source code repository at https://github.com/jupyterlab/ extension-examplesitHub. Click  $\checkmark$ , enter the repository address, and click **OK** to start cloning. After the cloning is complete, the code library folder is displayed in the navigation pane of JupyterLab.

Figure 6-28 Using the Git plug-in to clone a GitHub open-source code repository



### **Cloning a GitHub Private Code Repository**

When you clone a GitHub private code repository, a dialog box will be displayed, asking you to enter your personal credentials. In this case, enter the personal access token in GitHub.

# Git credentials required

#### Enter credentials for remote repository

username	
password / personal access	tol
Cancel	ок

To obtain a personal access token, perform the following operations:

- 1. Log in to **GitHub** and open the configuration page.
- 2. Click Developer settings.
- 3. Choose Personal access tokens > Generate new token.
- 4. Verify the account.
- 5. Describe the token, select permissions to access the private repository, and click **Generate token** to generate a token.
- 6. Copy the generated token to CloudBuild.

### NOTICE

- Save the token securely once it is generated. It will be unavailable after you refresh the page. If it is not obtained, generate a new token.
- Enter a valid token description so that it can be easily identified. If the token is deleted by mistake, the building will fail.
- Delete the token when it is no longer used to prevent information leakage.

**Figure 6-29** Cloning a GitHub private code repository (only authorization using a personal access token is supported)



Search or ju	github.com/pints 22.2 Pull	jit requests Issues Codespaces Marketplace Explo	re	کا	2 ☆ 券 □ @(
Code 💿 Iss.	es 11 Pull requests ③ Action	ns 🗄 Projects 🛈 Security 🗠 Insights 😵	Settings		Set status
	Y main →     P 1 branch      O	tags 225129@qq.com Test private push	Go to file Add file ▼ <> Code → 885c786 5 days ago ③2 commits	About No description, website, or topics pr	Your repositories Your projects Your stars Your gists
	README.md     test.py	Initial commit Test private push	2 years ago 5 days ago	<ul> <li>✿ 0 stars</li> <li>④ 1 watching</li> <li>♥ 0 forks</li> </ul>	Your sponsors Upgrade Try Enterprise
	learngit		0	Releases No releases published Create a new release	Feature preview Help Settings
				Packages No packages published Publish your first package	July Cort
				Languages  Python 100.0%	

Figure 6-30 Obtaining a personal access token

# Viewing a Code Repository

In the list under **Name**, double-click the folder you want to use and click the Git plug-in icon on the left to access the code repository corresponding to the folder.

Figure 6-31 Opening the folder and starting the Git plug-in

$\bowtie$	File	Edit	View	Run	Kernel	Git	Tabs	Settings	Help			
		+		٥		<u>+</u>		C	♦\$+		🛛 Launcher	
Ø	Nam	ne						La	ast Modi	fied		
•		extensi	on-exan	nples				:	seconds	ago		
�												
ß												

You can view the information current code repository, such as the repository name, branch, and historical submission records.

-			-		-	-				
$\bowtie$	File Edit	View	Run	Kernel	Git	Tabs	Settings	Help		
				_			٩	Þ	G	🛛 Launcher
œ	Current	t Reposi <b>ion-exa</b>	tory <b>mples</b>							
0	Current master	t Branch	I			_		_	<b>~</b>	
-		Chang	es				History			
•	Frédéric Col	llonval	9c3	5013		3 we	eeks ago		•	1
	working m	aster o	rigin/HE	AD origi	in/maste	er			- 1	
B	Fix contenth	neader d	lev dep	endencie	s (#186	)				
	Ahmed Fasi	h	f05f	b90		3 we	eeks ago		•	
°	Example for	r MainAr	eaWidg	get's cont	entHea	der (#1	85)			
	Jeremy Tulo	oup	c02	81bc		3 m	onths ago		•	
	Merge pull	request	#184 fr	om krass	owski/p	atch-1				
	Michał Kras Update doc	sowski sumenta	fcc9 tion exc	0b3 cerpt		3 m	onths ago		•	
=	Jeremy Tulo Merge pull	oup request	dcb #181 fr	1fa6 om jupyt	erlab/fi	5 m x/upda	onths ago te-3.1		•	

### Figure 6-32 Viewing a code repository

### **NOTE**

By default, the Git plug-in clones the master branch. To switch another branch, click **Current Branch** to expand all branches and click the target branch name.

$\bigwedge$	File	Edit	View	Run	Kernel	Git	Tabs	Settings	Help		
								٩	٢	G	🛛 Launcher
(3	Ţ	Current extens	t Reposi ion-exa	tory <b>mples</b>							
0	ų	Current <b>master</b>	t Branch	I						•	
U			Branch	nes				Tags			
•	Filt	er							lew Bra	nch	
<b>F</b> Q	<b>p</b> 2	.x									
•	۲ v	naster									
Q.,	₽ origin/1.x										
4	۶c	origin/2	.х								
	۴o	origin/H	IEAD								
	ษ	origin/fo	collonva	l/issue1	20						

# **Viewing Modifications**

If a file in the code repository has been modified, you can view the modified file under **Changed** on the **Changes** tab page. Click **Diff this file** on the right of the file name to view the modifications.

### Figure 6-33 Viewing modifications

1994	File Edit View Kun Kernel Git Tabs	Settings Help		
-		@ @ C	I Launcher X I README.md X ■ README.md X	
0	Current Repository extension-examples		HEAD	WORKING
0	P Current Branch master	-	() 52 conda activate jupyterlab-extension-examples 51	() 52 conde activate jupyterlab-extension-examples 53
	Changes	History	54 # go to the hello world example 55 cd hello world	54 # go to the Launcher example 55 cd leurcher
۲	> Staged	(0)	56 57 # install the extension in editable mode	56 57 # install the extension in editable mode
#0.	- Changed	(1)	()	()
	README.md			
°0	Untracked	(0)		

### **Committing Modifications**

After confirming that the modifications are correct, click **Stage this change** on the right of the file name, which is equivalent to running the **git add** command. The file enters the **Staged** state. Enter the message to be committed in the lower left corner and click **Commit** that is equivalent to running the **git commit** command.

Figure 6-34 Committing modifications

A File Edit View Run Kernel Git Tabs Settings Hele

80		@ @ C	I Launcher X 🗈 README.md X	
0	Current Repository extension-examples		HEAD	INDEX
0	P Current Branch master	-	() conda activate jupyterlab-extension-examples 53	() 50 53
	Changes	History	54 = # go to the hello world example 55 = cd hello-world	54 ± # go to the launcher example 55 ± cd launcher
٠	- Staged	(1)	1) 57 # install the extension in editable mole	56 57 # install the extension in editable mode
-0	🖉 README.md	м	M ()	()
B	> Changed	(0)	D)	
°¢	+ Untracked	(0)	<u>n</u>	
_				
E.	-			
	Modify README.md			
	push to GitHuk demo			
	Commit			

On the **History** tab page, view the committing status.

Figure 6-35 Checking whether the committing is successful



Click the **push** icon, which is equivalent to running the **git push** command, to push the code to the GitHub repository. After the pushing is successful, the message "Successfully completed" is displayed. If the token used for OAuth authentication has expired, a dialog box is displayed asking you to enter the user token or account information. Enter the information as prompted. This section describes the authorization using a personal access token. If you use a password for authorization but the password becomes unavailable, perform the operations described in **What Do I Do If the Git Plug-in Password Is Invalid?** 

-		5	
ile Edit View Run Kernel	Git Tabs Settings Help	C Disordar X DiffaDMEnd X	
Current Repository			R
extension-examples		HEAD	INDEX
P Current Branch master		* 52 () conda activate jupyterlab-extension-examples	() conda activate jupyteriab-extension-examples
Changes	History	53 54 # pp to the bells world example	54 # go to the launcher example
marcingin b3384ec	24 seconds ago	SS 1 cd bellowerld SS 56	55 ± of Launcher 56
working master		57 # install the extension in editable mode ()	57 # install the extension in editable mode ()
Modify README.md			
Frédéric Collorival 9c35013	3 weeks ago		
Fix contentheader dev dependenci	es (#186)		
Ahmed Fasih R56b90	3 weeks ago	·	
Example for MainAreaWidget's cor	tentHeader (#185)		
Jeremy Tuloup c0281bc Merce pull request #104 from kter	3 months ago sowiki/patch-1		
Michał Krassowski fcc90b3	3 months ago		
Update documentation excerpt			
Jeremy Tuloup dcb1fa6	5 months ago		
Fahldeis Collorval 080678	6 months and		
Fix context-menu and documents			
Frédéric Collorval e5b003d	6 months ago		
More fix	6		
Fix tests and linter	o montra ago		
Frédéric Collorval 7cd15ba	6 months ago		
Fix linter			
Frédéric Collorival d550b01 Undate the examples to use the ne	6 months ago w menu definitions farm settions		
Carlos Herrero fe27afb	6 months ago		
Adds a new example about docum	ents using shared models (#163)		
Frédéric Collorival 9e43taa Upgrade playwright to 1.13.1 (#17	6 months ago		
R Ely 64112c1	6 months ago		
Oustom completer example (#169)			
Carlos Herrero 09dilde7	6 months ago		Successfully p

Figure 6-36 Pushing code to the GitHub repository

After the preceding operations are complete, on the **History** tab page of the JupyterLab Git plug-in page, you can see that **origin/HEAD** and **origin/master** point to the latest push. In addition, you can find the corresponding information in the committing records of the GitHub repository.

# 6.4.6 Visualized Model Training

# 6.4.6.1 Introduction to Training Job Visualization

ModelArts notebook of the new version supports TensorBoard and MindInsight for visualizing training jobs. In the development environment, use small datasets to train and debug algorithms, during which you can check algorithm convergence and detect issues to facilitate debugging.

You can create visualization jobs of TensorBoard and MindInsight types on ModelArts.

Both TensorBoard and MindInsight effectively display the change trend of a training job and the data used in the training.

• TensorBoard

TensorBoard effectively displays the computational graph of TensorFlow in the running process, the trend of all metrics in time, and the data used in the training. For more details about TensorBoard, see **TensorBoard official website**.

TensorBoard visualization training jobs support only CPU and GPU flavors based on TensorFlow 2.1, and PyTorch 1.4 and 1.8 images. Select images and flavors based on the site requirements.

• MindInsight

MindInsight visualizes information such as scalars, images, computational graphs, and model hyperparameters during training. It also provides functions such as training dashboard, model lineage, data lineage, and performance debugging, helping you train and debug models efficiently. MindInsight supports MindSpore training jobs. For more information about MindInsight, see **MindSpore official website**.

The following shows the images and flavors supported by MindInsight visualization training jobs, and select images and flavors based on the site requirements.

- MindSpore 1.2.0 (CPU or GPU)
- MindSpore 1.5.x or later (Ascend)

You can use the summary file generated during model training to create a visualization job in Notebook of DevEnviron.

- For details about how to create a MindInsight visualization job in a development environment, see MindInsight Visualization Jobs.
- For details about how to create a TensorBoard visualization job in a development environment, see **TensorBoard Visualization Jobs**.

### 6.4.6.2 MindInsight Visualization Jobs

ModelArts notebook of the new version supports MindInsight visualization jobs. In a development environment, use a small dataset to train and debug an algorithm. This is used to check algorithm convergence and detect training issues, facilitating debugging.

MindInsight visualizes information such as scalars, images, computational graphs, and model hyperparameters during training. It also provides functions such as training dashboard, model lineage, data lineage, and performance debugging, helping you train and debug models efficiently. MindInsight supports MindSpore training jobs. For more information about MindInsight, see MindSpore official website.

MindSpore allows you to save data into the summary log file and obtain the data on the MindInsight GUI.

### Prerequisites

When using MindSpore to edit a training script, add the code for collecting the summary record to the script to ensure that the summary file is generated in the training result.

For details, see Collecting Summary Record.

### Note

- To run a MindInsight training job in a development environment, start MindInsight and then the training process.
- Only one-card single-node training is supported.

### Creating a MindInsight Visualization Job in a Development Environment

Step 1 Create a Development Environment and Access It Online

Step 2 Upload the Summary Data

Step 3 Start MindInsight

Step 4 View Visualized Data on the Training Dashboard

# **Step 1 Create a Development Environment and Access It Online**

Log in to ModelArts management console, choose **DevEnviron** > **Notebook**, and create a development environment instance for the MindSpore engine. After the instance is created, click **Open** in the **Operation** column of the instance to access it online.

The images and resource types supported by MindInsight visualization training jobs are as follows:

- MindSpore 1.2.0 (CPU or GPU)
- MindSpore 1.5.x or later (Ascend)

# Step 2 Upload the Summary Data

Summary data is required for MindInsight visualization in a development environment.

Upload the summary data to the **/home/ma-user/work/** directory in a development environment or store it in an OBS parallel file system.

- For details about how to upload the summary data to **/home/ma-user/** work/, see Uploading and Downloading Data in Notebook.
- To store the summary data in an OBS parallel file system that is mounted to a notebook instance, upload the summary file generated during model training to the OBS parallel file system. When MindInsight is started in a notebook instance, the notebook instance automatically reads the summary data from the mounted OBS parallel file system.

# Step 3 Start MindInsight

Choose a way you like to start MindInsight in JupyterLab.

κ, μ	ile Edit	View Run Kernel Git Tabs	Settings Help							
	+	Activate Command Palette	Ctrl+Shift+C							
C	Filter file:	Simple Interface Entry 3 Presentation Mode	Ctrl+Shift+D							
. –	-/	✓ Show Status Bar		Notebook						
≥ ≡	Name .model: lost+fo Untitler	<ul> <li>✓ Show Left Sidebar</li> <li>Show Right Sidebar</li> <li>✓ Show Hidden Files</li> </ul>	Ctrl+B	Μ	Entry 1 P					
		Show Line Numbers Match Brackets Wrap Words		MindSpore	python-3.7.10					
		Collapse Selected Code Collapse Selected Outputs		Console						
		Collapse All Code Collapse All Outputs		MindSpore	P python-3.7.10					
		Expand Selected Code Expand Selected Outputs Expand All Code								
		Expand All Outputs		S_ Other E	ntry 4		Entry 2			
		Render Side-by-side	Shift+R					NA		
		Text Editor Syntax Highlighting	•	\$_	X	[M]*	E		Y:	2
		Show Log Console		Terminal	VS Code	MindInsight	Text File	Markdown File	Pipeline As Code	Python File

Figure 6-37 Starting MindInsight in JupyterLab

### Method 1

Μ

- 1. Click <sup>MindSpore</sup> to go to the JupyterLab development environment. An IPYNB file will be automatically created.
- Enter the following command in the dialog box: %reload\_ext mindinsight %mindinsight --port {PORT} --summary-base-dir {SUMMARY\_BASE\_DIR}

Parameters:

- port {PORT}: web service port for visualization, which defaults to 8080. If the default port 8080 has been used, specify a port ranging from 1 to 65535.
- summary-base-dir{SUMMARY\_BASE\_DIR}: data storage path in the development environment
  - Local path to the development environment: ./work/xxx (relative path) or /home/ma-user/work/xxx (absolute path)
  - Path to the OBS parallel file system bucket: obs://xxx/

#### For example:

# If the summary data is stored in **/home/ma-user/work/** of a development environment, run the following command:

%mindinsight --summary-base-dir /home/ma-user/work/xxx Or

# If the summary data is stored in an OBS parallel file system, run the following command. Then, the development environment will automatically mount the storage path to the OBS parallel file system and read data from the path.

%mindinsight --summary-base-dir obs://xxx/

### Figure 6-38 MindInsight page (1)



### Method 2

	[ <b>M</b> ] <sup>s</sup>	
	MindInsight	
Click		to go to the MindInsight page.

Data is read from /home/ma-user/work/ by default.

If there are two projects or more, select the target project to view its logs.

### Figure 6-39 MindInsight page (2)

Untitled.ip	ynb	×	📕 Untitled1.ipynb	×	Mr MindInsight 1	×						
[M]	Mindlnsi	ght	Summary List	Мо	del Explanation							
Sumn	nary List (c	urrei	nt folder:work)						Linea	ge Analysis	Comparison Analy	sis
No.	Summary Path					Update Time		Operation				
1	./summary					2021	6	Training Dasht	board	Profiling	Parameter Details	
2	./summary_dir					2021-	3	Training Dasht	board	Profiling	Parameter Details	

### Method 3

1. Choose View > Activate Command Palette, enter MindInsight in the search box, and click Create a new MindInsight.

Figure 6-40 Create a new MindInsight

🖾 Launcher	MindInsight		۹
	Create a new Mind	llnsight	
	Notebook	(	_
	Μ	Р	
	MindSpore	python-3.7.10	

- 2. Enter the path to the summary data or the storage path to the OBS parallel file system, and click **CREATE**.
  - Local path to the development environment: ./summary (relative path) or /home/ma-user/work/summary (absolute path)
  - Path to the OBS parallel file system: obs://xxx/

### Figure 6-41 Path to the summary data



### Figure 6-42 MindInsight page (3)



### **NOTE**

A maximum of 10 MindInsight instances can be started using method 2 or 3.

#### Method 4



Click Terminal and run the following command (the UI will not be displayed):

mindinsight start --summary-base-dir ./summary\_dir

Figure 6-43 Opening MindInsight through Terminal

Mr MindInsight 1	×	MindInsight 2	×	s. Terminal 2	×
sh-4.4 mindinsight sta Workspace: /home/ma-us Summary base dir: /hom Web address: http://12 service start state: s sh-4.4\$	rtsum er/mindi e/ma-use 7.0.0.1: uccess	umary-base-dir ./summa nsight sr/summary_dir 8080	ary_dir		

# Step 4 View Visualized Data on the Training Dashboard

The training dashboard is important for MindInsight visualization. It allows visualization for scalars, parameter distribution, computational graphs, dataset graphs, images, and tensors.

For more information, see **Viewing Training Dashboard** on the MindSpore official website.

# **Related Operations**

To stop a MindInsight instance, use one of the following methods:

- Method 1: Enter the following command in the .ipynb file window of JupyterLab. in which the port number is configured in Start MindInsight (8080 by default): !mindinsight stop --port 8080
- Method 2: Click . The MindInsight instance management page is displayed, which shows all started MindInsight instances. Click **SHUT DOWN** next to the target instance to stop it.

### Figure 6-44 Stopping an instance

0	MINDINSIGHT SESSIONS	SHUT DOWN
3	TERMINAL SESSIONS	×
	KERNEL SESSIONS	×
in i		0

• Method 3: Click in the following figure to close all started MindInsight instances.

### Figure 6-45 Stopping all started MindInsight instances

		G	Untitled1.ipynb × (M)* MindInsight 0 ×
	KERNEL SESSIONS	×	
۲	TERMINAL SESSIONS	×	[M] <sup>®</sup> MindInsight Summary List
•	MINDINSIGHT SESSIONS	×	
	[M]' MindInsights/0	SHUT DOWN	Summary List (Current folder:work)
8	[M] <sup>*</sup> MindInsights/1	SHUT DOWN	No. Summary Path

• Method 4 (not recommended): Close the MindInsight window on JupyterLab. In this way, only the visualization window is closed, but the instance is still running on the backend.

# 6.4.6.3 TensorBoard Visualization Jobs

ModelArts supports TensorBoard for visualizing training jobs. TensorBoard is a visualization tool package of TensorFlow. It provides visualization functions and tools required for machine learning experiments.

TensorBoard effectively displays the computational graph of TensorFlow in the running process, the trend of all metrics in time, and the data used in the training.

### Prerequisites

When you write a training script, add the code for collecting the summary record to the script to ensure that the summary file is generated in the training result.

For details about how to add the code for collecting the summary record to a TensorFlow-powered training script, see **TensorFlow official website**.

### Process of Creating a TensorBoard Visualization Job in a Development Environment

Step 1 Create a Development Environment and Access It Online

Step 2 Upload the Summary Data

Step 3 Start TensorBoard

**Step 4 View Visualized Data on the Training Dashboard** 

### Step 1 Create a Development Environment and Access It Online

On the ModelArts management console, choose **DevEnviron** > **Notebook**, and create an instance using a TensorFlow or PyTorch image. After the instance is created, click **Open** in the **Operation** column of the instance to access it online.

Only CPU and GPU flavors with TensorFlow2.1, PyTorch1.4, or PyTorch1.8 and later images can support TensorBoard visualization for training jobs. Select images and flavors based on the site requirements.

# Step 2 Upload the Summary Data

Summary data is required for using TensorBoard visualization functions in DevEnviron.

You can upload the summary data to the **/home/ma-user/work/** directory in the development environment or store it in the OBS parallel file system.

- For details about how to upload the summary data to the notebook path / home/ma-user/work/, see Uploading Files to JupyterLab.
- To store the summary data in an OBS parallel file system that is mounted to a notebook instance, upload the summary file generated during model training to the OBS parallel file system. When TensorBoard is started in a notebook instance, the notebook instance automatically mounts the OBS parallel file system directory and reads the summary data.

# Step 3 Start TensorBoard

There are multiple methods to open TensorBoard in JupyterLab in the development environment. Select one based on your habits.

File Edit	View Run Kernel Git Tabs	Settings Help				
+	Activate Command Palette	Ctrl+Shift+C				
Filter files	Simple Interface Method Presentation Mode	3 Ctrl+Shift+D				
• /	✓ Show Status Bar		Notebook	Method 2		
Name .modela lost+fo	<ul> <li>Show Left Sidebar</li> <li>Show Right Sidebar</li> <li>Show Hidden Files</li> </ul>	Ctrl+B	Τ	Ρ		
	Show Line Numbers Match Brackets Wrap Words		TensorFlow-2.1	python-3.7.10		
	Collapse Selected Code Collapse Selected Outputs Collapse All Code Collapse All Outputs		T	Р		
	Expand Selected Code Expand Selected Outputs Expand All Code Expand All Outputs		TensorFlow-2.1	python-3.7.10 hod 4	Method 1	
	Render Side-by-side	Shift+R				
	Text Editor Syntax Highlighting	•	\$_	X		
	Show Log Console		Terminal	VS Code	Tensorboard	Text File

Figure 6-46 Starting TensorBoard in JupyterLab

### Method 1 (recommended):

1. Open JupyterLab, in the navigation pane on the left, create the **summary** folder, and upload data to **/home/ma-user/work/summary**. The folder name must be **summary**.



2. Go to the **summary** folder and click TensorBoard page. See **Figure 6-47**.

to go to the

#### Figure 6-47 TensorBoard page (1)



### Method 2

### NOTICE

You can upgrade TensorBoard to any version except 2.4.0. After the upgrade, the new version of TensorBoard is used only in method 2. For other methods, use TensorBoard 2.1.1.



The .ipynb file is automatically created.

 Enter the following command in the dialog box: %reload\_ext ma\_tensorboard %ma\_tensorboard --port {PORT} --logdir {BASE\_DIR}

Parameters:

1.

- port {PORT}: web service port for visualization, which defaults to 8080. If the default port 8080 has been used, specify a port ranging from 1 to 65535.
- **logdir** {*BASE\_DIR*}: data storage path in the development environment
  - Local path of the development environment: ./work/xxx (relative path) or /home/ma-user/work/xxx (absolute path)
  - Path of the OBS parallel file system: obs://xxx/

Example:

# If the summary data is stored in **/home/ma-user/work/** of the development environment, run the following command:

%ma\_tensorboard --port {PORT} --logdir /home/ma-user/work/xxx or

# If the summary data is stored in the OBS parallel file system, run the following command and the development environment automatically mounts the storage path of the OBS parallel file system and

#### reads data.

%ma\_tensorboard --port {PORT} --logdir obs://xxx/

### Figure 6-48 TensorBoard page (2)



### Method 3

 Choose View > Activate Command Palette, enter TensorBoard in the search box, and click Create a new TensorBoard.

Figure 6-49 Create a new TensorBoard

	TensorBoard	Q
	TENSORBOARD	
0	Create a new TensorBoard	
	NOTEBOOK OPERATIONS	
0	Exp <b>or</b> t Note <b>bo</b> ok to M <b>arkd</b> own	
ß		
°¢		

2. Enter the path of the summary data you want to view or the storage path of the OBS parallel file system.
- Local path of the development environment: ./summary (relative path) or /home/ma-user/work/summary (absolute path)
- Path of the OBS parallel file system bucket: **obs://xxx/**

### Figure 6-50 Entering the summary data path

nput the logair Path to cr	eate a new TensorBoard
Logdir Path:	
./summary	
	Cancel CREATE

### Figure 6-51 TensorBoard page (3)

Terminal 1 X 🖸 La	uncher	X TensorBoard 1 X
TensorBoard SCALA	rs images	GRAPHS DISTRIBUTIONS HISTOGRAMS
Show data download links		<b>Q</b> Filter tags (regular expressions supported)
Ignore outliers in chart scali	ing	accuracy
Tooltip sorting method: default	· ·	acturacy
Smoothing	0.6	0.3
Horizontal Axis           STEP         RELATIVE         WALL		0.2
Runs		
Write a regex to filter runs		
✓ ○ .		cost
TOGGLE ALL RUNS		
/home/ma-user/work/log		cross_entropy tag: cost/cross_entropy

### Method 4

Click



Terminal and run the following command. The UI will not be displayed.

tensorboard --logdir ./log

### Figure 6-52 Opening TensorBoard through Terminal

```
sh-4.41ped
/home/maruser
h-4.45tenestboard --logdir ./log
2021-10-18 20 34.53.506976 I temporflow/stream_executor/platform/defuelt/dso_loader.cc 44) Successfully opened dynamic library libsvinfer_plugin.co.6
2021-10-18 20 34.53.508972 I temporflow/stream_executor/platform/defuelt/dso_loader.cc 44) Successfully opened dynamic library libsvinfer_plugin.co.6
2021-10-18 20 34.53.508972 I temporflow/stream_executor/platform/defuelt/dso_loader.cc 44) Successfully opened dynamic library libsvinfer_plugin.co.6
Serving TemporBoard 2.1.1 at http://localhost.6008/ (Press CTML+C to quit)
```

# Step 4 View Visualized Data on the Training Dashboard

For TensorBoard visualization, you need the training dashboard. It lets you visualize scalars, images, and computational graphs.

For more functions, see Get started with TensorBoard.

# **Related Operations**

To stop a TensorBoard instance, use any of the following methods:

• Method 1: Click <sup>O</sup>. The TensorBoard instance management page is displayed, which shows all started TensorBoard instances. Click **SHUT DOWN** next to an instance.

#### Figure 6-53 Clicking SHUT DOWN to stop an instance

AA	File	Edit	View	Run	Kernel	Git	Tabs	Settings	Help	)					
									G	~	Tensorboard	d 1	×		
	OPE	N TABS						Close A	JI I						
0	ß	Tensort	board 1							Т	ensorBoa	ard	9	SCALARS	TE
~	KERI	NELS					Shu	ut Down A	Ш	_			_		
•>	TERM	MINALS	;				Shu	ut Down A	.11	۵	Show da	ita down	nload	links	
≣	TEN	SORBO	ARD				Shu	ut Down A	.11	6	Ignore o	utliers ir	n chai	rt scaling	
	° <b>f</b>	tensorb	ooards/*	l					×	Т	「ooltip sortin	ng metho	od: de	efault	*

- Method 2: Enter the following command in the .ipynb file window in JupyterLab (Obtain PID on the startup screen or using the command ps -ef | grep tensorboard): !kill PID
- Method 3: Click \_\_\_\_\_ as shown in the following figure to stop all started TensorBoard instances.

Figure 6-54 Stopping all started TensorBoard instances

AA	File Edit View Run Kernel Tabs Settings Help		
		C 🛛 Launcher 🛛 🗙 👎	TensorBoard 2
	KERNEL SESSIONS	×	
0	TERMINAL SESSIONS	× TensorBoard SCA	ALARS IMAG
	s_ terminals/1	SHUT DOWN	
0	TENSORBOARD SESSIONS	× Show data download lin	ks
50	🎓 TensorBoards/1	SHUT DOWN	caling
E	🎓 TensorBoards/2	SHUT DOWN	
°o		Tooltip sorting method: defa	iult 👻

• (Not recommended) Method 4: Close the TensorBoard window in JupyterLab. This method closes only the window, but the instance is still running on the backend.

# 6.4.7 Uploading and Downloading Data in Notebook

# 6.4.7.1 Uploading Files to JupyterLab

### 6.4.7.1.1 Scenarios

Easy and fast file uploading is a common requirement in AI development.

Before the optimization, ModelArts only allowed local files not exceeding 100 MB to be directly uploaded to a notebook instance. However, the files to be uploaded are not all stored locally, which may be from an open-source repository of GitHub, an open-source dataset (https://nodejs.org/dist/v12.4.0/node-v12.4.0-linux-x64.tar.xz), or OBS. Additionally, ModelArts did not show the file uploading progress or speed.

ModelArts has been optimized for better file uploading experience. It not only provides more file upload functions, but also displays more file upload details.

Optimized file uploading:

- Supports local files.
- Supports cloning files from open-source repositories in GitHub.
- Supports OBS files.
- Supports remote files.
- Supports visualized upload progress.

### 6.4.7.1.2 Uploading Files from a Local Path to JupyterLab

### **Upload Scenarios and Entries**

JupyterLab provides multiple methods for uploading files.

# Methods for Uploading a File

• For a file that does not exceed 100 MB, directly upload it, and details such as the file size, upload progress, and upload speed are displayed.

- For a file that exceeds 100 MB but does not exceed 5 GB, upload the file to OBS (an object bucket or a parallel file system), and then download the file from OBS to a notebook instance. After the download is complete, the file is deleted from OBS.
- For a file that exceeds 5 GB, upload it by calling ModelArts SDK or MoXing.
- For a file that shares the same name with an existing file in the current directory of a notebook instance, overwrite the existing file or cancel the upload.
- A maximum of 10 files can be uploaded at a time. The other files are in awaiting upload state. No folders can be uploaded. If a folder is required, compress it into a package, upload the package to notebook, and decompress the package in Terminal.

unzip xxx.zip # Directly decompress the package in the path where the package is stored.

For more details, search for the decompression command in mainstream search engines.

• When multiple files are uploaded in a batch, the total number of files to be uploaded and the number of files that have been uploaded are displayed at the bottom of the JupyterLab window.



# Prerequisites

You have used JupyterLab to open a running notebook environment.

# Upload Entry 1: Dragging a File to the File Browser Window

Drag the file to the blank area on the left of the JupyterLab window and upload it.



# Upload Entry 2: Clicking the File Upload Icon and Uploading a File

Click <sup>1</sup> in the navigation bar on the top of the window. In the displayed dialog box, drag or select a local file and upload it.



# Figure 6-56 File upload page

Add fi	les to Notebook	—	×
Local			
Git			
OBS	Drag and drop local files to upload. Compress the folder first		
P Remote			
	Or		
	SELECT FILES		
ତ			

# Uploading a Local File Less Than 100 MB to JupyterLab

For a file not exceeding 100 MB, directly upload it to the target notebook instance. Detailed information, such as the file size, upload progress, and upload speed are displayed.



Add fi	les to Notebook		_	$\times$
Local	Uploaded files (0/1)		SELE FILE	ECT
$\Box$	Name	Status		
Git	B typora-0-11-18-Windows.rar (67.7MB)	-	4% 7M	B/s
<b>∂</b> BS		Rows per page: 10 ▼	1-1 of 1 🛛 🕹	> >
СР Remote				

A message is displayed after the file is uploaded.

Figure	e 6-58 Uploaded				
Add fi	les to Notebook			-	×
Local	Uploaded files (1/1)		.,	SELE(	CT
$\Box$	Name	Status			
Git	urlCheck_green.zip (29.2MB)	<ul> <li>uploaded successfully</li> </ul>			
<b>∂</b> OBS		Rows per page: 10 ▼ 1-1 of 1	Κ	< >	· >I
8 Remote					

# Uploading a Local File with a Size Ranging from 100 MB to 5 GB to JupyterLab

For a file that exceeds 100 MB but does not exceed 5 GB, upload the file to OBS (an object bucket or a parallel file system), and then download the file from OBS to the target notebook instance. After the download is complete, the file is automatically deleted from OBS.

For example, in the scenario shown in the following figure, upload the file through OBS.

Add fi	les to Notebook	- ×
	Uploaded files (0/1)	C. SELECT FILES
Ø	Name	Status
Git	D data-set.zip (163.5MB)	10 The file size exceeds 100MB OBS TRANSIT
<b>∂</b> OBS		Rows per page: 10 ▼ 1–1 of 1  < < > >
& Remote		

Figure 6-59 Uploading a large file through OBS

To upload a large file through OBS, set an OBS path.

### Figure 6-60 Uploading a file through OBS

Add fi	les to Notebook	$ \times$
D	< RETURN AND CANCEL	3
Local	Set the OBS transit path	USE DEFAULT
57	Enter the OBS transit path, select a path from OBS File Browser, or use the default path	CONFIRM
OBS OBS Remote	OPEN OBS FILE BROWSER V 📀	

### **NOTE**

Set an OBS path for uploading local files to JupyterLab. After the setting, this path is used

by default in follow-up operations. To change the path, click in the lower left corner of the file upload window.

Method 1: Enter a valid OBS path in the text box and click **OK**. •

### Figure 6-61 Configuring an OBS path

D		í
Local	Set the OBS transit path	USE DEFAULT
Git	obs://test-	
<b>∂</b> OBS	OPEN OBS FILE BROWSER V	
P Remote		

Method 2: Select an OBS path in **OBS File Browser** and click **OK**. •

### Figure 6-62 OBS File Browser

Add fi	les to Notebook				$ \times$
D	< RETURN AND CAN	NCEL			
Local	Set the OBS transi	t path		U	SE DEFAULT
$\left( \begin{array}{c} \\ \\ \\ \\ \\ \end{array} \right)$	obs://cn-north-7-dev			×	CONFIRM
	CLOSE OBS FILE BE				
Ct)	obs				
-2	Back		Enter a name for query		C
Remote		Name	Last Modified	Туре	Size
	O 06024:		-	8	-
	0 06024:		-	8	-
	O automa		-	8	-
	O automa		-	8	-
	Ocn-north-7-dev		-	8	-
	O cqiaomeiyan-112		-	8	

Method 3: Use the default path. •

### Figure 6-63 Using the default path to upload a file

#### Add files to Notebook $\times$ **<** RETURN AND CANCEL USE DEFAULT Set the OBS transit path Ð Enter the OBS transit path, select a path from OBS File Browser, or use the default path CONFIRM Git OPEN OBS FILE BROWSER $\checkmark$ ብ OBS $\mathscr{S}$ Remote

### Figure 6-64 Setting an OBS path for uploading a local file

Add f	Add files to Notebook						
D	RETURN AND CANCEL						
Local	Set the OBS transit path	USE DEFAULT					
57	Enter the OBS transit path, select a path from OBS File Browser, or use the default path	CONFIRM					
Git	OPEN OBS FILE BROWSER 🗸						
0							
C Remote							
ଡ							

After the OBS path is set, upload a file.

Figure 6-65 Uploading a file

Add fi	les to Notebook		_	-	×
	Uploaded files (0/1)	5	SE F	LEC	
Git	Name	Status			
	D data-set.zip (163.5MB)	5	1% 5	50MB/	s
<b>⊕</b> OBS		Rows per page: 10 ▼ 1–1 of 1	< <	>	>
8 Remote					

### Decompressing a package

After a large file is uploaded to Notebook JupyterLab as a compressed package, you can decompress the package in Terminal.

unzip xxx.zip # Directly decompress the package in the path where the package is stored.

For more details, search for the decompression command in mainstream search engines.

# Uploading a Local File Larger Than 5 GB to JupyterLab

A file exceeding 5 GB cannot be directly uploaded to JupyterLab.

Figure 6-66 Failed to upload a file over 5 GB

已上传文件 (0/1)	□ 选择文件
名称	状态
Downloads0.rar (7GB)	④ 不支持上传超过5GB的文件

To upload files exceeding 5 GB, upload them to OBS. Then, call the ModelArts MoXing or SDK API in the target notebook instance to read and write the files in OBS.



Figure 6-67 Uploading and downloading large files in a notebook instance

The procedure is as follows:

- 1. Upload the file from a local path to OBS.
- 2. Download the file from OBS to the notebook instance by calling the ModelArts SDK or MoXing API.
  - Method 1: Call the ModelArts SDK to download a file from OBS.
     Example code:

from modelarts.session import Session session = Session() session.obs.copy("obs://*bucket-name/obs\_file.txt*","/home/ma-user/work/")

 Method 2: Call the ModelArts MoXing API for reading an OBS file. import moxing as mox

# Download the OBS folder sub\_dir\_0 from OBS to a notebook instance. mox.file.copy\_parallel('obs://bucket\_name/sub\_dir\_0', '/home/ma-user/work/sub\_dir\_0') # Download the OBS file obs\_file.txt from OBS to a notebook instance. mox.file.copy('obs://bucket\_name/obs\_file.txt', '/home/ma-user/work/obs\_file.txt')

If a .zip file is downloaded, run the following command on the terminal to decompress the package:

unzip xxx.zip # Directly decompress the package in the path where the package is stored.

After the code is executed, open the terminal shown in **Figure 6-68** and run the **ls /home/ma-user/work** command to view the file downloaded to the notebook instance. Alternatively, view the downloaded file in the left navigation pane of Jupyter. If the file is not displayed, refresh the page.

### Figure 6-68 Opening the terminal



M	File	Edit	View	Run	Kernel	Git	Tabs	Settings	Help
		+			1		C	4\$ <sup>*</sup>	
0	Fil	ter file:	s by nai	me					Q
0	Nam	ie					•	Last M	odified
		4,						secor	nds ago
	ß	1.txt						22 minu	tes ago
		202204	08-0856	547(''-	i ne inclui	g		an ho	our ago

Figure 6-69 File downloaded to a notebook instance

### **Error Handling**

If you download a file from OBS to your notebook instance and the system displays error message "Permission denied", perform the following operations for troubleshooting:

- Ensure that the target OBS bucket and notebook instance are in the same region. If the OBS bucket and notebook instance are in different regions, the access to OBS is denied.
- Ensure that the notebook account has the permission to read data in the OBS bucket.

### 6.4.7.1.3 Cloning an Open-Source Repository in GitHub

Files can be cloned from a GitHub open-source repository to JupyterLab.

- 1. Use JupyterLab to open a running notebook instance.
- 2. Click <sup>1</sup> in the navigation bar on the top of the JupyterLab window. In the

displayed dialog box, click Git on the left to go to the page for cloning files from a GitHub open-source repository.

#### Figure 6-70 File upload icon

+		<u>±</u>		C	\$
Name			*		Last Modified
• 🖪 Untitled.i	ipynb				an hour ago

#### Figure 6-71 Page for cloning files from a GitHub open-source repository

Add fi	iles to Notebook -	- ×	
<b>□</b>	GitHub open-source repository clone		
Local	Enter the GitHub open-source repository URL	CLONE	
Git			
OBS			
P Remote			

3. Enter a valid address of a GitHub open-source repository, select files from the displayed files and folders, and click **Clone**.

GitHub open-source repository address: https://github.com/jupyterlab/ extension-examples

Figure 6-72 Entering a valid address of a GitHub open-source repository

L+	GitHub open-source repository clone		
Local	https://github.com/jupyterlab/extension-examples	×	CI
$\overline{\mathbf{Q}}$	Branch: master 👻		
Git	Files preview		
<b>A</b>	.github		
063	command-palette		
Pomoto	commands		
Remote	■ completer		
View	the clone process.		
Figur	e 6-73 Process of cloning a repository		

extensior	-examples is being cloned
	(,

Begin clone

- Complete the clone.
- 2024-04-30

5.

Figure 6-74 Repository cloned



extension-e	examples 克隆成功
	返回

# **Error Handling**

• Failing to clone the repository may be caused by network issues. In this case, run the **git clone https://github.com/jupyterlab/extension-examples.git** command on the **Terminal** page to test the network connectivity.

Figure 6-75 Failed to clone the repository

上传文件到Notebook

Х



extension-examples仓库Clone失败

fatal: unable to access 'https://github.com/jupyterlab/extension-exa...

- 2	-	C =	
- 1	~	11	
- 2	$\sim$	-	-

• If the repository already exists in the current directory of the notebook instance, the system displays a message indicating that the repository name already exists. In this case, you can overwrite the existing repository or click

to cancel the cloning.

# 6.4.7.1.4 Uploading OBS Files to JupyterLab

In JupyterLab, you can download files from OBS to a notebook instance. Ensure that the file is not larger than 10 GB. Otherwise, the upload will fail.

- 1. Use JupyterLab to open a running notebook instance.
- 2. Click <sup>1</sup> in the navigation bar on the top of the JupyterLab window. In the

displayed window, click  $\curvearrowleft$  on the left to go to the OBS file upload page.

### Figure 6-76 File upload icon

+		<u>±</u>	G	${\bf O}^{+}$
Name				Last Modified
• 📃 Untitled.i	pynb			an hour ago

### Figure 6-77 OBS file upload

Add fi	files to Notebook –	×
Ę	OBS file upload	
Local	Enter the OBS file path, or select a path from OBS File Browser	PLOAD
Git	OPEN OBS FILE BROWSER V	
OBS		
& Remote		

- 3. Set an OBS file path in either of the following ways:
  - Method 1: Enter a valid OBS file path in the text box and click **Upload**.

### Figure 6-78 Entering a valid OBS file path

Add files to Notebook				
Local Git	OBS file upload	UPLOAD		
OBS				
8 Remote				

### **NOTE**

Enter an OBS file path instead of a folder path. Otherwise, the upload fails.

• Method 2: Open OBS File Browser, select an OBS file path, and click Upload.

### Figure 6-79 Uploading an OBS File

Add fi	iles to Not	ebook		- ×
L.	OBS file u	pload		
Local	obs://(	be00d5092fbdc0013d201342/f9937afa	a-26cb-4a1e-a002-a376897dbbbc-2022-07-28-15-43-4	X UPLOAD
Git	CLOSE OB	304bo00d5 / f0937afa 26	ch /a	
ብ	Back	504be0005 / 19957818-20	Enter a name for query	(
OBS		Name	Last Modified Type	e Size
S	⊙ idealU-		Thu, 28 Jul 2022 07:43:45 G 🗈	686MB
Remote	0	.)	Thu, 28 Jul 2022 08:43:34 G 🗅	109MB

# Figure 6-80 File uploaded

### 上传文件到Notebook



# **Error Handling**

There are three typical scenarios in which uploading a file failed.

• Scenario 1

Figure 6-81 File uploading failure



Possible causes:

- The OBS path is set to a folder instead of a file path.
- The file in OBS is encrypted. In this case, go to the OBS console and ensure that the file is encrypted.

Objects are k	vania unita of data d	storago In ORS (	Floc and foldors	are treated as	objecto Any file	hung can be unleaded and manage
You can use	OBS Browser+ to r	move an object to	any other folde	er in this bucket	objects. Any me	type can be uprodued and manage
For security r	easons, files cann	ot be previewed o	nline when you	access them fi	rom a browser. To	o preview files online, see How Do
Upload C	Object Cre	ate Folder	Delete	More 💌		
Na	me	Storage Cla	Size	1≡	Encrypted	Restoration Status

- The OBS bucket and notebook instance are not in the same region.
   Ensure that the OBS bucket to be read is in the same region as the notebook instance. You cannot access an OBS bucket in another region.
- The account does not have the permission to access the OBS bucket. In this case, ensure that the notebook account has the permission to read data in the OBS bucket.
- The OBS file has been deleted. In this case, make sure that the OBS file to be uploaded is available.
- Scenario 2

Figure 6-82 File uploading failure

#.wget-log3 uploading failure	
Contain the following invalid characters < > ' " ; \ ` = # \$ %% ^ &, file_name: #.wget-log3	
RETURN	

Possible causes:

The file name contains special characters such as <>'";\`=#\$%^&.

• Scenario 3

### Figure 6-83 File uploading failure



Possible causes: The uploaded file exceeded 10 GB.

### 6.4.7.1.5 Uploading Remote Files to JupyterLab

Files can be downloaded through remote file addresses to JupyterLab.

Method: Enter the URL of a remote file in the text box of a browser, and the file is directly downloaded.

- 1. Use JupyterLab to open a running notebook instance.
- 2. Click  $\stackrel{1}{=}$  in the navigation bar on the top of the JupyterLab window. In the

displayed window, click  ${\mathscr O}$  on the left to go to the remote file upload page.

#### Figure 6-84 File upload icon

+		<b>±</b>	C	$\mathbf{O}^{+}$
<b>I</b> /				
Name				Last Modified
• 🖪 Untitled.	ipynb			an hour ago

### Figure 6-85 Remote file upload page

Add f	iles to Notebook	- ×
Ę	Remote file upload	
Local	Enter the remote file URL	UPLOAD
Git	• The URL can be used to download files directly from the browser. For example: http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz	
OBS		
P Remote		

3. Enter a valid remote file URL, and the system automatically identifies the file name. Then, click **Upload**.

Figure 6-86 Entering a valid remote file URL



Figure 6-87 Remote file uploaded 上传文件到Notebook



# **Error Handling**

Failing to upload the remote file may be caused by network issues. In this case, enter the URL of the remote file in the text box of a browser to check whether the file can be downloaded.



# 6.4.7.2 Downloading a File from JupyterLab to a Local Path

Files created in JupyterLab can be downloaded to a local path.

- If a file is less than or equal to 100 MB, directly download it from JupyterLab. For details, see **Downloading a File Less Than or Equal to 100 MB**.
- If a file is larger than 100 MB, use OBS to transfer it to your local path. For details, see **Downloading a File Larger Than 100 MB**.

# Downloading a File Less Than or Equal to 100 MB

In the JupyterLab file list, right-click the file to be downloaded and choose **Download** from the shortcut menu. The file is downloaded to your browser's downloads folder.





# Downloading a File Larger Than 100 MB

Use OBS to transfer the file from the target notebook instance to the local path. To do so, perform the following operations:

 In the notebook instance, create an IPYNB file larger than 100 MB and use MoXing to upload it to OBS. Example code is as follows: import moxing as mox mox.file.copy('/home/ma-user/work/obs\_file.txt', 'obs://bucket\_name/obs\_file.txt')

**/home/ma-user/work/obs\_file.txt** is the path to the file stored in the notebook instance. **obs://bucket\_name/obs\_file.txt** is the path of the file uploaded to OBS, where **bucket\_name** is the name of the bucket created in OBS, and **obs\_file.txt** is the uploaded file.

- 2. Use OBS or ModelArts SDK to download the file from OBS to the local path.
  - Method 1: Use OBS to download the file.
  - Download **obs\_file.txt** from OBS to the local path. If a large amount of data is to be downloaded, use OBS Browser+ to download.
  - Method 2: Use ModelArts SDK to download the file.
    - i. Download and install the SDK locally.
    - ii. Authenticate sessions.
    - iii. Download the file from OBS to the local path. Example code is as follows:

from modelarts.session import Session

# Hardcoded or plaintext AK/SK is risky. For security, encrypt your AK/SK and store them in the configuration file or environment variables. # In this example, the AK/SK are stored in environment variables for identity authentication. Before running this example, set environment variables HUAWEICLOUD\_SDK\_AK and HUAWEICLOUD\_SDK\_SK. \_\_AK = os.environ["HUAWEICLOUD\_SDK\_AK"] \_\_SK = os.environ["HUAWEICLOUD\_SDK\_SK"] # Decrypt the password if it is encrypted. session = Session(access\_key=\_\_AK,secret\_key=\_\_SK, project\_id='\*\*\*', region\_name='\*\*\*')

session.download\_data(bucket\_path="/bucket\_name/obs\_file.txt",path="/home/user/ obs\_file.txt")

# 6.5 Local IDE

# 6.5.1 Operation Process in a Local IDE

ModelArts allows you to remotely access notebook instances from a local IDE to develop AI models based on PyTorch, TensorFlow, or MindSpore. The following figure shows the operation process.

1. Configure a local IDE.

Configure a local IDE on your PC.

You can use **PyCharm**, **VS Code**, or **SSH tools** to access a notebook instance from a local IDE. PyCharm and VS Code can be automatically configured using plug-ins or manually configured.

### 2. Create a notebook instance.

On the ModelArts management console, create a notebook instance with a proper AI engine and remote SSH enabled.

- 3. Use the local IDE to remotely access ModelArts DevEnviron.
- 4. Upload data and code to the development environment.
  - Copy the code to the local IDE, which will automatically synchronize the code to the in-cloud development environment.
  - If the data is less than or equal to 500 MB, directly copy the data to the local IDE.
  - When creating a training job, if the volume of data is greater than 500 MB, upload the data to OBS and then to EVS.
- 5. Upload the training script and dataset to the OBS directory.
- 6. Submit a training job.
  - Submit a training job in the local IDE.
  - Submit a training job on the ModelArts management console. .

# 6.5.2 Local IDE (PyCharm)

# 6.5.2.1 Connecting to a Notebook Instance Through PyCharm Toolkit

### 6.5.2.1.1 PyCharm Toolkit

AI developers use PyCharm tools to develop algorithms or models. Therefore, ModelArts provides PyCharm Toolkit to help AI developers quickly submit locally developed code to a training environment on ModelArts. With PyCharm Toolkit, developers can quickly upload code, submit training jobs, and obtain training logs for local display so that they can better focus on local code development. For details about how to download and install PyCharm Toolkit, see **Installing Through Marketplace**.

# Constraints

- Currently, only PyCharm 2019.2 or later is supported, including the community and professional editions.
- Only PyCharm of the professional edition can be used to access the notebook development environment. PyCharm Toolkit cannot be used to remotely access notebook instances using RightCloud accounts.
- You can use a community or professional edition of PyCharm Toolkit to submit training jobs. The latest version of PyCharm Toolkit can be used only to submit training jobs of the new version.
- PyCharm Toolkit supports PyCharm of the Window version.

# **Available Functions**

Function	Description	Reference
Remote SSH	The notebook development environment can be accessed through remote SSH.	Connecting to a Notebook Instance Through PyCharm Toolkit
Model training	Code developed locally can be quickly submitted to ModelArts and a training job of the new version is automatically created. During the running of the training job, training logs can be obtained and displayed on a local host.	<ul> <li>Submitting a Training Job (New Version)</li> <li>Stopping a Training Job</li> <li>Viewing Training Logs</li> </ul>
OBS-based upload and download	Local files or folders can be uploaded to OBS and files or folders can be downloaded from OBS to a local directory.	Uploading Data to a Notebook Instance Using PyCharm

**Table 6-8** Toolkit functions of the latest version

# 6.5.2.1.2 Downloading and Installing PyCharm Toolkit

Before using PyCharm Toolkit, install and configure it in PyCharm by following the instructions provided in this section.

# **Prerequisites**

PyCharm community or professional 2019.2 or later has been installed locally.

- Only PyCharm of the professional edition can be used to access the notebook development environment.
- You can use a community or professional edition of PyCharm Toolkit to submit training jobs. PyCharm Toolkit 2.x can be used to submit only the old version of training jobs, and the latest version of PyCharm Toolkit can be used to submit only the new version of training jobs.

# Installing Through Marketplace

In PyCharm, choose **File** > **Settings** > **Plugins**, search for **ModelArts** in Marketplace, and click **Install**.

### Figure 6-90 Installing through Marketplace

-	settings				
[0			Plugins	Marketplace	e Installed
>	Appearance & Behavior		Q+ ModelArts		م و
	Keymap		Search Results (1)	Sort By: Relevance 🔫	
>	Editor		9 P El Madal Arta		
	Plugins				
>	Version Control				
>	Project: models				Plugin hor
>	Build, Execution, Deployme	nt			Al develop
>	Languages & Frameworks				ModelArts
>	Tools				With the P

### D NOTE

- The version installed in Marketplace is the latest version.
- If ModelArts cannot be found in Marketplace, your network may be restricted. Ensure that you can access the Internet.

### 6.5.2.1.3 Connecting to a Notebook Instance Through PyCharm Toolkit

ModelArts provides the PyCharm plug-in PyCharm Toolkit for you to remotely access a notebook instance through SSH, upload code, submit a training job, and obtain training logs for local display.

# Prerequisites

PyCharm professional 2019.2 or later has been installed locally. Remote SSH applies only to the PyCharm professional edition. **Download PyCharm** and install it.

PyCharm Toolkit cannot be used to remotely access notebook instances using RightCloud accounts.

### Step 1 Create a Notebook Instance

Create a notebook instance with remote SSH enabled and whitelist configured. Ensure that the instance is running. For details, see **Creating a Notebook Instance**.

# Step 2 Download and Install PyCharm Toolkit

In PyCharm, choose File > Settings > Plugins, search for ModelArts in Marketplace, and click Install. For details, see Downloading and Installing PyCharm Toolkit.

# Step 3 Add More Regions

- 1. On the PyCharm interface, choose **ModelArts > Edit Credential**. The **Edit Credential** dialog box is displayed.
- Contact the region operations company to obtain the YAML configuration file and host information. In the Edit Credential dialog box, click Config to import the downloaded YAML file. After the file is imported, the message Import successful is displayed, indicating that the region information is configured.

# Step 4 Log In to the Plug-in

To use the AK/SK pair for login authentication, perform the following steps:

1. Open PyCharm with Toolkit installed. Choose **ModelArts > Edit Credential** from the menu bar.

### Figure 6-91 Edit Credential

P	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efa	ctor	R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> indow	<u>M</u> oo	lelArts	<u>H</u> elp	<b>b</b>
P	/thonF	roject	👌 📥 ma	ain.py										denti	
ಭ	📄 Р	roject	•		€	÷	Ŧ	۵	- 👔	main.	py ×	л т	Noteboo raining	ok Joh	> >
Proj	~	pytho	nProje	<b>ct</b> C:\Users					roje 1	Ę	# This	1S 8	samp	Le P	ytho

- 2. In the displayed dialog box, select the region where ModelArts is located, enter the AK and SK, and click **OK**. For details about how to obtain the AK and SK, see **How Do I Obtain an Access Key**?.
  - Region: Select a region from the drop-down list. It must be the same as the region of the ModelArts console.
  - **Project**: After the region is selected, the project is automatically filled.
  - Access Key ID: Enter the AK.
  - Secret Access Key: Enter the SK.
- 3. View the verification result.

In the **Event Log** area, if information similar to the following is displayed, the access key has been successfully added:

16:01Validate Credential Success: The credential is valid.

If an authentication fails, refer to **What Should I Do If an Error Occurs When I Edit a Credential in PyCharm Toolkit?** for the solution.

# Step 5 Automatically Configure PyCharm Toolkit

 In the local PyCharm development environment, choose ModelArts > Notebook > Remote Config... and configure PyCharm Toolkit. Figure 6-92 Remotely connecting to PyCharm Toolkit

,	M	odelArts	<u>H</u> elp		pythonProject - main.py	
	🗸 Edit Credential					
		Notebook			Remote Config	
		Training	Job	>		

2. Choose the target instance from the drop-down list, where all notebook instances with remote SSH enabled under the account are displayed.

### Figure 6-93 Notebook list

Notebook List	×
Notebook:	notebook-b7d1 👻
RunningStatus:	RUNNING
Flavor:	modelarts.vm.cpu.2u
ImageName:	PyTorch1.4-CUDA10.1-cuDNN7-Ubuntu18.04
SshUrl:	ssh://ma-user@dev-modelarts
KeyPairName:	KeyPair-3x-bj4
KeyPair:	
PathMappings:	/home/ma-user/work/pythonProject
	Apply Cancel

- KeyPair: Select the locally stored key pair of the notebook instance for authentication. The key pair created during the notebook instance creation is saved in your browser's default downloads folder.
- PathMappings: Synchronization directory for the local IDE project and notebook, which defaults to /home/ma-user/work/Project name and is adjustable.
- 3. Click **Apply**. After the configuration is complete, restart the IDE for the configuration to take effect.

After the restart, it takes about 20 minutes to update the Python interpreter for the first time.

# Step 6 Access a Notebook Instance Through PyCharm Toolkit

Click the notebook instance name and connect it to the local IDE as prompted. The connection is kept for 4 hours by default.

### Figure 6-94 Starting the connection



To interrupt the connection, click the notebook name and disconnect it from the local IDE as prompted.

#### Figure 6-95 Interrupting the connection



# Step 7 Upload Local Files to the Notebook Instance

Code in a local file can be copied to the local IDE, which will automatically synchronize the code to the in-cloud development environment.

### Initial synchronization

In the **Project** directory of the local IDE, right-click **Deployment** and choose **Upload to** *Notebook name* from the shortcut menu to upload the local project file to the specified notebook instance.

	<u> </u>		- ,	g									
	Pr	oject 🔻				€	ŧ	Ť	\$	-	6	trair	_mn
~		Recogni		New	Þ								fro
	~	train > In da	彩 「回	Cu <u>t</u> <u>C</u> opy Copy Path Paste	Ctrl+X Ctrl+C Ctrl+V								ses ttt pri
	1111	cu te ڈی te ڈی tra External		Find <u>U</u> sages Find in Files Repl <u>a</u> ce in Files Inspect Code	Alt+F7 Ctrl+Shift+F Ctrl+Shift+R						6		pri
	70	Scratche		 <u>R</u> efactor Clean Python Compi	► iled Files								
				Add to F <u>a</u> vorites	•								
				<u>R</u> eformat Code Optimi <u>z</u> e Imports	Ctrl+Alt+L Ctrl+Alt+O								
				Open In	•								
			G	Local <u>H</u> istory Reload from Disk	+								
			÷	Compare With	Ctrl+D								
				Mark Directory as									
			†ţ	Deployment	•	<u>∓ U</u> p	load	to No	ote-el	EbdZ	7		
				Remove BOM		<u>+ D</u> o	wnlo	ad fro	om N	ote-	eEbd	Z	
			<b>□ ○</b> + +	Diagrams Create Gist Shutdown Kernel ModelArts Upload ModelArts Downloa	►								

Figure 6-96 Synchronizing local data to a notebook instance

### Follow-up synchronization

After modifying the code, press **Ctrl+S** to save it. The local IDE will automatically synchronize the modification to the specified notebook instance.

After PyCharm Toolkit is installed, **Automatic Upload** is automatically enabled in the local IDE for automatically uploading the files in the local directory to the target notebook instance. If **Automatic Upload** is not enabled, enable it by referring to the following figure.



Figure 6-97 Enabling Automatic Upload

# Step 8 Remotely Debug the Code

Click **Interpreter** in the lower right corner of the local IDE and select a notebook Python interpreter.

### Figure 6-98 Selecting a Python interpreter

Python Interpreter
🔩 🙀 Note-eEbdZ Python 3.7.10 (sftp://ma-user@10.155.1a-user/anaconda3/envs/TensorFlow-2.1.0/bin/python)
Interpreter Settings
Add Interpreter
🔋 Note-eEbdZ is disconnected Note-eEbdZ Python 3.7.10sorFlow-2.1.0/bin/python) 偹 476 of 1979M

Run the code in the notebook instance. The logs are displayed locally.

Figure 6-99 Runtime logs

D	DocTest ) codes ) 🐞 train_mnist they						
g			💰 customize_service.py 🛛 🚜 train_mnist_th.py 👋 📇 README.md 🗵				
■ I: Structure II: Proje	I BOOFest (ADDeTest I as I as I aconfigion I aconfigi	Que     AA     T     S     H     S     T       43     builder.add_meta_graph_and_variables(       44     sess, [fr.jaaved_model.tag_constants.SERVING],       45     graphinture_def_mant_(       46     'promotict_images':       47     promotiction signature.					
			Y = [st1 - manenaln:       28     tf_maneain_main_main_				
	Run: 💣 train mnist tf 🛛						
	ssh://ma-user@10.155.101.174: Training model	38881/hone/na-	user/miniconda3/envs/Tensorflow-1.15.0/bin/python -u /home/ma-user/work/testssh/testok/codes/train_mnist_tf.py				

Click **Run/Debug Configurations** in the upper right corner of the local IDE to set runtime parameters.

### Figure 6-100 Setting runtime parameters (1)



Select the Python interpreter that remotely connects to the target notebook instance.

Run/Debug Configurations			
+ - ि ☐ ♪ ▲ ▼ № ↓3	Name: Statistics		
∲ mnist_tutorial ∳resnet > ∦ Templates	Script path:   Parameters:  Environment	Eljäheityenetyenetääheityene <u>n</u> tääheityene <u>n</u> tääheityenen <u>a</u> EP,py	
	Environment variables:	PYTHONUNBUFFERED=1	
	Python interpreter: Interpreter options:	देश Project Default (Note-RKBYR Python 3.7.6 (sftp://ma-user@tere=besister)	1.05892/hor ▼
	Working directory:	ENELSTALSTALSTRESTALSTA	
	Path mappings:	рутнопратн	
	Add source roots to F	YTHONPATH	
	Run with Python Cons		
	Redirect input from:		
	▼ Before launch		

Figure 6-101 Setting runtime parameters (2)

To debug code, set breakpoints and run the program in debug mode.

inguic o ioz Kanning the program in acoug mou	Figure	6-102	Running	the	program	in	debug	mode
---	--------	-------	---------	-----	---------	----	-------	------

🛃 cust	nize_service.py 🛛 🔥 train_mnist_tf.py 👋 🏭 README.md 🗵	
	TTOM LENSONTLOW.CAMMPLES.LUCUITALS.MNIISL IMPOLE INPUL_UALA	
	tf.flags.DEFINE_integer('max_steps', 10, 'number of training iterations.')	
	tf.flags.DEFINE_string('data_url', '/home/ma-user/work/ <u>testssh</u> /train/data/ <u>Mni</u>	<u>s</u> 1
	t <mark>f.flags.DEFINE_string(</mark> 'train_url', '/home/jnn/temp/delete', 'saved model dir	
	FLAGS = tf.flags.FLAGS	
	def main(*args):	
	# Train model	
	ss = 100	
	print(ss)	
21 🔵	<pre>print('Training model')</pre>	

In debug mode, the code execution is suspended in the specified line, and you can obtain variable values.

def main(\*args): args: ['/home/ma-user/work/testssh/testsk/codes/train\_mnist\_tf.py']
# Train model
# Train mo

### Figure 6-103 Viewing variable values in debug mode

# 6.5.2.2 Manually Connecting to a Notebook Instance Through PyCharm

A local IDE supports PyCharm and VS Code. You can use PyCharm or VS Code to remotely connect the local IDE to the target notebook instance on ModelArts for running and debugging code.

This section describes how to use PyCharm to access a notebook instance.

# Prerequisites

- PyCharm professional 2019.2 or later has been installed locally. The PyCharm professional edition is available because remote SSH applies only to the professional edition.
- A notebook instance has been created with remote SSH enabled. Ensure that the instance is running. For details, see **Creating a Notebook Instance**.
- The address and port number of the development environment are available. To obtain this information, go to the notebook instance details page.

### Figure 6-104 Instance details page

Address	ssh://ma-user@ <mark>dev-modelarts-</mark>	.com:32651
	Access address of the	Port
Authentication	KeyPair-9a64 🖉 development environmen	t number

• The key pair is available.

A key pair is automatically downloaded after you create it. Securely store your key pair. If an existing key pair is lost, create a new one.

# Step 1 Configure SSH

- 1. In your local PyCharm development environment, choose **File** > **Settings** > **Tools** > **SSH Configurations** and click **+** to add an SSH configuration.
  - Host: address for accessing the cloud development environment. Obtain the address on the page providing detailed information of the target notebook instance.
  - Port: port number for accessing the cloud development environment.
     Obtain the port number on the page providing detailed information of the target notebook instance.
  - **User name**: consistently set to **ma-user**.
  - Authentication type: key pair
  - Private key file: locally stored private key file of the cloud development environment. It is the key pair file automatically downloaded when you created the notebook instance.
- 2. Click low to rename the connection. Then, click **OK**.
- 3. After the configuration is complete, click **Test Connection** to test the connectivity.
- 4. Select **Yes**. If "Successfully connected" is displayed, the network is accessible. Then, click **OK**.
- 5. Click **OK** at the bottom to save the configuration.

# Figure 6-105 Configuring SSH

The Face Ten Tenders Hare Brann, 18, Tens and Thursen Tenners Tab			
DocTest Malabed			
¥ ■ Project + ② ÷ ♥ ー			¢ - s
			±a× ů
Image: A state of the state			
t modes 0 mabod		<ul> <li>motebook-samples-1610346734</li> <li>mosek</li> </ul>	
di di as		<ul> <li>ipynb_checkpoints</li> </ul>	1
af assar		🕨 🖿 theia	8.
g, conspyton		► Bit assesseda?	
& READWE md			20
C testmind.py			
🛙 qi Ke osu funor (oʻta)		<ul> <li>Im images</li> <li>Im images</li> </ul>	
Illi External Libraries			
Scratches and Consoles	Conto Ello Chill (Shift) N	► 🖿 model1	
		<ul> <li>monotebook data</li> <li>monotets</li> </ul>	
	Navigation Bar, Alt+Home	<ul> <li>Battest (</li> <li>Battest (</li> </ul>	
		ascend tf_sample.ipynb	
	Drop files here to open		
		Gillion and State and Stat	
		g, lusion, resakjson Æ image buildjourb	
			pynb
		spip.conf	
		🚯 t10k-labels-ldx1-ubyte.gz	
		train-labels-idx1-ubyte.gz	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
& Run:  testmind (1) >>			¢ -
<pre>np_resource = np.otype([("resource", np.uoyte, 1)]) ok</pre>	BY 14:26		emote host 10
Process finished with exit code 0	p 1417	Deployment configuration to 10.155.101.174 has been created.	. t.
III ≜ 1000 ▶ § Nun 17 File Transfer ♦ Python Console 12 Terminal Deployment configuration to 10.155.101.174 has been created. // Configure (34 misutes ago)	🔿 Updating skeletons.	Remote Python 3.6.4 (stuensorflow	Event Log -1.3/bir/python)

# Step 2 Obtain the Path to the Virtual Environment Built in the Development Environment

- 1. Choose **Tools** > **Start SSH Session** to access the cloud development environment.
- 2. Run the following command to view the Python virtual environments built in the current environment in the **README** file in **/home/ma-user/**: cat /home/ma-user/README

- 3. Run the **source** command to switch to a specific Python environment.
- 4. Run **which python** to obtain the Python path and copy it for configuring the Python interpreter on the cloud.

**Figure 6-106** Obtaining the path to the virtual environment built in the development environment



# Step 3 Configure a Python Interpreter

1. Choose **File** > **Settings** > **Project**: *Python project* > **Python Interpreter**. Then,

click was and **Add** to add an interpreter.

- 2. Select **Existing server configuration**, choose the SSH configuration from the drop-down list, and click **Next**.
- 3. Configure the Python interpreter.
  - Interpreter: Enter the Python path copied in step 1, for example, / home/ma-user/anaconda3/envs/Pytorch-1.0.0/bin/python.

If the path is **~/anaconda3/envs/Pytorch-1.0.0/bin/python**, replace **~** with **/home/ma-user**.

- Sync folders: Set this parameter to a directory in the cloud development environment for synchronizing local project directory files. A directory in / home/ma-user is recommended, for example, /home/ma-user/work/ projects, because other directories may be prohibited from accessing.
- 4. Click ! on the right and select **Automatically upload** so that the locally modified file can be automatically uploaded to the container.
- 5. Click Finish.

The local project file has been automatically uploaded to the cloud environment. Each time a local file is modified, the modification is automatically synchronized to the cloud environment.

In the lower right corner, the current interpreter is displayed as a remote interpreter.



Figure 6-107 Configuring a Python interpreter

# Step 4 Install the Dependent Library for the Cloud Environment

After accessing the development environment, you can use different virtual environments, such as TensorFlow and PyTorch. However, in actual development, you need to install dependency packages. Then, you can access the environment through the terminal to perform operations.

Choose **Tools** > **Start SSH Session** and select the configured development environment. Run the **pip install** command to install the required dependency packages.



# Step 5 Debug Code in the Development Environment

You have accessed the cloud development environment. Then, you can write, debug, and run the code in the local PyCharm. The code is actually executed in the cloud development environment, and the Ascend AI resources on the cloud are used. In this way, you compile and modify code locally and run the code in the cloud.

Run the code in the local IDE. The logs can be displayed locally.

### Figure 6-108 Debugging code



Click **Run/Debug Configurations** in the upper right corner of the local IDE to set runtime parameters.

Figure 6-109 Setting runtime parameters



To debug code, set breakpoints and run the program in debug mode.

### Figure 6-110 Code breakpoint



### Figure 6-111 Debugging in debug mode



In debug mode, the code execution is suspended in the specified line, and you can obtain variable values.

Figure 6-112 Debug mode



Before debugging code in debug mode, ensure that the local code is the same as the cloud code. If they are different, the line where a breakpoint is added locally may be different from the line of the cloud code, leading to errors.

When configuring a Python interpreter in the cloud development environment, select **Automatically upload** so that any local file modification can be automatically uploaded to the cloud. If you do not select **Automatically upload**, manually upload the directory or code after you modify the local code. For details, see **Step 7 Upload Local Files to the Notebook Instance**.

# 6.5.2.3 Submitting a Training Job Using PyCharm Toolkit

# 6.5.2.3.1 Submitting a Training Job (New Version)

You can use PyCharm Toolkit of the latest version to quickly submit the locally developed training code to ModelArts for training.

# Prerequisites

- A training code project exists in the local PyCharm.
- You have created a bucket and folders in OBS for storing datasets and trained models. Data used by the training job has been uploaded to OBS.
- The credential has been configured. For details, see Using Access Keys for Login.
- PyCharm Toolkit of the latest version is available for submitting a training job of the new version only.

# **Configuring Training Job Parameters**

1. In PyCharm, open the training code project and training boot file, and choose **ModelArts** > **Training Job** > **New...** on the menu bar.

ModelArts Help Edit Credential Notebook > Training Job > New... Stop

Figure 6-113 Edit training job configuration

2. In the displayed dialog box, configure the training job parameters. For details, see **Table 6-9**.
# Table 6-9 Training job parameters

Parameter	Description
Job Name	<ul><li>Name of a training job</li><li>The system automatically generates a name. You can rename it based on the following naming rules:</li><li>The name contains 1 to 64 characters.</li></ul>
	<ul> <li>Letters, digits, hyphens (-), and underscores (_) are allowed.</li> </ul>
Job Description	Brief description of a training job
Algorithm Source	Source of the training algorithm. The options are <b>Frequently-used</b> and <b>Custom</b> .
	<b>Frequently-used</b> refers to the frequently-used AI engines supported by ModelArts Training Management. If the AI engine you use is not in the supported list, you are advised to create a training job using a custom image.
Al Engine	Select the AI engine and the version used in code. The supported AI engines are the same as on the ModelArts management console.
Boot File Path	Training boot file. The selected boot file must be a file in the current PyCharm training project. This parameter is displayed if <b>Algorithm Source</b> is set to <b>Frequently-used</b> .
Code Directory	Training code directory. The system automatically sets this parameter to the directory where the training boot file is located. You can change the parameter value to a directory that is in the current project and contains the boot file.
	If the algorithm source is a custom image and the training code has been built in the image, this parameter can be left blank.
Image Path(optional)	URL of the SWR image
Boot Command	Command for starting a training job, for example, bash /home/work/run_train.sh python { <i>Python boot</i> <i>file and parameters</i> }. This parameter is displayed if Algorithm Source is set to Custom.
	If the command does not contain the <b>data_url</b> or <b></b> <b>train_url</b> parameter, the tool automatically adds the two parameters to the end of the command when submitting the training job. The two parameters correspond to the OBS path for storing training data and the OBS path for storing training output, respectively.

Parameter	Description
Data OBS Path	OBS path for storing training data, for example, <b>/test-</b> modelarts2/mnist/dataset-mnist/, in which test- modelarts2 indicates a bucket name.
Training OBS Path	OBS path. A directory is automatically created in the path for storing a trained model and training logs.
Running Parameters	Running parameters. If you want to add some running parameters to your code, add them here. Separate multiple running parameters with semicolons (;), for example, <b>key1=value1;key2=value2</b> . This parameter can be left blank.
Specifications	Type of resources used for training. Currently, public resource pools and dedicated resource pools are supported.
	Dedicated Resource Pool specifications are identified by Dedicated Resource Pool.
Compute Nodes	Number of compute nodes. If this parameter is set to <b>1</b> , the system runs in standalone mode. If this parameter is set to a value greater than 1, the distributed computing mode is used at the background.
Available/Total Nodes	When <b>Specifications</b> is set to a dedicated resource pool, the number of available nodes and the total number of nodes are displayed. The value of <b>Compute Nodes</b> cannot exceed the number of available nodes.



📕 Edit Training Job (	Configurations					
JobName:	MA-new-models-07-0	04-15-06-7				
Job Description:						
	Frequently-used C	Custom				
	Al Engine:	TensorFlow		TF-1.13.1-python3.0		
Algorithm Source:	Boot File Path:	D <sup>3</sup>	,models\official\c	v\deeplabv3\train.py		
	Code Directory:		\models\official			
	Image Path(optional):					
Data OBS Path:						
Training OBS Path:						
Running Parameters:						
Specifications:	CPU:2*vCPUs 16GB					
Compute Nodes:						
				Apply and Run	Cancel	

inguie o ins	Conniganing	li all'illig job	parameter	(acalcated )	esource po	01)
🖺 Edit Training Job C	Configurations					×
JobName:	MA-new-models-07-0	04-15-12-606				
Job Description:						
	Frequently-used C	Custom				
	Al Engine:	TensorFlow		TF-1.13.1-python3.6		•
Algorithm Source:	Boot File Path:	D:1		¦∖train.py		
	Code Directory:	D:\	\models\official			
	Image Path(optional):					
Data OBS Path:						
Training OBS Path:	obs://					
Running Parameters:						27
* Specifications:	pool-maostest-noteb	ook: CPU:8vCPUs 32				
Compute Nodes:						
				Apply and Run	Cancel	Apply

Figure 6-115 Configuring training job parameter (dedicated resource pool)



Edit Training Job C	Configurations	×
* JobName:	MA-new-models-07-04-15-14-100	
Job Description:		
	Frequently-used Custom	
	Image Path: swr.4	
* Algorithm Source:	pythen run py Boot Command:	
	Code Directory:	
* Data OBS Path:		
Training OBS Path:		
Running Parameters:		
* Specifications:	CPU:2*vCPUs 16GB	
* Compute Nodes:		
	Apply and Run Cancel	

3. After setting the parameters, click **Apply and Run**. Then, local code is automatically uploaded to the cloud and training is started. The training job running status is displayed in the **Training Log** area in real time. If information similar to **Current training job status: Successful** is displayed in the training log, the training job has been successfully executed.

#### 

- After you click Apply and Run, the system automatically executes the training job. To stop the training job, choose ModelArts > Training Job > Stop on the menu bar.
- If you click **Apply**, the job is not started directly, and the training job settings are saved instead. To start the job, click **Apply and Run**.

Figure 6-117 Training log example

ModelArts Event Log Event Log 🗢 🖛	ModelArts Training Logjob-ma-iceberg-11-22-16.0
Begin to check training configuration. Begin to upload training code.	Log Name: job-ma-iceberg-11-22-16.0
MA-iceberg-11-22-16/code/src/train_iceberg_new.py: 100% Files are uploaded successfully.	INFO:Current training job status: Initializing INFO:Current training job status: Running
Begin to get training job pre version.	INFO:Current training job status: Successful
Begin to create training job. Training job is created successfully.	<ul> <li>Restarting DNS forwarder and DHCP server dnsmasq</li> <li>done.</li> </ul>
Job id: 411903, version id: 620356 Begin to get training job log.	[Modelarts Service Log]user: uid=1101(work) gid=1101(work) groups=1101(work) [Modelarts Service Log]pwd: /home/work

# 6.5.2.3.2 Stopping a Training Job

You can stop a running training job.

# Stopping a Job

When a training job is running, choose **ModelArts > Training Job > Stop** on the PyCharm menu bar to stop the job.

Figure	6-118	Stopping	a job
--------	-------	----------	-------

<u>M</u> odelArts	<u>H</u> elp		ру	/thonProj
🗸 Edit Cree	dential			
Noteboo	ok	▶_		
Training	Job	►		New
our shell scrip	ots?			Stop

#### 6.5.2.3.3 Viewing Training Logs

This section describes how to view training job logs.

# **Viewing Training Logs in OBS**

When you submit a training job, the system automatically creates a folder with the same name as the training job in the configured OBS path to store the model, logs, and code outputted after training is complete.

For example, when the **train-job-01** job is submitted, a folder named **train-job-01** is created in the **test-modelarts2** bucket. In this folder, three sub-folders (**output**, **log**, and **code**) are created to store the outputted model, logs, and training code, respectively. Sub-folders will be created in the **output** folder based on your training job version. The following is an example of the folder structure: test-modelarts2

|---train-job-01 |---output |---log |---code

# Viewing Training Logs in Toolkit

In PyCharm, click **ModelArts Training Log** in the lower right corner of the page. The training logs are displayed.

Figure 6-119 Viewing Training Logs

ModelArts Training Logjob-ma-iceberg-11-22-16.0	
Log Name: job-ma-iceberg-11-22-16.0	
INFO:Current training job status: Initializing INFO:Current training job status: Running INFO:Current training job status: Successful * Restarting DNS forwarder and DHCP server dnsmasq dome.	
<pre>[Modelarts Service Log]user: uid=1101(work) gid=1101(work) groups=1101(work) [Modelarts Service Log]pwd: /home/work [Modelarts Service Log]oot_file: sr//rain_iceberg_new.py [Modelarts Service Log]loot_file: sr//train_iceberg_122-16.log [Modelarts Service Log]command: sr//train_iceberg_new.py -/data_url=s3://cnnorth1-job-test/iceberg/iceberg/num_gpus=</pre>	1train
[Modelarts Service Log]MODELARTS_IPOIB_DEVICE: [Modelarts Service Log]dependencies_file_dir: /home/work/user-job-dir/src [Modelarts Service Log][modelarts_create_log] modelarts-pipe found	
2 Event Log ModelArts Tr	raining Log

# 6.5.2.4 Uploading Data to a Notebook Instance Using PyCharm

If the data is less than or equal to 500 MB, directly copy the data to the local IDE.

If the data is larger than 500 MB, upload the code to OBS and then to the notebook instance.

- 1. Upload data to OBS.
- 2. Call the **mox.file.copy\_parallel** MoXing API provided by ModelArts in the terminal of the local IDE to transfer data from OBS to the notebook instance.

Figure 6-120 Uploading data to a notebook Instance through OBS



The following shows how to enable terminal in PyCharm.



Figure 6-121 Enabling the terminal in PyCharm

The following shows how to use MoXing in the terminal of the local IDE to download files from OBS to a development environment:

```
# Manually access the development environment.
cat /home/ma-user/README
# Select the source environment.
source /home/ma-user/miniconda3/bin/activate MindSpore-python3.7-aarch64
# Enter python and press Enter to enter the Python environment.
python
# Use MoXing for access.
import moxing as mox
# Download a folder from OBS to EVS.
mox.file.copy_parallel('obs://bucket_name/sub_dir_0', '/tmp/sub_dir_0')
```

# 6.5.3 Local IDE (VS Code)

# 6.5.3.1 Connecting to a Notebook Instance Through VS Code

After creating a notebook instance with remote SSH enabled, you can use VS Code to access the development environment in any of the following ways:

 Connecting to a Notebook Instance Through VS Code with One Click (Recommended)

In this mode, click **Access VS Code** in the **Operation** column of a notebook instance on the ModelArts console to open VS Code and connect to the instance.

• Connecting to a Notebook Instance Through VS Code Toolkit (Recommended)

In this mode, log in to the ModelArts VS Code Toolkit plug-in and use it to connect to an instance.

# Manually Connecting to a Notebook Instance Through VS Code In this mode, use the VS Code Remote-SSH plug-in to configure connection information and connect to an instance.

# 6.5.3.2 Installing VS Code

#### Download URL:

Download address: https://code.visualstudio.com/updates/v1\_85

#### Figure 6-122 VS Code download URL



Update 1.85.1: The update addresses these issues.

Update 1.85.2: The update addresses these issues.

Downloads: Windows: x64 Arm64 | Mac: Universal Intel silicon | Linux: deb rpm tarball Arm snap

#### VS Code version requirements:

You are advised to use VS Code 1.85.2 or the latest version for remote connection.

#### VS Code installation guide:

In Linux, run the command **sudo dpkg -i code\_1.85.2-1705561292\_amd64.deb** to install VS Code.

**NOTE** 

Linux system users must install VS Code as a non-root user.

# 6.5.3.3 Connecting to a Notebook Instance Through VS Code with One Click

# Prerequisites

- The notebook instance with remote SSH enabled is running. For details, see **Creating a Notebook Instance**.
- You have downloaded the key file of the instance to a following local directory or its subdirectory based on your operating system:
   Windows: C:\Users\{{user}}
   Mac or Linux: Users/{{user}}

# Procedure

- Step 1 Log in to the ModelArts management console. In the left navigation pane, choose DevEnviron > Notebook.
- Step 2 The created notebook instance is running. Access a VS Code connection in either of the following ways: Click More in the Operation column and choose Access VS Code from the drop-down list. Alternatively, click Open in the Operation column. On the Launcher tab of the JupyterLab page, click VS Code. The Access VS Code dialog box is displayed.

🖾 Laund



Figure 6-123 Accessing VS Code on the management console

Figure 6-124 Accessing VS Code on the launcher page

static					
Notebo	ok				
D	D				
PyTorch-1.8	python-3.7.10				
Canaal					
Z- Console	,				
Р	Р				
PyTorch-1.8	python-3.7.10				
\$_ Other					
6			=	Μ	2,
ې Terminal	VS Code	Tensorboard	Text File	Markdown File	Python File

Step 3 If you have installed VS Code, click Access VS Code. The Visual Studio Code page is displayed.

Figure 6-125 Opening Visual Studio Code



If VS Code has not been installed, click Windows or other OS as required to download and install VS Code. For details about how to install VS Code, see Installing VS Code.



Figure 6-126 Downloading and Installing VS Code

**Step 4** If the ModelArts VS Code plug-in has not been installed, click **Install and Open**. If you have installed the plug-in, perform **5**.

Figure 6-127 Installing the VS Code plug-in

<b>X</b>	File Edit Selection	View Go	Run Terminal	Help	Get Started - Visual Studio C	ode		0	🗖 🔲   08	-	×
Ð	🔀 Get Started 🗙										
	Vi	isual S	tudio	Code							
æ	Ed	liting ev	olve <mark>v</mark> isual S	tudio Code			×				
				Extension 'ModelA like to install the e	Arts-HuaweiCloud' is r extension and open th	not installed. Wou is URL?	ild you				
				ModelArts-HuaweiCloud URL:	d (huaweicloud-ei.modelarts	vscode-toolkit) wants	to open a	ns to make VS Code	e yours.		
			itory	://huaweicloud-ei.mod	lelarts-vscode-toolkit/devco	ntainer/remote?id%3D	f9937afa				
						Install and Open	Cancel	et an overview of th	ne must-have		
	Rec										
			D:\Code\extensi	on\vscode-toolkit							
			ode\vscode			😁 Boost your Proc	ductivity				

The installation takes about 1 to 2 minutes. After the installation is complete, a dialog box is displayed in the lower right corner. Then, click **Reload Window and Open**.

#### **NOTE**

This section uses VS Code 1.78.2 as an example. The **Reload Window and Open** dialog box may not be displayed when you install other versions of VS Code. In this case, perform **5**.

<b>N</b>	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>R</u> un <u>I</u>	erminal <u>H</u> elp	Getting Started - Visual Studio Code - E	
Ch		🔀 Getting Started 🗙	⊳	
	〜 OPEN EDITORS 🛛 🖧 🗗 🗐			
Q	X Getting Started			
go	You have not yet opened a folder.	Start	Getting Started	
	Open Folder	Dpen File	🖇 Get Started with VS Code	
æ>	You can clone a repository locally.	<ul> <li>Open Folder</li> <li>Clone Git Reposito</li> </ul>	PLearn the Fundamentals	
<u>_</u> ⊘	Clone Repository		Roost your Productivity	
82	To learn more about how to use git and source control in VS Code read our docs.	Recent		
			Code\vscode_toolkits\op	
			n D:\Code\vscode_tool	
M		.ssh C:\Users\d006044 d00604475 C:\Users	<ul> <li>Would you like to reload the window and open the URL 'vscode://huaweicloud-ei.modelarts-vscode-</li> </ul>	
A			toolkit/devcontainer/remote?id%3Df9937afa-07e6-48e7-9314- 5a68829e6262%26host%3Dnotebook-	
			dc%26accessUrl%3Dssh%3A%2F%2Fma-user%40dev-modelart	
8			cnnorth7.ulanqab.huawei.com%3A30830%26keypairName%3D BKeyPaird00604475%5D'?	%5
£63-	AUTURE		Reload Window and	d Open
× (	So A o			ब्र <b>ा</b>
	00000			~ +

Figure 6-128 Reload Window and Open

In the displayed dialog box, select **Don't ask again for this extension** and click **Open**.

×	File Editt Selection View Go Run Terminal Help	Ger Stanled - Visual Studio Code	∎∎∎ ® - a ×
Ð			
	~ NO FOLDER OPENED		
~	You have not yet opened a folder.		
P	Open Folder		
₽	Opening a folder will close all currently open editors. To keep them open, add a folder instead.		
<u>н</u> 9	You can done a repository locally.	Visual Studio Code	
Ë	Clane Repository	Editing evolved	
40	To learn more about how to use git and source control in VS Code read our docs.		
М			
		Alow an extension to open this URI?     Discover the text outcombations to make VS Code yours.	
		Medelate-FuscalCioud (susceinted-stradelate-social-toolid) watte to open a	
		URI: France and the multi-base for the multi-base for the multi-base for them.	
		Disert and the disert of the d	
		🔁 Don't ask again for this extension. Doet Cancel 💀 Boost your Productivity	
		Viciose extension UV collegates consistent - bound	
8			
@		🗸 Strawalizme page en startup	
	TWELNE		

**Step 5** Remotely connect to a notebook instance.

• Before the remote connection is executed, the system automatically searches for the key file. If the key is found, a new window will be displayed and the system connects to the instance. In this case, you do not need to select the key.

	,					
				548] 🔷 🔲 🖾 🗍 🛛 🖉 –		×
Ω		• < Get Started 🗙				
	V WORK [SSH: MODELARTS-NOTEBOOK-3548]			Walkthroughs		
	> .vscode-server	Dittart				
		Cpen File.		Get Started with VS Code Discover the best customizations to make VS		
				Code yours.		
₫>						
A <mark>R</mark>				Learn the Fundamentals lump right into VS Code and get an oveniew		
		Recent		of the must-have features.		
			on D:\Code\vscode_toolkits\isAdaptLi ielArts-Note-OPGcd)			
				😥 Boost your Productivity		
			DEBUG CONSOLE TERMINAL PORTS	∑ bash + ~ □	î ^	×
		<u> </u>	□   \ -/     //\			
		Using user ma-user				
		Tips:				
		<ol> <li>avigate to the t</li> <li>Copy (Ctrl+C) and</li> </ol>	d paste (Ctrl+V) on the jupyter	r terminal.		
		3) Store your data 3 O (PyTorch-1.8) [ma-us	in /home/ma-user/work, to which ser work]\$[]	h a persistent volume is mounted.		
	> TIMELINE					
≫ SSF	: ModelArts-notebook-3548 🛞 0 🛆 0 👷 0				R	¢.
				🕕 [KeyPair-1a97] under 😫		k
				C:\Users\c \Downloads\KeyPair-1a97.pem has		
ଞ				been used for remote connection.		
		✓ Show	welcome page on startup	Source: ModelArts-HuaweiCloud (Extension)		
* (	0000					¢

Figure 6-129 Remotely connecting to a notebook instance

• If the key file is not found, a dialog box is displayed. Select the correct key as prompted.

#### **NOTE**

The key file name cannot contain Chinese characters.

#### Figure 6-130 Selecting a key file



• If an incorrect key is selected, a message will be displayed. Then, select the correct key as prompted.





When the information shown in the following figure is displayed, the instance is accessed.

Figure 6-132 Connection successful



The following error message indicates that accessing the instance failed. In this case, close the dialog box and view the output logs in the **OUTPUT** window. Then, check the **FAQs** and locate the cause.

# Figure 6-133 Connection failed



----End

# 6.5.3.4 Connecting to a Notebook Instance Through VS Code Toolkit

This section describes how to use the ModelArts VS Code Toolkit plug-in to remotely connect to a notebook instance.

# Prerequisites

You have downloaded and installed VS Code. For details, see Installing VS Code.

# Step 1 Install the VS Code Plug-in

1. Search for ModelArts in the EXTENSIONS text box and click Install.



Figure 6-134 Installing the VS Code plug-in

2. Wait for about 1 to 2 minutes.

#### Figure 6-135 Installation process



3. After the installation is complete, check the message displayed in the lower

right corner. If the ModelArts icon and remote SSH icon are displayed in the navigation pane on the left, the VS Code plug-in is installed.

#### Figure 6-136 Installation completion message

Completed installing ModelArts extension from VSIX.

#### Figure 6-137 Installation completed



Network issues may cause an installation failure. If this occurs, proceed with follow-up operations. After 1 in **Step 5 Access the Notebook Instance** is performed, the system will automatically display a dialog box shown in the following figure. In this case, click **Install and Reload**.

#### Figure 6-138 Reconnecting remote SSH



# Step 2 Add More Regions

1. Import the configuration file in the VS Code plug-in.

Contact the region operations company to obtain the YAML configuration file

and host information. Open the VS Code plug-in, click , choose **Import Region Profile**, click **From local file** in the lower right corner, enter the path to the local YAML file, and press **Enter**.

2. Log in to the VS Code plug-in to use more functions.

After the configuration file is imported, the region changes to your region. Enter the account name and AK/SK to log in to the plug-in.



# Step 3 Log In to the VS Code Plug-in

1. In the local VS Code development environment, click and **User Settings**, and configure the login information.

Figure 6-139 Logging in to the plug-
--------------------------------------

<b>ModelArts Login</b>	
Account Name	
AccessKey Id	<u>Get Your AK/SK</u>
SecretAccess Key	
· · · · · · · · · · · · · · · · · · ·	Get More Region 0
Log in	

Enter the login information and click Log in.

- Name: Custom username, which is displayed only on the VS Code page and is not associated with any account.
- AK and SK: Access key pair. To create a key pair, choose My Credentials > API Credentials > Access Keys, and click Create Access Key.
- Region: must be the same as that of the notebook instance to be remotely connected. Otherwise, the connection will fail.
- 2. After the login, check the notebook instance list.

#### Figure 6-140 Login succeeded



# Step 4 Create a Notebook Instance

#### 

• Create a notebook instance with remote SSH enabled, and download the key file to either of the following directories based on your OS:

Windows: C:\Users\{{user}}

macOS or Linux: Users/{{user}}

• A key pair is automatically downloaded after you create it. Securely store your key pair. If an existing key pair is lost, create a new one.

Create a notebook instance with remote SSH enabled. For details, see **Creating a Notebook Instance**.

# Step 5 Access the Notebook Instance

1. In the local VS Code development environment, right-click the instance name and choose **Connect to Instance** from the shortcut menu to start and connect to the notebook instance.

The notebook instance can either be running or stopped. If it is stopped, the VS Code plug-in starts the instance and then connects to it.



#### Figure 6-141 Connecting to a notebook instance

Alternatively, click the instance name. On the instance details page, click **Connect**. Then, the system automatically starts and connects to the notebook instance.

Figure 6-142 Viewing details about a notebook instance

r of instances	M @ A C							
tebook-324d (RUNNING)								
ag		╵╧╯╵	otebook-324d		Connect	Start	Stop	Refresh
99	imageTest_ku97m3	(STOPPED)						
ag te			notebook-324d					
ag te			RUNNING					
te								
te te								

2. When you connect to a notebook instance for the first time, the system prompts you in the lower right corner to configure the key file. In this case, select the local .pem key file and click **OK**.

Figure 6-143 Configuring the key file



3. Wait for about 1 to 2 minutes until the notebook instance is accessed. After information similar to the following is displayed in the lower left corner of the VS Code environment, the connection is succeeded.

Figure 6-144 Connection succeeded



# **Related Operations**

For details about uninstalling the VS Code plug-in, see Figure 6-145.



Figure 6-145 Uninstalling the VS Code plug-in

# 6.5.3.5 Manually Connecting to a Notebook Instance Through VS Code

A local IDE supports PyCharm and VS Code. You can use PyCharm or VS Code to remotely connect the local IDE to the target notebook instance on ModelArts for running and debugging code.

This section describes how to use VS Code to access a notebook instance.

# Prerequisites

- You have downloaded and installed VS Code. For details, see Installing VS Code.
- Python has been installed on your local PC or server. For details, see VS Code official documentation.
- A notebook instance has been created with remote SSH enabled. Ensure that the instance is running. For details, see **Creating a Notebook Instance**.
- The address and port number of the development environment are available. To obtain the information, go to the notebook instance details page.

#### Figure 6-146 Instance details page



• The key pair is available.

A key pair is automatically downloaded after you create it. Securely store your key pair. If an existing key pair is lost, create a new one.

# Step 1 Add the Remote-SSH Plug-in

In the local VS Code development environment, click **I**, enter **SSH** in the search box, and click **install** of the Remote-SSH plug-in to install the plug-in.

Figure 6-147 Adding the Remote-SSH plug-in

Û	EXTENSIONS: MARKETP 🍸 📰 …
~	SSH
γ	Remote - SSH 0.56.0
ego	Open any folder on a remote machine Microsoft
₽ D	SSH FS 1.18.3 File system provider using SSH Kelvin Schoofs Install
۵	Remote - SSH: Editing Config 0.56.0 Edit SSH configuration files Microsoft
₿	Remote - SSH (Nightly) 2020.11.24164 Open any folder on a remote machine Microsoft

# Step 2 Configure SSH

1. In the local VS Code development environment, click 6 on the left, select **SSH Targets** from the drop-down list box, and click 6. The SSH configuration file path is displayed.

Figure 6-148 Configuring SSH Targets



2. Click the SSH configuration path and configure SSH.

#### Figure 6-149 SSH configuration file path



#### HOST remote-dev

hostname <*Instance connection host*> port <*Instance connection port*> user **ma-user** IdentityFile **~/.ssh/test.pem** UserKnownHostsFile=/dev/null StrictHostKeyChecking no

- **HOST**: name of the cloud development environment
- HostName: address for accessing the cloud development environment. Obtain the address on the page providing detailed information of the target notebook instance.
- Port: port number for accessing the cloud development environment.
   Obtain the port number on the page providing detailed information of the target notebook instance.
- user: ma-user
- **IdentityFile**: locally stored private key file of the cloud development environment. It is the key pair file in **Prerequisites**.
- 3. Choose File > Preference > Settings > Extensions > Remote-SSH. On the Remote Platform page, click Add Item, set Item and Value, and click OK.

#### Figure 6-150 Configuring Remote Platform

HTML	common install location	ins.						
Jake								
JavaScript Debugger								
JSON								
Jupyter 😫	Remote.SSH: Remote	Platform						
LESS	A map of the remote h	A map of the remote hostname to the platform for that remote. Valid values:						
Markdown	usel ocalServer is disal	s. Note - this setting will bled so it is currently bei	soon be required w	nen or successful				
Merge Conflict	connections, but is no	t currently used.	ng aaropopalatea ti					
Node debug	lá a m	Value						
Npm	item	Value						
РНР	remote-dev	linux						
Python	test	linux	~ ОК	Cancel				
Reference Search V								
Remote - Containers	Demote COL Demote							
Remote - SSH	Remote.SSH: Remote	Server Listen On Socket						

**Item**: host name configured in SSH configuration **Value**: remote development environment platform 4. Go back to the **SSH Targets** page and click **b** on the right. Then, click the development environment name to open the development environment.

Figure 6-151 Opening the development environment



After the page shown in the following figure is displayed, the connection is succeeded.







#### Figure 6-153 Complete configuration example

# Step 3 Install the Python Plug-in in the Cloud Development Environment



On the displayed VS Code page, click **D** on the left, enter **Python** in the search box, and click **Install**.

#### Figure 6-154 Installing the Python plug-in in the cloud development environment



If the Python plug-in fails to be installed on the cloud, install it using an offline package.

# Step 4 Install the Dependent Library for the Cloud Environment

After accessing the container environment, you can use different virtual environments, such as TensorFlow and PyTorch. However, in actual development, you need to install dependency packages. Then, you can access the environment through the terminal to perform operations.

1. In VS Code, press Ctrl+Shift+P.

- 2. Search for **Python: Select Interpreter** and select the target Python.
- 3. Choose **Terminal > New Terminal**. The CLI of the remote container is displayed.
- 4. Run the following command to install the dependency package: pip install spacy

# 6.5.3.6 Remotely Debugging in VS Code

# Prerequisites

A notebook instance has been accessed through VS Code.

# **Step 1 Upload Local Code to the Cloud Development Environment**

1. On the VS Code page, choose **File** > **Open Folder** to access the cloud path.

Figure 6-155 Open Folder

Ð	EXPLORER ····	
. <mark>Ш</mark> ,	$\sim$ NO FOLDER OPENED	
Q	Connected to remote.	
2º	Open Folder	
å	You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.	
Ē	Clone Repository	

2. Select a path and click **OK**.

#### Figure 6-156 Selecting a file path

Open Folder		
/home/ma-user/	ОК	Show Local
.astropy		
.cache		
.conda		
.config		
.jupyter		
.local		
.modelarts		

3. In the displayed directory structure on the left of the IDE, drag the code and files you want to upload to the corresponding folders. Then, the code is uploaded to the cloud development environment.

# Step 2 Debug Code Remotely

Open the code file to be debugged in VS Code. Before running the code, click the default Python version in the lower left part and select a version as required.

	EVOLODED			
L,	LAFLOREN	Current: ./anaconda3/envs/iensor+iow-1.8/bin/python		
	✓ MA-USER [SSH: REMOTE-DI	Enter interpreter path		
Q	> .ssh	Enter path or find an existing interpreter		
	> .vscode	Python 2.7.12 64-bit		
20	> .vscode-server	/usr/bin/python		
5	> .yarn	Python 3.5.2 64-bit		
	> anaconda3	/usr/bin/python3		
$\geq$	> env_script	Python 3.6.12 64-bit (conda)		
	> log	./anaconda3/envs/R-3.6.1/bin/python		
7	> modelarts-sdk	Python 3.6.2 64-bit (conda)		
:09	> notebook-exts	./anaconda3/envs/PySpark-2.3.2/bin/python		
	> notebook-samples	Python 3.6.2 64-bit (conda)		
Б	> notebook-samples-16	/anaconda3/envs/XGBoost-Sklearn/bin/python		
	∨ test_456064			
	> PythonPractise			
	① README.md			
	🔹 test.py			
	> work			
	.bash_history			
	🔲 .bash_logout			
	.bashrc			
	🚥 .npmrc	PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2:	Python	~ +
	📑 .profile			
	■ .python_history	(base) sn-4.3\$/nome/ma-user/anaconda3/envs/lensorFiow-	1.8/bin/python /	nome/ma-u
	≣ .viminfo	1		
	.varnrc	2		
	≣ core.9193	3 (hana) ah 4 200		
	≣ core.9556	(base) sn-4.33		
2	S core.9751			
	≡ core.9929			
22	≡ core.10103			
5				
SSF	I: remote-dev Python 3.6.4 64	-bit (conda) 🛞 0 🛆 0 👾 No Ports Available 🤣	Ln 4, Col 1 Spaces:	4 UTF-8

Figure 6-157 Selecting a Python version

- Click the execution button to run the code. The code output is shown on the **TERMINAL** tab page.
- If a training job takes a long time to execute, run the job at the backend through the nohup command. This prevents the disconnection of an SSH session or a network failure from affecting job execution. The following shows an example nohup command: nohup your\_train\_job.sh > output.log 2>&1 & tail -f output.log
- To debug the code, perform the following operations:
  - a. Choose **Run > Run and Debug** on the left.
  - b. Select the default Python code file.
  - c. Click on the left of the code to set breakpoints.
  - d. Debug the code according to the debug procedure which is displayed above the code, and the debug information is displayed on the left of the page.

# 6.5.3.7 Uploading and Downloading Files in VS Code

# Uploading Data from a Local IDE to a Notebook Instance

If the data is less than or equal to 500 MB, directly copy the data to the local IDE.

If the data is larger than 500 MB, upload it to OBS and then to the notebook instance.





#### Procedure

1. Upload data to OBS. Alternatively, use ModelArts SDK on a local VS Code terminal.

Open the terminal in the local VS Code environment.



Figure 6-159 Opening the terminal in the local VS Code environment

Enter **python** and press **Enter** to access the Python environment.

In the terminal of the local VS Code, use ModelArts SDK to upload the target local file to OBS. For details, "OBS Management" > "Transferring Files (Recommended)" in *SDK Reference*.

2. Use ModelArts SDK in the terminal of the remote VS Code environment to download the file from OBS to a development environment.



Figure 6-160 Opening the terminal in the remote VS Code environment

 # Manually access the development environment using the source command. cat /home/ma-user/README
 # Select the target environment. source /home/ma-user/miniconda3/bin/activate MindSpore-python3.7-aarch64 # Enter **python** and press **Enter** to access the Python environment. python

Then, perform OBS transfer operations by referring to "OBS Management" > "Transferring Files (Recommended)" in *SDK Reference*.

# Downloading Files from a Notebook Instance to a Local Directory

Files created in Notebook can be downloaded to a local path. In the **Project** directory of the local IDE, right-click the **Notebook2.0** project and choose **Download** from the shortcut menu to download the project file to the local PC.

**Figure 6-161** Downloading files from a notebook instance to a local directory in VS Code



# 6.5.4 Local IDE (Accessed Using SSH)

This section describes how to use PuTTY to remotely log in to a notebook instance on the cloud in the Windows environment.

# **Prerequisites**

- You have created a notebook instance with remote SSH enabled and whitelist configured. Ensure that the instance is running. For details, see **Creating a Notebook Instance**.
- The address and port number of the development environment are available. To obtain this information, go to the notebook instance details page.

#### Figure 6-162 Instance details page

Address	ssh://ma-user@ <mark>dev-modelarts-</mark>	.com:32651
	Access address of the	Port
Authentication	KeyPair-9a64 🖉 development environment	number

• The key pair is available.

A key pair is automatically downloaded after you create it. Securely store your key pair. If an existing key pair is lost, create a new one.

# Step 1 Install the SSH Tool

Download and install the SSH remote access tool, for example, PuTTY.

# Step 2 Use PuTTYgen to Convert the .pem Key Pair File to a .ppk Key Pair File

- 1. Download PuTTYgen and double-click it to run it.
- 2. Click **Load** to load the .pem key file created and saved during notebook instance creation.
- 3. Click **Save private key** to save the generated .ppk file. The file name can be customized, for example, **key.ppk**.

Figure 6-163 Converting the .pem key pair file to a .ppk key pair file

PutTY Key Generator ? ×   e Key Conversions Help     Key   Public key for pasting into OpenSSH authorized_keys file:     ssh-rsa   Actives file:   ssh-rsa     Ssh-rsa     Actives file:     Ssh-rsa        Ssh-rsa								
PutTY Key Generator       ?         File       Key         Public key for pasting into OpenSSH authorized_keys file:								
Key								
Public key for pastin	ig into Open SSH aut	horized_keys file:						
ssh-rsa ନ୍ଦ୍ଧାନ କ୍ରିଡିହାନ୍ତ କ୍ରିମ୍ବାନ କ୍ରି ଜାନିକେଙ୍କ ଅଭିଜନିକାଳ	EVARIAN SARAA MARAAN SARAA MARAAN SARAAN	le Ber The Hangerig Ruffich Bergerig Muffich Bergerig and Angel	neoriseanstainstain The M	<b>3&amp;\$</b> /5#6253&37584748	f			
X2NEEDAE	ROMAN CONTRACTOR	CARLES AND	SECHARTONIA FONDATO XI		¥			
Key fingerprint:	ssh-rsa 2048 SHA256:Rady #2309 สีหลุก R.#2605/Wape-JRBn 24/305/ #20/54/sageU							
Key comment:	Key comment: imported-openssh-key							
Key passphrase:								
Confirm passphrase	:							
Actions								
Generate a public/p	rivate key pair			Generate				
Load an existing priv	vate key file			Load				
Save the generated	key		Save public key	Save private key				
Parameters								
Type of key to gene RSA	rate: ODSA	◯ ECDSA	CEdDSA	O SSH-1 (RSA)				
Number of hits in a (	enerated kev:			2048				

# Step 3 Use SSH to Connect to a Notebook Instance

- 1. Run PuTTY.
- 2. Click **Session** and set the following parameters:
  - a. **Host Name (or IP address)**: address for accessing the in-cloud notebook instance. Obtain the address on the page providing detailed information of the target notebook instance .

I

- b. **Port**: port number for accessing the in-cloud notebook instance. Obtain the port number on the page providing detailed information of the target notebook instance, for example, **32701**.
- c. Connection type: SSH

- -

d. **Saved Sessions**: task name, which can be clicked for remote access when you use PuTTY next time

#### Figure 6-164 Configuring Session

🔀 PuTTY Configuration		?	×
PuTTY Configuration         Category:         Session        Logging        Logging	Basic options for your PuTTY se Specify the destination you want to conne Host Name (or IP address) b dev-modelarts .co Connection type: SSH Serial Other: Telne Load, save or delete a stored session Saved Sessions Saved Sessions Saved Sessions Saved Sessions Saved Sessions	? ession ect to Port r 32701 et Load Save	×
Connection     Data     Proxy     SSH     Serial     Telnet     Rlogin     SUPDUP	Close window on exit:	Load Save Delet	e ie
About Help	Always     Never     Only on c     Open	clean exit Cance	el

3. Choose **Window** > **Translation** and select **UTF-8** from the drop-down list box in the **Remote character set** area.

🕵 PuTTY Configuration	? ×
Category:	
<ul> <li>Session</li> <li>Logging</li> <li>Terminal</li> <li>Keyboard</li> <li>Bell</li> <li>Features</li> <li>Window</li> <li>Appearance</li> <li>Behaviour</li> <li>Translation</li> <li>Selection</li> <li>Colours</li> <li>Connection</li> <li>Data</li> <li>Proxy</li> <li>SSH</li> <li>Serial</li> <li>Telnet</li> <li>Rlogin</li> <li>SUPDUP</li> </ul>	Options controlling character set translation         Character set translation         Remote character set:         UTF-8         (Codepages supported by Windows but not listed here, such as CP866 on many systems, can be entered manually)         Treat CJK ambiguous characters as wide         Caps Lock acts as Cyrillic switch         Adjust how PuTTY handles line drawing characters         Handling of line drawing characters:         Use Unicode line drawing code points         Poor man's line drawing (+, - and I)         Font has XWindows encoding         Use font in both ANSI and OEM modes         Use font in OEM mode only         Copy and paste line drawing characters as lqqqk         Enable VT100 line drawing even in UTF-8 mode
About Help	Open Cancel

Figure 6-165 Setting the character format

4. Choose **Connection** > **Data** and enter **ma-user** for **Auto-login username**.

🕵 PuTTY Configuration			?	$\times$	
Category:					
<ul> <li>Session         <ul> <li>Logging</li> <li>Terminal</li> <li>Keyboard</li> <li>Bell</li> <li>Features</li> </ul> </li> <li>Window         <ul> <li>Appearance</li> <li>Behaviour</li> <li>Translation</li> <li>Selection</li> <li>Colours</li> </ul> </li> <li>Connection         <ul> <li>Data</li> <li>Proxy</li> <li>SSH</li> <li>Serial</li> <li>Telnet</li> <li>Rlogin</li> </ul> </li> </ul>	Data to send to the server Login details				
	Auto-login usemame When usemame is not spec	ma-user			
	Prompt OUse system usemame (000443458)				
	Terminal-type string xterm				
	Environment variables Variable Add Value				
				move	
About Help		Open	Can	cel	

# Figure 6-166 Entering a username

5. Choose **Connection** > **SSH** > **Auth**, click **Browse**, and select the .ppk file generated in **step 2**.



6. Click **Open**. If you are logging in to the instance for the first time, PuTTY displays a security warning dialog box, asking if you want to accept the instance security certificate. Click **Accept** to save the certificate to your local registry.

Figure 6-167 Asking if you want to accept the instance security certificate



7. Connect to the notebook instance.

#### Figure 6-168 Connecting to a notebook instance



# 6.6 Using Notebook to Develop Ascend Operators

#### Overview

In training and inference scenarios, you need to develop your own operator if the original operator is not supported when a third-party framework is used. During debugging, if the performance of some operators combined is low, you need to develop high-performance operators. You can use VS Code to connect to notebooks on the cloud and use cloud resources to develop and debug operators on VS Code. The environment has been configured in the notebook instance. You can perform project-based development without installing the CANN software or configuring environment variables.

#### **NOTE**

This document provides a set of operator project sample code, which you can directly use. For details about the programming model of the Ascend operator, see **Quick Start**. The environment has been configured in notebook. You do not need to install the CANN software or configure environment variables. You can directly perform operator analysis and subsequent operations in the remote VS Code environment.

# Prerequisites

- You have downloaded the operator sample from Ascend samples and uploaded it to the OBS bucket.
- You have created a notebook instance based on the mindspore\_2.2.0cann\_7.0.1-py\_3.9-euler\_2.10.7-aarch64-snt9b engine and enabled remote SSH development. The notebook instance must be in the **Running** state. For details, see **Creating a Notebook Instance**.

#### **NOTE**

This document uses the mindspore\_2.2.0-cann\_7.0.1-py\_3.9-euler\_2.10.7-aarch64-snt9b engine as an example to describe how to debug operators. If other AI engines are used, an error may be reported.

• Open JupyterLab and click to upload the sample files in the OBS bucket to notebook. For details, see **Uploading OBS Files to JupyterLab**.


# Connecting to a Notebook Instance on the Cloud Through VS Code

- After a notebook is created and is in the Running state, locate it in the list, and click More > Access VS Code in the Operation column. For details about how to connect to a cloud development environment, see Connecting to a Notebook Instance Through VS Code with One Click.
- 2. After the cloud development environment is connected, projects downloaded from the cloud are displayed on the VS Code page, as shown in Figure 6-169.

Figure 6-169 Project directory



# Debugging the Add Operator in VS Code

- 1. In Terminal, run the following command to access the directory where the Add operator is stored: cd samples/cplusplus/level1\_single\_api/4\_op\_dev/6\_ascendc\_custom\_op/kernel\_invocation/Add
- 2. Run the following commands to compile and run the script.
  - a. Run the following command in CPU mode: bash run.sh add\_custom ascend910B1 VectorCore cpu

**add\_custom** indicates the operator to be run, **ascend910B1** indicates the AI processor model, **VectorCore** indicates that the operator runs on VectorCore, and **cpu** indicates that the operator runs in CPU mode.

The output is shown in the following figure. In this example, md5sum is used to compare all output bin files. If the values of **md5** are the same, the actual output data is in consistent with the true data.

PROBLEMS	OUTPUT	TERMINAL	PORTS	DEBUG CONSOLE		🔲 bash - Add	I Ш~	Ξ	Û		<u>^</u>
/home/ma	-user/wor	k/samples/	cplusplu	us/level1_single_ap	oi/4_0	op_dev/6_as	cendc	_cus	tom_	_op/I	kerne
1_invocat	tion/Add										
INFO: con	mpile op	on cpu suc	ceed!								
[INFO] BU	JSIN_VER	<ul> <li>get_conf.</li> </ul>	igfile -	<pre>- config_file_name</pre>	: As	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	<ul> <li>get_conf.</li> </ul>	igfile -	<pre>config_file_name</pre>	: As	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	<ul> <li>get_confi</li> </ul>	igfile -	<pre>config_file_name</pre>	: As	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	<ul> <li>get_conf</li> </ul>	igfile -	<pre>config_file_name</pre>	: As	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	<ul> <li>get_conf.</li> </ul>	igfile -	<pre>config_file_name</pre>	: Aso	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	<ul> <li>get_conf.</li> </ul>	igfile -	<pre>config_file_name</pre>	: As	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	- get_conf	igfile -	<pre>config_file_name</pre>	: As	cend910C_pv	.toml				
[INFO] BU	JSIN_VER	<ul> <li>get_conf.</li> </ul>	igfile -	<pre>config_file_name</pre>	: As	cend910C_pv	.toml				
pid 4502	5 exits s	tatus 0!									
pid 4502	5 exits s	tatus 0!									
pid 4502	7 exits s	tatus 0!									
pid 4502	8 exits s	tatus 0!									
pid 45029	exits s	tatus 0!									
pid 45030	exits s	tatus 0!									
pid 4503	l exits s	tatus 0!									
pid 45033	2 exits s	tatus 0!									
INFO: exc	ecute op (	on cpu suc	ceed!								
md5sum:											
cbb7ee04	85ca67263	d68eb6cbb1	9a4de d	output/golden.bin							
cbb7ee04	85ca67263	d68eb6cbb1	9a4de d	output/output_z.bin	i i						
1113-10-00	A N Frank in	A.4.1.4∏									

b. Run the following command in NPU mode: bash run.sh add\_custom ascend910B1 VectorCore npu

The output is shown in the following figure. In this example, md5sum is used to compare all output bin files. If the values of **md5** are the same, the actual output data is in consistent with the true data.

Figure 6-171	I Command	output ir	ו NPU	mode
--------------	-----------	-----------	-------	------

	PROBLEMS OUTPU	JT TERMINAL	PORTS	DEBUG CONSOLE	🔲 bash - Add	<b>□</b> ~ □	Ū ··	•• ^	×
ľ	/home/ma-user/w	work/samples/	cplusplu	s/level1_single_ap	oi/4_op_dev/6_aso	cendc_cust	com_op	o/kern	e
	1_invocation/Ad	ad							
	INFO: compile o	op on cpu suce	ceed!						
	[INFO] BUSIN_V	ER - get_conf:	igfile -	config_file_name	: Ascend910C_pv	toml			
	[INFO] BUSIN_V	ER - get_conf:	igfile -	<pre>config_file_name</pre>	: Ascend910C_pv	toml			
	[INFO] BUSIN_V	ER - get_conf:	igfile -	config_file_name	: Ascend910C_pv	toml			
	[INFO] BUSIN_V	ER - get_conf:	igfile -	config_file_name	: Ascend910C_pv.	toml			
	[INFO] BUSIN_V	ER - get_conf:	igfile -	config_file_name	: Ascend910C_pv.	toml			
	[INFO] BUSIN_V	ER - get_conf:	igfile -	<pre>config_file_name</pre>	: Ascend910C_pv.	toml			
	[INFO] BUSIN_VE	ER - get_conf:	igfile -	config_file_name	: Ascend910C_pv	toml			
	[INFO] BUSIN_V	ER - get_conf:	igfile -	config_file_name	: Ascend910C_pv	toml			
	pid 45025 exits	s status 0!							
	pid 45026 exits	s status 0!							
	pid 45027 exits	s status 0!							
	pid 45028 exits	s status 0!							
	pid 45029 exits	s status 0!							
	nid 45030 exit	s status Ø!							
	nid 45031 exit	s status Øl							
	nid 45032 exits	s status Øl							
	TNEO: execute (		Ihee						
Γ.	md5cum:	op on epu suco							
	shb7aa0495aa67		Dadda a	utnut/goldon hin					
	cob/ee0485ca6/		Jaque o	ucpuc/goiden.bin					
1	cbb/ee0485ca672	263068eb6cbb19	Ja4de o	utput/output z.bin	E				

#### Debugging the matmul Operator in VS Code

- The matmul operator is stored in the samples/cplusplus/level1\_single\_api/ 4\_op\_dev/6\_ascendc\_custom\_op/kernel\_invocation/Matmul directory.
- 2. Run the following command in the **work** directory:

cd samples/cplusplus/level1\_single\_api/4\_op\_dev/6\_ascendc\_custom\_op/kernel\_invocation/Matmul

3. Run the following command to modify the **main.cpp** file, which is the application file for calling the operator: vim main.cpp

Set param4FileSize to 192.

#### Figure 6-172 Setting param4FileSize to 192



4. Run the following command to modify the **vim matmul\_custom.cpp** file: vim matmul\_custom.cpp

Change tiling.K in matmul\_custom.cpp to tiling.Ka.

Figure 6-173 Changing tiling.Ka to tiling.Ka



5. Create a folder **output** in the **Matmul** directory.



Figure 6-174 Creating the output folder

- 6. Run the following commands to compile and run the script.
  - a. Run the following command in CPU mode: bash run.sh matmul\_custom ascend910B1 AiCore cpu ONBOARD CUSTOM\_TILING

The output is shown in the following figure. In this example, md5sum is used to compare all output bin files. If the values of **md5** are the same, the actual output data is in consistent with the true data.

#### Figure 6-175 Command output in CPU mode

PROBLEMS	OUTPUT	TERMINAL	PORTS	DEBUG CONSOLE	🔲 bash - Matr	nul [[] ~ [[	Û		^ ;
gmake[3]:	: Entering	directory	'/home	/ma-user/work/s	amples/cplusplus	/level1_si	ngle_	_api/	(4_op
_dev/6_as	scendc_cus	tom_op/kerr	nel_inv	ocation/Matmul,	/build'				
[ 50%] <mark>B</mark> l	uilding CX	X object cm	nake/ti	ling/CMakeFiles	<pre>s/matmul_custom_t</pre>	iling.dir/	1_	/cus	stom_
tiling/ma	ain.cpp.o								
[100%] Li	inking CXX	executable	e/	//matmul_cust	tom_tiling				
gmake[3]:	Leaving	directory '	/home/	ma-user/work/sa	amples/cplusplus/	level1_sin	gle_a	pi/4	_op_
dev/6_aso	endc_cust	om_op/kerne	el_invo	cation/Matmul/M	ouild'				
[100%] Bu	uilt targe	t matmul_cu	ustom_t	iling					
gmake[2]:	: Leaving	directory '	/home/	ma-user/work/sa	amples/cplusplus/	level1_sin	gle_a	ipi/4	_op_
dev/6_aso	endc_cust	om_op/kerne	el_invo	cation/Matmul/	ouild'				
gmake[1]:	Leaving	directory '	/home/	ma-user/work/sa	amples/cplusplus/	level1_sin	gle_a	pi/4	_op_
dev/6_aso	endc_cust	om_op/kerne	el_invo	cation/Matmul/N	ouild'				
/home/ma-	user/work	/samples/cp	olusplu	s/level1_single	e_api/4_op_dev/6_	ascendc_cu	stom_	op/k	<u>kerne</u>
l_invocat	tion/Matmu	<u>1</u>							
INFO: CON	npile op o	n cpu succe	ed!	c: c:a	10100				
[INFO] BU	JSIN_VER -	get_config	gtile -	config_file_na	ame : Ascend910C_	pv.tom1			
p1d 62677	exits st	atus 0!							
p10 62678	s exits st	atus 0!							
μ10 02075 TNFO1 02075	exits su		odl						
INFU: exe	ecute op o	m cpu succe	edi						
abdf5420	£400424b2	hdcabo75ch	20d -	utput/goldon b	in				
Sebur 5425	F40043403		280 0	utput/golden.b.	in .				
Sebur5425	140043403	00000875006	200 0	ucpuc/oucpuc.b.					

b. Run the following command in NPU mode:

bash run.sh matmul\_custom ascend910B1 AiCore npu ONBOARD CUSTOM\_TILING

The output is shown in the following figure. In this example, md5sum is used to compare all output bin files. If the values of **md5** are the same, the actual output data is in consistent with the true data.

Figure 6-176 Command output in NPU mode

PROBLEMS	OUTPUT	TERMINAL	PORTS	DEBUG CONSOLE	🛄 bash - Matmul	□ ~ □ 🛍 … ^ ×
dev/6 as	scendc cu	stom op/ker	nel inv	ocation/Matmul/N	build'	
Scanning	dependen	cies of tar	get mat	mul_custom_tilin	ng	
gmake[3]:	Leaving	directory	'/home/	ma-user/work/sam	mples/cplusplus/lev	/el1_single_api/4_op_
dev/6_aso	endc_cus	tom_op/kern	el_invo	cation/Matmul/bu	uild'	
gmake[3]:	Enterin	g directory	'/home	/ma-user/work/sa	amples/cplusplus/le	evel1_single_api/4_op
_dev/6_as	cendc_cu	stom_op/ker	nel_inv	ocation/Matmul/N	build'	
[ 50%] <mark>B</mark> l	ilding C	XX object c	make/ti	ling/CMakeFiles,	/matmul_custom_tili	ing.dir///custom_
tiling/ma	ain.cpp.o					
[100%] Li	inking CX	X executabl	e/	//matmul_custo	om_tiling	
gmake[3]:	Leaving	directory	'/home/	ma-user/work/sar	mples/cplusplus/lev	/el1_single_api/4_op_
dev/6_asc	endc_cus	tom_op/kern	el_invo	cation/Matmul/bu	uild'	
[100%] Bi	ilt targ	et matmul_c	ustom_t	iling		
dev/6 asc	E Leaving	directory tom op/kern	'/home/ el invo	ma-user/work/sam cation/Matmul/bu	mples/cplusplus/lev uild'	/ell_single_api/4_op_
gmake[1]:	Leaving	directory	'/home/	ma-user/work/sar	mples/cplusplus/lev	/el1 single api/4 op
dev/6_asc	endc_cus	tom_op/kern	el_invo	cation/Matmul/bu	uild'	
/home/ma-	user/wor	k/samples/c	plusplu	s/level1_single_	_api/4_op_dev/6_asc	endc_custom_op/kerne
l_invocat	ion/Matm	ul				
INFO: con	npile op	on npu succ	eed!			
INFO: exe	ecute op	on npu succ	eed!			
md5sum:						
161404d0d	:7635c7a5	9e0cb937e68	fe7d o	output/golden.bi	า	
161404d0d	:7635c7a5	9e0cb937e68	fe7d o	utput/output.bi	า	

### **Generating a Profile**

After NPU debugging, the **matmul\_custom\_npu** executable file is generated in the project directory. Run the following command to generate a profile:

msprof --application="matmul\_custom\_npu" --output="./output"

Figure 6-177 Generating a profile

[INFO] Export al [INFO] Start que	ll data in PROF_0 ery data in PROF_	000001_2024012610 _000001_202401262	03713957_0B 03713957_08	BJCJNMKIJGEACC done. BBJCJNMKIJGEACC.		
Job Info	Device ID	Dir Name	Collection	Time	Model	IDI
teration Number	Top Time Iterat:	ion Rank ID				
NA	0	device_0	2024-01-26	10:37:13.959103	N/A	Ν
/A	N/A	0				
NA		host	2024-01-26	10:37:13.959103	N/A	Ν
/A	N/A	0				
<pre>[INFO] Query all data in PROF_000001_20240126103713957_OBBJCJNMKIJGEACC done. [INFO] Profiling finished. [INFO] Process profiling data complete. Data is saved in /home/ma-user/work/samples/cplus plus/level1_single_api/4_op_dev/6_ascendc_custom_op/kernel_invocation/Matmul/output/PROF_ 000001_20240126103713957_OBBJCJNMKIJGEACC</pre>						

#### Backing Up Files Before Stopping a Notebook Instance

After a notebook instance is stopped, the corresponding container environment is deleted. Only the content in the **/home/ma-user/work** directory is persistently stored. Modifications in other directories are lost.

#### Backup method

Copy the files to the **/home/ma-user/work** directory before stopping a notebook instance.

The files to be copied are as follows:

- 1. Self-built projects in the /home/ma-user/ AscendProjects directory
- 2. OM file, configuration file, and evaluation report in the **/home/ma-user/ modelzoo/** directory after model conversion
- 3. SSH configuration in the /home/ma-user/.mindstudio directory
- 4. Other modified content

After a notebook instance is restarted, copy the above files to the original directory so that the instance can run properly.

# 6.7 ModelArts CLI Command Reference

# 6.7.1 ModelArts CLI Overview

#### Description

ModelArts CLI, also called ma-cli, is a cross-platform command line tool used to connect to ModelArts and run management commands on ModelArts resources. You can use the interactive command prompt or script to run commands on a terminal. ma-cli allows you to interact with cloud services through ModelArts notebook and on-premises VMs. You can run ma-cli commands for command autocomplete and authentication, as well as creating images, submitting ModelArts training jobs and DLI Spark jobs, and copying OBS data.

### **Application Scenarios**

- ma-cli has been integrated into ModelArts notebook and can be directly used.
   Log in to the ModelArts console, choose **DevEnviron** > **Notebook**, create a notebook instance, start a terminal, and run ma-cli commands.
- In local Windows or Linux, install ma-cli and then use it on a local terminal. For details, see (Optional) Installing ma-cli Locally.

- ma-cli cannot be used in Git Bash.
- Terminals such as Linux Bash, Zsh, Fish, WSL, and PowerShell are recommended. To ensure the security of your sensitive information, it is important to prevent any potential leakage when using terminals.

#### **Command Preview**

```
$ ma-cli -h
Usage: ma-cli [OPTIONS] COMMAND [ARGS]...
Options:
 -V, -v, --version
                     1.2.1
 -C, --config-file TEXT Configure a file path for authorization.
                     Debugging mode, in which the full stack trace will be displayed when an error occurs.
 -D, --debug
 -P, --profile TEXT
-h, -H, --help
                      CLI connection profile to be used. The default profile is DEFAULT.
                     Show the help information and exit.
Commands:
 configure
                Configure authentication and endpoints for the CLI.
 image
               Obtain registered images, register or unregister images, debug images, and create images in
Notebook.
 obs-copy
                Copy files or directories between OBS and a local path.
               Submit ModelArts jobs and obtain jod details.
 ma-job
              Submit DLI spark jobs and obtain jod details.
 dli-iob
 auto-completion Auto complete ma-cli command in terminal, support "bash(default)/zsh/fish".
```

Among the preceding parameters, parameters **-C**, **-D**, **-P**, and **-h** are globally optional.

- -C indicates that you can manually specify the authentication configuration file when running this command. By default, the ~/.modelarts/ma-cliprofile.yaml configuration file is used.
- -P indicates a group of authentication information in the authentication file. The default value is **DEFAULT**.
- **-D** indicates whether to enable the debugging mode (disabled by default). After the debugging mode is enabled, the error stack information of the command will be printed. If this mode is disabled, only the error information will be printed.
- -h indicates that the help information about the command will be displayed.

# Commands

Table 6-10 ma-cli commands

Command	Description
configure	ma-cli authentication using a username and password or an SK/SK
image	ModelArts image creation, registration, and registered image query
obs-copy	Copying files or folders between a local path and OBS
ma-job	Managing ModelArts training jobs, including job submission and resource query
dli-job	DLI Spark job submission and resource management
auto- completion	Command autocomplete

# 6.7.2 (Optional) Installing ma-cli Locally

#### **Application Scenarios**

This document describes how to install ma-cli on Windows.

#### Step 1: Install ModelArts SDKs

Install ModelArts SDKs by referring to ModelArts SDK Reference > Preparations > Installing the ModelArts SDK Locally.

#### Step 2: Download ma-cli

- 1. .
- 2. Verify the software package signature.
  - a.
  - b. Install OpenSSL and run the following command to verify the signature: openssl cms -verify -binary -in D:\ma\_cli-latest-py3-none-any.whl.cms -inform DER -content D:\ma\_cli-latest-py3-none-any.whl -noverify > ./test

#### D NOTE

In this example, the software package is stored in **D**:\. Replace it with the actual path.

\$openssl cms -verify -binary -in package.tar.gz.cms -signer "root" -inform DER -content package.tar.gz -noverify > ./te

#### Step 3: Install ma-cli

- Run python --version in the command prompt of your local environment to check whether Python has been installed. The Python version must be later than 3.7.x and earlier than 3.10.x. Version 3.7.x is recommended. C:\Users\xxx>python --version Python \*.\*.\*
- Run pip --version to check whether the general package management tool pip is available. C:\Users\xxx>pip --version
- pip \*\*.\*\* from c:\users\xxx\appdata\local\programs\python\python\*\*\lib\site-packages\pip (python \*.\*)
- 3. Install ma-cli.

# pip install {*Path to the ma-cli software package*}\ma\_cli-latest-py3-none-any.whl

C:\Users\xxx>pip install C:\Users\xxx\Downloads\ma\_cli-latest-py3-none-any.whl

Successfully installed ma\_cli.\*.\*.\*

When ma-cli is installed, dependency packages are installed by default. If message "Successfully installed" is displayed, ma-cli has been installed.

#### **NOTE**

If an error message is displayed during the installation, indicating that a dependency package is missing, run the following command to install the dependency package as prompted:

#### pip install xxxx

*xxxx* is the name of the dependency package.

# 6.7.3 Autocompletion for ma-cli Commands

CLI autocomplete enables you to get a list of supported **ma-cli** commands by typing a command prefix and pressing **Tab** on your terminal. Autocomplete for **ma-cli** commands needs to be enabled in Terminal. After running the **ma-cli auto-completion** command, you can copy and run the commands as prompted on the current terminal to automatically complete the **ma-cli** commands. Bash, Fish, and Zsh shells are supported. The default shell is Bash.

Take the Bash command as an example. Run the **eval** "\$ (\_MA\_CLI\_COMPLETE=bash\_source ma-cli)" command in Terminal to enable autocomplete.

eval "\$(\_MA\_CLI\_COMPLETE=bash\_source ma-cli)"

Run the **ma-cli auto-completion Zsh** or **ma-cli auto-completion Fish** command to view the autocomplete command in Zsh or Fish.

#### Available Commands

\$ ma-cli auto-completion -h
Usage: ma-cli auto-completion [OPTIONS] [[Bash|Zsh|Fish]]

Auto complete ma-cli command in terminal.

Example:

# print bash auto complete command to terminal ma-cli auto-completion Bash

Options: -H, -h, --help Show this message and exit.

# By default, the autocomplete command for Bash is displayed.

\$ ma-cli auto-completion

Tips: please paste following shell command to your terminal to activate auto complation.

[ OK ] eval "\$(\_MA\_CLI\_COMPLETE=bash\_source ma-cli)"

# After the preceding command is executed, autocomplete has been enabled on the terminal.

\$ eval "\$(\_MA\_CLI\_COMPLETE=bash\_source ma-cli)"

# The autocomplete command for Fish is displayed.\$ ma-cli auto-completion FishTips: please paste following shell command to your terminal to activate auto complation.

[ OK ] eval (env \_MA\_CLI\_COMPLETE=fish\_source ma-cli)

# 6.7.4 ma-cli Authentication

#### **Overview**

- VMs and personal computers require the configuration of authentication. Both a username and password (default) and an AK/SK can be used for authentication.
- When using an account for authentication, specify a username and password. When using an IAM account for authentication, specify an account, username, and password.
- In ModelArts notebook, you do not need to manually configure authentication because an agency is used for authentication by default.
- If you have configured authentication in ModelArts notebook, the specified authentication is preferentially used.

#### 

To ensure the security of your sensitive information, it is important to prevent any potential leakage during authentication.

#### **CLI Parameters**

# \$ ma-cli configure -h Usage: ma-cli configure [OPTIONS]

#### Options:

K ROMA] Authentication type.
H ModelArts region file path.
Account of an IAM user.
Username of an IAM user.
Password of an IAM user
User access key.
User secret key.
The region you want to visit.
User project id.
Configure file path for authorization.
Debug Mode. Shows full stack trace when error occurs.
CLI connection profile to use. The default profile is "DEFAULT".
Show this message and exit.

Parameter	Туре	Man dator y	Description
-auth / auth	String	No	Authentication mode, which can be <b>PWD</b> (username and password) or <b>AKSK</b> (AK/SK). The default value is <b>PWD</b> .
-rp / region- profile	String	No	ModelArts region configuration file
-a / account	String	No	IAM tenant account, which needs to be specified when authentication using an IAM account is used. It is required in authentication using a username and password.
-u / username	String	No	Username, which is a username or an IAM username for authentication using an account or an IAM account. It is required in authentication using a username and password.
-p / password	String	No	Password, which is required in authentication using a username and password
-ak / access-key	String	No	Access key, which is required in authentication using an AK/SK
-sk / secret-key	String	No	Secret key, which is required in authentication using an AK/SK
-r /region	String	No	Region name. If this parameter is left blank, the value of the <b>REGION_NAME</b> environment variable will be used by default.
-pi / project-id	String	No	Project ID. If this parameter is left blank, the <b>region</b> value (default) or the value of the <b>PROJECT_ID</b> environment variable will be used.
-P /profile	String	No	Authentication configuration, which defaults to <b>DEFAULT</b>
-C /config- file	String	No	Local path to the configuration file, which defaults to ~/.modelarts/ma-cli-profile.yaml

Table 6-11 Authentication CLI parameters

# Authentication Using Username and Password

The following describes how to use the **ma-cli configure** command on a VM to configure authentication using the user name and password. On a local VM,

specify the YAML file and region endpoint. Obtain the information from the region operations company. The usage is as follows:

#### **NOTE**

In the following example, any string with \${} is a variable. You can specify a value.

For example, \${your\_password} indicates that you need to type your password.

# The DEFAULT authentication configuration is used by default. You need to type the account, username, and password one by one. If the account and username are not required, press Enter to skip them. \$ ma-cli configure --auth PWD --region \${your\_region} --region-profile \${your\_region-profile path} account: \${your\_account} username: \${your\_username} password: \${your\_password} # The input is not displayed on the console.

**\$**{*your\_region-profile path*} indicates the local relative path of the YAML file, for example, ./ModelArts-region-profile.yaml.

#### Authentication Using an AK/SK

This command uses an AK/SK for authentication, which means you have to enter them interactively. Your AK/SK will not be visible on the console.

#### 

In the following example, any string with **\${}** is a variable. You can specify a value. For example, you need to replace **\${access key}** with your access key.

```
ma-cli configure --auth AKSK
access key [***]: ${access key}
secret key [***]: ${secret key}
```

After the authentication command is executed, the authentication information will be saved in the ~/.modelarts/ma-cli-profile.yaml configuration file.

# 6.7.5 ma-cli Image Building Command

#### 6.7.5.1 ma-cli Image Building Command

The **ma-cli image** command can be used to obtain registered images, obtain or load image creation templates, create images using Dockerfiles, obtain or clear image creation caches, register or deregister images, and debug whether images can be used in notebook instances. For details, run the **ma-cli image -h** command.

# **Commands for Creating an Image**

\$ ma-cli image -h
Usage: ma-cli image [OPTIONS] COMMAND [ARGS]...
Obtain registered images, register or unregister images, debug images, and create images in Notebook.
Options:

-H, -h, --help Show this message and exit.

Commands:

add-template, at List build-in dockerfile templates.

build	Build docker image in Notebook.
debug	Debug SWR image as a Notebook in ECS.
df	Query disk usage.
get-image, g	gi Query registered image in ModelArts.
get-templat	e, gt List build-in dockerfile templates.
prune	Prune image build cache.
register	Register image to ModelArts.
unregister	Unregister image from ModelArts.

Table 6-12 Commands for creating an image

Comma nd	Description
get- templat e	Obtain an image creation template.
add- templat e	Load an image creation template.
get- image	Obtain registered ModelArts images.
register	Register SWR images with ModelArts image management.
unregist er	Deregister a registered image from ModelArts image management.
build	Build an image using a Dockerfile (only supported in ModelArts Notebook).
df	Obtain image creation cache, which can only be used in ModelArts notebook.
prune	Clear image creation cache, which can only be used in ModelArts notebook.
debug	Debug an SWR image on an ECS to check whether the image can be used in ModelArts notebook. (Only the ECSs with Docker installed can be used.)

### 6.7.5.2 Obtaining an Image Creation Template

ma-cli provides some common image creation templates, in which the guidance for developing Dockerfiles on ModelArts notebook is provided.

\$ ma-cli image get-template -h Usage: ma-cli image get-template [OPTIONS] List build-in dockerfile templates. Example: # List build-in dockerfile templates ma-cli image get-template [--filer <filter\_info>] [--page-num <yourPageNum>] [--page-size <yourPageSize>]

Options:	
filter TEXT	filter by keyword.
-pn,page-num INTE	GER RANGE Specify which page to query. [x>=1]
-ps,page-size INTEG	ER RANGE The maximum number of results for this query. [x>=1]
-D,debug	Debug Mode. Shows full stack trace when error occurs.
-P,profile TEXT	CLI connection profile to use. The default profile is "DEFAULT".
-H, -h,help	Show this message and exit.
(PyTorch-1.4) [ma-user	work]\$

#### Table 6-13 Parameters

Parameter	Туре	Mandatory	Description
filter	String	No	Filter templates based on the template name keyword.
-pn /page- num	Int	No	Image page index. The default value is page 1.
-ps /page- size	Int	No	Number of images displayed on each page. The default value is <b>20</b> .

#### **Examples**

Obtain an image creation template.

ma-cli image get-template

(PvTorch-1.8) [ma-user work]\$ma-cli image get-templa	te
Template Name	Description
<pre>customize_from_ubuntu_18.04_to_modelarts</pre>	Add ma-user, apt install packages and create a new conda environment with pip based on scratch ubuntu 18.04
upgrade_current_notebook_apt_packages	Install apt packages like ffmpeg, gcc-8, g++-8 based on current Notebook image
migrate 3rd party image to modelarts	General template for migrating your own or open source image to ModelArts
migrate_official_torch_110_cu113_image_to_modelarts	Reconstructing and migrating the official torch 1.10.0 with cuda11.3 image to ModelArts
build handwritten number inference application	Create a new AI application, used to generate an image to deploy and infer in ModelArts
update_dli_image_pip_package	Install pip packages based on DLI image
forward compationed 11 image to modelants	Migrate and forward compaticuda-11 x to ModelArts by ungrading only user-mode CUDA components

#### 6.7.5.3 Loading an Image Creation Template

The **add-template** command is used to load image templates to a specified folder. By default, the path where the current command is located is used,

for example, **\${current\_dir}/.ma/\${template\_name}/**. You can also run the **--dest** command to specify the path. If a template folder with the same name already exists in the target path, run the **--force | -f** parameter to forcibly overwrite the existing template folder.

\$ ma-cli image add-template -h Usage: ma-cli image add-template [OPTIONS] TEMPLATE\_NAME Add buildin dockerfile templates into disk. Example: # List build-in dockerfile templates ma-cli image add-template customize\_from\_ubuntu\_18.04\_to\_modelarts --force Options: --dst TEXT target save path.

-f, --forceOverride templates that has been installed.-D, --debugDebug Mode. Shows full stack trace when error occurs.

-P, --profile TEXT CLI connection profile to use. The default profile is "DEFAULT". -h, -H, --help Show this message and exit.

Paramete r	Туре	Mandat ory	Description
dst	String	No	Load templates to a specified path. The current path is used by default.
-f / force	Bool	No	Whether to forcibly overwrite an existing template with the same name. By default, the template is not overwritten.

Table 6-14 Parameters
-----------------------

#### **Examples**

Load the customize\_from\_ubuntu\_18.04\_to\_modelarts image creation template.

ma-cli image add-template customize\_from\_ubuntu\_18.04\_to\_modelarts

(P/Torch-1.8) [ma-user work]\$ma-cli image add-template customize from ubuntu 18.04 to modelarts [ OK ] Successfully add comfinumation template [ customize from ubuntu 18.04 to modelarts ] under folder [ /home/ma-user/work/.ma/customize from ubuntu 18.04 to modelart

#### 6.7.5.4 Obtaining Registered ModelArts Images

A path to a base image is provided in a Dockerfile typically. Public images and SWR public or private images can be obtained from open-source image repositories such as Docker Hub. ma-cli allows you to obtain ModelArts preset images and registered images and their SWR addresses.

\$ma-cli image get-image -h
Usage: ma-cli image get-image [OPTIONS]
Get registered image list.

Example:

# Query images by image type and only image id, show name and swr\_path ma-cli image get-image --type=DEDICATED

# Query images by image id ma-cli image get-image --image-id \${image\_id}

# Query images by image type and show more information ma-cli image get-image --type=DEDICATED -v

# Query images by image name ma-cli image get-image --filter=torch

Options:

```
-t, --type [BUILD_IN|DEDICATED|ALL]
Image type(default ALL)
```

```
-f, --filter TEXT Image name to filter

-v, --verbose Show detailed information on image.

-i, --image-id TEXT Get image details by image id

-n, --image-name TEXT Get image details by image name

-wi, --workspace-id TEXT The workspace where you want to query image(default "0")

-pn, --page-num INTEGER RANGE Specify which page to query [x>=1]

-ps, --page-size INTEGER RANGE The maximum number of results for this query [x>=1]
```

-C,config-file PATH	Configure file path for authorization.
-D,debug	Debug Mode. Shows full stack trace when error occurs.
-P,profile TEXT	CLI connection profile to use. The default profile is "DEFAULT".
-H, -h,help	Show this message and exit.

#### Table 6-15 Parameters

Parameter	Туре	Man dato ry	Description	
-t /type	String	No	Type of the images to be obtained. The options are <b>BUILD_IN</b> , <b>DEDICATED</b> , and <b>ALL</b> .	
			BUILD_IN: preset images	
			• <b>DEDICATED</b> : custom images registered with ModelArts	
			ALL: all images	
-f /filter	String	No	Keyword of an image name, which is used to filter images	
-v / verbose	Bool	No	Whether to display detailed information. This function is disabled by default.	
-i / image-id	String	No	Obtain details about an image with a specified ID.	
-n / image- name	String	No	Obtain details about an image with a specified name.	
-wi / workspace -id	String	No	Obtain images in a specified workspace.	
-pn / page-num	Int	No	Image page index. The default value is page 1.	
-ps / page-size	Int	No	Number of images displayed on each page. The default value is <b>20</b> .	

# Examples

Obtain custom images registered with ModelArts.

ma-cli image get-image --type=DEDICATED

(PyTorch	-1.8) [ma-user	work]\$ma-cli image get-im	nagetype=DE	DICATED	
INDEX		IMAGE ID	NAME		SWR PATH
1	c857e5a8	fc5e3d002f	0314test		huaweicloud.com/notebook_test/0314test:1.0.0
2	193b2557	d39093a811	0328	+ 	7.myhuaweicloud.com/notebook_test/0328:1
3	171fe036	I3b37e9aa7c	0926	+ 	aweicloud.com/ei_modelarts_y00218826_05/0926:1
4	1b48bb0a	 689b0a7267	0926	+   sv	weicloud.com/ei_modelarts_y00218826_05/0926:111
5	c8667cf0	.d2e3563107	1	+ 	huaweicloud.com/ei_modelarts_y00218826_05/1:6
6	3e6cda6a	1a360eea80e	1	+ 	huaweicloud.com/ei_modelarts_y00218826_05/1:1
7	42e86ca5	'ec198be968	111	+ 	.myhuaweicloud.com/notebook_test/111:1227
8	0f349cef	'c411011ef2	11111110801	+   <u></u>	aweicloud.com/notebook_test/11111110801:111111
9	3a082e32	l4f485aadb6	112121	swr	eicloud.com/ei_modelarts_y00218826_05/112121:123
10	db0d02f6	:74eb00e1ce	1203	+ 	myhuaweicloud.com/notebook_test/1203:1.2.3
11	031dc02e	ld92cd457d8	1227	+ 	.myhuaweicloud.com/notebook_test/1227:111
12	f7d95648	.7aaec8b1cc	1227	+ 	.myhuaweicloud.com/notebook_test/1227:888
13	2f720610	:a1d1db9d7d	1227	+	.myhuaweicloud.com/notebook_test/1227:6666
14	42221bf2	22d726d270	1229	+	.myhuaweicloud.com/notebook_test/1229:123
15	70deea1e	170b2414ae7	123	+	myhuaweicloud.com/mindspore-dis-train/123:2
16	e6cc5414	:ce318069 <del>f</del> 4	123	1	.myhuaweicloud.com/notebook_test/123:45678
17	6e7a86c9	319fb3bb28	1234	1	.myhuaweicloud.com/notebook_test/1234:666
18	ec036306	8c9dc6b391	1234	1	7.myhuaweicloud.com/notebook_test/1234:1
19	b37f8f3b	7a9941c978	441211		.myhuaweicloud.com/notebook_test/441211:11
20	d5acd51b	lef16534d68	aaa		.myhuaweicloud.com/notebook_test/aaa:1.1.1

#### 6.7.5.5 Creating an Image in ModelArts Notebook

Run the **ma-cli image build** command to create an image based on a specified Dockerfile. This command is available only in ModelArts notebook instances.

\$ ma-cli image build -h Usage: ma-cli image build [OPTIONS] FILE\_PATH

Build docker image in Notebook.

Example:

# Build a image and push to SWR ma-cli image build .ma/customize\_from\_ubuntu\_18.04\_to\_modelarts/Dockerfile -swr my\_organization/ my\_image:0.0.1

# Build a image and push to SWR, dockerfile context path is current dir ma-cli image build .ma/customize\_from\_ubuntu\_18.04\_to\_modelarts/Dockerfile -swr my\_organization/ my\_image:0.0.1 -context .

# Build a local image and save to local path and OBS ma-cli image build .ma/customize\_from\_ubuntu\_18.04\_to\_modelarts/Dockerfile --target ./build.tar -obs\_path obs://bucket/object --swr-path my\_organization/my\_image:0.0.1

Options:

-t, --target TEXTName and optionally a tag in the 'name:tag' format.-swr, --swr-path TEXTSWR path without swr endpoint, eg:organization/image:tag. [required]--context DIRECTORYbuild context path.-arg, --build-arg TEXTbuild arg for Dockerfile.-obs, --obs-path TEXTOBS path to save local built image.-f, --forceForce to overwrite the existing swr image with the same name and tag.-C, --config-file PATHConfigure file path for authorization.-D, --debugDebug Mode. Shows full stack trace when error occurs.-P, --profile TEXTCLI connection profile to use. The default profile is "DEFAULT".-H, -h, --helpShow this message and exit.

Table 6-16 Para	meters
-----------------	--------

Parameter	Туре	Ma nda tory	Description
FILE_PATH	String	Yes	Directory where the Dockerfile is stored
-t / target	String	No	Local path for storing the generated TAR package. The current directory is used by default.
-swr / swr-path	String	Yes	SWR image name, which is in the format of "organization/image_name:tag". This parameter can be omitted when a TAR package is saved for creating an image.
context	String	No	Path of the context information for data copying when creating a Dockerfile
-arg / build-arg	String	No	Parameter for creating an image. If there are multiple parameters, runbuild-arg VERSION=18.04build-arg ARCH=X86_64.
-obs / obs-path	String	No	Automatically upload the generated TAR package to OBS.
-f /force	Bool	No	Whether to forcibly overwrite an existing SWR image with the same name. By default, the SWR image is not overwritten.

# Examples

Create an image in ModelArts notebook.

ma-cli image build .ma/customize\_from\_ubuntu\_18.04\_to\_modelarts/Dockerfile -swr notebook\_test/my\_image:0.0.1

In this command, **.ma/customize\_from\_ubuntu\_18.04\_to\_modelarts/Dockerfile** is the path where the Dockerfile is stored, and **notebook\_test/my\_image:0.0.1** is the SWR path of the new image.

(PyTorch-1.8) [ma-user work]\$ma-cli image build .ma/customize_from_ubuntu_18.04_to_modelarts/Doc	<pre>serfile -swr notebook_test/my_image:0.0.1</pre>					
[+] Building 4.3s (8/8) FINISHED						
-> [internal] load .dockerignore						
-> -> transferring context: 2B						
-> [internal] load build definition from Dockerfile						
-> -> transferring dockerfile: 3.29kB						
-> [internal] load metadata for swr.cn-north-7.myhuaweicloud.com/atelier/ubuntu:18.04						
=> [auth] atelier/ubuntu:,pull token for swr.cn-north-7.myhuaweicloud.com						
=> [1/2] FROM swr.cn-north-7.myhuaweicloud.com/atelier/ubuntu:18.04@sha256:b58746c8a89938b8c9f5						
=> => resolve swr.cn-north-7.myhuaweicloud.com/atelier/ubuntu:18.04@sha256:b58746c8a89938b8c9f5						
=> sha256:2910811b6c4227c2f42aaea9a3dd5f53b1d469f67e2cf7e601f631b119b61ff7 847B / 847B						
=> sha256:bc38caa0f5b94141276220daaf428892096e4afd24b05668cd188311e00a635f 35.37kB / 35.37kB						
=> sha256:36505266dcc64eeb1010bd2112e6f73981e1a8246e4f6d4e287763b57f101b0b 161B / 161B						
=> sha256:23884877105a7ff84a910895cd044061a4561385ff6c36480ee080b76ec0e771 26.69MB / 26.69MB						
=> extracting sha256:23884877105a7ff84a910895cd044061a4561385ff6c36480ee080b76ec0e771						
=> => extracting sha256:bc38caa0f5b94141276220daaf428892096e4afd24b05668cd188311e00a635f						
=> extracting sha256:2910811b6c4227c2f42aaea9a3dd5f53b1d469f67e2cf7e601f631b119b61ff7						
=> extracting sha256:36505266dcc64eeb1010bd2112e6f73981e1a8246e4f6d4e287763b57f101b0b						
=> [2/2] RUN default_user=\$(getent passwd 1000   awk -F ':' '{print \$1}')    echo "uid: 1000 do	=> [2/2] RUN default_user=\$(getent passwd 1000   awk -F ':' '{print \$1}')    echo "uid: 1000 does not exist" && default_group=\$(getent group 100   awk -F ':' '{pr 0.					
=> exporting to image						
=> => exporting layers						
=> exporting manifest sha256:b239078457df7c75d57a45989cf8d9d08e6fd9dc882a4ede6d4311bc487d80e						
=> exporting config sha256:6794fa8ae0cc9464b7f3102345237559fb82a377296309b954841b1340cd51db						
=> => pushing layers						
=> => pushing manifest for swr.cn-north-7.myhuaweicloud.com/notebook_test/my_image:0.0.1@sha256						
=> [auth] notebook_test/my_image:,pull,push token for swr.cn-north-7.myhuaweicloud.com						
* Summary Board						
* Image Build Time: 4.3s						
* Repository: swr.cn-north-7.myhuaweicloud.com/notebook_test/my_image						
* Tag: 0.0.1						
* Compressed Image Size: 25MB						
<ul> <li>SWR Download Command: docker pull swr.cn-north-7.myhuaweicloud.com/notebook_test/my_image:0.0.</li> </ul>						
(PyloPch-1.8) [ma-User Work]\$						

# 6.7.5.6 Obtaining Image Creation Caches in ModelArts Notebook

Run the **ma-cli image df** command to obtain image creation caches. This command is available only in ModelArts notebook instances.

\$ ma-cli image df -h Usage: ma-cli image df [OPTIONS]
Query disk usage used by image-building in Notebook.
Example:
# Query image disk usage ma-cli image df
Options:
-v, --verbose Show detailed information on disk usage.
-D, --debug Debug Mode. Shows full stack trace when error occurs.

-h, -H, --help Show this message and exit.

 Table 6-17
 Parameters

Parameter	Туре	Mandator y	Description
-v /verbose	Bool	No	Whether to display detailed information. This function is disabled by default.

#### **Examples**

• View all image caches in ModelArts notebook. ma-cli image df

(PyTorch-1.8)	[ma-user work]\$ma-cli image df			
ID		RECLAIMABLE	SIZE	LAST ACCESSED
iwrrwsi9pdcjaf	e1ij6d0r918	true	98.50MB	
cp52c4q81ud2ab	u2vp7sj5vyt	true	1.04MB	
4jbo6v06r2w157	5ddq3w8g12e	true	139.68kB	
ojdjw5mok71s1n	h2cauant051	true	86.86kB	
k2jm6g061n5twmz7gmonmqjsh		true	16.55kB	
efu5kwgig1ve44	fe7smbrncnh*	true	8.19kB	
uzikwqk5taxnslvajm14jrbje*		true	4.10kB	
2g8p0qcb014g3q	va7ucawkv87*	true	4.10kB	
Reclaimable:	99.80MB			
Total:	99 80MR			

• View details about an image. ma-cli image df --verbose



### 6.7.5.7 Clearing Image Creation Caches in ModelArts Notebook

Run the **ma-cli image prune** command to clear image creation caches. This command is available only in ModelArts notebook instances.

\$ ma-cli image prune -h
Usage: ma-cli image prune [OPTIONS]

Prune image build cache by image-building in Notebook.

Example:

# Prune image build cache ma-cli image prune

#### Options:

-ks, --keep-storage INTEGER Amount of disk space to keep for cache below this limit (in MB) (default: 0). -kd, --keep-duration TEXT Keep cache newer than this limit, support second(s), minute(m) and hour(h) (default: 0).

-v, --verboseShow more verbose output.-D, --debugDebug Mode. Shows full stack trace when error occurs.-h, -H, --helpShow this message and exit.

Parameter	Туре	Mand atory	Description
-ks /keep- storage	Int	No	Size of the cache to be retained, in MB. The default value is <b>0</b> , indicating that all caches will be cleared.
-kd /keep- duration	String	No	Whether to retain the latest caches and clear only historical caches. The unit can be <b>s</b> (second), <b>m</b> (minute), or <b>h</b> (hour). The default value is <b>0</b> , indicating that all caches will be cleared.
-v / verbose	Bool	No	Whether to display detailed information. This function is disabled by default.

#### Table 6-18 Parameters

#### Examples

Retain 1 MB of image cache when clearing caches.

ma-cu image prune -ks	na-cli	image	prune	-ks	1
-----------------------	--------	-------	-------	-----	---

(PyTorch-1.8) [ma-user work]\$ma-cli image prune -ks 1		
ID	RECLAIMABLE	SIZE LAST ACCESSED
uzikwqk5taxnslvajm14jrbje*	true	4.10kB
4jbo6v06r2w1575ddq3w8g12e	true	139.68kB
k2jm6g061n5twmz7gmonmqjsh	true	16.55kB
ojdjw5mok71s1nh2cauant051	true	86.86kB
cp52c4q81ud2abu2vp7sj5vyt	true	1.04MB
iwrrwsi9pdcjafe1ij6d0r918	true	98.50MB
Total: 99.79MB		

### 6.7.5.8 Registering SWR Images with ModelArts Image Management

After an image is debugged, run the **ma-cli image register** command to register it with ModelArts image management so that the image can be used in ModelArts.

```
$ma-cli image register -h
Usage: ma-cli image register [OPTIONS]
 Register image to ModelArts.
 Example:
 # Register image into ModelArts service
 ma-cli image register --swr-path=xx
 # Share SWR image to DLI service
 ma-cli image register -swr xx -td
 # Register image into ModelArts service and specify architecture to be 'AARCH64'
 ma-cli image register --swr-path=xx --arch AARCH64
Options:
                             SWR path without swr endpoint, eg:organization/image:tag. [required]
 -swr, --swr-path TEXT
 -a, --arch [X86_64|AARCH64]
                                Image architecture (default: X86_64).
 -s, --service [NOTEBOOK|MODELBOX]
                       Services supported by this image(default NOTEBOOK).
 -rs, --resource-category [CPU|GPU|ASCEND]
                      The resource category supported by this image (default: CPU and GPU).
 -wi, --workspace-id TEXT
                              The workspace to register this image (default: "0").
 -v, --visibility [PUBLIC|PRIVATE]
                       PUBLIC: every user can use this image. PRIVATE: only image owner can use this
image (Default: PRIVATE).
 -td, --to-dli
                         Register swr image to DLI, which will share SWR image to DLI service.
 -d, --description TEXT
                             Image description (default: "").
 -C, --config-file PATH
                             Configure file path for authorization.
 -D, --debug
                          Debug Mode. Shows full stack trace when error occurs.
 -P, --profile TEXT
                           CLI connection profile to use. The default profile is "DEFAULT".
 -h, -H, --help
                          Show this message and exit.
```

#### Table 6-19 Parameters

Parameter	Туре	Mand atory	Description
-swr /swr- path	String	Yes	SWR path to the image to be registered

Parameter	Туре	Mand atory	Description
-a /arch	String	No	Architecture of the registered image. The value can be <b>X86_64</b> or <b>AARCH64</b> . The default value is <b>X86_64</b> .
-s /service	String	No	Service type of the registered image. The value can be <b>NOTEBOOK</b> or <b>MODELBOX</b> . The default value is <b>NOTEBOOK</b> . You can also specify both values, <b>-s</b> <b>NOTEBOOK -s MODELBOX</b> .
-rs /resource- category	String	No	Resource type that can be used by the registered image. The value can be <b>CPU</b> , <b>GPU</b> , or <b>ASCEND</b> . The default value is <b>CPU</b> and <b>GPU</b> .
-wi / workspace-id	String	No	Register an image into a specified workspace. The default workspace ID is <b>0</b> .
-v /visibility	Bool	No	Available scope of the registered image. The value can be <b>PRIVATE</b> (available only to the image owner) or <b>PUBLIC</b> (available to all users). The default value is <b>PRIVATE</b> .
-td /to-dli	Bool	No	Register an image with DLI.
-d/ description	String	No	Describe an image. By default, this parameter is left blank.

# Examples

Register an SWR image with ModelArts.

ma-cli image register --swr-path=xx

(PyTorch-1.8) [ma-user work]≸ma-cli image registerswr-path=swr.cn- yhuaweicloud.com/notebook :/my_image:0.0.1
You are now in a notebook or devcontainer and cannot use 'ImageManagement.debug' to check your image. If you need to debug it, please use a workstation.
[ OK ] Successfully registered this image and image information is
{
"arch":"x86_64",
"create_at":1680006812157,
"dev_services":[
"NOTEBOOK",
"SSH"
],
"id":"85: 0a66748",
"name":"my_image",
"namespace": "notebook_test",
"origin":"CUSTOMIZE",
"resource_categories":[
"GPU",
"CPU"
],
"service_type":"UNKNOWN",
"size":26735097,
"status":"ACTIVE",
"swr_path":"swr.cn-n myhuaweicloud.com/notebo /my_image:0.0.1",
"tag":"0.0.1",
"tags":[],
"type":"DEDICATED",
"update_at":1680006812157,
"visibility":"PRIVATE",
"workspace_id":"0"

# 6.7.5.9 Deregistering a Registered Image from ModelArts Image Management

Run the **ma-cli image unregister** command to deregister a registered image from ModelArts.

\$ ma-cli image unregister -h Usage: ma-cli image unregister [OPTIONS]

Unregister image from ModelArts.

Example:

# Unregister image ma-cli image unregister --image-id=xx

# Unregister image and delete it from swr ma-cli image unregister --image-id=xx -d

Options:

-i, --image-id TEXT Unregister image details by image id. [required]

- -d, --delete-swr-image Delete the image from swr.
- -C, --config-file PATH Configure file path for authorization.
- -D, --debug Debug Mode. Shows full stack trace when error occurs.
- -P, --profile TEXT CLI connection profile to use. The default profile is "DEFAULT".
- -h, -H, --help Show this message and exit.

Table 6-20 Parameters

Parameter	Туре	Manda tory	Description
-i / -image-id	String	Yes	ID of the image to be deregistered
-d /delete-swr- image	Bool	No	Whether to delete a deregistered SWR image. This function is disabled by default.

#### **Examples**

Deregister a registered image from ModelArts image management.

ma-cli image unregister --image-id=xx



# 6.7.5.10 Debugging an SWR Image on an ECS

ma-cli allows you to debug an SWR image on an ECS to determine whether to use the image in a ModelArts development environment.

```
ma-cli image debug -h
Usage: ma-cli image debug [OPTIONS]
 Debug SWR image as a Notebook in ECS.
 Example:
 # Debug cpu notebook image
 ma-cli image debug ---swr-path=xx ---service=NOTEBOOK ---region=
 # Debug gpu notebook image
 ma-cli image debug —swr-path=xx —service=NOTEBOOK —region= ______ _gpu
Options:
 -swr, --swr-path TEXT
                              SWR path without SWR endpoint, eg:organization/image:tag. [required]
 -r, -region TEXT
                               Region name. [required]
 -s, --service [NOTEBOOK|MODELBOX]
                               Services supported by this image(default NOTEBOOK).
 -a, —arch [X86_64|AARCH64] Image architecture(default X86_64).
 -g, —gpu
                              Use all gpus to debug.
                              Debug Mode. Shows full stack trace when error occurs.
 -D, -debug
 -P, -profile TEXT
                               CLI connection profile to use. The default profile is "DEFAULT".
 -h, -H, -help
                               Show this message and exit.
```

#### Table 6-21 Parameters

Paramet er	Туре	Manda tory	Description
-swr / swr-path	String	Yes	SWR path to the image to be debugged
-r / region	String	Yes	Region where the image to be debugged is located
-s / service	String	No	Service type of the debugged image. The value can be <b>NOTEBOOK</b> or <b>MODELBOX</b> . The default value is <b>NOTEBOOK</b> .
-a / arch	String	No	Architecture of the debugged image. The value can be <b>X86_64</b> or <b>AARCH64</b> . The default value is <b>X86_64</b> .
-g /gpu	Bool	No	GPU debugging status. This function is disabled by default.

# 6.7.6 Using the ma-cli ma-job Command to Submit a ModelArts Training Job

#### 6.7.6.1 ma-cli ma-job Command Overview

Run the **ma-cli ma-job** command to submit training jobs, obtain training job logs, events, used AI engines, and resource specifications, and stop training jobs.

\$ ma-cli ma-job -h Usage: ma-cli ma-job [OPTIONS] COMMAND [ARGS]... ModelArts job submission and query jod details. Options: -h, -H, --help Show this message and exit. Commands: delete Delete training job by job id. get-engine Get job engines. get-event Get job running event. get-flavor Get job flavors. get-job Get job details. get-job Get job log details. get-pool Get job log details. get-pool Get job engines. stop Stop training job by job id. submit Submit training job.

Command	Description
get-job	Obtain ModelArts training jobs and their details.
get-log	Obtain runtime logs of a ModelArts training job.
get-engine	Obtain ModelArts AI engines for training.
get-event	Obtain ModelArts training job events.
get-flavor	Obtain ModelArts resource specifications for training.
get-pool	Obtain ModelArts resource pools dedicated for training.
stop	Stop a ModelArts training job.
submit	Submit a ModelArts training job.
delete	Delete a training job with a specified job ID.

Table C 22	Commonde	cump ortod	h.,	training	:-	4~
	Commanus	supported	DV	traininu	10	DS
			- ,		J - 1	

## 6.7.6.2 Obtaining ModelArts Training Jobs

Run the **ma-cli ma-job get-job** command to view training jobs or details about a specific job.

\$ ma-cli ma-job get-job -h Usage: ma-cli ma-job get-job [OPTIONS]

Get job details.

Example:

# Get train job details by job name ma-cli ma-job get-job -n \${job\_name}

# Get train job details by job id ma-cli ma-job get-job -i \${job\_id}

# Get train job list ma-cli ma-job get-job --page-size 5 --page-num 1

Options:

-i,job-id TEXT -n,job-name TEXT -pn,page-num INTEGE	Get training job details by job id. Get training job details by job name. R Specify which page to guery. [x>=1]
-ps,page-size INTEGER	RANGE The maximum number of results for this query. [1<=x<=50]
-v,verbose	Show detailed information about training job details.
-C,config-file TEXT	Configure file path for authorization.
-D,debug	Debug Mode. Shows full stack trace when error occurs.
-P,profile TEXT	CLI connection profile to use. The default profile is "DEFAULT".
-h, -H,help	Show this message and exit.

#### Table 6-23 Description

Parameter	Туре	Mandato ry	Description
-i /job-id	String	No	Obtain details about a training job with a specified job ID.
-n /job- name	String	No	Obtain a training job with a specified job name or filter training jobs by job name.
-pn /page- num	Int	No	Page number. The default value is page 1.
-ps /page- size	Int	No	Number of training jobs displayed on each page. The default value is <b>10</b> .
-v / verbose	Bool	No	Whether to display detailed information. This function is disabled by default.

#### **Examples**

• Obtain a training task job a specified job ID. ma-cli ma-job get-job -i *b63e90xxx* 

(FyToFcff=1:4) [ma-user work]pma-cii	ma-job get-job =1 0036300a-31						
id	name	status	user_name	duration	create_time	start_time	descripti
b63e90ba-1	workflow_created_job_ed3a963f-5438-4a99-9a19-c97ce 88c48b8	Complet ed	ei_modela	00h:01m:1 6s	2023-03-29 03:41:21	2023-03-29 03:41:30	

• Filter training jobs by job name **auto**. ma-cli ma-job get-job -n *auto* 

inde		id	name	status	user_name	duration	create_time	start_time	
	9b495c		autotest_ohe278-copy- 4582	Complete d	ei_modelarts_y0021882 6_05	00h:01m:3 1s	2023-03-29 07:03:08	2023-03-29 07:05:20	"
	af2147f5-		autotest_ohe278-copy- ae52	Terminat ed	ei_modelarts_y0021882 6_05	00h:10m:4 95	2023-03-29 06:52:16	2023-03-29 06:52:32	1
3	2c1855b1-	2	autotest_nv487q	Failed	ei_modelarts_y0021882 6_05	00h:37m:2 9s	2023-03-29 03:22:31	2023-03-29 03:22:58	16
4	4525b3c9-	F	autotest_x2cjf6	Failed	ei_modelarts_y0021882 6_05	00h:00m:0 1s	2023-03-29 03:19:41	2023-03-29 03:19:49	16
5	4234455d-	)	autotest_sx71zc	Terminat ed	ei_modelarts_y0021882 6_05	00h:00m:0 0s	2023-03-29 02:25:18	N/A	ĺť
6	9810ae49-	l	autotest_s62gz3	Terminat ed	ei_modelarts_y0021882 6_05	00h:09m:0 6s	2023-03-29 02:19:49	2023-03-29 02:20:13	6 
7	90c7de89-	1	autotest_wf8z2g	Abnormal	ei_modelarts_y0021882 6_05	00h:00m:0 0s	2023-03-29 01:43:18	N/A	
8	fc740dc5-	1	autotest_g17mit	Terminat ed	ei_modelarts_y0021882 6_05	00h:00m:0 05	2023-03-29 01:22:19	N/A	
9	5d16fdfe-		autotest_02dfd46i	Terminat ed	ei_modelarts_y0021882 6_05	00h:00m:0  0s	2023-03-29 01:11:26	N/A	
10	3737e56d-	F	autotest_c1utp0	Complete d	ei_modelarts_y0021882 6_05	00h:05m:5 9s	2023-03-29 00:59:28	2023-03-29 01:04:20	

#### 6.7.6.3 Submitting a ModelArts Training Job

Run the **ma-cli ma-job submit** command to submit a ModelArts training job.

Before running this command, configure **YAML\_FILE** to specify the path to the configuration file of the target job. If this parameter is not specified, the configuration file is empty. The configuration file is in YAML format, and its parameters are the **option** parameter of the command. If you specify both the **YAML\_FILE** configuration file and the **option** parameter in the CLI, the value of the **option** parameter will overwrite that in the configuration file.

\$ma-cli ma-job submit -h Usage: ma-cli ma-job submit [OPTIONS] [YAML\_FILE] ... Submit training job. Example: ma-cli ma-job submit --code-dir obs://your\_bucket/code/ --boot-file main.py --framework-type PyTorch --working-dir /home/ma-user/modelarts/user-job-dir/code --framework-version pytorch\_1.8.0-cuda\_10.2-py\_3.7-ubuntu\_18.04-x86\_64 --data-url obs://your\_bucket/dataset/ --log-url obs://your\_bucket/logs/ --train-instance-type modelarts.vm.cpu.8u --train-instance-count 1 Options: --name TEXT Job name. --description TEXT Job description. --image-url TEXT Full swr custom image path. Uid for custom image (default: 1000). --uid TEXT --working-dir TEXT ModelArts training job working directory. --local-code-dir TEXT ModelArts training job local code directory. --user-command TEXT Execution command for custom image. --pool-id TEXT Dedicated pool id. --train-instance-type TEXT Train worker specification. --train-instance-count INTEGER Number of workers. --data-url TEXT OBS path for training data. --log-url TEXT OBS path for training log. --code-dir TEXT OBS path for source code. Training output parameter with OBS path. --output TEXT --input TEXT Training input parameter with OBS path.

env-variables TEXT parameters TEXT	Env variables for training job. Training job parameters (only keyword parameters are supported).
boot-file TEXT	Training job boot file path behinds `code_dir`.
framework-type TEXT	Training job framework type.
framework-version TEX	T Training job framework version.
workspace-id TEXT	The workspace where you submit training job(default "0")
policy [regular economi	c turbo auto]
Trai	ning job policy, default is regular.
volumes TEXT	Information about the volumes attached to the training job.
-q,quiet E	xit without waiting after submit successfully.
-C,config-file PATH	Configure file path for authorization.
-D,debug	Debug Mode. Shows full stack trace when error occurs.
-P,profile TEXT	CLI connection profile to use. The default profile is "DEFAULT".
-H, -h,help	Show this message and exit.

#### Table 6-24 Parameters

Parameter	Туре	Ma nd ato ry	Description
YAML_FILE	Strin g	No	Configuration file of a training job. If this parameter is not specified, the configuration file is empty.
code-dir	Strin g	Yes	OBS path to the training source code
data-url	Strin g	Yes	OBS path to the training data
log-url	Strin g	Yes	OBS path to training logs
train- instance- count	Strin g	Yes	Number of compute nodes in a training job. The default value is <b>1</b> , indicating a standalone node.
boot-file	Strin g	No	Boot file specified when you use a preset command is used to submit a training job. This parameter can be omitted when you use a custom image or command to submit a training job.
name	Strin g	No	Name of a training job
description	Strin g	No	Description of a training job
image-url	Strin g	No	SWR URL of a custom image, which is in the format of "organization/image_name:tag".
uid	Strin g	No	Runtime UID of a custom image. The default value is <b>1000</b> .
working- dir	Strin g	No	Work directory where an algorithm is executed

Parameter	Туре	Ma nd ato ry	Description
local-code- dir	Strin g	No	Local directory to the training container to which the algorithm code directory is downloaded
user- command	Strin g	No	Command for executing a custom image. The directory must be under <b>/home</b> . When <b>code-dir</b> is prefixed with <b>file://</b> , this parameter does not take effect.
pool-id	Strin g	No	Resource pool ID selected for a training job. To obtain the ID, do as follows: Log in to the ModelArts management console, choose <b>Dedicated Resource Pools</b> in the navigation pane on the left, and view the resource pool ID in the dedicated resource pool list.
train- instance- type	Strin g	No	Resource flavor selected for a training job
output	Strin g	No	Training output. After this parameter is specified, the training job will upload the output directory of the training container corresponding to the specified output parameter in the training script to a specified OBS path. To specify multiple parameters, useoutput output1=obs://bucket/ output1output output2=obs://bucket/output2.
input	Strin g	No	Training input. After this parameter is specified, the training job will download the data from OBS to the training container and transfer the data storage path to the training script through the specified parameter. To specify multiple parameters, useinput data_path1=obs:// bucket/data1input data_path2=obs://bucket/data2.
env- variables	Strin g	No	Environment variables input during training. To specify multiple parameters, useenv-variables ENV1=env1env-variables ENV2=env2.
 parameters	Strin g	No	Training input parameters. To specify multiple parameters, useparameters "epoch 0 pretrained".
 framework- type	Strin g	No	Engine selected for a training job

Parameter	Туре	Ma nd ato ry	Description
 framework- version	Strin g	No	Engine version selected for a training job
-q /quiet	Bool	No	After a training job is submitted, the system exits directly and does not print the job status synchronously.
 workspace- id	Strin g	No	Workspace where a training job is deployed. The default value is <b>0</b> .
policy	Strin g	No	Training resource specification mode. The options are <b>regular</b> , <b>economic</b> , <b>turbo</b> , and <b>auto</b> .
volumes	Strin g	No	Mount EFS disks. To specify multiple parameters, usevolumes. "local_path=/xx/yy/ zz;read_only=false;nfs_server_path=xxx.xxx.xxx.xxx:/ " -volumes "local_path=/xxx/yyy/ zzz;read_only=false;nfs_server_path=xxx.xxx.xxx.xxx:/ "

#### Submitting a Training Job Based on a Preset ModelArts Image

Submit a training job by specifying the options parameter in the CLI.

ma-cli ma-job submit --code-dir obs://your-bucket/mnist/code/ \

- --boot-file main.py \
  - --framework-type PyTorch \
  - --working-dir /home/ma-user/modelarts/user-job-dir/code \
  - --framework-version pytorch\_1.8.0-cuda\_10.2-py\_3.7-ubuntu\_18.04-x86\_64 \
  - --data-url obs://your-bucket/mnist/dataset/MNIST/ \
  - --log-url obs://your-bucket/mnist/logs/ \
  - --train-instance-type modelarts.vm.cpu.8u \
  - --train-instance-count 1 \
  - -q

#### The following is an example of train.yaml using a preset image:

# Example .ma/train.yaml (preset image) # pool\_id: pool\_xxxx train-instance-type: modelarts.vm.cpu.8u train-instance-count: 1 data-url: obs://your-bucket/mnist/dataset/MNIST/ code-dir: obs://your-bucket/mnist/code/ working-dir: /home/ma-user/modelarts/user-job-dir/code framework-type: PyTorch framework-version: pytorch\_1.8.0-cuda\_10.2-py\_3.7-ubuntu\_18.04-x86\_64 boot-file: main.py log-url: obs://your-bucket/mnist/logs/

##[Optional] Uncomment to set uid when use custom image mode

uid: 1000 ##[Optional] Uncomment to upload output file/dir to OBS from training platform output: name: output\_dir obs\_path: obs://your-bucket/mnist/output1/ ##[Optional] Uncomment to download input file/dir from OBS to training platform input: - name: data url obs\_path: obs://your-bucket/mnist/dataset/MNIST/ ##[Optional] Uncomment pass hyperparameters parameters: - epoch: 10 - learning\_rate: 0.01 - pretrained: ##[Optional] Uncomment to use dedicated pool pool\_id: pool\_xxxx ##[Optional] Uncomment to use volumes attached to the training job volumes: - efs: local\_path: /xx/yy/zz

#### Using a Custom Image to Create a Training Job

nfs\_server\_path: xxx.xxx.xxx.xxx./

read\_only: false

Submit a training job by specifying the **options** parameter in the CLI.

ma-cli ma-job submit --image-url atelier/pytorch\_1\_8:pytorch\_1.8.0-cuda\_10.2-py\_3.7-ubuntu\_18.04x86\_64-20220926104358-041ba2e \

--code-dir obs://your-bucket/mnist/code/ \

--user-command "export LD\_LIBRARY\_PATH=/usr/local/cuda/compat:\$LD\_LIBRARY\_PATH && cd /home/ma-user/modelarts/user-job-dir/code && /home/ma-user/anaconda3/envs/PyTorch-1.8/bin/ python main.py"

--data-url obs://your-bucket/mnist/dataset/MNIST/ \

--log-url obs://your-bucket/mnist/logs/ \

- --train-instance-type modelarts.vm.cpu.8u \
- --train-instance-count 1 \
- -a

The following is an example of **train.yaml** using a custom image:

```
# Example .ma/train.yaml (custom image)
image-url: atelier/pytorch_1_8:pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-
x86 64-20220926104358-041ba2e
user-command: export LD_LIBRARY_PATH=/usr/local/cuda/compat:$LD_LIBRARY_PATH && cd /home/ma-
user/modelarts/user-job-dir/code && /home/ma-user/anaconda3/envs/PyTorch-1.8/bin/python main.py
train-instance-type: modelarts.vm.cpu.8u
train-instance-count: 1
data-url: obs://your-bucket/mnist/dataset/MNIST/
code-dir: obs://your-bucket/mnist/code/
log-url: obs://your-bucket/mnist/logs/
##[Optional] Uncomment to set uid when use custom image mode
uid: 1000
##[Optional] Uncomment to upload output file/dir to OBS from training platform
output:
   - name: output_dir
```

obs\_path: obs://your-bucket/mnist/output1/

##[Optional] Uncomment to download input file/dir from OBS to training platform input:

- name: data url

obs\_path: obs://your-bucket/mnist/dataset/MNIST/

##[Optional] Uncomment pass hyperparameters

- parameters:
  - epoch: 10learning\_rate: 0.01
  - pretrained:

##[Optional] Uncomment to use dedicated pool pool\_id: pool\_xxxx

##[Optional] Uncomment to use volumes attached to the training job volumes:

- efs:

```
ers:
local_path: /xx/yy/zz
read_only: false
nfs_server_path: xxx.xxx.xxx.xxx./
```

#### Examples

• Submit a training job based on a YAML file. ma-cli ma-job submit ./train-job.yaml



• Submit a training job using preset image **pytorch1.8-cuda10.2-cudnn7-ubuntu18.04** through the CLI.

ma-cli ma-job submit --code-dir obs://automation-use-only/Original/TrainJob/TrainJob-v2/ pytorch1.8.0\_cuda10.2/code/ \

- --boot-file test-pytorch.py \
- --framework-type PyTorch \
- --working-dir /home/ma-user/modelarts/user-job-dir/code \
- --framework-version pytorch\_1.8.0-cuda\_10.2-py\_3.7-ubuntu\_18.04-x86\_64 \
- --data-url obs://automation-use-only/Original/TrainJob/TrainJob-v2/
- pytorch1.8.0\_cuda10.2/data/ \

--log-url obs://automation-use-only/Original/TrainJob/TrainJob-v2/ pytorch1.8.0\_cuda10.2/data/logs/ \ --train-instance-type modelarts.vm.cpu.8u \

--train-instance-type modelarts.vm

(PyTorch-1.4)	ma-user work]\$ma-cli ma-job submitcode-dir obs://au	/Original/TrainJob/T ?' ?/pytorch1.8.0 cuda10.2/code
>	boot-file test-pytorch.py \	
>	framework-type PyTorch \	
	working-dir /home/ma-user/modelarts/user-job-dir/co	de \
	framework-version pytorch_1.8.0-cuda_10.2-py_3.7-ub	ountu_18.04-x86_64 \
>	data-url obs://autom `````````}y/Original/TrainJo	b/TrainJob-v2/pytorch1.8.0_cuda10.2/data/ \
	log-url obs://automa //Original/TrainJob	/TrainJob-v2/pytorch1.8.0_cuda10.2/data/logs/ \
	train-instance-type modelarts.vm.cpu.8u \	
	train instance count 1 \	

#### 6.7.6.4 Obtaining ModelArts Training Job Logs

OK ] Creating OK ] Running

Run the ma-cli ma-job get-log command to obtain ModelArts training job logs.

\$ ma-cli ma-job get-log -h Usage: ma-cli ma-job get-log [OPTIONS]

Get job log details.

Example:

# Get job log by job id ma-cli ma-job get-log --job-id \${job\_id}

Options:-i,job-id TEXTGet training job details by job id. [required]-t,task-id TEXTGet training job details by task id (default "worker-0")C,config-file TEXTConfigure file path for authorizationD,debugDebug Mode. Shows full stack trace when error occursP,profile TEXTCLI connection profile to use. The default profile is "DEFAULT"h, -H,helpShow this message and exit.				
Parameter	Туре	Mandatory	Description	
-i /job-id	String	Yes	Obtain logs of a training job with a specified job ID.	
-t /task-	String	No	Obtain logs of a specified task, which	

#### **Examples**

Obtain logs of a training job with a specified job ID.

ma-cli ma-job get-log --job-id *b63e90baxxx* 

(PyTorch-1.4) [ma-user work]\$ma-cli ma-job get-logjob-id b	963e90ba-
time="2023-03-29T11:41:26+08:00" level=info msg="init logger	successful" file="init.go:55" Command=bootstrap/init Component=ma-training-toolkit Platform=ModelArts-Service
time="2023-03-29T11:41:26+08:00" level=info msg="current user	1000:1000" file="init.go:57" Command-bootstrap/init Component=ma-training-toolkit Platform-ModelArts-Service
time="2023-03-29T11:41:27+08:00" level=info msg="report even	nt-ma-training-toolkit Platform=ModelArts-Servi
time="2023-03-29T11:41:27+08:00" level=info msg="init command	code/'" file="init.go:81" Command=bootstrap/ini
aining-toolkit Platform=ModelArts-Service	
time="2023-03-29T11:41:27+08:00" level=info msg="scc is alre	ModelArts-Service
time="2023-03-29T11:41:27+08:00" level=info msg="[init] tool	vice
time="2023-03-29T11:41:27+08:00" level=info msg="[init] runn	Service
time="2023-03-29T11:41:27+08:00" level=info msg="[init] ip o	ice
time="2023-03-29T11:41:27+08:00" level=info msg="local dir =	Component-ma-training-toolkit Platform=ModelAr
time="2023-03-29T11:41:27+08:00" level=info msg="obs dir = s	<pre>file="upload.go:209" Command=obs/upload Compon</pre>
oolkit Platform-ModelArts-Service Task-	
time="2023-03-29T11:41:27+08:00" level=info msg="num of worke	rs = 8" file="upload.go:214" Command=obs/upload Component=ma-training-toolkit Platform=ModelArts-Service Task=
time="2023-03-29T11:41:27+08:00" level=info msg="start the pe	riodic upload task, upload Period = 5 seconds " file="upload.go:220" Command=obs/upload Component=ma-training-toolki
rts-Service Task=	
time="2023-03-29T11:41:27+08:00" level=info msg="report event	DetectStart success" file="event.go:63" Command=report Component=ma-training-toolkit Platform=ModelArts-Service

# 6.7.6.5 Obtaining ModelArts Training Job Events

Run the ma-cli ma-job get-event command to view ModelArts training job events.

\$ ma-cli ma-job get-event -h Usage: ma-cli ma-job get-event [OPTIONS]

Get job running event.

Example:

# Get training job running event ma-cli ma-job get-event --job-id \${job\_id}

Options:

-i, --job-id TEXT Get training job event by job id. [requi -C, --config-file TEXT Configure file path for authorization. Get training job event by job id. [required]

-D, --debug Debug Mode. Shows full stack trace when error occurs.

-P, --profile TEXT CLI connection profile to use. The default profile is "DEFAULT".

-H, -h, --help Show this message and exit.

Parameter	Туре	Mandatory	Description
-i /job-id	String	Yes	Obtain events of a training job with a specified job ID.

# Examples

Obtain events of a training job with a specified job ID.

ma-cli ma-job get-event --job-id *b63e90baxxx* 



# 6.7.6.6 Obtaining ModelArts AI Engines for Training

Run the **ma-cli ma-job get-engine** command to obtain ModelArts AI engines for training.

\$ ma-cli ma-job get-engine -h Usage: ma-cli ma-job get-engine [OPTIONS]
Get job engine info.
Example:
# Get training job engines ma-cli ma-job get-engine
Options:-v,verboseShow detailed information about training enginesC,config-file TEXTConfigure file path for authorizationD,debugDebug Mode. Shows full stack trace when error occursP,profile TEXTCLI connection profile to use. The default profile is "DEFAULT"H, -h,helpShow this message and exit.

Table 6-25 Parameters

Parameter	Туре	Mandatory	Description
-v /verbose	Bool	No	Whether to display detailed information. This function is disabled by default.

# Examples

View the AI engine of a training job.

ma-cli ma-job get-engine

(PyTorch-1.4) [ma-user work]\$ma-cli ma-job get-engine					
index	engine id	engine name	run user		
1	caffe-1.0.0-python2.7	Caffe			
2	horovod-cp36-tf-1.16.2	Horovod			
3	horovod_0.20.0-pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	Horovod	1102		
4	horovod_0.20.0-tensorflow_2.1.0-cuda_10.1-py_3.7-ubuntu_18.04-x86_64	Horovod	1102		
5	kungfu-0.2.2-tf-1.13.1-python3.6	KungFu			
6	mindspore_1.3.0-cuda_10.1-py_3.7-ubuntu_1804-x86_64	MPI	1102		
7	<pre>mindspore_1.7.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64</pre>	Ascend-Powered-Engine	1000		
8	<pre>mindspore_1.8.0-cann_5.1.2-py_3.7-euler_2.8.3-aarch64</pre>	Ascend-Powered-Engine	1000		
9	<pre>mindspore_1.9.0-cann_6.0.0-py_3.7-euler_2.8.3-aarch64</pre>	Ascend-Powered-Engine	1000		
10	mxnet-1.2.1-python3.6	MXNet			
11	optverse_0.2.0-pygrassland_1.1.0-py_3.7-ubuntu_18.04-x86_64	OR	1000		
12	pytorch-cp36-1.0.0	PyTorch			
13	pytorch-cp36-1.3.0	PyTorch			
13   14	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0	PyTorch PyTorch			
13   14   15	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64	PyTorch PyTorch PyTorch Ascend-Powered-Engine	1000		
13   14   15   15   16	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64	PyTorch PyTorch Ascend-Powered-Engine PyTorch	1000		
13   14   15   15   16   17	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64 pytorch_1.8.1-cann_5.1.2-py_3.7-euler_2.8.3-aarch64	PyTorch PyTorch Ascend-Powered-Engine PyTorch Ascend-Powered-Engine	1000 1000 1000		
13 +	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64 pytorch_1.8.1-cann_5.1.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cann_6.0.0-py_3.7-euler_2.8.3-aarch64	PyTorch PyTorch Ascend-Powered-Engine PyTorch Ascend-Powered-Engine Ascend-Powered-Engine	1900 1900 1900 1900		
13   14   15   16   16   17   18   18   19	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.0-cuda_10.2-py_3.7-ubuntu_18.04-x86_64 pytorch_1.8.1-cann_5.1.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cann_6.0.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cuda_11.1-py_3.7-ubuntu_18.04-x86_64	PyTorch PyTorch Ascend-Powered-Engine PyTorch Ascend-Powered-Engine Ascend-Powered-Engine PyTorch	1000 1000 1000 1000 1000		
13   14   15   15   16   17   17   18   19   20	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.0-cuda_10.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cann_5.1.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cann_6.0.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cuda_11.1-py_3.7-euler_1.8.04-x86_64 pytorch_1.8.2-cuda_10.2-py_3.7-eulentu_18.04-x86_64	PyTorch PyTorch Ascend-Powered-Engine PyTorch Ascend-Powered-Engine Ascend-Powered-Engine PyTorch PyTorch	1000 1000 1000 1000 1000 1000		
13   14   15   16   17   18   19   20   21	pytorch-cp36-1.3.0 pytorch-cp36-1.4.0 pytorch_1.8.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.0-cuda_10.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cann_5.1.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cann_6.0.0-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.1-cuda_11.1-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.2-cuda_10.2-py_3.7-euler_2.8.3-aarch64 pytorch_1.8.2-cuda_11.1-py_3.7-euler_1.8.04-x86_64 pytorch_1.9.1-cuda_11.1-py_3.7-eulentu_18.04-x86_64	PyTorch PyTorch Ascend-Powered-Engine PyTorch Ascend-Powered-Engine Ascend-Powered-Engine PyTorch PyTorch PyTorch	1000 1000 1000 1000 1000 1000 1000		

# 6.7.6.7 Obtaining ModelArts Resource Specifications for Training

Run the **ma-cli ma-job get-flavor** command to obtain ModelArts resource specifications for training.

\$ ma-cli ma-job get-flavor -h Usage: ma-cli ma-job get-flavor [OPTIONS]

Get job flavor info.

Example:

# Get training job flavors ma-cli ma-job get-flavor

Options:

-t,flavor-type [CPU	GPU[Ascend]
	Type of training job flavor.
-v,verbose	Show detailed information about training flavors.
-C,config-file TEXT	Configure file path for authorization.
-D,debug	Debug Mode. Shows full stack trace when error occurs.
-P,profile TEXT	CLI connection profile to use. The default profile is "DEFAULT
-H, -h,help	Show this message and exit.
Table 6-26
 Parameters

Parameter	Туре	Mandatory	Description
-t /flavor- type	String	No	Resource flavor. If this parameter is not specified, all resource flavors are returned by default.
-v /verbose	Bool	No	Whether to display detailed information. This function is disabled by default.

# **Examples**

View the resource flavor and type of a training job.

ma-cli ma-job get-flavor

(PyTorch	(PyTorch-1.4) [ma-user work]\$ma-cli ma-job get-flavor				
index	flavor id	flavor name	flavor type		
1	modelarts.kat1.8xlarge	Computing NPU(8*Ascend) instance	Ascend		
2	modelarts.kat1.xlarge	Computing NPU(Ascend) instance	Ascend		
3	modelarts.vm.cpu.2u	Computing CPU(2U) instance	CPU		
4	modelarts.vm.cpu.8u	Computing CPU(8U) instance	CPU		
5	modelarts.vm.cpu.8u16g.119	Computing CPU(8U) instance	CPU		
6	modelarts.vm.v100.large	Computing GPU(V100) instance	GPU		
7	modelarts.vm.v100.large.free	Computing GPU(V100) instance	GPU		

# 6.7.6.8 Stopping a ModelArts Training Job

Run the **ma-cli ma-job stop** command to stop a training job with a specified job ID.

\$ ma-cli ma-job stop -h Usage: ma-cli ma-job stop [OPTIONS] Stop training job by job id. Example: Stop training job by job id ma-cli ma-job stop --job-id \${job\_id}

Options:	
-i,job-id TEXT	Get training job event by job id. [required]
-y,yes (	Confirm stop operation.
-C,config-file TEX	KT Configure file path for authorization.
-D,debug	Debug Mode. Shows full stack trace when error occurs.
-P,profile TEXT	CLI connection profile to use. The default profile is "DEFAULT"
-H, -h,help	Show this message and exit.

#### Table 6-27 Parameters

Parameter	Туре	Mandatory	Description
-i /job-id	String	Yes	Training job ID
-y /yes	Bool	No	Whether to forcibly stop a specified training job

#### Examples

Stop a running training job.

ma-cli ma-job stop --job-id efd3e2f8xxx

# 6.7.7 Using ma-cli to Copy OBS Data

Run the ma-cli obs-copy [SRC] [DST] command to copy a local file to an OBS folder or an OBS file or folder to a local path.

\$ma-cli obs-copy -h Usage: ma-cli obs-copy [OPTIONS ] SRC DST

Copy file or directory between OBS and local path. Example:

# Upload local file to OBS path ma-cli obs-copy ./test.zip obs://your-bucket/copy-data/

# Upload local directory to OBS path ma-cli obs-copy ./test/ obs://your-bucket/copy-data/

# Download OBS file to local path ma-cli obs-copy obs://your-bucket/copy-data/test.zip ./test.zip

# Download OBS directory to local path ma-cli obs-copy obs://your-bucket/copy-data/ ./test/

Options:

-d, --drop-last-dir Whether to drop last directory when copy folder. if True, the last directory of the source folder will not copy to the destination folder. [default: False]

-C, --config-file PATH Configure file path for authorization.

-D, --debug Debug Mode. Shows full stack trace when error occurs. CLI connection profile to use. The default profile is "DEFAULT".

-P, --profile TEXT

-H, -h, --help Show this message and exit.

 Table 6-28
 Parameters

Parameter	Туре	Mandat ory	Description
-d /drop-last- dir	Bool	No	If you specify this parameter, the last- level directory of the source folder will not be copied to the destination folder. This parameter is valid only for copying folders.

## Examples

# Upload a file to OBS.

\$ ma-cli obs-copy ./test.csv obs://\${your\_bucket}/test-copy/ [ OK ] local src path: [ /home/ma-user/work/test.csv ] [ OK ] obs dst path: [ obs://\${your\_bucket}/test-copy/ ]

#### # Upload a folder to obs://\${your\_bucket}/test-copy/data/.

\$ ma-cli obs-copy /home/ma-user/work/data/ obs://\${your\_bucket}/test-copy/ [ OK ] local src path: [ /home/ma-user/work/data/ ] [ OK ] obs dst path: [ obs://\${your\_bucket}/test-copy/ ]

# # Upload a folder to **obs://\${your\_bucket}/test-copy/** with **--drop-last-dir** specified.

\$ ma-cli obs-copy /home/ma-user/work/data/ obs://\${your\_bucket}/test-copy/ --drop-last-dir [ OK ] local src path: [ /home/ma-user/work/data ] [ OK ] obs dst path: [ obs://\${your\_bucket}/test-copy/ ]

#### # Download a folder from OBS to a local disk.

\$ ma-cli obs-copy obs://\${your\_bucket}/test-copy/ ~/work/test-data/ [ OK ] obs src path: [ obs://\${your\_bucket}/test-copy/ ] [ OK ] local dst path: [ /home/ma-user/work/test-data/ ]

# **7** Training Management

# 7.1 Introduction to Model Development

AI modeling involves two stages:

- Development: Prepare and configure the environment, and debug code for training based on deep learning. ModelArts DevEnviron is recommended for code debugging.
- Experiment: Optimize the datasets and hyperparameters, and obtain an ideal model through multiple rounds of experiments. The ModelArts training platform is recommended for training.

In the two stages, code is designed, developed and tested in repeated cycles. In the development stage, when the code becomes stable, the modeling process enters the experiment stage, during which hyperparameters are continuously optimized to iterate the model. In the experiment stage, when the training performance can be optimized, the modeling process returns to the development stage for optimizing code.

Figure 7-1 Model development process



ModelArts provides model training, which allows you to view training results and tune model parameters based on the training results. You can select resource pools with different instance flavors for model training. The following guides you to train models on ModelArts:

- Upload the labeled data to OBS. For details, see Preparing Data.
- Follow the instructions provided in **Preparing Algorithms** to use an algorithm for model training.
- Create a training job. You can perform this operation on the ModelArts console. For details, see **Creating a Training Job**.
- Follow the instructions provided in **Training Job Logs** to view training job logs and training resource usage.
- Follow the instructions provided in **Stopping**, **Rebuilding**, or **Searching for a Training Job** to stop or delete a training job.
- Troubleshoot if you encounter any problem during training. For details, see **Troubleshooting**.

# 7.2 Preparing Data

ModelArts uses OBS to store data, and backs up and takes snapshots for models, achieving secure, reliable storage at low costs.

- OBS
- Obtaining Training Data

#### OBS

OBS provides stable, secure, and efficient cloud storage service that lets you store virtually any volume of unstructured data in any format. Bucket and objects are basic concepts in OBS. A bucket is a container for storing objects in OBS. Each bucket is specific to a region and has specific storage class and access permissions. A bucket is accessible through its domain name over the Internet. An object is the basic unit of data storage in OBS.

OBS is a data storage center for ModelArts. All the input data, output data, and cache data during AI development can be stored in OBS buckets for reading.

Before using ModelArts, create an OBS bucket and folders for storing data.



# **Obtaining Training Data**

Use either of the following methods to obtain ModelArts training data:

• Datasets stored in OBS buckets

After labeling and preprocessing your dataset, upload it to an OBS bucket. When you create a training job, set **Input** to the path of the OBS bucket where the training data is stored.

• Datasets in data management

If your dataset has not labeled or requires preprocessing, import it to ModelArts data management for data preprocessing.

#### **NOTE**

ModelArts data management is being upgraded and is invisible to users who have not used data management. It is recommended that new users store their training data in OBS buckets.

#### Figure 7-3 Preparing data



# 7.3 Preparing Algorithms

# 7.3.1 Introduction to Algorithm Preparation

Machine learning explores general rules from limited volume of data and uses these rules to predict unknown data. To obtain more accurate prediction results, select a proper algorithm to train your model. ModelArts provides a large number of algorithm samples for different scenarios. This section describes algorithm sources and learning modes.

## **Algorithm Sources**

You can use one of the following methods to build a ModelArts model:

• Using a preset image

To use a custom algorithm, use a framework built in ModelArts. ModelArts supports most mainstream AI engines. For details, see **Built-in Training Engines**. These built-in engines pre-load some extra Python packages, such as NumPy. You can also use the **requirements.txt** file in the code directory to install dependency packages. For details about how to create a training job using a preset image, see **Using a Preset Image (Custom Script)**.

• Using a custom image (For details about the new version of training, see Using a Custom Image to Train a Model.)

The subscribed algorithms and built-in frameworks can be used in most training scenarios. In certain scenarios, ModelArts allows you to create custom images to train models. Custom images can be used to train models in ModelArts only after they are uploaded to the Software Repository for Container (SWR). Customizing an image requires a deep understanding of containers. Use this method only if the subscribed algorithms and custom scripts cannot meet your requirements.

## Algorithm Learning Modes

ModelArts allows you to train models in different modes as required.

• Offline learning

Offline learning is the most fundamental mode for model training. In this mode, all data required for training must be provided at a time, and optimizing the objective function stops when the training is complete. The advantage of this mode is that the trained models are stable, facilitating model verification and evaluation.

• Incremental learning

Incremental learning is a continuous learning process. Compared with offline learning, it does not need to store all training data at a time, which alleviates the problem of limited storage resources. In addition, it saves a large amount of compute power and time, and reduces economic costs in retraining.

# 7.3.2 Using a Preset Image (Custom Script)

# 7.3.2.1 Overview

If the subscribed algorithms cannot meet your requirements or you want to migrate local algorithms to ModelArts for training, use the ModelArts preset images to create algorithms. This method is also called using a preset image.

This section describes how to use a preset image to create an algorithm.

- For details about ModelArts built-in engines and models, see **Built-in Training Engines**.
- To migrate local algorithms to ModelArts, perform code adaptation. For details, see **Developing a Custom Script**.
- For details about how to use a preset image to create an algorithm on the ModelArts console, see **Creating an Algorithm**.

## Built-in Training Engines

The following table lists the training engines and their versions supported by ModelArts.

**NOTE** 

Supported AI engines vary depending on regions.

Runtime Environmen t	System Archite cture	System Version	AI Engine and Version	Supported CUDA or Ascend Version
Ascend- Powered- Engine	aarch6 4	Euler2.8	mindspore_2.0.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0
PyTorch	aarch6 4	Euler2.8	pytorch_1.11.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0

Table 7-1 AI engines supported by training jobs

Runtime Environmen t	System Archite cture	System Version	AI Engine and Version	Supported CUDA or Ascend Version
TensorFlow	aarch6 4	Euler2.8	tensorflow_1.15.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0

# 7.3.2.2 Developing a Custom Script

Before you use a preset image to create an algorithm, develop the algorithm code. This section describes how to modify local code for model training on ModelArts.

When creating an algorithm, set the code directory, boot file, input path, and output path. These settings enable the interaction between your codes and ModelArts.

Code directory

Specify the code directory in the OBS bucket and upload training data such as training code, dependency installation packages, or pre-generated model to the directory. After you create the training job, ModelArts downloads the code directory and its subdirectories to the container.

Take OBS path **obs://obs-bucket/training-test/demo-code** as an example. The content in the OBS path will be automatically downloaded to **\$ {MA\_JOB\_DIR}/demo-code** in the training container, and **demo-code** (customizable) is the last-level directory of the OBS path.

Do not store training data in the code directory. When the training job starts, the data stored in the code directory will be downloaded to the backend. A large amount of training data may lead to a download failure. It is recommended that the size of the code directory does not exceed 50 MB.

• Boot file

The boot file in the code directory is used to start the training. Only Python boot files are supported.

Input path

The training data must be uploaded to an OBS bucket or stored in the dataset. In the training code, **the input path** must be parsed. ModelArts automatically downloads the data in the input path to the local container directory for training. Ensure that you have the read permission to the OBS bucket. After the training job is started, ModelArts mounts a disk to the / **cache** directory. You can use this directory to store temporary files. For details about the size of the **/cache** directory, see What Are Sizes of the **/cache Directories for Different Resource Specifications in the Training Environment**?

• Output path

You are advised to set an empty directory as the training output path. In the training code, **the output path** must be parsed. ModelArts automatically uploads the training output to the output path. Ensure that you have the write and read permissions to the OBS bucket.

The following section describes how to develop training code in ModelArts.

## (Optional) Introducing Dependencies

- 1. If your model references other dependencies, place the required file or installation package in **Code Directory** you set during algorithm creation.
  - For details about how to install the Python dependency package, see
     How Do I Create a Training Job When a Dependency Package Is
     Referenced by the Model to Be Trained?
  - For details about how to install a C++ dependency library, see How Do I Install a Library That C++ Depends on?
  - For details about how to load parameters to a pre-trained model, see
     "How Do I Load Some Well Trained Parameters During Job Training?" in FAQs.

#### **Parsing Input and Output Paths**

When a ModelArts model reads data stored in OBS or outputs data to a specified OBS path, perform the following operations to configure the input and output data:

1. Parse the input and output paths in the training code. The following method is recommended:

```
import argparse
# Create a parsing task.
parser = argparse.ArgumentParser(description='train mnist')
# Add parameters.
parser.add_argument('--data_url', type=str, default="./Data/mnist.npz", help='path where the dataset
is saved')
parser.add_argument('--train_url', type=str, default="./Model", help='path where the model is saved')
# Parse the parameters.
```

# Parse the parameters.
args = parser.parse\_args()

After the parameters are parsed, use **data\_url** and **train\_url** to replace the paths to the data source and the data output, respectively.

2. When creating a training job, set the input and output paths.

Select the OBS path or dataset path as the training input, and the OBS path as the output.

## Editing Training Code and Saving the Model

Training code and the code for saving the model are closely related to the AI engine you use. The following uses the TensorFlow framework as an example. Before using this case, you need to **download** the **mnist.npz** file and upload it to the OBS bucket. The training input is the OBS path where the **mnist.npz** file is stored.

```
import os
import argparse
import tensorflow as tf
parser = argparse.ArgumentParser(description='train mnist')
parser.add_argument('--data_url', type=str, default="./Data/mnist.npz", help='path where the dataset is
saved')
parser.add_argument('--train_url', type=str, default="./Model", help='path where the model is saved')
args = parser.parse_args()
```

# 7.3.2.3 Creating an Algorithm

Your locally developed algorithms or algorithms developed using other tools can be uploaded to ModelArts for unified management. Note the following when creating a custom algorithm:

- 1. Prerequisites
- 2. Accessing the Algorithm Creation Page
- 3. Setting Basic Parameters
- 4. Setting the Boot Mode
- 5. Configuring Pipelines
- 6. Defining Hyperparameters
- 7. Supported Policies
- 8. Adding Training Constraints
- 9. Runtime Environment Preview
- 10. Follow-up Operations

#### Prerequisites

- Data is available either by creating a dataset in ModelArts or by uploading the dataset used for training to the OBS directory.
- Your training script has been uploaded to the OBS directory. For details about how to develop a training script, see **Developing a Custom Script**.
- At least one empty folder has been created in OBS for storing the training output.

#### Accessing the Algorithm Creation Page

- 1. Log in to the ModelArts management console and click **Algorithm Management** in the left navigation pane.
- 2. On the **My Algorithms** page, click **Create**. The **Create Algorithm** page is displayed.

# **Setting Basic Parameters**

Enter the basic algorithm information, including Name and Description.

## Setting the Boot Mode

Select a preset image to create an algorithm.

Set **Image**, **Code Directory**, and **Boot File** based on the algorithm code. Ensure that the framework of the AI image you select is the same as the one you use for editing algorithm code. For example, if TensorFlow is used for editing algorithm code, select a TensorFlow image when you create an algorithm.

Parameter	Description		
Boot Mode > Preset image	Select a preset image and its version used by the algorithm.		
Code Directory	OBS path for storing the algorithm code. The files required for training, such as the training code, dependency installation packages, and pre-generated models, are uploaded to the code directory.		
	Do not store training data in the code directory. When the training job starts, the data stored in the code directory will be downloaded to the backend. A large amount of training data may lead to a download failure.		
	After you create the training job, ModelArts downloads the code directory and its subdirectories to the container.		
	Take OBS path <b>obs://obs-bucket/training-test/demo-code</b> a an example. The content in the OBS path will be automatical downloaded to <b>\${MA_JOB_DIR}/demo-code</b> in the training container, and <b>demo-code</b> (customizable) is the last-level directory of the OBS path.		
	NOTE		
	<ul><li>Any programming language is supported.</li><li>The number of files (including files and folders) cannot exceed</li></ul>		
	<ul><li>1,000.</li><li>The total size of files cannot exceed 5 GB.</li></ul>		
Boot File	The file must be stored in the code directory and end with .py. ModelArts supports boot files edited only in Python.		
	The boot file in the code directory is used to start a training job.		

Table 7-2 Parameters

# **Configuring Pipelines**

A preset image-based algorithm obtains data from an OBS bucket or dataset for model training. The training output is stored in an OBS bucket. The input and output parameters in your algorithm code must be parsed to enable data exchange between ModelArts and OBS. For details about how to develop code for training on ModelArts, see **Developing a Custom Script**.

When you use a preset image to create an algorithm, configure the input and output pipelines.

• Input configurations

 Table 7-3 Input configurations

Paramete r	Description
Parameter Name	Set the name based on the data input parameter in your algorithm code. The code path parameter must be the same as the training input parameter parsed in your algorithm code. Otherwise, the algorithm code cannot obtain the input data.
	For example, If you use <b>argparse</b> in the algorithm code to parse <b>data_url</b> into the data input, set the data input parameter to <b>data_url</b> when creating the algorithm.
Descriptio n	Customizable description of the input parameter,
Obtained from	Source of the input parameter. You can select Hyperparameters (default) or Environment variables.
Constraint s	Whether data is obtained from a storage path or ModelArts dataset.
	If you select the ModelArts dataset as the data source, the following constraints are added:
	• Labeling Type: For details, see Creating a Labeling Job.
	• <b>Data Format</b> , which can be <b>Default</b> , <b>CarbonData</b> , or both. <b>Default</b> indicates the manifest format.
	<ul> <li>Data Segmentation: available only for image classification, object detection, text classification, and sound classification datasets.</li> <li>Possible values are Segmented dataset, Dataset not segmented, and Unlimited. For details, see Publishing a Data Version.</li> </ul>
Add	Multiple data input sources are allowed.

• Output configurations

Parameter	Description
Parameter Name	Set the name based on the data output parameter in your algorithm code. The code path parameter must be the same as the training output parameter parsed in your algorithm code. Otherwise, the algorithm code cannot obtain the output path.
	For example, If you use <b>argparse</b> in the algorithm code to parse <b>train_url</b> into the data output, set the data output parameter to <b>train_url</b> when creating the algorithm.
Descriptio n	Customizable description of the output parameter,
Obtained from	Source of the output parameter. You can select Hyperparameters (default) or Environment variables.
Add	Multiple data output paths are allowed.

#### Table 7-4 Output configurations

# **Defining Hyperparameters**

When you use a preset image to create an algorithm, ModelArts allows you to customize hyperparameters so you can view or modify them anytime. After the hyperparameters are defined, they are displayed in the startup command and transferred to your boot file as CLI parameters.

1. Import hyperparameters.

You can click Add hyperparameter to manually add hyperparameters.

2. Edit hyperparameters.

For details, see **Table 7-5**.

Table	7-5	Hyperparameters
-------	-----	-----------------

Parame ter	Description
Name	Hyperparameter name Enter 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Туре	Type of the hyperparameter, which can be <b>String</b> , <b>Integer</b> , <b>Float</b> , or <b>Boolean</b>
Default	Default value of the hyperparameter, which is used for training jobs by default
Constrai nts	Click <b>Restrain</b> . Then, set the range of the default value or enumerated value in the dialog box displayed.

Parame ter	Description
Require	Select <b>Yes</b> or <b>No</b> .
d	• If you select <b>No</b> , you can delete the hyperparameter on the training job creation page when using this algorithm to create a training job.
	• If you select <b>Yes</b> , you cannot delete the hyperparameter on the training job creation page when using this algorithm to create a training job.
Descript	Description of the hyperparameter
ion	Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), and periods (.) are allowed.

# **Supported Policies**

Only the pytorch\_1.8.0-cuda\_10.2-py\_3.7-ubuntu\_18.04-x86\_64 and tensorflow\_2.1.0-cuda\_10.1-py\_3.7-ubuntu\_18.04-x86\_64 images are available for auto search.

# **Adding Training Constraints**

You can add training constraints of the algorithm based on your needs.

- **Resource Type**: Select the required resource types.
- **Multicard Training**: Choose whether to support multi-card training.
- **Distributed Training**: Choose whether to support distributed training.

## **Runtime Environment Preview**



When creating an algorithm, click the arrow on in the lower right corner of the page to know the path of the code directory, boot file, and input and output data in the training container.

## **Follow-up Operations**

After an algorithm is created, use it to create a training job. For details, see **Creating a Training Job**.

# 7.3.3 Using Custom Images

The subscribed algorithms and preset images can be used in most training scenarios. In certain scenarios, ModelArts allows you to create custom images to train models.

Customizing an image requires a deep understanding of containers. Use this method only if the subscribed algorithms and preset images cannot meet your requirements. Custom images can be used to train models in ModelArts only after they are uploaded to the Software Repository for Container (SWR).

You can use custom images for training on ModelArts in either of the following ways:

• Using a preset image with customization

If you use a preset image to create a training job and you need to modify or add some software dependencies based on the preset image, you can customize the preset image. In this case, select a preset image and choose **Customize** from the framework version drop-down list box.

• Using a custom image

You can create an image based on the ModelArts image specifications, select your own image and configure the code directory (optional) and boot command to create a training job.

#### **NOTE**

When you use a custom image to create a training job, the boot command must be executed in the **/home/ma-user** directory. Otherwise, the training job may run abnormally.

## Using a Preset Image with Customization

The only difference between this method and creating a training job totally based on a preset image is that you must select an image. You can create a custom image based on a preset image. For details about how to create a custom image based on a preset framework, see **Using a Base Image to Create a Training Image**.

★ Boot Mode	1 Preset image	Custom image	
		Customize	•
★ Image			Select
* Code Directory 🕐			Select
★ Boot File ⑦			Select

Figure 7-4 Creating an algorithm using a preset image with customization

The process of this method is the same as that of creating a training job based on a preset image. For example:

- The system automatically injects environment variables.
  - PATH=\${MA\_HOME}/anaconda/bin:\${PATH}
  - LD\_LIBRARY\_PATH=\${MA\_HOME}/anaconda/lib:\${LD\_LIBRARY\_PATH}
  - PYTHONPATH=\${MA\_JOB\_DIR}:\${PYTHONPATH}
- The selected boot file will be automatically started using Python commands. Ensure that the Python environment is correct. The PATH environment variable is automatically injected. Run the following commands to check the Python version for the training job:

- export MA\_HOME=/home/ma-user; docker run --rm {image} \$ {MA\_HOME}/anaconda/bin/python -V
- docker run --rm {image} \$(which python) -V
- The system automatically adds hyperparameters associated with the preset image.

## Using a Custom Image

#### Figure 7-5 Creating an algorithm using a custom image

* Boot Mode	Preset image	Custom image	
* Image			Select
Code Directory			Select
* Boot Command ⑦	1		

For details about how to use custom images supported by the new-version training, see Using a Custom Image to Create a CPU- or GPU-based Training Job.

If all used images are customized, do as follows to use a specified Conda environment to start training:

Training jobs do not run in a shell. Therefore, you are not allowed to run the **conda activate** command to activate a specified Conda environment. In this case, use other methods to start training.

For example, Conda in your custom image is installed in the **/home/ma-user/ anaconda3** directory, the Conda environment is **python-3.7.10**, and the training script is stored in **/home/ma-user/modelarts/user-job-dir/code/train.py**. Use a specified Conda environment to start training in one of the following ways:

• Method 1: Configure the correct **DEFAULT\_CONDA\_ENV\_NAME** and **ANACONDA\_DIR** environment variables for the image.

Run the **python** command to start the training script. The following shows an example:

python /home/ma-user/modelarts/user-job-dir/code/train.py

- Method 2: Use the absolute path of Conda environment Python.
  - Run the **/home/ma-user/anaconda3/envs/python-3.7.10/bin/python** command to start the training script. The following shows an example: /home/ma-user/anaconda3/envs/python-3.7.10/bin/python /home/ma-user/modelarts/user-job-dir/ code/train.py
- Method 3: Configure the path environment variable.

Configure the bin directory of the specified Conda environment into the path environment variable. Run the **python** command to start the training script. The following shows an example:

export PATH=/home/ma-user/anaconda3/envs/python-3.7.10/bin:\$PATH; python /home/ma-user/ modelarts/user-job-dir/code/train.py

• Method 4: Run the **conda run -n** command.

Run the **/home/ma-user/anaconda3/bin/conda run -n python-3.7.10** command to execute the training. The following shows an example: /home/ma-user/anaconda3/bin/conda run -n python-3.7.10 python /home/ma-user/modelarts/userjob-dir/code/train.py

#### **NOTE**

If there is an error indicating that the .so file is unavailable in the **\$ANACONDA\_DIR/envs/ \$DEFAULT\_CONDA\_ENV\_NAME/lib** directory, add the directory to **LD\_LIBRARY\_PATH** and place the following command before the preceding boot command:

export LD\_LIBRARY\_PATH=\$ANACONDA\_DIR/envs/\$DEFAULT\_CONDA\_ENV\_NAME/ lib:\$LD\_LIBRARY\_PATH;

For example, the example boot command used in method 1 is as follows:

export LD\_LIBRARY\_PATH=\$ANACONDA\_DIR/envs/\$DEFAULT\_CONDA\_ENV\_NAME/ lib:\$LD\_LIBRARY\_PATH; python /home/ma-user/modelarts/user-job-dir/code/train.py

# 7.3.4 Viewing Algorithm Details

- 1. Log in to the ModelArts console.
- 2. In the navigation pane, choose **Algorithm Management**. The **My algorithm** page is displayed.
- 3. In the algorithm list, click the target algorithm name to go to the algorithm details page.
  - On the **Basic Information** tab, you can view the algorithm information.

Parameter	Description
Name	Algorithm name.
ID	Unique ID of an algorithm.
Description	Algorithm description. You can click the edit icon to update the description.
Preset image	Preset image and its version used by an algorithm. This parameter is available only for algorithms created using a preset image.
Custom image	Container image used by an algorithm. This parameter is available only for algorithms created using a custom engine version or custom image.
Code Directory	OBS directory for storing the algorithm code.
Boot File	OBS directory for storing the boot file.
Input	Input parameters of an algorithm.

 Table 7-6 Basic algorithm information

Parameter	Description
Output	Output parameters of an algorithm.
Hyperparamete r	Hyperparameter information of an algorithm.
Supported Policies	Auto search policy of an algorithm. If this parameter is left blank, auto search is not supported. Otherwise, auto search parameters are displayed.
Training Constraint	Training constraints of an algorithm. If <b>No</b> is displayed, there is no constraint. Otherwise, the supported resource types and training scenarios are displayed.

- On the **Training** tab, you can view the information about the training jobs that use the algorithm, such as the training job name and status.
- 4. On the **Basic Information** tab, click **Edit** to modify algorithm information except the name and ID. After the modification, click **Save**.

# 7.3.5 Searching for an Algorithm

ModelArts allows you to quickly search for algorithms by performing the following operations.

Operation 1: Search for jobs by name, image, code directory, description, and creation time.

Operation 2: Click the refresh button in the upper right corner to refresh the algorithm list.

Operation 3: Configure the custom columns and other basic settings.

Figure 7-6 Searching for an algorithm

My algorithm	My subscription	
Create		
I Specify filter c	riteria.	Q C 8

To sort algorithms in a column, click  $\ominus$  in the table header of the algorithm list.

# 7.3.6 Deleting an Algorithm

## **Deleting Your Algorithm**

Choose **Algorithm Management** > **My algorithm** and click **Delete** in the **Operation** column of the target algorithm. In the displayed dialog box, confirm the deletion.

# 7.4 Performing a Training

# 7.4.1 Creating a Training Job

Model training continuously iterates and optimizes model weights. ModelArts training management allows you to create training jobs, view training status, and manage training versions. Through model training, you can test various combinations of model structures, data, and hyperparameters to obtain the optimal model structure and weight.

## Prerequisites

- The data used for training has been uploaded to an OBS directory.
- At least one empty folder has been created in OBS for storing the training output.

#### **NOTE**

OBS buckets are not encrypted. ModelArts does not support encrypted OBS buckets. When creating an OBS bucket, do not enable bucket encryption.

- Access authorization has been configured. For details, see Configuring Access Authorization (Global Configuration).
- (Optional) An algorithm is available in Algorithm Management if you want to use it to create a training job. For details, see Introduction to Algorithm Preparation.
- (Optional) A custom image has been uploaded to SWR if you want to use it to create a training job. For details, see How Can I Log In to SWR and Upload Images to It?

#### **Operation Procedure**

To create a training job, follow these steps:

- Step 1 Access the page for creating a training job. For details, see Accessing the Page for Creating a Training Job .
- **Step 2** Configure basic information about the training job. For details, see **Configuring Basic Information About a Training Job**.
- **Step 3** Select an algorithm type for creating the training job.
  - Use a preset image to create a training job by referring to Choosing a Boot Mode (Preset Image).
  - Use a custom image to create a training job by referring to Choosing a Boot Mode (Custom Image).
  - Use an existing algorithm to create a training job by referring to **Choosing an** Algorithm Type (My Algorithm).
- **Step 4** Configure training parameters, including the input, output, hyperparameters, and environment variables. For details, see **Configuring Training Parameters**.
- **Step 5** Select a resource pool as required. A dedicated resource pool is recommended.
  - Configuring a Resource Pool (Public Resource Pool)
  - Configuring a Resource Pool (Dedicated Resource Pool)

- **Step 6** Select a training mode. For details, see **(Optional) Selecting a Training Mode**. When a MindSpore engine and Ascend resources are used for a training job, you can select the training mode.
- **Step 7** Perform follow-up procedure. For details, see **Follow-Up Procedure**.

----End

## Accessing the Page for Creating a Training Job

- 1. Log in to the ModelArts console.
- 2. In the navigation pane, choose **Training Management** > **Training Jobs**. The training job list is displayed.
- 3. Click Create Training Job. The Create Training Job page is displayed.

## **Configuring Basic Information About a Training Job**

On the **Create Training Job** page, set parameters.

Parameter	Description
Name	Name of a training job, which is mandatory.
	The system automatically generates a name. You can rename it based on the following naming rules:
	• The name contains 1 to 64 characters.
	<ul> <li>Letters, digits, hyphens (-), and underscores (_) are allowed.</li> </ul>
Description	Job description, which helps you learn about the job information in the training job list.
Experiment	Experiment for classifying and managing the job.
	<ul> <li>If you select Create new, enter the experiment name and description.</li> </ul>
	<ul> <li>If you select Use existing, select an experiment name.</li> </ul>
	<ul> <li>If you select Not required, this job will not be managed in any experiment.</li> </ul>

Table 7-7 Basic information for creating a training job

## Choosing a Boot Mode (Preset Image)

If you use a preset image to create a training job, select a boot mode by referring to **Table 7-8**.

Parameter	Description	
Algorithm Type	Select <b>Custom algorithm</b> . This parameter is mandatory.	
Boot Mode	Select <b>Preset image</b> and select the preset image engine and engine version to be used by the training job. If you select <b>Customize</b> for the engine version, select a	
	custom image from <b>Image</b> .	
Image	This parameter is displayed and mandatory only when the preset image version is set to <b>Customize</b> .	
	You can set the container image path in either of the following ways:	
	<ul> <li>To select your image or an image shared by others, click Select on the right and select a container image for training. The required image must be uploaded to SWR beforehand.</li> </ul>	
	<ul> <li>To select a public image, enter the address of the public image in SWR. Enter the image path in the format of "Organization name/Image name:Version name". Do not contain the domain name (swr.<region>.xxx.com) in the path because the system will automatically add the domain name to the path. For example, if the SWR address of a public image is swr.<region>.xxx.com/test-image/tensorflow2_1_1:1.1.1, enter test-images/tensorflow2_1_1:1.1.1.</region></region></li> </ul>	
Code Directory	Select the OBS directory where the training code file is stored. This parameter is mandatory.	
	• Upload code to the OBS bucket beforehand. The total size of files in the directory cannot exceed 5 GB, the number of files cannot exceed 1000, and the folder depth cannot exceed 32.	
	<ul> <li>The training code file is automatically downloaded to the \${MA_JOB_DIR}/demo-code directory of the training container when the training job is started. demo-code is the last-level OBS directory for storing the code. For example, if Code Directory is set to /test/code, the training code file is downloaded to the \${MA_JOB_DIR}/code directory of the training container.</li> </ul>	
Boot File	Select the Python boot script of the training job in the code directory. This parameter is mandatory.	
	ModelArts supports only the boot file written in Python. Therefore, the boot file must end with .py.	

Table 7-8 Creating a training job using a preset image

Parameter	Description
Local Code Directory	Specify the local directory of a training container. When a training starts, the system automatically downloads the code directory to this directory.
	The default local code directory is <b>/home/ma-user/</b> modelarts/user-job-dir. This parameter is optional.
Work Directory	During training, the system automatically runs the <b>cd</b> command to execute the boot file in this directory.

# Choosing a Boot Mode (Custom Image)

If you use a custom image to create a training job, select a boot mode by referring to **Table 7-9**.

Parameter	Description
Algorithm Type	Select <b>Custom algorithm</b> . This parameter is mandatory.
Boot Mode	Select <b>Custom image</b> . This parameter is mandatory.
Image	Container image path. This parameter is mandatory. You can set the container image path in either of the following ways:
	• To select your image or an image shared by others, click <b>Select</b> on the right and select a container image for training. The required image must be uploaded to SWR beforehand.
	• To select a public image, enter the address of the public image in SWR. Enter the image path in the format of "Organization name/Image name:Version name". Do not contain the domain name (swr. <region>.xxx.com) in the path because the system will automatically add the domain name to the path. For example, if the SWR address of a public image is swr.<region>.xxx.com/test-image/ tensorflow2_1_1:1.1.1, enter test-images/ tensorflow2_1_1:1.1.1.</region></region>

Table 7-9 Creating a training job using a custom image

Parameter	Description
Code Directory	Select the OBS directory where the training code file is stored. If the custom image does not contain training code, you need to set this parameter. If the custom image contains training code, you do not need to set this parameter.
	• Upload code to the OBS bucket beforehand. The total size of files in the directory cannot exceed 5 GB, the number of files cannot exceed 1000, and the folder depth cannot exceed 32.
	<ul> <li>The training code file is automatically downloaded to the \${MA_JOB_DIR}/demo-code directory of the training container when the training job is started. demo-code is the last-level OBS directory for storing the code. For example, if Code Directory is set to /test/code, the training code file is downloaded to the \${MA_JOB_DIR}/code directory of the training container.</li> </ul>
User ID	User ID for running the container. The default value 1000 is recommended.
	If the UID needs to be specified, its value must be within the specified range. The UID ranges of different resource pools are as follows:
	Public resource pool: 1000 to 65535
	Dedicated resource pool: 0 to 65535
Boot Command	Command for booting an image. This parameter is mandatory.
	When a training job is running, the boot command is automatically executed after the code directory is downloaded.
	• If the training boot script is a .py file, <b>train.py</b> for example, the boot command is as follows. python \${MA_JOB_DIR}/demo-code/train.py
	<ul> <li>If the training boot script is a .sh file, main.sh for example, the boot command is as follows.</li> <li>bash \${MA JOB DIR}/demo-code/main.sh</li> </ul>
	You can use semicolons (;) and ampersands (&&) to combine multiple commands. <b>demo-code</b> in the command is the last-level OBS directory where the code is stored. Replace it with the actual one.
Local Code Directory	Specify the local directory of a training container. When a training starts, the system automatically downloads the code directory to this directory.
	The default local code directory is <b>/home/ma-user/</b> modelarts/user-job-dir. This parameter is optional.

Parameter	Description
Work Directory	During training, the system automatically runs the <b>cd</b> command to execute the boot file in this directory.

# Choosing an Algorithm Type (My Algorithm)

Set **Algorithm Type** to **My algorithm** and select an algorithm from the algorithm list. If no algorithm meets the requirements, you can create an algorithm. For details, see **Creating an Algorithm**.

## **Configuring Training Parameters**

Data is obtained from an OBS bucket or dataset for model training. The training output is also stored in an OBS bucket. When creating a training job, you can configure parameters such as input, output, hyperparameters, and environment variables by referring to Table 7-10.

#### **NOTE**

The input, output, and hyperparameter parameters of a training job vary depending on the algorithm type selected during training job creation. If a parameter value is dimmed, the parameter has been configured in the algorithm code and cannot be modified.

Paramete r	Sub- Paramete r	Description
Input	Paramete r name	The algorithm code reads the training input data based on the input parameter name.
		The recommended value is <b>data_url</b> . The training input parameters must match the input parameters of the selected algorithm. For details, see <b>Table 7-3</b> .
	Dataset	Click <b>Dataset</b> and select the target dataset and its version in the ModelArts dataset list.
		When the training job is started, ModelArts automatically downloads the data in the input path to the training container.
		<b>NOTE</b> ModelArts data management is being upgraded and is invisible to users who have not used data management. It is recommended that new users store their training data in OBS buckets.

Paramete r	Sub- Paramete r	Description
	Data path	Click <b>Data path</b> and select the storage path to the training input data from an OBS bucket. When the training job is started, ModelArts automatically downloads the data in the input path to the training container.
	Obtained from	<ul><li>The following uses training input data_path as an example.</li><li>If you select Hyperparameters, use this code to</li></ul>
		obtain the data: import argparse parser = argparse.ArgumentParser() parser.add_argument('data_path') args, unknown = parser.parse_known_args() data_path = args.data_path
		<ul> <li>If you select Environment variables, use this code to obtain the data: import os data_path = os.getenv("data_path", "")</li> </ul>
Output	Paramete r name	The algorithm code reads the training output data based on the output parameter name. The recommended value is <b>train_url</b> . The training output parameters must match the output parameters of the selected algorithm. For details, see <b>Table 7-4</b> .
	Data path	Click <b>Data path</b> and select the storage path to the training output data from an OBS bucket. During training, the system automatically synchronizes files from the local code directory of the training container to the data path. <b>NOTE</b> The data path can only be an OBS path. To prevent any issues with data storage, choose an empty directory as the data
	Obtained from	<ul> <li>The following uses the training output train_url as an example.</li> <li>If you select Hyperparameters, use this code to</li> </ul>
		obtain the data: import argparse parser = argparse.ArgumentParser() parser.add_argument('train_url') args, unknown = parser.parse_known_args() train_url = args.train_url
		<ul> <li>If you select Environment variables, use this code to obtain the data: import os train_url = os.getenv("train_url", "")</li> </ul>

Paramete r	Sub- Paramete r	Description
	Predownl oad	Indicates whether to pre-download the files in the output directory to a local directory.
		• If you set <b>Predownload</b> to <b>No</b> , the system does not download the files in the training output data path to a local directory of the training container when the training job is started.
		<ul> <li>If you set Predownload to Yes, the system automatically downloads the files in the training output data path to a local directory of the training container when the training job is started. The larger the file size, the longer the download time. To avoid excessive training time, remove any unneeded files from the local code directory of the training container as soon as possible. If you want to use resumable training and incremental training, you must select Yes.</li> </ul>
Hyperpar ameter	N/A	Used for training tuning. This parameter is determined by the selected algorithm. If hyperparameters have been defined in the algorithm, all hyperparameters in the algorithm are displayed.
		Hyperparameters can be modified and deleted. The status depends on the hyperparameter constraint settings in the algorithm. For details, see <b>Defining Hyperparameters</b> .
Environm ent Variable	N/A	Add environment variables based on service requirements. For details about the environment variables preset in the training container, see Viewing Environment Variables of a Training Container.
Auto Restart	N/A	Number of retries for a failed training job. If this parameter is enabled, a failed training job will be automatically re-delivered and run. On the training job details page, you can view the number of retries for a failed training job.
		<ul> <li>This function is disabled by default.</li> <li>If you enable this function, set the number of retries. The value ranges from 1 to 3 and cannot be changed.</li> </ul>

# Configuring a Resource Pool (Public Resource Pool)

If you use a public resource pool to create a training job, configure the public resource pool by referring to **Table 7-11**.

Parameter	Description
Resource Pool	Select Public resource pool.
Resource Type	Select the resource type required for training. This parameter is mandatory. If a resource type has been defined in the training code, select a proper resource type based on algorithm constraints. For example, if the resource type defined in the training code is CPU and you select other types, the training fails. If some resource types are invisible or unavailable for selection, they are not supported.
Specifications	Select the required resource specifications based on the resource type.
	If <b>Data path</b> is selected for <b>Input</b> , you can click <b>Check</b> <b>Input Size</b> on the right to ensure the storage is larger than the input data size.
	* Specifications Specifications Ensure the storage is larger than the input data size.
	<b>NOTICE</b> The resource flavor <b>GPU</b> : <i>n</i> * <b>nvidia-t4</b> ( <i>n</i> indicates a specific number) does not support multi-process training.
Compute Nodes	Select the number of compute nodes as required. The default value is <b>1</b> .
	• If only one compute node is used, a single-node training job is created. ModelArts starts one training container on this node. The training container exclusively uses the compute resources of the selected flavor.
	• If more than one compute nodes are used, a distributed training job is created. For more information about distributed training configurations, see <b>Distributed Training</b> .
Persistent Log Saving	If you select CPU or GPU flavors, <b>Persistent Log</b> <b>Saving</b> is available for you to set.
	• This function is disabled by default. ModelArts automatically stores the logs for 30 days. You can download all logs on the job details page to a local path.
	• After this function is enabled, set <b>Job Log Path</b> . The system permanently stores training logs to the specified OBS path.

#### Table 7-11 Creating a public resource pool for training jobs

Parameter	Description
Job Log Path	When enabling <b>Persistent Log Saving</b> or selecting Ascend resources, select an empty OBS directory for <b>Job Log Path</b> to store log files generated by the training job. Ensure that you have read and write permissions to the selected OBS directory.
Event Notification	Indicates whether to enable event notification.
	• This function is disabled by default, which means SMN is disabled.
	• After this function is enabled, you will be notified of specific events, such as job status changes or suspected suspensions, via an SMS or email. Notifications will be billed based on SMN pricing. In this case, you must configure the topic name and events.
	<ul> <li>Topic: topic of event notifications. Click Create</li> <li>Topic to create a topic on the SMN console.</li> </ul>
	<ul> <li>Event: events you want to subscribe to. Examples: JobStarted, JobCompleted, JobFailed, JobTerminated, and JobHanged.</li> </ul>
	NOTE
	<ul> <li>After you create a topic on the SMN console, add a subscription to the topic, and confirm the subscription. Then, you will be notified of events.</li> </ul>
	• Only training jobs using GPUs support <b>JobHanged</b> events.
Auto Stop	<ul> <li>This function is disabled by default, the training job keeps running until the training is completed.</li> <li>If this function is enabled, configure the auto stop time. The value can be 1 hour, 2 hours, 4 hours, 6 hours, or Customize. The customized time must range from 1 hour to 720 hours. When you enable this function, the training stops automatically when the time limit is reached. The time limit does not count down when the training is paused.</li> </ul>

# Configuring a Resource Pool (Dedicated Resource Pool)

If you use a dedicated resource pool to create a training job, configure the dedicated resource pool by referring to **Table 7-12**.

_	
Parameter	Description
Resource Pool	Select a dedicated resource pool.
	If you select a dedicated resource pool, you can view the status, node specifications, number of idle/ fragmented nodes, number of available/total nodes, and number of cards of the resource pool. If the resource pool has available cards, hover over <b>View</b> in the <b>Idle/Fragmented Nodes</b> column to view fragment details and check whether the resource pool meets the training requirements.
Specifications	Select the required resource specifications based on the resource type.
	If <b>Data path</b> is selected for <b>Input</b> , you can click <b>Check</b> <b>Input Size</b> on the right to ensure the storage is larger than the input data size.
	* Specifications 32GB Check Input Size Ensure the storage is larger than the input data size.
	NOTICE The resource flavor GPU:n*nvidia-t4 (n indicates a specific number) does not support multi-process training.
Customized Specifications	Indicates whether to enable customized specifications. You can customize resource specifications for training jobs based on dedicated resource pool specifications to improve resource pool utilization.
	<ul> <li>This function is disabled by default, which means the dedicated resource pool specifications are used.</li> </ul>
	• When you enable this function, jobs run with custom specifications. The custom specifications should not exceed the node specifications of the dedicated resource pool that you set. For CPU specifications, you can only customize the number of vCPUs and memory. For GPU and Ascend specifications, you can customize the number of vCPUs, memory, and cards.
	<b>NOTE</b> If customized specifications are enabled, the <b>Specifications</b> parameter is invalid.

Table 7-12 Creating a dedicated resource pool for training jobs

Parameter	Description
Compute Nodes	Select the number of compute nodes as required. The default value is <b>1</b> .
	• If only one compute node is used, a single-node training job is created. ModelArts starts one training container on this node. The training container exclusively uses the compute resources of the selected flavor.
	<ul> <li>If more than one compute nodes are used, a distributed training job is created. For more information about distributed training configurations, see <b>Distributed Training</b>.</li> </ul>
Job Priority	When using a dedicated resource pool, you can set the priority of the training job. The value ranges from 1 to 3. The default priority is <b>1</b> , and the highest priority is <b>3</b> .
	• By default, the job priority can be set to <b>1</b> or <b>2</b> . After the permission to <b>set the highest job priority</b> is configured, the priority can be set to <b>1</b> to <b>3</b> .
	• If a training job is in the <b>Pending</b> state for a long time, you can change the job priority to reduce the queuing duration. For details, see <b>Priority of a Training Job</b> .

Parameter	Description
SFS Turbo	When ModelArts and SFS Turbo are directly connected, multiple SFS Turbo file systems can be mounted to a training job to store training data. Click <b>Add Mount</b> <b>Configuration</b> and set the following parameters:
	• File System: Select an SFS Turbo file system.
	• <b>Mount Path</b> : Enter the SFS Turbo mounting path in the training container.
	• <b>Storage Location</b> : Specify the SFS Turbo storage location. If you have configured the folder control permission, select a storage location. If you have not configured the folder control permission, retain the default value / or customize a location.
	• <b>Mounting Mode</b> : Permission on the mounted SFS Turbo file system. This parameter is displayed as <b>Read/Write</b> or <b>Read-only</b> based on the permission of the SFS Turbo storage location. If you have not configured the folder control permission, this parameter is unavailable.
	NOTE
	• A file system can be mounted only once and to only one path. Each mount path must be unique. A maximum of 8 disks can be mounted to a training job.
	<ul> <li>To mount an SFS Turbo file system to a training job, you need to configure network passthrough between ModelArts and the SFS Turbo file system. For details, see .</li> </ul>
	<ul> <li>The mounting path cannot be a / directory or a default mounting path, such as /cache and /home/ma-user/ modelarts.</li> </ul>
Persistent Log Saving	If you select CPU or GPU flavors, <b>Persistent Log</b> <b>Saving</b> is available for you to set.
	• This function is disabled by default. ModelArts automatically stores the logs for 30 days. You can download all logs on the job details page to a local path.
	• After this function is enabled, set <b>Job Log Path</b> . The system permanently stores training logs to the specified OBS path.
Job Log Path	When enabling <b>Persistent Log Saving</b> or selecting Ascend resources, select an empty OBS directory for <b>Job Log Path</b> to store log files generated by the training job.
	Ensure that you have read and write permissions to the selected OBS directory.

Parameter	Description
Event Notification	Indicates whether to enable event notification.
	• This function is disabled by default, which means SMN is disabled.
	• After this function is enabled, you will be notified of specific events, such as job status changes or suspected suspensions, via an SMS or email. Notifications will be billed based on SMN pricing. In this case, you must configure the topic name and events.
	<ul> <li>Topic: topic of event notifications. Click Create</li> <li>Topic to create a topic on the SMN console.</li> </ul>
	<ul> <li>Event: events you want to subscribe to. Examples: JobStarted, JobCompleted, JobFailed, JobTerminated, and JobHanged.</li> </ul>
	NOTE
	<ul> <li>After you create a topic on the SMN console, add a subscription to the topic, and confirm the subscription. Then, you will be notified of events.</li> </ul>
	• Only training jobs using GPUs support <b>JobHanged</b> events.
Auto Stop	• This function is disabled by default, the training job keeps running until the training is completed.
	• If this function is enabled, configure the auto stop time. The value can be <b>1 hour</b> , <b>2 hours</b> , <b>4 hours</b> , <b>6 hours</b> , or <b>Customize</b> . The customized time must range from 1 hour to 720 hours. When you enable this function, the training stops automatically when the time limit is reached. The time limit does not count down when the training is paused.

# (Optional) Selecting a Training Mode

When a MindSpore engine and Ascend resources are used for a training job, you can select the training mode. ModelArts provides three training modes for you to select. You can obtain different diagnosis information based on the actual scenario. For details, see **Selecting a Training Mode**.

- Common mode: It is the default training scenario.
- High performance mode: In this mode, certain O&M functions will be adjusted or even disabled to accelerate the running speed, but this will deteriorate fault locating. This mode is suitable for stable networks requiring high performance.
- Fault diagnosis mode: In this mode, certain O&M functions will be enabled or adjusted to collect more information for locating faults. This mode provides fault diagnosis. You can select a diagnosis type as required.

# Follow-Up Procedure

After parameter setting for creating a training job, click **Submit**. On the **Confirm** dialog box, click **OK**.

A training job runs for a period of time. You can go to the training job list to view the basic information about the training job.

- In the training job list, **Status** of a newly created training job is **Pending**.
- When the status of a training job changes to **Completed**, the training job is finished, and the generated model is stored in the corresponding output path.
- If the status is **Failed** or **Abnormal**, click the job name to go to the job details page and view logs for troubleshooting.

# 7.4.2 Viewing Training Job Details

- 1. Log in to the ModelArts management console.
- 2. In the navigation pane on the left, choose **Training Management** > **Training Jobs**.
- 3. In the training job list, click a job name to switch to the training job details page.
- 4. On the left of the training job details page, view basic job settings and algorithm parameters.
  - Basic job settings

Parameter	Description
Job ID	Unique ID of a training job
Status	Training job status
Created	Time when the training job is created
Duration	Running duration of a training job
Retries	Number of times that a training job automatically restarts upon a fault. This parameter is available only when <b>Auto Restart</b> is enabled during training job creation.
Description	Description of a training job. You can click the edit icon to update the description of a training job.

 Table 7-13 Basic job settings

Algorithm parameters

Parameter	Description
Name	algorithm used in a training job you can click the algorithm name to go to the algorithm details page.
Preset images	Preset image used by a training job
Code	OBS path to the code directory of a training job
Directory	You can click <b>Edit Code</b> on the right to edit the training script code in <b>OBS Online Editor</b> . <b>OBS Online</b> <b>Editor</b> is not available for a training job in the <b>Pending</b> , <b>Creating</b> , or <b>Running</b> status.
	OBS Online Editor ×
	Only .txt, py, and sh files can be edited online.     X
	Enter a job name. Q test-modelarts-xxx/pytorth/mnist-code/train.py
	Infer     Language Python v Interne Light v     Undo Save     Inform     Infer     Infer
	2 from _future_ import print_function
	5 ingert os 6 ingert çip 7 ingert codes
	8 import arguman 9 from typing import ID, Valon 10
	11 import mongo as ng 12 13 import torch
	14 import torch.nn as nn 15 import torch.nn (mutinal as F 16 import torch.sein as oth
	17 from terchrision inport datasets, transforms 18 from terch optim.lr_scheduler inport Step12 19
	20 import shatil 21 22
	23 ~ elsus Her(m. Models): 24 ~ definit(sif): 76 ~ (sif):
	NOTE If you use the algorithm subscribed in AI Hub to create a training job, then this parameter is not supported.
Boot File	Location where a boot file is stored.
	<b>NOTE</b> If you use the algorithm subscribed in AI Hub to create a training job, then this parameter is not supported.
User ID	ID of the user who runs the container.
Local Code Directory	Path to the training code in the training container
Work Directory	Path to the training startup file in the training container
Compute Nodes	Number of compute nodes
Dedicated resource pool	Dedicated resource pool information. This parameter is available only when a training job uses a dedicated resource pool.
Specifications	Training specifications used in a training job
Input - Input Path	OBS path where the input data is stored

#### Table 7-14 Algorithm parameters

Parameter	Description
lnput - Parameter Name	Algorithm code parameter specified by the input path
lnput - Obtained from	Method of obtaining the training job input.
Input - Local Path (Training Parameter Value)	Path for storing the input data in the ModelArts backend container. After the training is started, ModelArts downloads the data stored in OBS to the backend container.
Output - Output Path	OBS path where the output data is stored
Output - Parameter Name	Algorithm code parameter specified by the output path
Output - Obtained from	Method of obtaining the training job output.
Output - Local Path (Training Parameter Value)	Path for storing the output data in the ModelArts backend container
Hyperparamet er	Hyperparameters used in a training job
Environment Variable	Environment variables for a training job

# 7.4.3 Viewing Training Job Events

Any key event of a training job will be recorded at the backend after the training job is displayed for you. You can check events on the training job details page.

This helps you better understand the running process of a training job and locate faults more accurately when a task exception occurs. The following job events are supported:

- Training job created.
- Training job failures:
- Preparations timed out. The possible cause is that the cross-region algorithm synchronization or creating shared storage timed out.
- The training job is queuing and awaiting resource allocation.
- Failed to be queued.
- The training job starts to run.
- Training job executed.
- Failed to run the training job.
- The training job is preempted.
- The system detects that your training job may be suspended. Go to the job details page to view the cause and handle the issue.
- The training job has been restarted.
- The training job has been manually stopped.
- The training job has been stopped. (Maximum running duration: 1 hour)
- The training job has been stopped. (Maximum running duration: 3 hours)
- The training job has been manually deleted.
- Billing information synchronized.
- [worker-0] The training environment is being pre-checked.
- [worker-0] [Duration: second] Pre-check completed.
- [worker-0] [Duration: second] Pre-check failed. Error: xxx
- [worker-0] [Duration: second] Pre-check failed. Error: xxx
- [worker-0] The training code is being downloaded.
- [worker-0] [Duration: second] Training code downloaded.
- [worker-0] [Duration: second] Failed to download the training code. Failure cause:
- [worker-0] The training input is being downloaded.
- [worker-0] [Duration: second] Training input (parameter: xxx) downloaded.
- [worker-0] [Duration: second] Failed to download the training input (parameter: xxx). Failure cause:
- [worker-0] Python dependency packages are being installed. Import the following files:
- [worker-0] [Duration: second] Python dependency packages installed. Import the following files:
- [worker-0] The training job starts to run.
- [worker-0] Training job executed.
- [worker-0] The training input is being uploaded.
- [worker-0] [Duration: second] Training output (parameter: xxx) uploaded.

During the training process, key events can be manually or automatically refreshed.

#### Procedure

- 1. On the ModelArts console, choose **Training Management** > **Training Jobs** from the navigation pane.
- 2. In the training job list, click the name of the target job to go to the training job details page.
- 3. Click **Events** to view events.

## 7.4.4 Training Job Logs

## 7.4.4.1 Introduction to Training Job Logs

### Overview

Training logs record the runtime process and exception information of training jobs and provide useful details for fault location. The standard output and standard error information in your code are displayed in training logs. If you encounter an issue during the execution of a ModelArts training job, view logs first. In most scenarios, you can locate the issue based on the error information reported in logs.

Training logs include common training logs and Ascend logs.

- Common Logs: When resources other than Ascend are used for training, only common training logs are generated. Common logs include the logs for piprequirement.txt, training process, and ModelArts.
- Ascend Logs: When Ascend resources are used for training, device logs, plog logs, proc log for single-card training logs, MindSpore logs, and common logs are generated.

#### Figure 7-7 ModelArts training logs



#### 

Separate MindSpore logs are generated only in the MindSpore+Ascend training scenario. Logs of other AI engines are contained in common logs.

#### **Retention Period**

Logs are classified into the following types based on the retention period:

- Real-time logs: generated during training job running and can be viewed on the ModelArts training job details page.
- Historical logs: After a training job is completed, you can view its historical logs on the ModelArts training job details page. ModelArts automatically stores the logs for 30 days.

Permanent logs: These logs are dumped to your OBS bucket. When creating a training job, you can enable persistent log saving and set a job log path for dumping. For Ascend training, you need to configure the OBS path for storing training logs by default. You need to manually enable Persistent Log Saving for training jobs using other resources.

#### Figure 7-8 Enabling Persistent Log Saving

	Persistent Log Saving	
		Logs will be deleted after 30 days. Click this button to save logs to a specified OBS path or download all logs on the job detail page to a local PC.
*	Job Log Path	Select

Real-time logs and historical logs have no difference in content. In the Ascend training scenario, permanent logs contain Ascend logs, which are not displayed on ModelArts.

### **Related Chapters**

- On the ModelArts training job details page, you can preview logs, download logs, and search for logs by keyword in the log pane. For details, see Viewing Training Job Logs.
- ModelArts also enables you to quickly locate and rectify training faults. For details, see Locating Faults by Analyzing Training Logs.

### 7.4.4.2 Common Logs

Common logs include the logs for **pip-requirement.txt**, training process, and ModelArts.

#### Log Type

#### Table 7-15 Log type

Туре	Description
Training process log	Standard output of your training code
Installation logs for <b>pip-requirement.txt</b>	If <b>pip-requirement.txt</b> is defined in training code, PIP package installation logs are generated.
ModelArts logs	ModelArts logs are used by O&M personnel to locate service faults.

#### File Format

The format of a common log file is as follows. **task id** is the node ID of a training job.

Unified log format: modelarts-job-[job id]-[task id].log Example: log/modelarts-job-95f661bd-1527-41b8-971c-eca55e513254-worker-0.log

- Single-node training jobs generate a log file, and task id defaults to worker-0.
- Distributed training generates multiple node log files, which are distinguished by **task id**, such as **worker-0** and **worker-1**.

Common logs include the logs for **pip-requirement.txt**, training process, and ModelArts.

### **ModelArts Logs**

ModelArts logs can be filtered in the common log file **modelarts-job-[job id]-**[task id].log using the following keywords: [ModelArts Service Log] or Platform=ModelArts-Service.

- Type 1: [ModelArts Service Log] xxx [ModelArts Service Log][init] download code\_url: s3://dgg-test-user/snt9-test-cases/mindspore/lenet/
- Type 2: time="xxx" level="xxx" msg="xxx" file="xxx" Command=xxx Component=xxx Platform=xxx time="2021-07-26T19:24:11+08:00" level=info msg="start the periodic upload task, upload period = 5 seconds " file="upload.go:46" Command=obs/upload Component=ma-training-toolkit Platform=ModelArts-Service

## 7.4.4.3 Ascend Logs

#### Description

Ascend logs are generated when Ascend resources are used to for training. When Ascend resources are used for training, device logs, plog logs, proc logs for single-card training logs, MindSpore logs, and common logs are generated.

Common logs in the Ascend training scenario include the logs for **piprequirement.txt**, **ma-pre-start**, **davincirun**, training process, and ModelArts.

The following is an example of	the Ascend log structu	ure:
obs://dgg-test-user/snt9-test-cases/log-o modelarts-job-9ccf15f2-6610-42f9-a ascend process log	ut/ # b99-059ba049a41e	Job log path
rank_0	# Plog logs	
device-0	# device logs	
 — mindspore — modelarts-job-95f661bd-1527-41b8- — modelarts-job-95f661bd-1527-41b8- single-card training logs	# MindSp -971c-eca55e513254-worker- -971c-eca55e513254-proc-rar	ore logs 0.log           # Common logs nk-0-device-0.txt    # proc log for

Туре	Description	Name
device logs	User process AICPU and HCCP logs generated on the device and sent back to the host (training container) After the training process ends, the logs are uploaded to the ~/ascend/log/ directory of the ModelArts training container. If any of the following situations occur, device logs cannot be obtained: • The compute node restarts unexpectedly. • The compute node actively stops.	<pre>~/ascend/log/device-{device-id}/ device-{pid}_{timestamp}.log In the preceding command, pid indicates the user process ID on the host. Example log: device-166_20220718191853764.l og</pre>
plog logs	User process logs, for example, ACL/GE Plog logs are printed by default, and no log file is generated for <b>ascend/log</b> or related files.	<pre>~/ascend/log/plog/plog- {pid}_{timestamp}.log In the preceding command, pid indicates the user process ID on the host. Example log: plog-166_20220718191843620.log</pre>

### Table 7-16 Ascend log description

Туре	Description	Name
proc log	<b>proc log</b> is a redirection file of single-node training logs, helping you quickly obtain logs of a compute node.	<ul> <li>[modelarts-job-uuid]-proc-rank- [rank id]-device-[device logic id].txt</li> <li>device id indicates the ID of the NPU used in the training job. The value is 0 for a single NPU and 0 to 7 for eight NPUs. For example, if the Ascend specification is 8*Snt9, the value of device id ranges from 0 to 7. If the Ascend specification is 1*Snt9, the value of device id is 0.</li> <li>rank id indicates the global NPU ID of the training job. The value ranges from 0 to the number of compute nodes multiplied by the number of NPUs minus 1. If a single compute node is used, the value of rank id is the same as that of device id.</li> <li>Example log: modelarts- job-95f661bd-1527-41b8-971c- eca55e513254-proc-rank-0- device-0.txt</li> </ul>
MindSpore logs	Separate MindSpore logs are generated in the MindSpore +Ascend training scenario.	For details about MindSpore logs, visit the MindSpore official website.

Туре	Description	Name
Type Common training logs	<ul> <li>Description</li> <li>Logs for ma-pre-start (specific to Ascend training): If the ma-pre- start script is defined, the script execution log is generated.</li> <li>Logs for davincirun (specific to Ascend training): log generated when the Ascend training process is started using the davincirun.py file</li> <li>Training process logs: standard output of user training code</li> <li>Logs for pip- requirement.txt: If pip- requirement.txt is defined in training code. pip</li> </ul>	Name Contained in the modelarts-job- [job id]-[task id].log file. task id indicates the compute node ID. If a single node is used, the value is worker-0. If multiple nodes are used, the value is worker-0, worker-1,, or worker-{n-1}: n indicates the number of compute nodes. Example log: modelarts- job-95f661bd-1527-41b8-971c- eca55e513254-worker-0.log
in train packag are ge	in training code, pip package installation logs are generated.	
	<ul> <li>ModelArts logs: used by O&amp;M personnel to locate service faults.</li> </ul>	

#### Log Path

In the Ascend training scenario, after the training process exits, ModelArts uploads the log files in the training container to the OBS directory specified by **Job Log Path**.

### **Environment Variables Settings**

You can run the **ma-pre-start** script to modify the default environment variable configurations.

```
ASCEND_GLOBAL_LOG_LEVEL=3 # Log level, 0 for debug, 1 for info, 2 for warning, and 3 for error
ASCEND_SLOG_PRINT_TO_STDOUT=1 # Whether to display plog logs. The value 1 indicates that plog logs
are displayed by default.
ASCEND_GLOBAL_EVENT_ENABLE=1 # Event log level, 0 for disabling event logging and 1 for enabling
event logging
```

Place the **ma-pre-start.sh** or **ma-pre-start.py** script in the directory at the same level as the training boot file.

Before the training boot file is executed, the system executes the **ma-pre-start** script in **/home/work/user-job-dir/**. This method can be used to update the Ascend RUN package installed in the container image or set some additional global environment variables required for training.

## 7.4.4.4 Viewing Training Job Logs

On the training job details page, you can preview logs, download logs, search for logs by keyword, and filter system logs in the log pane.

• Previewing logs

You can preview training logs on the system log pane. If multiple compute nodes are used, you can choose the target node from the drop-down list on the right.

Figure 7-9 Viewing logs of different compute nodes



If a log file is oversized, the system displays only the latest logs in the log pane. To view all logs, click the link in the upper part of the log pane, which will direct you to a new page. Then you will be redirected to a new page.

#### Figure 7-10 Viewing all logs

server-0	▼ 🛃 ALL 🖏
Enter a keyword.	Q   Aa <u>Abi</u> _*   ∧ ∨

#### **NOTE**

- If the total size of all logs exceeds 500 MB, the log page may be frozen. In this case, download the logs to view them locally.
- A log preview link can be accessed by anyone within one hour after it is generated. You can share the link with others.
- Ensure that no privacy information is contained in the logs. Otherwise, information leakage may occur.
- Downloading logs

Training logs are retained for only 30 days. To permanently store logs, click the download icon in the upper right corner of the log pane. You can download the logs of multiple compute nodes in a batch. You can also enable **Persistent Log Saving** and set a log path when you create a training job. In this way, the logs will be automatically stored in the specified OBS path.

If a training job is created on Ascend compute nodes, certain system logs cannot be downloaded in the training log pane. To obtain these logs, go to the **Job Log Path** you set when you created the training job.

#### Figure 7-11 Downloading logs

server-0	▼ 上 ALL 🝾
Enter a keyword.	Q   Aa <u>Abi</u> _*   ^ ∨

• Searching for logs by keyword

In the upper right corner of the log pane, enter a keyword in the search box to search for logs.

The system will highlight the keyword and redirect you between search results. Only the logs loaded in the log pane can be searched for. If the logs are not fully displayed (see the message displayed on the page), obtain all the logs by downloading them or clicking the full log link and then search for the logs. On the page redirected by the full log link, press **Ctrl+F** to search for logs.

• Filtering system logs

#### Figure 7-12 System logs

Events	Logs	Resource Usages	Evaluation Results	Tags
Susta	m logs	Log file size: 1.00MB		
Syste	in togs			

If **System logs** is selected, system logs and user logs are displayed. If **System logs** is deselected, only user logs are displayed.

### 7.4.4.5 Locating Faults by Analyzing Training Logs

If you encounter an issue during the execution of a ModelArts training job, view logs first. In most scenarios, you can locate the issue based on the error information reported in logs.

If a training job fails, ModelArts automatically identifies the failure cause and displays a message on the log page. The message consists of possible causes, recommended solutions, and error logs (marked in red).

Figure 7-13 Identifying training faults



ModelArts provides possible causes (for reference only) and solutions for some common training faults. Not all faults can be identified. For a distributed job, only the analysis result of the current node is displayed. To obtain the failure cause of a training job, check the analysis results of all nodes used by the training job.

To rectify common training faults, perform the following steps:

- 1. Rectify the fault based on the analysis and suggestions provided on the log page.
  - Solution 1: A troubleshooting document is provided for you to follow.
  - Solution 2: Rebuild the training job and run it again.
- 2. If the fault persists, analyze the error information in the logs to locate and rectify the fault.
- 3. If the provided solutions cannot rectify your fault, you can submit a service ticket for technical support.

## 7.4.5 Cloud Shell

## 7.4.5.1 Logging In to a Training Container Using Cloud Shell

#### **Application Scenario**

You can use Cloud Shell provided by the ModelArts console to log in to a running training container.

#### Constraints

Only dedicated resource pools support Cloud Shell. The training job must be in the **Running** state.

## Preparation: Assigning the Cloud Shell Permission to an IAM User

- 1. Log in to the management console as a tenant user, hover the cursor over your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
- On the IAM console, choose Permissions > Policies/Roles from the navigation pane, click Create Custom Policy in the upper right corner, and configure the following parameters.
  - Policy Name: Enter a custom policy name, for example, Using Cloud Shell to access a running job.
  - Policy View: Select Visual editor.
  - **Policy Content**: Select **Allow**, **ModelArts Service**, **modelarts:trainJob:exec**, and default resources.

 Potck/Kdd / Create Custom Policies to supplement system-defined policies for fine-grained permissions management. Image in the system defined policies for fine-grained permissions management. Image in the system defined policies for fine-grained permissions management. Image in the system defined permissions management. Image is a process arunning permission with the service in the service in the description.

 Image: Image in the system defined policies of the system defined permissions

 Image: Image in the system defined policies of the system defined permissions

 Image: Image in the system defined policies of the system defined permissions

 Image: Image in the system defined policies of the system defined permissions

 Image: Image in the system defined policies of the system defined

Figure 7-14 Creating a custom policy

3. In the navigation pane, choose **User Groups**. Then, click **Authorize** in the **Operation** column of the target user group. On the **Authorize User Group** page, select the custom policies created in **2**, and click **Next**. Then, select the scope and click **OK**.

After the configuration, all users in the user group have the permission to use Cloud Shell to log in to a running training container.

If no user group is available, create a user group, add users using the user group management function, and configure authorization. If the target user is not in a user group, you can add the user to a user group through the user group management function.

### Using Cloud Shell

- 1. Configure parameters based on **Preparation: Assigning the Cloud Shell Permission to an IAM User**.
- 2. On the ModelArts console, choose **Training Management** > **Training Jobs** from the navigation pane.
- 3. In the training job list, click the name of the target job to go to the training job details page.
- 4. On the training job details page, click the **Cloud Shell** tab and log in to the training container.

Verify that the login is successful, as shown in the following figure.

#### Figure 7-15 Cloud Shell page



If the job is not running or the permission is insufficient, Cloud Shell cannot be used. In this case, locate the fault as prompted.

#### **NOTE**

An exception may occur when some users log in to the Cloud Shell page. Click **Enter** to rectify the fault.

Figure 7-16 Abnormal path

ind/model/1\$ @97c6-b87f-4410-9f74-18a8b1d0ff9d-59x451kz-6548f94565-lrjgs:/home/mi

### 7.4.5.2 Keeping a Training Job Running

You can only log in to Cloud Shell when the training job is in **Running** state. This section describes how to log in to a running training container through Cloud Shell.

#### Using the sleep Command

• For training jobs using a preset image

When creating a training job, set **Algorithm Type** to **Custom algorithm** and **Boot Mode** to **Preset image**, add **sleep.py** to the code directory, and use the script as the boot file. The training job keeps running for 60 minutes. You can access the container through Cloud Shell for debugging.

Example of sleep.py import os os.system('sleep 60m')

• For training jobs using a custom image

When creating a training job, set **Algorithm Type** to **Custom algorithm** and **Boot Mode** to **Custom image**, and enter **sleep 60m** in **Boot Command**. The training job keeps running for 60 minutes. You can access the container through Cloud Shell for debugging.

#### Keeping a Failed Job Running

When creating a training job, add **|| sleep 5h** at the end of the boot command and start the training job. Run the following command: cmd || sleep 5h

If the training fails, the **sleep** command is executed. In this case, you can log in to the container image through Cloud Shell for debugging.

#### D NOTE

To debug a multi-node training job in Cloud Shell, you need to switch between worker-0 and worker-1 in Cloud Shell and run the boot command on each node. Otherwise, the task will wait for other nodes to join.

### 7.4.5.3 Preventing Cloud Shell Session from Disconnection

To run a job for a long time, you can use the **screen** command to prevent the job from failing due to disconnection.

- 1. If screen is not installed in the image, run apt-get install screen to install it.
- 2. Create a screen terminal. # Use -S to create a screen terminal named name. screen -S name
- 3. View the created screen terminals.

screen -ls There are screens on: 2433.pts-3.linux (2013-10-20 16:48:59) (Detached) 2428.pts-3.linux (2013-10-20 16:48:05) (Detached) 2284.pts-3.linux (2013-10-20 16:14:55) (Detached) 2276.pts-3.linux (2013-10-20 16:13:18) (Detached) 4 Sockets in /var/run/screen/S-root.

- 4. Connect to the screen terminal whose screen\_id is 2276. screen -r 2276
- 5. Press **Ctrl+A+D** to exit the screen terminal. After the exit, the screen session is still active and can be reconnected at any time.

For details about how to use screens, see Screen User's Manual.

### 7.4.5.4 Analyzing the Call Stack of the Suspended Process Using the py-spy Tool and Locating the Suspended Problem By Analyzing Code

#### **Scenarios**

If a process is suspended, you can analyze the call stack of the process with the py-spy tool and locate the suspended problem by analyzing code.

#### Procedure

- Step 1 On the ModelArts console, choose Training Management > Training Jobs.
- **Step 2** Click the target training job to go to its details page. On the page that appears, click the **Cloud Shell** tab and log in to the training container (the training job must be in the **Running** state).
- **Step 3** Install the py-spy tool.

# Use the **utils.sh** script to automatically configure the Python environment. source /home/ma-user/modelarts/run/utils.sh

# Install py-spy.
pip install py-spy

# If the message "connection broken by 'ProxyError('Cannot connect to proxy.')" is displayed, disable the proxy. export no\_proxy=\$no\_proxy,repo.myhuaweicloud.com (Replace it with the pip source address of the corresponding site.)' pip install py-spy

# **Step 4** Check the stacks. For details about how to use the py-spy tool, see the **py-spy official document**.

# Find the PID of the training process.
ps -ef
# Check the process stack of process 12345.
# For a training job using eight cards, run the following command to check the stacks of the eight processes started by the main process in sequence.
py-spy dump --pid 12345

----End

## 7.4.6 Viewing the Resource Usage of a Training Job

### Operations

- 1. On the ModelArts console, choose **Training Management** > **Training Jobs** from the navigation pane.
- 2. In the training job list, click the name of the target job to go to the training job details page.
- 3. On the training job details page, click the **Resource Usages** tab to view the resource usage of the compute nodes. The data of at most the last three days can be displayed. When the resource usage window is opened, the data is loading and refreshed periodically.

Operation 1: If a training job uses multiple compute nodes, choose a node from the drop-down list box to view its metrics.

Operation 2: Click **cpuUsage**, **gpuMemUsage**, **gpuUtil**, **memUsage**, **npuMemUsage**, or **npuUtil** to show or hide the usage chart of the parameter. Operation 3: Hover the cursor on the graph to view the usage at the specific time.

Parameter	Description
cpuUsage	CPU usage
gpuMemUs age	GPU memory usage
gpuUtil	GPU usage
memUsage	Memory usage
npuMemUs age	NPU memory usage
npuUtil	NPU usage

Table 7-17 Parameters

### Alarms of Job Resource Usage

You can view the job resource usage on the training job list page. If the average GPU/NPU usage of the job's worker-0 instance is lower than 50%, an alarm is displayed in the training job list.



Figure 7-17 Job resource usage in the job list

The job resource usage here involves only GPU and NPU resources. The method of calculating the average GPU/NPU usage of a job's worker-0 instance is: Summarize the usage of each GPU/NPU accelerator card at each time point of the job's worker-0 instance and calculate the average value.

## Improving Job Resource Utilization

- Increasing the value of **batch\_size** increases GPU and NPU usage. You must decide the batch size that will not cause a memory overflow.
- If the time for reading data in a batch is longer than the time for GPUs or NPUs to calculate data in a batch, GPU or NPU usage may fluctuate. In this case, optimize the performance of data reading and data augmentation. For example, read data in parallel or use tools such as NVIDIA Data Loading Library (DALI) to improve the data augmentation speed.
- If a model is large and frequently saved, GPU or NPU usage is affected. In this case, do not save models frequently. Similarly, make sure that other non-GPU/NPU operations, such as log printing and training metric saving, do not affect the training process for too much time.

## 7.4.7 Evaluation Results

After a training job has been executed, ModelArts evaluates your model and provides optimization diagnosis and suggestions.

- When you use a built-in algorithm to create a training job, you can view the evaluation result without any configurations. The system automatically provides optimization suggestions based on your model metrics. Read the suggestions and guidance on the page carefully to further optimize your model.
- For a training job created by writing a training script or using a custom image, you need to add the evaluation code to the training code so that you can view the evaluation result and diagnosis suggestions after the training job is complete.

#### **NOTE**

- Only validation sets of the image type are supported.
- You can add the evaluation code only when the training scripts of the following frequently-used frameworks are used:
  - TF-1.13.1-python3.6
  - TF-2.1.0-python3.6
  - PyTorch-1.4.0-python3.6

This section describes how to use the evaluation code in a training job. To adapt and modify the training code, three steps are involved, Adding the Output Path, Copying the Dataset to the Local Host, and Mapping the Dataset Path to OBS.

#### Adding the Output Path

The code for adding the output path is simple. That is, add a path for storing the evaluation result file to the code, which is called **train\_url**, that is, the training output path on the console. Add **train\_url** to the analysis function and use **save\_path** to obtain **train\_url**. The sample code is as follows:

```
FLAGS = tf.app.flags.FLAGS
tf.app.flags.DEFINE_string('model_url', ", 'path to saved model')
tf.app.flags.DEFINE_string('data_url', ", 'path to output files')
tf.app.flags.DEFINE_string('adv_param_json',
                                '{"attack_method":"FGSM","eps":40}',
                               'params for adversarial attacks')
FLAGS(sys.argv, known_only=True)
...
# analyse
res = analyse(
    task_type=task_type,
    pred_list=pred_list,
    label_list=file_name_list,
    label_map_dict=label_dict,
    save_path=FLAGS.train_url)
```

### Copying the Dataset to the Local Host

Copying a dataset to the local host is to prevent the OBS connection from being interrupted due to long-time access. Therefore, copy the dataset to the local host before performing operations.

There are two methods for copying datasets. The recommended method is to use the OBS path.

• OBS path (recommended)

Call the copy\_parallel API of MoXing to copy the corresponding OBS path.

Dataset in ModelArts data management (manifest file format)

Call the copy\_manifest API of MoXing to copy the file to the local host and obtain the path of the new manifest file. Then, use SDK to parse the new manifest file.

#### **NOTE**

ModelArts data management is being upgraded and is invisible to users who have not used data management. It is recommended that new users store their training data in OBS buckets.

```
if data_path.startswith('obs://'):
```

```
if '.manifest' in data_path:
    new_manifest_path, _ = mox.file.copy_manifest(data_path, '/cache/data/')
    data_path = new_manifest_path
else:
    mox.file.copy_parallel(data_path, '/cache/data/')
    data_path = '/cache/data/'
print('------ download dataset success ------')
```

#### Mapping the Dataset Path to OBS

The actual path of the image file, that is, the OBS path, needs to be entered in the JSON body. Therefore, after analysis and evaluation are performed on the local host, the original local dataset path needs to be mapped to the OBS path, and the new list needs to be sent to the analysis API.

If the OBS path is used as the input of **data\_url**, you only need to replace the string of the local path.

```
if FLAGS.data_url.startswith('obs://'):
    for idx, item in enumerate(file_name_list):
        file_name_list[idx] = item.replace(data_path, FLAGS.data_url)
```

If the manifest file is used, the original manifest file needs to be parsed again to obtain the list and then the list is sent to the analysis API.

```
if or FLAGS.data_url.startswith('obs://'):
    if 'manifest' in FLAGS.data_url:
        file_name_list = []
        manifest, _ = get_sample_list(
        manifest_path=FLAGS.data_url, task_type='image_classification')
        for item in manifest:
            if len(item[1]) != 0:
                file_name_list.append(item[0])
```

An example code for image classification that can be used to create training jobs is as follows:

```
import json
import logging
import os
import sys
import tempfile
import h5py
import numpy as np
from PIL import Image
import moxing as mox
import tensorflow as tf
from deep_moxing.framework.manifest_api.manifest_api import get_sample_list
from deep_moxing.model_analysis.api import analyse, tmp_save
from deep_moxing.model_analysis.common.constant import TMP_FILE_NAME
logging.basicConfig(level=logging.DEBUG)
FLAGS = tf.app.flags.FLAGS
tf.app.flags.DEFINE_string('model_url', '', 'path to saved model')
tf.app.flags.DEFINE_string('data_url', '', 'path to output files')
tf.app.flags.DEFINE_string('train_url', '', 'path to output files')
tf.app.flags.DEFINE_string('adv_param_json',
                     '{"attack_method":"FGSM","eps":40}',
                     'params for adversarial attacks')
FLAGS(sys.argv, known_only=True)
def _preprocess(data_path):
   img = Image.open(data_path)
   img = img.convert('RGB')
   img = np.asarray(img, dtype=np.float32)
   img = img[np.newaxis, :, :, :]
   return img
def softmax(x):
 x = np.array(x)
```

```
orig_shape = x.shape
  if len(x.shape) > 1:
     # Matrix
     x = np.apply\_along\_axis(lambda x: np.exp(x - np.max(x)), 1, x)
     denominator = np.apply_along_axis(lambda x: 1.0 / np.sum(x), 1, x)
     if len(denominator.shape) == 1:
        denominator = denominator.reshape((denominator.shape[0], 1))
     x = x * denominator
  else:
     # Vector
     x_max = np.max(x)
     x = x - x_max
     numerator = np.exp(x)
     denominator = 1.0 / np.sum(numerator)
     x = numerator.dot(denominator)
  assert x.shape == orig_shape
  return x
def get_dataset(data_path, label_map_dict):
  label list = []
  img_name_list = []
  if 'manifest' in data_path:
     manifest, _ = get_sample_list(
        manifest_path=data_path, task_type='image_classification')
     for item in manifest:
        if len(item[1]) != 0:
          label_list.append(label_map_dict.get(item[1][0]))
          img_name_list.append(item[0])
        else:
          continue
  else:
     label_name_list = os.listdir(data_path)
     label_dict = {}
     for idx, item in enumerate(label_name_list):
        label_dict[str(idx)] = item
        sub_img_list = os.listdir(os.path.join(data_path, item))
        img_name_list += [
          os.path.join(data_path, item, img_name) for img_name in sub_img_list
        label_list += [label_map_dict.get(item)] * len(sub_img_list)
  return img_name_list, label_list
def deal_ckpt_and_data_with_obs():
  pb dir = FLAGS.model url
  data_path = FLAGS.data_url
  if pb_dir.startswith('obs://'):
     mox.file.copy_parallel(pb_dir, '/cache/ckpt/')
     pb_dir = '/cache/ckpt'
     print('----- download success ------')
  if data_path.startswith('obs://'):
     if '.manifest' in data_path:
        new_manifest_path, _ = mox.file.copy_manifest(data_path, '/cache/data/')
        data_path = new_manifest_path
     else:
        mox.file.copy_parallel(data_path, '/cache/data/')
        data_path = '/cache/data/'
     print('----- download dataset success ------')
  assert os.path.isdir(pb_dir), 'Error, pb_dir must be a directory'
  return pb_dir, data_path
def evalution():
  pb_dir, data_path = deal_ckpt_and_data_with_obs()
  index_file = os.path.join(pb_dir, 'index')
  try:
     label_file = h5py.File(index_file, 'r')
```

```
label_array = label_file['labels_list'][:].tolist()
  label_array = [item.decode('utf-8') for item in label_array]
except Exception as e:
  logging.warning(e)
  logging.warning ('index file is not a h5 file, try json.')
  with open(index_file, 'r') as load_f:
     label_file = json.load(load_f)
   label_array = label_file['labels_list'][:]
label_map_dict = {}
label_dict = {}
for idx, item in enumerate(label_array):
   label_map_dict[item] = idx
  label_dict[idx] = item
print(label_map_dict)
print(label_dict)
data_file_list, label_list = get_dataset(data_path, label_map_dict)
assert len(label_list) > 0, 'missing valid data'
assert None not in label list, 'dataset and model not match'
pred_list = []
file_name_list = []
img_list = []
for img_path in data_file_list:
  img = _preprocess(img_path)
  img_list.append(img)
  file_name_list.append(img_path)
config = tf.ConfigProto()
config.gpu_options.allow_growth = True
config.gpu_options.visible_device_list = '0'
with tf.Session(graph=tf.Graph(), config=config) as sess:
  meta_graph_def = tf.saved_model.loader.load(
     sess, [tf.saved_model.tag_constants.SERVING], pb_dir)
  signature = meta_graph_def.signature_def
  signature_key = 'predict_object'
  input_key = 'images'
  output_key = 'logits'
  x_tensor_name = signature[signature_key].inputs[input_key].name
  y_tensor_name = signature[signature_key].outputs[output_key].name
  x = sess.graph.get_tensor_by_name(x_tensor_name)
  y = sess.graph.get_tensor_by_name(y_tensor_name)
  for img in img_list:
     pred output = sess.run([y], {x: img})
     pred_output = softmax(pred_output[0])
     pred_list.append(pred_output[0].tolist())
label_dict = json.dumps(label_dict)
task_type = 'image_classification'
if FLAGS.data_url.startswith('obs://'):
  if 'manifest' in FLAGS.data url:
     file_name_list = []
     manifest, _ = get_sample_list(
        manifest_path=FLAGS.data_url, task_type='image_classification')
     for item in manifest:
        if len(item[1]) != 0:
           file_name_list.append(item[0])
  for idx, item in enumerate(file_name_list):
     file_name_list[idx] = item.replace(data_path, FLAGS.data_url)
# analyse
res = analyse(
  task_type=task_type,
   pred_list=pred_list,
   label_list=label_list,
  name_list=file_name_list,
  label_map_dict=label_dict,
```

```
save_path=FLAGS.train_url)
if __name__ == "__main__":
evalution()
```

## 7.4.8 Viewing Fault Recovery Details

When a training job fault occurs (such as process-level recovery, POD-level rescheduling, and job-level rescheduling), the **Fault Recovery Details** tab appears on the job details page, recording the start and stop details of the training job.

- 1. On the ModelArts console, choose **Training Management** > **Training Jobs** from the navigation pane.
- 2. In the training job list, click the name of the target job to go to the training job details page.
- 3. On the training job details page, click the **Fault Recovery Details** tab to view the fault recovery information.

## 7.4.9 Viewing Environment Variables of a Training Container

### What Is an Environment Variable

This section describes environment variables preset in a training container. The environment variables include:

- Path environment variables
- Environment variables of a distributed training job
- Nvidia Collective multi-GPU Communication Library (NCCL) environment variables
- OBS environment variables
- Environment variables of the PIP source
- Environment variables of the API Gateway address
- Environment variables of job metadata

### **Configuring Environment Variables**

When you create a training job, you can add environment variables or modify environment variables preset in the training container.

#### **Environment Variables Preset in a Training Container**

The following tables list environment variables preset in a training container, including Table 7-18, Table 7-19, Table 7-20, Table 7-21, Table 7-22, Table 7-23, and Table 7-24.

The environment variable values are examples.

Table 7-18 Path	environment	variables
-----------------	-------------	-----------

Variable	Description	Example
PATH	Executable file paths	PATH=/usr/local/nvidia/bin:/usr/ local/cuda/bin:/usr/local/ sbin:/usr/local/bin:/usr/ sbin:/usr/bin:/sbin:/bin
LD_LIBRARY_P ATH	Dynamic load library paths	LD_LIBRARY_PATH=/usr/local/ seccomponent/lib:/usr/local/ cuda/lib64:/usr/local/cuda/ compat:/root/miniconda3/ lib:/usr/local/nvidia/lib:/usr/ local/nvidia/lib64
LIBRARY_PATH	Static library paths	LIBRARY_PATH=/usr/local/cuda/ lib64/stubs
MA_HOME	Main directory of a training job	MA_HOME=/home/ma-user
MA_JOB_DIR	Parent directory of the training algorithm folder	MA_JOB_DIR=/home/ma-user/ modelarts/user-job-dir
MA_MOUNT_P ATH	Path mounted to a ModelArts training container, which is used to temporarily store training algorithms, algorithm input, algorithm output, and logs	MA_MOUNT_PATH=/home/ma- user/modelarts
MA_LOG_DIR	Training log directory	MA_LOG_DIR=/home/ma-user/ modelarts/log
MA_SCRIPT_IN TERPRETER	Training script interpreter	MA_SCRIPT_INTERPRETER=
WORKSPACE	Training algorithm directory	WORKSPACE=/home/ma-user/ modelarts/user-job-dir/code

Table 7-19 Environment variables of a distributed training jo	зb
---	----

Variable	Description	Example
MA_CURRENT_ IP	IP address of a job container.	MA_CURRENT_IP=192.168.23.38
MA_NUM_GPU S	Number of accelerator cards in a job container.	MA_NUM_GPUS=8

Variable	e Description Example					
MA_TASK_NAM EName of a job container, for example:• worker in MindSpore and PyTorch.• learner or worker in reinforcement learning engines.• ps or worker in TensorFlow.		MA_TASK_NAME=worker				
MA_NUM_HOS TS	Compute nodes required for a training job.	MA_NUM_HOSTS=4				
VC_TASK_INDE X	Sequence number of a job container for multi- node training. The value of the first container is <b>0</b> .	VC_TASK_INDEX=0				
VC_WORKER_N UM	Compute nodes required for a training job.	VC_WORKER_NUM=4				
VC_WORKER_H OSTS	Domain name of each node for multi-node training. Use commas (,) to separate the domain names in sequence. You can obtain the IP address through domain name resolution.	VC_WORKER_HOSTS=modelarts -job- a0978141-1712-4f9b-8a83-0000 0000000-worker-0.modelarts- job- a0978141-1712-4f9b-8a83-0000 0000000,modelarts-job- a0978141-1712-4f9b-8a83-0000 0000000-worker-1.ob- a0978141-1712-4f9b-8a83-0000 0000000,modelarts-job- a0978141-1712-4f9b-8a83-0000 0000000-worker-2.modelarts- job- a0978141-1712-4f9b-8a83-0000 0000000,ob- a0978141-1712-4f9b-8a83-0000 0000000,ob- a0978141-1712-4f9b-8a83-0000 0000000-worker-3.modelarts- job- a0978141-1712-4f9b-8a83-0000 00000000-worker-3.modelarts- job-				

#### Table 7-20 NCCL environment variables

Variable Description		Example
NCCL_VERSION	NCCL version	NCCL_VERSION=2.7.8

Variable	Description Example		
NCCL_DEBUG	NCCL log level	NCCL_DEBUG=INFO	
NCCL_IB_HCA	InfiniBand NIC to use for communication	NCCL_IB_HCA=^mlx5_bond_0	
NCCL_SOCKET_ IFNAME	IP interface to use for communication	NCCL_SOCKET_IFNAME=bond0, eth0	

 Table 7-21 OBS environment variables

Variable	Description	Example
S3_ENDPOINT	OBS endpoint	S3_ENDPOINT=https:// obs.region.xxx.com
S3_VERIFY_SSL	Whether to use SSL to access OBS	S3_VERIFY_SSL=0
S3_USE_HTTPS	Whether to use HTTPS to access OBS	S3_USE_HTTPS=1

 Table 7-22
 Environment variables of the PIP source and API Gateway address

Variable	Description	Example
MA_PIP_HOST	Domain name of the PIP source	MA_PIP_HOST=repo.xxx.com
MA_PIP_URL	Address of the PIP source	MA_PIP_URL=http:// repo.xxx.com/repository/pypi/ simple/
MA_APIGW_EN DPOINT	ModelArts API Gateway address	MA_APIGW_ENDPOINT=https:/ /modelarts.region.xxx.xxx.com

|--|

Variable	Description	Example
MA_CURRENT_I NSTANCE_NAM E	Name of the current node for multi-node training	MA_CURRENT_INSTANCE_NAM E=modelarts-job- a0978141-1712-4f9b-8a83-000 000000000-worker-1

Variable	Description	Example
MA_SKIP_IMAGE _DETECT	Whether to enable ModelArts precheck. The default value is <b>1</b> , which indicates that the pre- check is enabled; the value <b>0</b> indicates that the pre-check is disabled.	1
	It is a good practice to enable precheck to detect node and driver faults before they affect services.	

	<b>Table 7-24</b>	Precheck	environment	variables
--	-------------------	----------	-------------	-----------

## 7.4.10 Stopping, Rebuilding, or Searching for a Training Job

### Saving As an Algorithm

To modify the algorithm of a training job, click **Save As Algorithm** in the upper right corner of the training job details page.

On the **Algorithms** page, the algorithm parameters for the last training job are automatically set. You can modify the settings.

#### **NOTE**

This function is not supported for algorithms subscribed in AI Hub.

#### **Stopping a Training Job**

In the training job list, click **Stop** in the **Operation** column of a training job that is in creating, pending, or running state to stop the job.

A training job in completed, failed, terminated, or abnormal state cannot be stopped.

#### Rebuilding a Training Job

If you are not satisfied with a created training job, click **Rebuild** in the **Operation** column to rebuild it. The page for creating a training job is displayed. On this page, the parameter settings for the previous training job are automatically retained. You only need to modify certain parameter settings.

#### Searching for a Training Job

If you log in to ModelArts using an IAM account, all training jobs under this account are displayed in the training job list. To quickly search for a training job, use the following methods:

Method 1: Click **Only my jobs**. Then, only jobs created under the current IAM user account are displayed in the training job list.

Method 2: Search for jobs by name, ID, job type, status, creation time, algorithm, and resource pool.

Method 3: Click the refresh button in the upper right corner of the job list to refresh it.

Method 4: Configure the custom columns and other basic settings.

Figure 7-18 Searching for a training job



## 7.4.11 Releasing Training Job Resources

Release resources of a training job when not in use.

- On the **Training Jobs** page, click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the training job.
- Go to OBS and delete the OBS bucket and files used by the training job.

After the resources are released, check the resource usage on the **Dashboard** page.

Figure 7-19 Checking the resource usage

Usage										
Te	xt Classificatio	n - ExeML	Image Classific	ation - ExeML	Training Jobs-	Beta New - Trai	AI Applicatio	n Management	Real-Time Se	rvices - Service
Ru O	inning	Projects 1	Running O	Projects 1	Running 0	Projects 23	AI App 20	AI Versions 20	Running 0	Services 1

## 7.5 Training Experiment

Users Details

## 7.5.1 Introduction to Experiment

An experiment is a job management capability provided by ModelArts. You can add training jobs to experiments for management.

Manage training jobs in an experiment by referring to the following instructions:

- For details about how to add a training job to an experiment, see Adding a Training Job to an Experiment.
- For details about how to view experiment information, see Viewing an Experiment.
- For details about how to delete an experiment, see **Deleting an Experiment**.

## 7.5.2 Adding a Training Job to an Experiment

To add a training job to an experiment, configure **Experiment** when creating a training job. The options are as follows:

- **Create new**: An experiment can only be created when you create a training job. If you select this option, enter a new experiment name. After the job is submitted, the experiment is created and the job is added to the new experiment. The experiment name will be checked. If the name is already in use, the job cannot be submitted.
- **Use existing**: Select an existing experiment from the drop-down list box to add the job to the existing experiment.
- **Not required**: Select this option if you do not want to manage your job through an experiment. The experiment tab page of Training Management does not display a job that has not been added to an experiment.

#### Creating a Job to Be Added to an Experiment

Log in to the ModelArts console, choose **Training Management** > **Training Jobs**, and click **Create Training Job** in the upper right corner.

On this page, configure **Experiment**. If you keep the default setting **Create new**, enter a name for the new experiment. Then, an experiment is created after you create the training job.

#### Figure 7-20 Creating a training job

Experiment	Create new	Use existing	Not required
* Experiment Name			
Description			Å
			0/256

#### Adding a Created Job to an Experiment

Log in to the ModelArts console, choose **Training Management** > **Training Jobs**, and click **Rebuild** in the **Operation** column of the target job. Alternatively, click the job name or ID in the job list. On the job details page, click **Rebuild** in the upper right corner.

- For a job that has not been added to an experiment, select **Create new** by default and enter a name for the new experiment. Then, an experiment is created after you create the training job.
- For a job that has been added to an experiment, select **Use existing** by default and select the experiment where the source job is.

#### Figure 7-21 Rebuilding a training job

Experiment	Create new	Use existing	Not required
Experiment Name			•

## 7.5.3 Viewing an Experiment

### Viewing the Experiment List

- 1. Log in to the ModelArts console. In the left navigation pane, choose **Training Management** > **Training Jobs**. The **Training Jobs** page is displayed.
- 2. Click **Experiments** to go to the **Experiments** tab page. The experiment list displays some basic experiment information.

Parameter	Description	
Experiment Name	Experiment name, which can be changed on the experiment details page	
Training Jobs	Number of training jobs in an experiment	
Created	Time when an experiment is created	
Modified At	<ul> <li>Time when any of the following occurs:</li> <li>Changing the experiment name</li> <li>Modifying the description of the experiment</li> <li>Adding a training job to or deleting a training job from the experiment</li> </ul>	
Description	Experiment description, which can be modified	
Operation	You can delete the experiment.	

Table 7-25 Basic experiment information

- You can search for experiments by experiment name, number of training jobs, creation time, modification time, and description.
- You can click the refresh button in the upper right corner of the job list to refresh the job list.
- You can click the setting button in the upper right corner of the experiment list to select items you want to display in the experiment list.
- You can click the arrow in the table header to sort experiments.

### **Viewing Experiment Details**

In the experiment list, click an experiment name to go to the experiment details page. Basic experiment information is displayed in the upper part of the experiment details page, and the job list of the experiment is displayed in the lower part of the experiment details page.

#### Figure 7-22 Viewing experiment details

Training / Experiments (j	ob-train-test)						Create	Training Job     Delete Experiment
Experiment Name	job-train-test 🖉	Training Jobs 0	Created Jun 01, 202	3 11:20:40 GMT+08:00	Modified At Jun 02	2023 10:12:21 GMT+08:00	Description	L
▼ Search by job	name by default.							Only my jobs C @
Name/ID	Job Type 🖓	Status 🖓	Created On 7	Algorithm	Resource Pool	Description	Created By	Operation

- You can click  $\swarrow$  to edit the name and description of an experiment.
- You can click **Only my jobs** to view the jobs that you have created and included in the experiment.

#### **NOTE**

By default, if an account has multiple IAM users, only the jobs of the current IAM user is displayed.

- You can search for jobs by name, ID, algorithm, status, creation time, job type, or resource pool.
- You can click the refresh button in the upper right corner of the job list to refresh the job list.
- You can click the setting button in the upper right corner of the job list to select items you want to display in the job list.

## 7.5.4 Deleting an Experiment

You can click **Delete** on the experiment list page or click **Delete Experiment** in the upper right corner of the experiment details page to delete an experiment. All jobs of the experiment are displayed on the **Delete Experiment** page. Enter **DELETE** and click **OK** to confirm the deletion.

#### 

After an experiment is deleted, all jobs in the experiment will be deleted accordingly and cannot be restored. Therefore, exercise cautions when performing this operation.

## 7.6 Advanced Training Operations

## 7.6.1 Selecting a Training Mode

If a MindSpore engine and Ascend resources are used for a training job, ModelArts provides three training modes: common mode, high-performance mode, and fault diagnosis mode. You can obtain different diagnosis information based on application scenarios.

### **Training Modes**

By default, a training job is in general mode. For details about debugging information in general mode, see **Training Job Logs**.

- High performance mode: In this mode, certain O&M functions will be adjusted or even disabled to accelerate the running speed, but this will deteriorate fault locating. This mode is suitable for stable networks requiring high performance.
- Fault diagnosis mode: In this mode, certain O&M functions will be enabled or adjusted to collect more information for locating faults. This mode provides fault diagnosis. You can select a diagnosis type as required.

 Table 7-26 details debugging information obtained in each mode.

Debugging Information	Gener al	High perfor manc e	Fault diagn osis	Description
MindSpore log levels	Info level	Error level	Info level	MindSpore framework runtime log
Running Data Recorder (RDR)	Disabl ed	Disabl ed	Enabl ed	If a running exception occurs, the recorded MindSpore data is automatically exported to help locate the exception cause. Different data is exported for different exceptions. For details about RDR, see MindSpore Documentation.
analyze_fail.dat	Enabled by default and uploaded to the training job log path		ult and path	Graph build failure information is automatically exported for inference process analysis.
Dump data	Enabled by default and uploaded to the training job log path		ult and path	Dump data is exported when an exception occurs during backend running.

**Table 7-26** Debugging information obtained in each mode

In the fault diagnosis mode, after the fault diagnosis function is enabled, you can view the following fault diagnosis data: The following data is stored in the OBS directory in the training log path.

Description of the training output log file in the fault diagnosis mode:

```
{obs-log-path}/
modelarts-job-{job-id}-worker-{index}.log # Displayed log summary
modelarts-job-{job-id}-proc-rank-{rank-id}-device-{device-id}.txt # Logs of each device are displayed.
modelarts-job-{job-id}/
ascend/
npu_collect/rank_{id}/ # Output path for TFAdapter DUMP GRAPH and GE DUMP GRAPH,
generated only for the TensorFlow framework
process_log/rank_{id}/ # Plog log path
msnpureport/{task-index}/ # msnpureport tool execution logs, which you do not need to pay
attention to
mindspore/
log/ # MindSpore framework logs and MindSpore fault diagnosis data
```

Category	Description
CANN framework logs and fault diagnosis data	Host logs of the INFO or higher levels, including CANN software stack logs and driver logs.
MindSpore	MindSpore framework logs of the INFO or higher levels.
framework logs and fault diagnosis data	RDR file.
	If a running exception occurs, the recorded MindSpore data is automatically exported to help locate the exception cause. Different data is exported for different exceptions.
	analyze_fail.dat. Graph build failure information is automatically exported for inference process analysis.
	Dump data, which is generated when an exception arises during backend operations.

#### Table 7-27 Fault diagnosis data of MindSpore

## Procedure

On the training job creation page, select a MindSpore engine and Ascend resources, and then choose a training mode.

#### Figure 7-23 Selecting an algorithm

* Created By	Custom algorithms	My algorithms	My subscriptions
* Boot Mode	Preset images	Custom images	
	Ascend-Powered-Engine	▼ mindspore_1.7.0-c	ann_5.1.0-py 🔻

#### Figure 7-24 Selecting a resource type

* Resource Pool	Public resource pool	Dedicated resource pool
* Resource Type	Ascend	
* Specifications	Ascend: 1* )(32GB	i)   ARM: 24 vCPUs 96GB 320 ▼

#### Figure 7-25 Enabling fault diagnosis

Training Mode	General	High performance	Fault diagnosis	
	In fault diagnosis r	node, certain O&M functions	will be enabled or adjuste	ed to collect more information for fault locating.

## 7.6.2 Automatic Recovery from a Training Fault

## 7.6.2.1 Training Fault Tolerance Check

During model training, a training failure may occur due to a hardware fault. For hardware faults, ModelArts provides fault tolerance check to isolate faulty nodes to improve user experience in training.

The fault tolerance check involves environment pre-check and periodic hardware check. If any fault is detected during either of the checks, ModelArts automatically isolates the faulty hardware and issues the training job again. In distributed training, the fault tolerance check will be performed on all compute nodes used by the training job.

The following shows four failure scenarios, among which the failure in scenario 4 is not caused by a hardware fault. You can enable fault tolerance in the other three scenarios to automatically resume the training job.

 Scenario 1: The environment pre-check fails, and the hardware is faulty. Then, ModelArts automatically isolates all faulty nodes and issues the training job again.



Figure 7-26 Pre-check failure and hardware fault

• Scenario 2: The environment pre-check fails but the hardware is functional. Then, ModelArts automatically isolates all faulty nodes and issues the training job again.



Figure 7-27 Pre-check failure but functional hardware

• Scenario 3: The environment pre-check is successful and the user service starts. A hardware fault occurs and the user service exits unexpectedly. Then, ModelArts automatically isolates all faulty nodes and issues the training job again.

Figure	7-28	Service	failure	and	hardware	fault
--------	------	---------	---------	-----	----------	-------



• Scenario 4: The environment pre-check is successful and the user service starts. The hardware is functional. A fault occurs in the user service, the training job ends in the failure state.



Figure 7-29 Service failure and functional hardware

After the faulty node is isolated, ModelArts creates a training job on new compute nodes. If the resources provided by the resource pool are limited, the re-issued training job will be queued with the highest priority. If the waiting time exceeds 30 minutes, the training job will automatically exit. This indicates that the resources are so limited that the training job cannot start. In this case, buy a dedicated resource pool to obtain dedicated resources.

If you use a dedicated resource pool to create a training job, the faulty nodes identified during the fault tolerance check will be removed. The system automatically adds healthy compute nodes to the dedicated resource pool. (This function is coming soon.)

More details of a fault tolerance check:

- 1. Enabling Fault Tolerance Check
- 2. Check Items and Conditions
- 3. Effect of a Fault Tolerance Check
- 4. After the environment pre-check is successful, any hardware fault will interrupt the user service. Add the reload ckpt code logic to the training so that the pre-trained model saved before the training is interrupted can be obtained. For details, see **Resumable Training and Incremental Training**.

#### **Enabling Fault Tolerance Check**

To enable fault tolerance check, enable auto restart when creating a training job.

• Configure fault tolerance check on the ModelArts management console:

Enable **Auto Restart** on the ModelArts management console. **Auto Restart** is disabled by default, indicating that the job will not be re-issued and the environment pre-check will not be enabled. After **Auto Restart** is enabled, the number of restart retries ranges from 1 to 3.

Configure fault tolerance check using an API:

Enable auto restart upon a fault using an API. When creating a training job, configure the **fault-tolerance/job-retry-num** field in **annotations** of the **metadata** field.

If the **fault-tolerance/job-retry-num** field is added, auto restart is enabled. The value can be an integer ranging from **1** to **3**. specifying the maximum number of times that a job can be re-issued. If this hyperparameter is not specified, the default value **0** is used, indicating that the job will not be reissued and the environment pre-check will not be enabled.

Figure 7-30 Setting the API

```
{
    wind": "job",
    metadata": 2
    wind": **
    metadata": 2
    wind": **
    wind": **
```

## **Check Items and Conditions**

Check Item	ltem (Log Keywor d)	Execution Condition	Requirements for a Check	
Domain name detection	dns	None	The domain names of the volcano containers in the .host file in <b>/etc/volcano</b> are successfully resolved.	
Disk size - Container root directory	disk-size root	None	The directory is greater than 32 GB.	
Disk size - /dev/shm	disk-size shm	None	The directory is greater than 1 GB.	
Disk size - / cache	disk-size cache	None	The directory is greater than 32 GB.	
ulimit check	ulimit	An IB network is used.	<ul> <li>Maximum locked memory &gt; 16000</li> <li>Open files &gt; 1000000</li> <li>Stack size &gt; 8000</li> <li>Maximum user processes &gt; 1000000</li> </ul>	
GPU check	gpu- check	GPU and the v2 training engine are used.	GPUs are detected.	

## Effect of a Fault Tolerance Check

• If the fault tolerance check is passed, the logs of the check items will be recorded, indicating that the check items are successful. You can search for

the keyword **item** in the log file. A fault tolerance check minimizes reported runtime faults.

```
[Modelarts Service Log][task] Detect
[Modelarts Service Log][INFO][detect] code: 0, message: ok, item: dns
[Modelarts Service Log][INFO][detect] code: 0, message: ok, item: disk-size root
[Modelarts Service Log][INFO][detect] code: 0, message: ok, item: disk-size shm
[Modelarts Service Log][INFO][detect] code: 0, message: ok, item: disk-size cache
[Modelarts Service Log][INFO][detect] code: 0, message: ok, item: disk-size cache
```

• If a fault tolerance check fails, check failure logs will be recorded. You can search for the keyword **item** in the log file to view the failure information.

```
[Modelarts Service Log][init] running
[Modelarts Service Log][init] ip of the pod: 172.16.0.160
[Modelarts Service Log][INFO]detect item: dns; json:{"code": 0, "message": "ok"}
[Modelarts Service Log][INFO][task][detect] code: 0, message: ok, item: dns
[Modelarts Service Log][INFO][task][detect] code: 0, message: i, "important disk space of the path
/"/" is 4892852224, which is less than 34359738366"}
[Modelarts Service Log][ERROR][task][detect] code: 13, message: the disk space of the path "/" is 4892852224, which is
less than 34359738366, item: disk-size root
[Modelarts Service Log][init] exiting...
[Modelarts Service Log][init] wait python processes exit...
```

If the number of job restarts does not reach the specified time, the job will be automatically issued again. You can search for keywords **error, exiting** to obtain the logs recording a restarted job that ends with a failure.

### Using reload ckpt to Resume an Interrupted Training

With fault tolerance enabled, if a training job is restarted due to a hardware fault, you can obtain the pre-trained model in the code to restore the training to the state before the restart. To do so, add reload ckpt to the code. For details, see **Resumable Training and Incremental Training**.

#### 7.6.2.2 Fault Dying Gasp

#### **Application Scenarios**

The sharp increase of model volumes and datasets requires a large-scale training set for training a large-scale neural network. During distributed training in a large-scale cluster, a chip or server in the cluster may be faulty. As a result, the distributed training job fails. A dying gasp indicates that an interrupted training job can be automatically resumed from the breakpoint where the previous training was interrupted.

#### Constraints

Resource Flavor	Ascend
Training Framewo rk	MindSpore

Table 7-28 Constraints

## **Working Principles**



The process of a dying gasp is as follows:

- 1. Create a training job on the ModelArts console.
- 2. ModelArts creates a training container and starts the training script.
- 3. After the training script is started, MindSpore is called to generate the hybrid parallel policy file **strategy.proto**. This file records the distribution of operators on NPUs in the hybrid parallel scenario.
- 4. After a training fault occurs, the ModelArts training component sends the SIGTERM signal to the affected service process.
- 5. The training script receives the SIGTERM signal and calls the elastic-agent module. This module then calls MindSpore to generate the dying gasp CKPT.
- 6. ModelArts restarts the training container and starts the training script.
- 7. The training script calls the elastic-agent module. This module generates a restoration policy file based on the NPU failure information in **configmap** and the **strategy.proto** file.
- 8. The training script loads the dying gasp CKPT to resume the training based on the restoration policy file.

In data parallel scenarios, the process is similar. The only difference is that the parallel policy file and restoration policy file are not required. You only need to save and load the dying gasp CKPT file.

### Procedure

#### 1. Install the binary dying gasp package.

Use ma\_pre\_start.sh to install the .whl package.

echo "[ma-pre-start] Enter the input directory" cd /home/ma-user/modelarts/inputs/data\_url\_0/ echo "[ma-pre-start] Start to install mindx-elastic 0.0.1" export PATH=/home/ma-user/anaconda/bin:\$PATH pip install ./mindx\_elastic-0.0.1-py3-none-any.whl echo "[ma-pre-start] Clean run package" sudo rm -rf ./script ./\*.run ./run\_package \*.whl echo "[ma-pre-start] Set ENV" export GLOG\_v=2 # If the diagnosis mode is used, set the log level to INFO. echo "[ma-pre-start] End"
#### 2. Create a training job.

- Ensure that MindSpore is 1.6.0 or later.
  - Add the following content to the sample code: # Load the dependency API. from mindx\_elastic.terminating\_message import ExceptionCheckpoint if args opt.do train: dataset = create\_dataset() loss cb = LossMonitor() $cb = [loss_cb]$ if int(os.getenv('RANK\_ID')) == 0: batch\_num = dataset.get\_dataset\_size() # Enable dying gasp saving. config\_ck = CheckpointConfig(save\_checkpoint\_steps=batch\_num, keep\_checkpoint\_max=35, async\_save=True, append\_info=[{"epoch\_num": cur\_epoch\_num}], exception\_save=True) ckpoint\_cb = ModelCheckpoint(prefix="train\_resnet\_cifar10", directory=args\_opt.train\_url,
    - config=config\_ck)
      # Define callback for dying gasp CKPT saving.
      ckpoint\_exp = ExceptionCheckpoint(
      prefix="train\_resnet\_cifar10",
      directory=args\_opt.train\_url,
      config=config\_ck)
      # Add callback for dying gasp CKPT saving.
      cb += [ckpoint\_cb, ckpoint\_exp]

# 7.6.3 Resumable Training and Incremental Training

#### **Overview**

Resumable training indicates that an interrupted training job can be automatically resumed from the checkpoint where the previous training was interrupted. This method is applicable to model training that takes a long time.

Incremental training is a method in which input data is continuously used to extend the existing model's knowledge to further train the model.

Checkpoints are used to resume model training or incrementally train a model.

During model training, training results (including but not limited to epochs, model weights, optimizer status, and scheduler status) are continuously saved. In this way, an interrupted training job can be automatically resumed from the checkpoint where the previous training was interrupted.

To resume a training job, load a checkpoint and use the checkpoint information to initialize the training status. To do so, add reload ckpt to the code.

#### **Resumable Training and Incremental Training in ModelArts**

To resume model training or incrementally train a model in ModelArts, configure **Training Output**.

When creating a training job, set the data path to the training output, save checkpoints in this data path, and set **Predownload** to **Yes**. If you set **Predownload** to **Yes**, the system automatically downloads the **checkpoint** file in

the training output data path to a local directory of the training container before the training job is started.

Enable fault tolerance check (auto restart) for resumable training. On the training job creation page, enable **Auto Restart**. If the environment pre-check fails, the hardware is not functional, or the training job fails, ModelArts will automatically issue the training job again.

#### reload ckpt for MindSpore

```
import os
import argparse
parser.add_argument("--train_url", type=str)
args = parser.parse_known_args()
# train_url is set to /home/ma-user/modelarts/outputs/train_url_0.
train_url = args.train_url
# Initially defined network, loss function, and optimizer
net = resnet50(args_opt.batch_size, args_opt.num_classes)
ls = SoftmaxCrossEntropyWithLogits(sparse=True, reduction="mean")
opt = Momentum(filter(lambda x: x.requires_grad, net.get_parameters()), 0.01, 0.9)
# Initial epoch value for the first training. The initial value of epoch_size will be customized in MindSpore
1.3 and later versions.
# cur_epoch_num = 0
# Check whether there is a model file in the OBS output path. If there is no file, the model will be trained
from the beginning by default. If there is a model file, the CKPT file with the maximum epoch value will be
loaded as the pre-trained model.
if os.listdir(train_url):
  last_ckpt = sorted([file for file in os.listdir(train_url) if file.endswith(".ckpt")])[-1]
  print('last_ckpt:', last_ckpt)
  last_ckpt_file = os.path.join(train_url, last_ckpt)
  # Load the checkpoint.
  param_dict = load_checkpoint(last_ckpt_file)
  print('> load last ckpt and continue training!!')
  # Load model parameters to the network.
  load param into net(net, param dict)
  # Load model parameters to the optimizer.
  load_param_into_net(opt, param_dict)
  # Obtain the saved epoch value. The model will continue to be trained based on the epoch value. This
function will be supported in MindSpore 1.3 and later versions.
  # if param_dict.get("epoch_num"):
       cur epoch num = int(param dict["epoch num"].data.asnumpy())
model = Model(net, loss_fn=ls, optimizer=opt, metrics={'acc'})
# as for train, users could use model.train
if args_opt.do_train:
  dataset = create_dataset()
  batch_num = dataset.get_dataset_size()
  config_ck = CheckpointConfig(save_checkpoint_steps=batch_num,
                         keep_checkpoint_max=35)
  # For append_info=[{"epoch_num": cur_epoch_num}], append_info will be supported in MindSpore
1.3 and later versions to save the epoch value at the current time.
  ckpoint_cb = ModelCheckpoint(prefix="train_resnet_cifar10",
                         directory=args_opt.train_url,
                         config=config_ck)
  loss_cb = LossMonitor()
  model.train(epoch_size, dataset, callbacks=[ckpoint_cb, loss_cb])
  # For model.train(epoch_size-cur_epoch_num, dataset, callbacks=[ckpoint_cb, loss_cb]), the training
resumed from the breakpoint will be supported in MindSpore 1.3 and later versions.
```

# 7.6.4 Detecting Training Job Suspension

#### **Overview**

A training job may be suspended due to unknown reasons. If the suspension cannot be detected promptly, resources cannot be released, leading to a waste. To

minimize resource cost and improve user experience, ModelArts provides suspension detection for training jobs. With this function, suspension can be automatically detected and displayed on the log details page. You can also enable notification so that you can be promptly notified of job suspension.

#### **Detection Rules**

Determine whether a job is suspended based on the monitored job process status and resource usage. A process is started to periodically monitor the changes of the two metrics.

- Job process status: If the process I/O of a training job changes, the next detection period starts. If the process I/O of the job remains unchanged in multiple detection periods, the resource usage detection starts.
- Resource usage: If the process I/O remains unchanged, the system collects the GPU usage within a certain period of time and determines whether the resource usage changes based on the variance and median of the GPU usage within the period. If the GPU usage is not changed, the job is suspended.

#### Constraints

Suspension can be detected only for training jobs that run on GPUs.

#### Procedure

Suspension detection is automatically performed during job running. No additional configuration is required. After detecting that a job is suspended, the system displays a message on the training job details page, indicating that the job may be suspended. If you want to be notified of suspension (by SMS or email), enable event notification on the job creation page.

#### Cases

Common cases and solutions to training job suspension are as follows:

**Data Replication Suspension** 

**Suspension Before Training** 

**Suspension During Training** 

Suspension in the Last Training Epoch

# 7.6.5 Priority of a Training Job

When using a new-version dedicated resource pool for training jobs, you can set the job priority when creating a training job or adjust the priority when a job is in the **Pending** state for a long time. By adjusting the job priority, you can reduce the job queuing duration.

#### Overview

Some training tasks, such as unimportant tests or experiments, are of low priority. In this case, you need to prioritize training tasks (jobs). A task with a higher priority is queued earlier than a task with a lower priority. You can adjust the job execution sequence by configuring the priority of training jobs to ensure normal running of important services at peak hours.

#### Constraints

- You can set the priority of a training job only if it is created using a new-version dedicated resource pool.
- The value ranges from 1 to 3. The default priority is **1**, and the highest priority is **3**. By default, the job priority can be set to **1** or **2**. After the permission to **set the highest job priority** is configured, the priority can be set to **1** to **3**.

#### **Configuring the Priority**

Set the priority when you create a training job. The value ranges from 1 to 3. The default priority is **1**, and the highest priority is **3**.

#### **Changing the Priority**

On the **Training Jobs** page, locate a training job in the **Pending** state and click

in the **Job Priority** column. In the dialog box that appears, change the priority and click **OK**.

# 7.6.6 Permission to Set the Highest Job Priority

You can configure the priority when you create a training job using a new-version dedicated resource pool. You can change the priority of a pending job. The value ranges from 1 to 3. The default priority is **1**, and the highest priority is **3**. By default, the job priority can be set to **1** or **2**. After the permission to set the highest job priority is configured, the priority can be set to **1** to **3**.

#### Assigning the Permission to Set the Highest Job Priority to an IAM User

- 1. Log in to the management console as a tenant user, hover the cursor over your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
- 2. On the IAM console, choose **Permissions** > **Policies/Roles** from the navigation pane, click **Create Custom Policy** in the upper right corner, and configure the following parameters.
  - Policy Name: Enter a custom policy name, for example, Allowing Users to Set the Highest Job Priority.
  - Policy View: Select Visual editor.
  - Policy Content: Select Allow, ModelArts Service, modelarts:trainJob:setHighPriority, and default resources.

	sual editor	JSON			•
·	Allow		ModelArts Service	5 L Actions: 1	D All
	Select all	nodelarts:trair	nJob:setHighPriority		
Sele	ct Existing Policy/	Role 🕂	Add Permissions		
Sele	ect Existing Policy/ ermission allows y	Role $(+)$	Add Permissions highest job priority.		

3. In the navigation pane, choose **User Groups**. Then, click **Authorize** in the **Operation** column of the target user group. On the **Authorize User Group** page, select the custom policies created in **2**, and click **Next**. Then, select the scope and click **OK**.

After the configuration, all users in the user group have the permission to use Cloud Shell to log in to a running training container.

If no user group is available, create a user group, add users using the user group management function, and configure authorization. If the target user is not in a user group, you can add the user to a user group through the user group management function.

# 7.7 Distributed Training

# 7.7.1 Distributed Training

ModelArts provides the following capabilities:

• Extensive built-in images, meeting your requirements

Figure 7-31 Creating a custom policy

- Custom development environments set up using built-in images
- Extensive tutorials, helping you quickly understand distributed training
- Distributed training debugging in development tools such as PyCharm, VS Code, and JupyterLab

#### Constraints

- The development environment refers to the new-version Notebook provided by ModelArts, excluding the old-version Notebook.
- If the notebook instance flavors are changed, you can only perform singlenode debugging. You cannot perform distributed debugging or submit remote training jobs.

- Only the PyTorch and MindSpore AI frameworks can be used for multi-node distributed debugging. If you want to use MindSpore, each node must be equipped with eight cards.
- The OBS paths in the debugging code should be replaced with your OBS paths.
- PyTorch is used to write debugging code in this document. The process is the same for different AI frameworks. You only need to modify some parameters.

#### **Related Chapters**

- **Single-Node Multi-Card Training Using DataParallel**: describes single-node multi-card training using DataParallel, and corresponding code modifications.
- Multi-Node Multi-Card Training Using DistributedDataParallel : describes multi-node multi-card training using DistributedDataParallel, and corresponding code modifications.
- **Distributed Debugging Adaptation and Code Example**: describes the procedure and code example of distributed debugging adaptation.
- Sample Code of Distributed Training: provides a complete code sample of distributed parallel training for the classification task of ResNet18 on the CIFAR-10 dataset.

# 7.7.2 Single-Node Multi-Card Training Using DataParallel

This section describes how to perform single-node multi-card parallel training based on the PyTorch engine.

For details about the distributed training using the MindSpore engine, see **the MindSpore official website**.

#### **Training Process**

The process of single-node multi-card parallel training is as follows:

- 1. A model is copied to multiple GPUs.
- 2. Data of each batch is distributed evenly to each worker GPU.
- 3. Each GPU does its own forward propagation and an output is obtained.
- 4. The master GPU with device ID 0 collects the output of each GPU and calculates the loss.
- 5. The master GPU distributes the loss to each worker GPU. Each GPU does its own backward propagation and calculates the gradient.
- 6. The master GPU collects gradients, updates parameter settings, and distributes the settings to each worker GPU.

The detailed flowchart is as follows.



#### Figure 7-32 Single-node multi-card parallel training

#### Advantages and Disadvantages

- Straightforward coding: Only one line of code needs to be modified.
- Bottlenecks in communication: The master GPU is used to update and distribute parameter settings, which causes high communication costs.
- Unbalanced GPU loading: The master GPU is used to summarize outputs, calculate loss, and update weights. Therefore, the GPU memory and usage are higher than those of other GPUs.

#### **Code Modifications**

Model distribution: DataParallel(model)

The code is slightly changed and the following is a simple example:

```
import torch
class Net(torch.nn.Module):
    pass
model = Net().cuda()
### DataParallel Begin ###
model = torch.nn.DataParallel(Net().cuda())
### DataParallel End ###
```

# 7.7.3 Multi-Node Multi-Card Training Using DistributedDataParallel

This section describes how to perform multi-node multi-card parallel training based on the PyTorch engine.

#### **Training Process**

Compared with DataParallel, DistributedDataParallel can start multiple processes for computing, greatly improving compute resource usage. Based on **torch.distributed**, DistributedDataParallel has obvious advantages over DataParallel in the distributed computing case. The process is as follows:

- 1. Initializes the process group.
- 2. Creates a distributed parallel model. Each process has the same model and parameters.
- 3. Creates a distributed sampler for data distribution to enable each process to load a unique subset of the original dataset in a mini batch.
- 4. Parameters are organized into buckets based on their shapes or sizes, which are generally determined by each layer of the network that requires parameter update in a neural network model.
- 5. Each process does its own forward propagation and computes its gradient.
- 6. After all parameter gradients at a bucket are obtained, communication is performed for gradient averaging.
- 7. Each GPU updates model parameters.

The detailed flowchart is as follows.

Figure 7-33 Multi-node multi-card parallel training



#### Advantages

- Fast communication
- Balanced load

• Fast running speed

#### **Code Modifications**

- Multi-process startup
- New variables such as rank ID and world\_size are used along with the TCP protocol.
- Sampler for data distribution to avoid duplicate data between different processes
- Model distribution: DistributedDataParallel(model)
- Model saved in GPU 0

import torch class Net(torch.nn.Module): pass

model = Net().cuda()

### DataParallel Begin ###
model = torch.nn.DataParallel(Net().cuda())
### DataParallel End ###

#### **Related Operations**

- For details about distributed debugging adaptation and code example, see **Distributed Debugging Adaptation and Code Example**.
- This document also provides a complete code sample of distributed parallel training for the classification task of ResNet18 on the cifar10 dataset. For details, see **Sample Code of Distributed Training**.

# 7.7.4 Distributed Debugging Adaptation and Code Example

In DistributedDataParallel, each process loads a subset of the original dataset in a batch, and finally the gradients of all processes are averaged as the final gradient. Due to a large number of samples, a calculated gradient is more reliable, and a learning rate can be increased.

This section describes the code of single-node training and distributed parallel training for the classification job of ResNet18 on the CIFAR-10 dataset. Directly execute the code to perform multi-node distributed training with CPUs or GPUs; comment out the distributed training settings in the code to perform single-node single-card training.

The training code contains three input parameters: basic training parameters, distributed parameters, and data parameters. The distributed parameters are automatically input by the platform. **custom\_data** indicates whether to use custom data for training. If this parameter is set to **true**, torch-based random data is used for training and validation.

#### Dataset

#### CIFAR-10 dataset

In notebook instances, torchvision of the default version cannot be used to obtain datasets. Therefore, the sample code provides three training data loading methods.

Click **CIFAR-10 python version** on the **download page** to download the CIFAR-10 dataset.

- Download the CIFAR-10 dataset using torchvision.
- Download the CIFAR-10 dataset based on the URL and decompress the dataset in a specified directory. The sizes of the training set and test set are (50000, 3, 32, 32) and (10000, 3, 32, 32), respectively.
- Use Torch to obtain a random dataset similar to CIFAR-10. The sizes of the training set and test set are (5000, 3, 32, 32) and (1000, 3, 32, 32), respectively. The labels are still of 10 types. Set **custom\_data** to **true**, and the training task can be directly executed without loading data.

#### **Training Code**

In the following code, those commented with *###* Settings for distributed training and ... *###* are code modifications for multi-node distributed training.

Do not modify the sample code. After the data path is changed to your path, multi-node distributed training can be executed on ModelArts.

After the distributed code modifications are commented out, the single-node single-card training can be executed. For details about the complete code, see **Sample Code of Distributed Training**.

#### • Importing dependency packages

import datetime import inspect import os import pickle import random

import argparse import numpy as np import torch import torch.distributed as dist from torch.distributed as dist from torch.utils.data import TensorDataset, DataLoader from torch.utils.data.distributed import DistributedSampler from sklearn.metrics import accuracy\_score

• **Defining the method and random number for loading data** (The code for loading data is not described here due to its large amount.)

def setup\_seed(seed): torch.manual\_seed(seed) torch.cuda.manual\_seed\_all(seed) np.random.seed(seed) random.seed(seed) torch.backends.cudnn.deterministic = True

def get\_data(path): pass

#### • Defining a network structure

class Block(nn.Module):

```
def __init__(self, in_channels, out_channels, stride=1):
    super().__init__()
    self.residual_function = nn.Sequential(
        nn.Conv2d(in_channels, out_channels, kernel_size=3, stride=stride, padding=1, bias=False),
        nn.BatchNorm2d(out_channels),
        nn.ReLU(inplace=True),
        nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1, bias=False),
        nn.BatchNorm2d(out_channels, kernel_size=3, padding=1, bias=False),
        nn.BatchNorm2d(out_channels, kernel_size=3, padding=1, bias=False),
        nn.BatchNorm2d(out_channels)
```

```
)
     self.shortcut = nn.Sequential()
     if stride != 1 or in_channels != out_channels:
        self.shortcut = nn.Sequential(
           nn.Conv2d(in_channels, out_channels, kernel_size=1, stride=stride, bias=False),
           nn.BatchNorm2d(out_channels)
        )
  def forward(self, x):
     out = self.residual_function(x) + self.shortcut(x)
     return nn.ReLU(inplace=True)(out)
class ResNet(nn.Module):
  def __init__(self, block, num_classes=10):
     super().__init__()
     self.conv1 = nn.Sequential(
        nn.Conv2d(3, 64, kernel_size=3, padding=1, bias=False),
        nn.BatchNorm2d(64),
        nn.ReLU(inplace=True))
     self.conv2 = self.make_layer(block, 64, 64, 2, 1)
     self.conv3 = self.make_layer(block, 64, 128, 2, 2)
     self.conv4 = self.make_layer(block, 128, 256, 2, 2)
     self.conv5 = self.make_layer(block, 256, 512, 2, 2)
     self.avg_pool = nn.AdaptiveAvgPool2d((1, 1))
     self.dense_layer = nn.Linear(512, num_classes)
  def make_layer(self, block, in_channels, out_channels, num_blocks, stride):
     strides = [stride] + [1] * (num_blocks - 1)
     layers = []
     for stride in strides:
        layers.append(block(in_channels, out_channels, stride))
        in_channels = out_channels
     return nn.Sequential(*layers)
  def forward(self, x):
     out = self.conv1(x)
     out = self.conv2(out)
     out = self.conv3(out)
     out = self.conv4(out)
     out = self.conv5(out)
     out = self.avg_pool(out)
     out = out.view(out.size(0), -1)
     out = self.dense_layer(out)
     return out
Training and validation
def main():
  file_dir = os.path.dirname(inspect.getframeinfo(inspect.currentframe()).filename)
  seed = datetime.datetime.now().year
  setup_seed(seed)
  parser = argparse.ArgumentParser(description='Pytorch distribute training',
                          formatter_class=argparse.ArgumentDefaultsHelpFormatter)
  parser.add_argument('--enable_gpu', default='true')
parser.add_argument('--lr', default='0.01', help='learning rate')
  parser.add_argument('--epochs', default='100', help='training iteration')
  parser.add_argument('--init_method', default=None, help='tcp_port')
  parser.add_argument('--rank', type=int, default=0, help='index of current task')
  parser.add_argument('--world_size', type=int, default=1, help='total number of tasks')
  parser.add_argument('--custom_data', default='false')
  parser.add_argument('--data_url', type=str, default=os.path.join(file_dir, 'input_dir'))
  parser.add_argument('--output_dir', type=str, default=os.path.join(file_dir, 'output_dir'))
  args, unknown = parser.parse_known_args()
```

```
args.enable_gpu = args.enable_gpu == 'true'
  args.custom_data = args.custom_data == 'true'
  args.lr = float(args.lr)
  args.epochs = int(args.epochs)
  if args.custom_data:
     print('[warning] you are training on custom random dataset, '
         'validation accuracy may range from 0.4 to 0.6.')
### Settings for distributed training. Initialize DistributedDataParallel process. The init_method,
rank, and world_size parameters are automatically input by the platform. ###
  dist.init_process_group(init_method=args.init_method, backend="nccl", world_size=args.world_size,
rank=args.rank)
### Settings for distributed training. Initialize DistributedDataParallel process. The init_method,
rank, and world_size parameters are automatically input by the platform. ###
  tr_set, val_set = get_data(args.data_url, custom_data=args.custom_data)
  batch_per_gpu = 128
  gpus_per_node = torch.cuda.device_count() if args.enable_gpu else 1
  batch = batch_per_gpu * gpus_per_node
  tr_loader = DataLoader(tr_set, batch_size=batch, shuffle=False)
### Settings for distributed training. Create a sampler for data distribution to ensure that different
processes load different data. ###
  tr sampler = DistributedSampler(tr set, num replicas=args.world_size, rank=args.rank)
  tr_loader = DataLoader(tr_set, batch_size=batch, sampler=tr_sampler, shuffle=False, drop_last=True)
### Settings for distributed training. Create a sampler for data distribution to ensure that different
processes load different data. ###
  val_loader = DataLoader(val_set, batch_size=batch, shuffle=False)
  lr = args.lr * gpus_per_node
  max_epoch = args.epochs
  model = ResNet(Block).cuda() if args.enable_gpu else ResNet(Block)
### Settings for distributed training. Build a DistributedDataParallel model. ###
  model = nn.parallel.DistributedDataParallel(model)
### Settings for distributed training. Build a DistributedDataParallel model. ###
  optimizer = optim.Adam(model.parameters(), lr=lr)
  loss_func = torch.nn.CrossEntropyLoss()
  os.makedirs(args.output_dir, exist_ok=True)
  for epoch in range(1, max_epoch + 1):
     model.train()
     train_loss = 0
### Settings for distributed training. DistributedDataParallel sampler. Random numbers are set for
the DistributedDataParallel sampler based on the current epoch number to avoid loading duplicate
data.###
     tr_sampler.set_epoch(epoch)
### Settings for distributed training. DistributedDataParallel sampler. Random numbers are set for
the DistributedDataParallel sampler based on the current epoch number to avoid loading duplicate
data. ###
     for step, (tr_x, tr_y) in enumerate(tr_loader):
        if args.enable_gpu:
          tr_x, tr_y = tr_x.cuda(), tr_y.cuda()
        out = model(tr_x)
        loss = loss_func(out, tr_y)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        train_loss += loss.item()
     print('train | epoch: %d | loss: %.4f' % (epoch, train_loss / len(tr_loader)))
```

```
val_loss = 0
     pred_record = []
     real_record = []
     model.eval()
     with torch.no_grad():
        for step, (val_x, val_y) in enumerate(val_loader):
           if args.enable_gpu:
             val_x, val_y = val_x.cuda(), val_y.cuda()
           out = model(val_x)
           pred_record += list(np.argmax(out.cpu().numpy(), axis=1))
           real_record += list(val_y.cpu().numpy())
           val_loss += loss_func(out, val_y).item()
     val_accu = accuracy_score(real_record, pred_record)
     print('val | epoch: %d | loss: %.4f | accuracy: %.4f' % (epoch, val_loss / len(val_loader), val_accu),
'\n')
     if args.rank == 0:
        # save ckpt every epoch
        torch.save(model.state_dict(), os.path.join(args.output_dir, f'epoch_{epoch}.pth'))
if __name__ == '__main__':
  main()
```

• Result comparison

100-epoch **cifar-10** dataset training is completed using two resource types respectively: single-node single-card and two-node 16-card. The training duration and test set accuracy are as follows.

Table 7-29 Training result comparison

Resource Type	Single-Node Single- Card	Two-Node 16-Card	
Duration	60 minutes	20 minutes	
Accuracy	80+	80+	

# 7.7.5 Sample Code of Distributed Training

The following provides a complete code sample of distributed parallel training for the classification task of ResNet18 on the CIFAR-10 dataset.

The content of the training boot file **main.py** is as follows (if you need to execute a single-node and single-card training job, delete the code for distributed reconstruction):

import datetime import inspect import os import pickle import random import logging

import argparse import numpy as np from sklearn.metrics import accuracy\_score import torch from torch import nn, optim import torch.distributed as dist from torch.utils.data import TensorDataset, DataLoader

```
from torch.utils.data.distributed import DistributedSampler
file dir = os.path.dirname(inspect.getframeinfo(inspect.currentframe()).filename)
def load_pickle_data(path):
  with open(path, 'rb') as file:
     data = pickle.load(file, encoding='bytes')
   return data
def _load_data(file_path):
   raw_data = load_pickle_data(file_path)
   labels = raw_data[b'labels']
   data = raw data[b'data']
   filenames = raw_data[b'filenames']
   data = data.reshape(10000, 3, 32, 32) / 255
   return data, labels, filenames
def load cifar data(root path):
   train root path = os.path.join(root path, 'cifar-10-batches-py/data batch ')
   train_data_record = []
   train_labels = []
   train_filenames = []
   for i in range(1, 6):
     train_file_path = train_root_path + str(i)
     data, labels, filenames = _load_data(train_file_path)
     train data record.append(data)
     train_labels += labels
     train filenames += filenames
   train_data = np.concatenate(train_data_record, axis=0)
   train_labels = np.array(train_labels)
  val file path = os.path.join(root path, 'cifar-10-batches-py/test batch')
   val data, val labels, val filenames = load data(val file path)
  val_labels = np.array(val_labels)
   tr data = torch.from numpy(train data).float()
   tr_labels = torch.from_numpy(train_labels).long()
   val data = torch.from numpy(val data).float()
   val labels = torch.from numpy(val labels).long()
   return tr data, tr labels, val data, val labels
def get_data(root_path, custom_data=False):
   if custom data:
     train samples, test samples, img size = 5000, 1000, 32
     tr label = [1] * int(train samples / 2) + [0] * int(train samples / 2)
     val_label = [1] * int(test_samples / 2) + [0] * int(test_samples / 2)
     random.seed(2021)
     random.shuffle(tr label)
     random.shuffle(val label)
     tr data, tr labels = torch.randn((train samples, 3, img size, img size)).float(),
torch.tensor(tr label).long()
     val data, val labels = torch.randn((test samples, 3, img size, img size)).float(),
torch.tensor(
        val label).long()
     tr_set = TensorDataset(tr_data, tr_labels)
     val_set = TensorDataset(val_data, val_labels)
     return tr_set, val_set
```

```
elif os.path.exists(os.path.join(root_path, 'cifar-10-batches-py')):
     tr_data, tr_labels, val_data, val_labels = load_cifar_data(root_path)
     tr set = TensorDataset(tr data, tr labels)
     val_set = TensorDataset(val_data, val_labels)
     return tr_set, val_set
  else:
     try:
        import torchvision
        from torchvision import transforms
        tr set = torchvision.datasets.CIFAR10(root='./data', train=True,
                                  download=True, transform=transforms)
        val_set = torchvision.datasets.CIFAR10(root='./data', train=False,
                                   download=True, transform=transforms)
        return tr_set, val_set
     except Exception as e:
        raise Exception(
           f"{e}, you can download and unzip cifar-10 dataset manually, "
           "the data url is http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz")
class Block(nn.Module):
  def init (self, in channels, out channels, stride=1):
     super().__init__()
     self.residual function = nn.Sequential(
        nn.Conv2d(in_channels, out_channels, kernel_size=3, stride=stride, padding=1,
bias=False).
        nn.BatchNorm2d(out_channels),
        nn.ReLU(inplace=True),
        nn.Conv2d(out channels, out channels, kernel size=3, padding=1, bias=False),
        nn.BatchNorm2d(out_channels)
     )
     self.shortcut = nn.Sequential()
     if stride != 1 or in channels != out channels:
        self.shortcut = nn.Sequential(
           nn.Conv2d(in channels, out channels, kernel size=1, stride=stride, bias=False),
           nn.BatchNorm2d(out_channels)
        )
  def forward(self, x):
     out = self.residual function(x) + self.shortcut(x)
     return nn.ReLU(inplace=True)(out)
class ResNet(nn.Module):
  def __init__(self, block, num_classes=10):
     super(), init ()
     self.conv1 = nn.Sequential(
        nn.Conv2d(3, 64, kernel_size=3, padding=1, bias=False),
        nn.BatchNorm2d(64),
        nn.ReLU(inplace=True))
     self.conv2 = self.make_layer(block, 64, 64, 2, 1)
     self.conv3 = self.make_layer(block, 64, 128, 2, 2)
     self.conv4 = self.make layer(block, 128, 256, 2, 2)
     self.conv5 = self.make layer(block, 256, 512, 2, 2)
     self.avg_pool = nn.AdaptiveAvgPool2d((1, 1))
     self.dense layer = nn.Linear(512, num classes)
  def make_layer(self, block, in_channels, out_channels, num_blocks, stride):
     strides = [stride] + [1] * (num_blocks - 1)
```

```
lavers = []
     for stride in strides:
        layers.append(block(in channels, out channels, stride))
        in_channels = out_channels
     return nn.Sequential(*layers)
  def forward(self, x):
     out = self.conv1(x)
     out = self.conv2(out)
     out = self.conv3(out)
     out = self.conv4(out)
     out = self.conv5(out)
     out = self.avg_pool(out)
     out = out.view(out.size(0), -1)
     out = self.dense layer(out)
     return out
def setup_seed(seed):
  torch.manual seed(seed)
  torch.cuda.manual_seed_all(seed)
  np.random.seed(seed)
  random.seed(seed)
  torch.backends.cudnn.deterministic = True
def obs_transfer(src_path, dst_path):
  import moxing as mox
  mox.file.copy_parallel(src_path, dst_path)
  logging.info(f"end copy data from {src path} to {dst path}")
def main():
  seed = datetime.datetime.now().year
  setup seed(seed)
  parser = argparse.ArgumentParser(description='Pytorch distribute training',
                         formatter_class=argparse.ArgumentDefaultsHelpFormatter)
  parser.add_argument('--enable_gpu', default='true')
  parser.add_argument('--lr', default='0.01', help='learning rate')
  parser.add_argument('--epochs', default='100', help='training iteration')
  parser.add_argument('--init_method', default=None, help='tcp_port')
  parser.add_argument('--rank', type=int, default=0, help='index of current task')
  parser.add argument('--world size', type=int, default=1, help='total number of tasks')
  parser.add argument('--custom data', default='false')
  parser.add argument('--data url', type=str, default=os.path.join(file dir, 'input dir'))
  parser.add argument('--output dir', type=str, default=os.path.join(file dir, 'output dir'))
  args, unknown = parser.parse known args()
  args.enable_gpu = args.enable_gpu == 'true'
  args.custom_data = args.custom_data == 'true'
  args.lr = float(args.lr)
  args.epochs = int(args.epochs)
  if args.custom data:
     logging.warning('you are training on custom random dataset, '
          'validation accuracy may range from 0.4 to 0.6.')
```

### Settings for distributed training. Initialize DistributedDataParallel process. The **init\_method**, **rank**, and **world\_size** parameters are automatically input by the platform. ###

```
dist.init_process_group(init_method=args.init_method, backend="nccl",
world_size=args.world_size, rank=args.rank)
### Settings for distributed training. Initialize DistributedDataParallel process. The
init_method, rank, and world_size parameters are automatically input by the platform. ###
  tr_set, val_set = get_data(args.data_url, custom_data=args.custom_data)
  batch_per_gpu = 128
  gpus_per_node = torch.cuda.device_count() if args.enable_gpu else 1
  batch = batch_per_gpu * gpus_per_node
  tr_loader = DataLoader(tr_set, batch_size=batch, shuffle=False)
### Settings for distributed training. Create a sampler for data distribution to ensure that
different processes load different data. ###
  tr sampler = DistributedSampler(tr_set, num_replicas=args.world_size, rank=args.rank)
  tr loader = DataLoader(tr set, batch size=batch, sampler=tr sampler, shuffle=False,
drop last=True)
### Settings for distributed training. Create a sampler for data distribution to ensure that
different processes load different data. ###
  val loader = DataLoader(val set, batch size=batch, shuffle=False)
  lr = args.lr * gpus_per_node * args.world_size
  max epoch = args.epochs
  model = ResNet(Block).cuda() if args.enable_gpu else ResNet(Block)
### Settings for distributed training. Build a DistributedDataParallel model. ###
  model = nn.parallel.DistributedDataParallel(model)
### Settings for distributed training. Build a DistributedDataParallel model. ###
  optimizer = optim.Adam(model.parameters(), lr=lr)
  loss_func = torch.nn.CrossEntropyLoss()
  os.makedirs(args.output dir, exist ok=True)
  for epoch in range(1, max epoch + 1):
     model.train()
     train loss = 0
### Settings for distributed training. DistributedDataParallel sampler. Random numbers are set
for the DistributedDataParallel sampler based on the current epoch number to avoid loading
duplicate data. ###
     tr sampler.set epoch(epoch)
### Settings for distributed training. DistributedDataParallel sampler. Random numbers are set
for the DistributedDataParallel sampler based on the current epoch number to avoid loading
duplicate data. ###
     for step, (tr x, tr y) in enumerate(tr loader):
        if args.enable gpu:
          tr_x, tr_y = tr_x.cuda(), tr_y.cuda()
        out = model(tr x)
        loss = loss_func(out, tr_y)
        optimizer.zero_grad()
        loss.backward()
```

print('train | epoch: %d | loss: %.4f' % (epoch, train\_loss / len(tr\_loader)))

```
val_loss = 0
pred_record = []
real_record = []
```

optimizer.step()
train loss += loss.item()

```
model.eval()
with torch.no_grad():
    for step, (val_x, val_y) in enumerate(val_loader):
        if args.enable_gpu:
            val_x, val_y = val_x.cuda(), val_y.cuda()
        out = model(val_x)
        pred_record += list(np.argmax(out.cpu().numpy(), axis=1))
        real_record += list(val_y.cpu().numpy())
        val_loss += loss_func(out, val_y).item()
        val_accu = accuracy_score(real_record, pred_record)
        print('val | epoch: %d | loss: %.4f | accuracy: %.4f' % (epoch, val_loss / len(val_loader),
        val_accu), '\n')
    if args.rank == 0:
        # save ckpt every epoch
        torch.save(model.state_dict(), os.path.join(args.output_dir, f'epoch_{epoch}.pth'))
```

```
if __name__ == '__main__':
main()
```

#### FAQs

#### 1. How Do I Use Different Datasets in the Sample Code?

• To use the CIFAR-10 dataset in the preceding code, **download** and decompress the dataset and upload it to the OBS bucket. The file directory structure is as follows:

```
DDP
|--- main.py
|--- input_dir
|----- cifar-10-batches-py
|------ data_batch_1
|------ data_batch_2
```

**DDP** is the code directory specified during training job creation, **main.py** is the preceding code example (the boot file specified during training job creation), and **cifar-10-batches-py** is the decompressed dataset folder (stored in the **input\_dir** folder).

#### \* Created By Custom algorithms My algorithms My subscriptions ★ Boot Mode Custom images PyTorch pytorch\_1.8.0-cuda\_10.2-py\_3.7... • Ŧ \* Code Directory 🕐 1/DDP/ Select 🛪 Boot File ( 🤉 /r /DDP/main.py Select Local Code Directory /home/ ma-user/modelarts/user-job-dir Work Directory /home/ma-user/modelarts/user-job-dir × Select

#### Figure 7-34 Creating a training job

• To use user-defined random data, change the value of **custom\_data** in the code example to **true**.

parser.add\_argument('--custom\_data', default='true')

Then, run **main.py**. The parameters for creating a training job are the same as those shown in the preceding figure.

#### 2. Why Can I Leave the IP Address of the Master Node Blank for DDP?

The **init method** parameter in **parser.add\_argument('--init\_method', default=None, help='tcp\_port')** contains the IP address and port number of the master node, which are automatically input by the platform.

# **8** Inference Deployment

# 8.1 Introduction to Inference

After an AI model is developed, you can use it to create an AI application and quickly deploy the application as an inference service. The AI inference capabilities can be integrated into your IT platform by calling APIs.



- Develop a model: Models can be developed in ModelArts or your local development environment. A locally developed model must be uploaded to OBS.
- Create an AI application: Import the model file and inference file to the ModelArts model repository and manage them by version. Use these files to build an executable AI application.
- Deploy as a service: Deploy the AI application as a container instance in the resource pool and register inference APIs that can be accessed externally.
- Perform inference: Add the function of calling the inference APIs to your application to integrate AI inference into the service process.

#### Deploying an AI Application as a Service

After an AI application is created, you can deploy it as a service on the **Deploy** page. ModelArts supports the following deployment types:

• Real-time service

Deploy an AI application as a web service with real-time test UI and monitoring supported.

• Batch service

Deploy an AI application as a batch service that performs inference on batch data and automatically stops after data processing is complete.

# 8.2 Managing AI Applications

# 8.2.1 Introduction to AI Application Management

Al development and optimization require frequent iterations and debugging. Modifications in datasets, training code, or parameters affect the quality of models. If the metadata of the development process cannot be centrally managed, the optimal model may fail to be reproduced.

ModelArts AI application management allows you to import all meta models obtained through training, meta models uploaded to OBS, and meta models in container images. In this way, you can centrally manage all iterated and debugged AI applications.

#### Constraints

• In an ExeML project, after a model is deployed, the model is automatically uploaded to the AI application management list. However, AI applications generated by ExeML cannot be downloaded and can be used only for deployment and rollout.

#### **Scenarios for Creating AI Applications**

- Imported from a training job: Create a training job in ModelArts and train a model. After obtaining a satisfactory model, use it to create an AI application and deploy the application as services.
- Imported from OBS: If you use a mainstream framework to develop and train a model locally, you can upload the model to an OBS bucket based on the model package specifications, import the model from OBS to ModelArts, and use the model to create an AI application for service deployment.
- Imported from a container image: If an AI engine is not supported by ModelArts, you can use it to build a model, import the model to ModelArts as a custom image, use the image to create an AI application, and deploy the AI application as services.

#### **Functions of AI Application Management**

Table o-I Functions of Al addition management	Table 8-1	Functions	of Al	application	managemen
---	-----------	-----------	-------	-------------	-----------

Supported Function	Description		
Creating an Al Application	Import the trained models to ModelArts and create AI applications for centralized management. The following provides the operation guide for each method of importing models.		
	Importing a Meta Model from a Training Job		
	Importing a Meta Model from a Container Image		
Viewing Details About an Al Application	After an AI application is created, you can view its information on the details page.		
Managing Al Application Versions	To facilitate traceback and model tuning, ModelArts provides the AI application version management function. You can manage AI applications by version.		

#### Supported AI Engines for ModelArts Inference

If you import a model from a template or OBS to create an AI application, the following AI engines and versions are supported.

#### **NOTE**

- Runtime environments marked with **recommended** are unified runtime images, which will be used as mainstream base inference images.
- Images of the old version will be discontinued. Use unified images.
- The base images to be removed are no longer maintained.
- Naming a unified runtime image: <AI engine name and version> <Hardware and version: CPU, CUDA, or CANN> - <Python version> - <OS version> - <CPU architecture>

#### Table 8-2 Supported AI engines and their runtime

Engine	Runtime		
TensorFlow	tensorflow_1.15.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64		
MindSpore	mindspore_2.0.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64		
PyTorch	pytorch_1.11.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64		

# 8.2.2 Creating an AI Application

#### 8.2.2.1 Importing a Meta Model from a Training Job

You can create a training job in ModelArts to obtain a satisfactory model. Then, you can import the model to **AI Application Management** for centralized management. In addition, you can quickly deploy the model as a service.

#### Constraints

- A model generated from a training job that uses subscribed algorithms can be directly imported to ModelArts without the need to use the inference code or configuration file.
- ModelArts of the Arm architecture does not support model import from training.
- If the meta model is from a container image, ensure the size of the meta model complies with Restrictions on the Size of an Image for Importing an AI Application.

#### Prerequisites

- The training job has been successfully executed, and the model has been stored in the OBS directory where the training output is stored (the input parameter is **train\_url**).
- If a model is generated from a training job that uses a frequently-used framework or custom image, upload the inference code and configuration file to the storage directory of the model by referring to Introduction to Model Package Specifications.
- The OBS directory you use and ModelArts are in the same region.

#### **Creating an AI Application**

- Log in to the ModelArts management console and choose AI Application Management > AI Applications in the left navigation pane. The AI Applications page is displayed.
- 2. Click **Create** in the upper left corner.
- 3. On the displayed page, set the parameters.
  - a. Set basic information about the AI application. For details about the parameters, see **Table 8-3**.

Parameter	Description
Name	Application name. The value can contain 1 to 64 visible characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Version	Version of the AI application to be created. For the first import, the default value is <b>0.0.1</b> .
	<b>NOTE</b> After an AI application is created, you can create new versions using different meta models for optimization.

<b>Table 0-3</b> Falameters of basic Al application information	Table 8-3	Parameters	of basic	AI ap	plication	information
---	-----------	------------	----------	-------	-----------	-------------

Parameter	Description
Description	Brief description of an AI application

b. Select the meta model source and set related parameters. Set **Meta Model Source** to **Training job**. For details about the parameters, see **Table 8-4**.

Table 8-4 Parameters of	the meta model so	ource
-------------------------	-------------------	-------

Parameter	Description			
Meta Model	Choose <b>Training Job</b> > <b>Training Jobs</b> or <b>Training Job</b> > <b>Training Jobs (New)</b> .			
Source	• Choose a training job that has executed under the current account and a training version.			
Al Engine	Inference engine used by the meta model. The engine is automatically matched based on the training job you select.			
Inference Code	Set inference code for an AI application. The code is used to customize the inference processing logic. Display the inference code URL. You can copy this URL directly.			
Runtime Dependenc y	List the dependencies of the selected model in the environment.			
AI Application Description	Provide AI application descriptions to help other AI application developers better understand and use your applications. Click <b>Add AI Application Description</b> and set the <b>Document name</b> and URL. A maximum of three AI application descriptions are supported.			
Deploymen t Type	Select the service types that the application can be deployed. When deploying a service, only the service types selected here are available. For example, if you only select <b>Real-time services</b> here, you can only deploy the AI application as a real-time service after it is created.			

c. Confirm the configurations and click **Create now**. The AI application is created.

In the AI application list, you can view the created AI application and its version. When the status changes to **Normal**, the AI application is successfully created. On this page, you can perform such operations as creating new versions and quickly deploying services.

#### **Follow-Up Procedure**

**Deploying an AI Application as a Service**: In the AI application list, click the option button on the left of the AI application name to display the version list at the bottom of the list page. Locate the row that contains the target version, click

**Deploy** in the **Operation** column to deploy the AI application as a service type selected during AI application creation.

#### 8.2.2.2 Importing a Meta Model from OBS

In scenarios where frequently-used frameworks are used for model development and training, you can import the model to ModelArts and use it to create an AI application for unified management.

#### Constraints

- The imported model for creating an AI application, inference code, and configuration file must comply with the requirements of ModelArts. For details, see Introduction to Model Package Specifications, Specifications for Editing a Model Configuration File, and Specifications for Writing Model Inference Code.
- If the meta model is from a container image, ensure the size of the meta model complies with **Restrictions on the Size of an Image for Importing an AI Application**.

#### Prerequisites

- The model has been developed and trained, and the type and version of the AI engine used by the model are supported by ModelArts. For details, see **Supported AI Engines for ModelArts Inference**.
- The trained model package, inference code, and configuration file have been uploaded to OBS.
- The OBS directory you use and ModelArts are in the same region.

#### **Creating an AI Application**

- Log in to the ModelArts management console, and choose AI Application Management > AI Applications in the left navigation pane. The AI Applications page is displayed.
- 2. Click **Create** in the upper left corner.
- 3. On the displayed page, set the parameters.
  - a. Set basic information about the AI application. For details about the parameters, see **Table 8-5**.

Parameter	Description
Name	Application name. The value can contain 1 to 64 visible characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Version	Version of the AI application to be created. For the first import, the default value is <b>0.0.1</b> .
	<b>NOTE</b> After an AI application is created, you can create new versions using different meta models for optimization.

Table 8-5	Parameters	of basic	AI application	n information
	rurumeters	or busic.	a application	1 millionnation

Parameter	Description
Description	Brief description of an AI application

b. Select the meta model source and set related parameters. Set **Meta Model Source** to **OBS**. For details about the parameters, see **Table 8-6**.

For the meta model imported from OBS, edit the inference code and configuration files by following **model package specifications** and place the inference code and configuration files in the **model** folder storing the meta model. If the selected directory does not comply with the model package specifications, the AI application cannot be created.

Table 8-6 Parameters of the meta model source

Parameter	Description
Meta Model	OBS path for storing the meta model. The OBS path cannot contain spaces. Otherwise, the AI application fails to be created.
Al Engine	The AI engine automatically associates with the meta model storage path you select.
	If <b>AI Engine</b> is set to <b>Custom</b> , you must specify the protocol and port number in <b>Container API</b> for starting the model. The request protocol is <b>HTTPS</b> , and the port number is <b>8080</b> .

Parameter	Description
Health Check	Health check on a model. After you select an AI engine that supports health check and runtime environment, this parameter is displayed. When <b>AI Engine</b> is set to <b>Custom</b> , you must configure health check in the image. Otherwise, the service deployment will fail.
	<ul> <li>Check Mode: Select HTTP request or Command. When a custom engine is used, you can select HTTP request or Command.</li> </ul>
	When a non-custom engine is used, you can select only <b>HTTP request</b> .
	<ul> <li>Health Check URL: This parameter is displayed when Check Mode is set to HTTP request. Enter the health check URL. The default value is /health.</li> </ul>
	• Health Check Command: This parameter is displayed when Check Mode is set to Command. Enter the health check command.
	• Health Check Period: Enter an integer ranging from 1 to 2147483647. The unit is second.
	• <b>Delay( seconds )</b> : specifies the delay for performing the health check after the instance is started. Enter an integer ranging from 0 to 2147483647.
	• <b>Maximum Failures</b> : Enter an integer ranging from 1 to 2147483647. During service startup, if the number of consecutive health check failures reaches the specified value, the service will be abnormal. During service running, if the number of consecutive health check failures reaches the specified value, the service will enter the alarm status.
	<b>NOTE</b> To use a custom engine to create an AI application, ensure that the custom engine complies with the specifications for custom engines. For details, see <i>Creating an AI Application Using a</i> <i>Custom Engine</i> .
	If health check is configured for an AI application, the deployed services using this AI application will stop 3 minutes after receiving the stop instruction.
Runtime Dependenc y	List the dependencies of the selected model in the environment.
AI Application Description	Provide AI application descriptions to help other AI application developers better understand and use your applications. Click <b>Add AI Application Description</b> and set the <b>Document name</b> and <b>URL</b> . You can add up to three AI application descriptions.

Parameter	Description	
Configurati on File	By default, the system associates the configuration file stored in OBS. After enabling this function, you can view and edit the model configuration file.	
	<b>NOTE</b> This function is to be taken offline. After that, you can modify the model configuration by setting <b>AI Engine</b> , <b>Runtime</b> <b>Dependency</b> , and <b>Apis</b> .	
Deploymen t Type	Select the service types that the application can be deployed. When deploying a service, only the service types selected here are available. For example, if you only select <b>Real-time services</b> here, you can only deploy the AI application as a real-time service after it is created.	
API Configurati on	After enabling this function, you can edit RESTful APIs to define the input and output formats of an AI application. The model APIs must comply with ModelArts specifications. For details, see <b>Specifications for Editing a Model Configuration File</b> . For details about the code example, see <b>Code Example of apis Parameters</b> .	

c. Check the information and click **Create now**. The AI application is created.

In the AI application list, you can view the created AI application and its version. When the status changes to **Normal**, the AI application is successfully created. On this page, you can perform such operations as creating new versions and quickly deploying services.

#### **Follow-Up Procedure**

**Deploying an AI Application as a Service**: In the AI application list, click the option button on the left of the AI application name to display the version list at the bottom of the list page. Locate the row that contains the target version, click **Deploy** in the **Operation** column to deploy the AI application as a service type selected during AI application creation.

#### 8.2.2.3 Importing a Meta Model from a Container Image

For AI engines that are not supported by ModelArts, you can import the models you compile to ModelArts from custom images.

#### Constraints

- For details about the specifications and description of custom images, see **Custom Image Specifications for Creating AI Applications**.
- The configuration file must be provided for a model that you have developed and trained. The file must comply with ModelArts specifications. For details, see Specifications for Editing a Model Configuration File. After the writing is completed, upload the file to the specified OBS directory.

 If the meta model is from a container image, ensure the size of the meta model complies with Restrictions on the Size of an Image for Importing an AI Application.

#### Prerequisites

The OBS directory you use and ModelArts are in the same region.

#### **Creating an AI Application**

- Log in to the ModelArts management console, and choose AI Application Management > AI Applications in the left navigation pane. The AI Applications page is displayed.
- 2. Click **Create** in the upper left corner.
- 3. On the displayed page, set the parameters.
  - a. Set basic information about the AI application. For details about the parameters, see **Table 8-7**.

Parameter	Description
Name	Application name. The value can contain 1 to 64 visible characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Version	Version of the AI application to be created. For the first import, the default value is <b>0.0.1</b> .
	<b>NOTE</b> After an AI application is created, you can <b>create new versions</b> using different meta models for optimization.
Description	Brief description of an AI application

<b>TAULE 0-1</b> FALAILIELEIS OF DASIC AF ADDUICATION INTOTTIATIC	Table 8-7	Parameters	of basic Al	application	informatior
---	-----------	------------	-------------	-------------	-------------

b. Select the meta model source and set related parameters. Set **Meta Model Source** to **Container image**. For details about the parameters, see **Table 8-8**.

Parameter	Description
Container Image Path	Click by to import the model image from the container image. The model is of the Image type, and you do not need to use <b>swr_location</b> in the configuration file to specify the image location.
	For details about operation guidance and requirements for creating a custom image, see <b>Custom Image Specifications for Creating Al Applications</b> .
	<b>NOTE</b> The model image you select will be shared with the system administrator, so ensure you have the permission to share the image (images shared with other accounts are not supported). When you deploy a service, ModelArts deploys the image as an inference service. Ensure that your image can be properly started and provide an inference API.
Container API	Protocol and port number for starting an AI application <b>NOTE</b> The default request protocol and port number provided by ModelArts are HTTP and 8080, respectively. Set them based on the actual custom image.
Image Replication	Indicates whether to copy the model image in the container image to ModelArts.
	• When this function is disabled, the model image is not copied, AI applications can be created quickly, but modifying or deleting images in the source directory of SWR may affect service deployment.
	• When this function is enabled, the model image is copied, AI applications cannot be created quickly, but you can modify or delete images in the source directory of SWR as that would not affect service deployment.
	<b>NOTE</b> You must enable this function if you want to use images shared by others. Otherwise, AI applications will fail to be created.

Table 8-8 Paramete	ers of the met	a model source
--------------------	----------------	----------------

Parameter	Description
Health Check	Health check on an AI application. This parameter is configurable only when the health check API is configured in the custom image. Otherwise, the AI application deployment will fail.
	<ul> <li>Check Mode: Select HTTP request or Command.</li> </ul>
	• Health Check URL: This parameter is displayed when Check Mode is set to HTTP request. Enter the health check URL. The default value is / health.
	• Health Check Command: This parameter is displayed when Check Mode is set to Command. Enter the health check command.
	• Health Check Period: Enter an integer ranging from 1 to 2147483647. The unit is second.
	• <b>Delay( seconds )</b> : specifies the delay for performing the health check after the instance is started. Enter an integer ranging from 0 to 2147483647.
	• <b>Maximum Failures</b> : Enter an integer ranging from 1 to 2147483647. During service startup, if the number of consecutive health check failures reaches the specified value, the service will be abnormal. During service running, if the number of consecutive health check failures reaches the specified value, the service will enter the alarm status.
	<b>NOTE</b> If health check is configured for an AI application, the deployed services using this AI application will stop 3 minutes after receiving the stop instruction.
AI Application Description	Provide AI application descriptions to help other AI application developers better understand and use your applications. Click <b>Add AI Application</b> <b>Description</b> and set the <b>Document name</b> and <b>URL</b> . You can add up to three AI application descriptions.
Deployment Type	Select the service types that the application can be deployed. When deploying a service, only the service types selected here are available. For example, if you only select <b>Real-time services</b> here, you can only deploy the AI application as a real- time service after it is created.
Start command	Customizable start command of a model

Parameter	Description
Apis	When you enable this function, you can edit RESTful APIs to define the AI application input and output formats. The model APIs must comply with ModelArts specifications. For details, see <b>Specifications for Editing a Model Configuration</b> <b>File</b> . For details about the code example, see <b>Code</b> <b>Example of apis Parameters</b> .

c. Check the information and click **Next**. The AI application is created.

In the AI application list, you can view the created AI application and its version. When the status changes to **Normal**, the AI application is successfully created. On this page, you can perform such operations as creating new versions and quickly deploying services.

#### **Follow-Up Procedure**

**Deploying an AI Application as a Service**: In the AI application list, click the option button on the left of the AI application name to display the version list at the bottom of the list page. Locate the row that contains the target version, click **Deploy** in the **Operation** column to deploy the AI application as a service type selected during AI application creation.

# 8.2.3 Viewing the AI Application List

You can view all created AI applications on the AI application list page. The AI application list page displays the following information.

Parameter	Description
AI Application Name	Name of an AI application.
Latest Version	Latest version of an AI application.
Status	Status of an AI application.
Deployment Type	Types of the services that an AI application can be deployed as.
Versions	Number of AI application versions.
Request Mode	Request mode of real-time services.
	<ul> <li>Synchronization Request: one-off inference with results returned synchronously (shorter than 60s). This mode is suitable for images and small videos.</li> </ul>
	• Asynchronous Request: one-off inference with results returned asynchronously (over 60s). This mode is suitable for real-time video inference and large videos.

 Table 8-9
 AI application list

Parameter	Description
Created	Time when an AI application is created.
Description	Description of an AI application.
Operation	• <b>Create Version</b> : Create an AI application version. The settings of the last version are used by default, except for the version. You can change the parameter settings.
	• <b>Delete</b> : Delete the AI application.
	<b>NOTE</b> If an AI application version has been deployed as a service, you must delete the associated service before deleting the AI application version. A deleted AI application cannot be recovered.

Click the check box next to the AI application name to display the hidden view at the bottom of the list, where you can see the version list. (If the view is not

displayed, click  $\uparrow$  in the bottom right corner.)

#### Figure 8-2 Version list

My Al Applications My Subscr Create Find Al Application	iptions Edge Subscri	ptions								
Q. Search or filter by AI application na AI Application Name	Latest Version	Status	Deployment Type	Westions	Request Mode	Created :		Description	Operation	C
۲	0.0.1	📀 Normal	Real-Time Services	1	asynchronous request	Dec 02, 2023 00:58 20 GMT+06:0		input,output@\$\$\$\$\$	Create Version   Delete	
	0.0.1	Normal	Real-Time Services/Batch Services/E	1	synchronous request	Dec 01, 2023 14:53:21 GMT+06:0			Create Version   Delete	
	0.0.1	<ul> <li>Normal</li> </ul>	Real-Time Services	1	s)nchronous request	Nov 30, 2023 22:03:57 GMT+08:0			Create Version   Delete	
	0.0.1	Normal	Real-Time Services	1	synchronous request.	Nov 30, 2023 15:09:37 GMT+08:0			Create Version   Delete	
Selected.model_iva_m/Wefl [Versi	ons]									
Version 0	Status : Deploym	ent Type 💠	Model Size 🗧	Model Source 0	Created 0		Description 0	Operation		
0.0.1	📀 Normal Real-Tim	e Services	261.81 MB	Custom algorithm	Dec 02, 2023 00:58	20 GMT+08.00		Deploy + Publish   Delete		
10 × Total Becords: 1 <										

The version list displays the following information.

	Table	8-10	Version	list
--	-------	------	---------	------

Parameter	Description
Version	Current version of an AI application.
Status	Status of an AI application.
Deployment Type	Types of the services that an AI application can be deployed as.
Model Size	Size of an AI application.
Model Source	Model source of an AI application.
Created	Time when an AI application is created.
Description	Description of an AI application.

Parameter	Description
Operation	<ul> <li>Deploy: Deploy an AI application as real-time services, batch services, or edge services.</li> </ul>
	Publish: Publish an AI application to AI Gallery.
	• <b>Delete</b> : Delete a version of an AI application.

# 8.2.4 Viewing Details About an AI Application

After an AI application is created, you can view its information on the details page.

- Log in to the ModelArts management console. In the navigation pane on the left, choose AI Application Management > AI Applications. The AI Applications page is displayed.
- 2. Click the name of the target AI application. The application details page is displayed.

On the application details page, you can view the basic information and model precision of the AI application, and switch tab pages to view more information.

Parameter	Description
Name	Name of an AI application
Status	Status of an AI application
Version	Current version of an AI application
ID	ID of an AI application
Description	Click the edit button to add the description of an AI application.
Deployment Type	Types of the services that an AI application can be deployed
Meta Model Source	Source of the meta model, which can be training jobs, OBS, or container images.
Training Name	Associated training job if the meta model comes from a training job. Click the training job name to go to its details page.
Training Version	Training job version if the meta model comes from an old-version training job.
Storage path of the meta model	Path to the meta model if the meta model comes from OBS.
Container Image Storage Path	Path to the container image if the meta model comes from a container image.

 Table 8-11
 Basic information about an AI application

Parameter	Description
Al Engine	AI engine if the meta model comes from a training job or OBS.
Engine Package Address	Engine package address if the meta model comes from OBS and <b>AI Engine</b> is <b>Custom</b> .
Runtime Environment	Runtime environment on which the meta model depends if the meta model comes from a training job or OBS and a preset AI engine is used.
Container API	Protocol and port number for starting the AI application if the meta model comes from OBS (AI Engine is Custom) or a container image.
Inference Code	Path to the inference code if the meta model comes from an olde-version training job.
Image Replication	Image replication status if the meta model comes from OBS or a container image.
Size	Size of an AI application
Health Check	Health check status if the meta model comes from OBS or a container image. If health check is enabled, the following parameters are displayed: <b>Check Mode</b> , <b>Health Check URL</b> , <b>Health Check Period</b> , <b>Delay</b> , and <b>Maximum Failures</b> .
AI Application Description	Description document added during the creation of an AI application.
Instruction Set Architecture	System architecture.
Inference Accelerator	Type of inference accelerator cards.

Table 8-12 Details p	page of an A	I application
----------------------	--------------	---------------

Parameter	Description
Model Precision	Model recall, precision, accuracy, and F1 score of an AI application
Parameter Configuration	API configuration, input parameters, and output parameters of an AI application
Runtime Dependency	Model dependency on the environment. If creating a job failed, edit the runtime dependency. After the modification is saved, the system will automatically use the original image to create the job again.

Parameter	Description
Events	The progress of key operations during AI application creation
	Events are stored for three months and will be automatically cleared then.
	For details about how to view events of an AI application, see Viewing Events of an AI Application.
Constraint	Displays the constraints of service deployment, such as the request mode, boot command, and model encryption, based on the settings during AI application creation. For AI applications in asynchronous request mode, parameters including the input mode, output mode, service startup parameters, and job configuration parameters can be displayed.
Associated Services	The list of services that an AI application was deployed. Click a service name to go to the service details page.

# 8.2.5 Managing AI Application Versions

To facilitate source tracing and repeated AI application tuning, ModelArts provides the AI application version management function. You can manage models based on versions.

#### Prerequisites

An AI application has been created in ModelArts.

#### **Creating a New Version**

On the AI Application Management > AI Applications page, click Create Version in the Operation column of the target AI application. On the Create Version page, set the parameters. For details, see Creating an AI Application. Click Create now.

#### **Deleting a Version**

On the **AI Application Management > AI Applications** page, click the option button on the left of the AI application name to display the application version list. In the application version list, click **Delete** in the **Operation** column to delete the corresponding version.

#### **NOTE**

If a service has been deployed for the AI application version, you need to delete the associated service before deleting the AI application version. A deleted version cannot be recovered. Exercise caution when performing this operation.
## **Deleting an AI Application**

In the navigation pane, choose AI Application Management > AI Applications. On the AI Applications page, click **Delete** in the **Operation** column to delete the target AI application.

#### **NOTE**

If a service has been deployed for the AI application version, you need to delete the associated service before deleting the AI application version. A deleted AI application cannot be recovered. Exercise caution when performing this operation.

# 8.2.6 Viewing Events of an AI Application

During the creation of an AI application, every key event is automatically recorded. You can view the events on the details page of the AI application at any time.

This helps you better understand the process of creating an AI application and locate faults more accurately when a task exception occurs. The following table lists the available events.

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Normal	The model starts to import. Start model import.	-
Abnormal	Failed to create the image. Failed to build the image.	Locate and rectify the fault based on the error information. FAQ
Abnormal	The custom image does not support specified dependencies. Customize model does not support dependencies.	The runtime dependencies cannot be configured when a custom image is imported. Install the pip dependency package in the Dockerfile that is used to create the image. FAQ
Abnormal	Only custom images support <b>swr_location</b> . Non-custom type models should not contain swr_location.	Delete the <b>swr_location</b> field from the model configuration file <b>config.json</b> and try again.
Abnormal	The health check API of a custom image must be <i>xxx</i> . The health check url of custom image model must be %s.	Modify the health check API of the custom image and try again.

Туре	Event ( <i>xxx</i> should be replaced with Solution the actual value.)	
Normal	The image creation task is in the xxx state. The status of the image building task is %s.	-
Abnormal	Label <i>xxx</i> does not exist in image <i>xxx</i> . Image %s does not have the %s tag.	Contact technical support.
Abnormal	Invalid parameter value xxx exists in the model configuration file. Invalid %s in config.json.	Delete invalid parameters from the model configuration file and try again.
Abnormal	Failed to obtain the labels of image <i>xxx</i> . Failed to obtain the tag list of image %s.	Contact technical support.
Abnormal	Failed to import data because <i>xxx</i> is larger than <i>xxx</i> GB. %s [%s] is larger than %dG and cannot be imported.	The size of the model or image exceeds the upper limit. Downsize the model or image and import it again. FAQ
Abnormal	User xxx does not have OBS permission obs:object:PutObjectAcl. User %s does not have obs:object:PutObjectAcl permission	The IAM user does not have the obs:object:PutObject Acl permission on OBS. Add the agency permission for the IAM user. FAQ
Abnormal	Creating the image timed out. The timeout duration is <i>xxx</i> minutes. Image building task timeout. The %s- minute limit is over.	There is a timeout limit for image building using ImagePacker. Simplify the code to improve efficiency. FAQ
Normal	Model description updated. Model description updated.	-
Normal	Model runtime dependencies not updated. Model running dependencies not updated.	-

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Normal	Model runtime dependencies updated. Recreating the image. Model running dependencies	-
Abnormal	SWR traffic control triggered. Try again later. The throttling threshold of swr has been reached.	SWR traffic control triggered. Try again later.
Normal	The system is being upgraded. Try again later. System is upgrading, please try again later.	
Abnormal	Failed to obtain the source image. An error occurred in authentication. The token has expired. Failed to access source image. Authenticate Error, token expired.	Contact technical support.
Abnormal	Failed to obtain the source image. Check whether the image exists. Failed to access source image. Check whether the image exists.	Contact technical support.
Normal	Source image size calculated. Source image size calculated successfully.	-
Normal	Source image shared. Source image shared successfully.	-
Abnormal	Failed to create the image due to traffic control. Try again later. Failed to build the image due to the threshold has been reached. Please try again later.	Traffic control triggered. Try again later.
Abnormal	Failed to send the image creation request. Failed to send image building request.	Contact technical support.

Туре	Event ( <i>xxx</i> should be replaced with Solution the actual value.)	
Abnormal	<ul> <li>Failed to share the source image. Check whether the image exists or whether you have the permission to share the image.</li> <li>Failed to share source image. Check whether the image exists or whether you have the share permission on the image.</li> </ul>	
Normal	The model importedModel imported successfully.	
Normal	Model file imported. Model file imported successfully.	-
Normal	Model size calculated. Model size calculated successfully.	-
Abnormal	Failed to import the model. Failed to import the model.	For details about how to locate and rectify the fault, see FAQ.
Abnormal	Failed to copy the model file. Check whether you have the OBS permission. Failed to copy model file due to obs exception. Please Check your obs access right.	Check whether you have the OBS permission. FAQ
Abnormal	Failed to schedule the image creation task. Image building task scheduling failed.	Contact technical support.
Abnormal	Failed to start the image creation task. Failed to start the image building task.	Contact technical support.
Abnormal	The Roman image has been created but cannot be shared with resource tenants. The ROMA image is successfully built but cannot be shared to resource tenants.	Contact technical support.
Normal	Image created. Image built successfully.	-

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Normal	The image creation task started. Start the image building task.	-
Normal	The environment image creation task started. Start the env image building task.	-
Normal	The request for creating an environment image received. Received another env image building request of the model.	-
Normal	The request for creating an image received. Received another image building request of the model.	-
Normal	An existing environment image is used. Use cached env image.	-
Abnormal	Failed to create the image. For details, see image creation logs. Failed to build the image. For details, view the building log.	View the build logs to locate and rectify the fault. FAQ
Abnormal	Failed to create the image due to an internal system error. Contact technical support. Failed to build the image due to system errors. Contact the administrator.	Contact technical support.
Abnormal	Failed to import model file <i>xxx</i> because it is larger than 5 GB. Model file %s is larger than 5G and cannot be imported.	The size of the model file xxx is greater than 5 GB. Downsize the model file and try again, or use dynamic loading to import the model file. FAQ
Abnormal	Failed to create the OBS bucket due to an internal system error. Contact technical support. Failed to create bucket due to system errors. Contact the administrator.	Contact technical support.

Event ( <i>xxx</i> should be replaced with the actual value.)	Solution	
Failed to calculate the model size. Subpath <i>xxx</i> does not exist in path <i>xxx</i> . Model size calculated failed.Can not find %s child directory in current model directory %s	Correct the subpath and try again, or contact technical support.	
Failed to calculate the model size. The model of the <i>xxx</i> type does not exist in path <i>xxx</i> . Model size calculated failed.Can not find %s file in current model directory %s.	Check the storage location of the model of the xxx type, correct the path, and try again, or contact technical support.	
Failed to calculate the model size. More than one xxx model file is stored in path xxx. Model size calculated failed.Find more than one %s file in current	-	
	Event (xxx should be replaced with the actual value.)Failed to calculate the model size. Subpath xxx does not exist in path xxx.Model size calculated failed.Can not find %s child directory in current model directory %s.Failed to calculate the model size. The model of the xxx type does not exist in path xxx.Model size calculated failed.Can not find %s file in current model directory %s.Failed to calculate the model size. The model size calculated failed.Can not find %s file in current model directory %s.Failed to calculate the model size. More than one xxx model file is stored in path xxx.Model size calculated failed.Find more than one %s file in current model directory %s.	

During AI application creation, key events can both be manually and automatically refreshed.

## **Viewing Events**

- In the navigation pane of the ModelArts management console, choose AI Application Management > AI Applications. In the AI application list, click the name of the target AI application to go to its details page.
- 2. View the events on the **Events** tab page.

# 8.3 Deploying an AI Application as a Service

# 8.3.1 Deploying AI Applications as Real-Time Services

## 8.3.1.1 Deploying as a Real-Time Service

After an AI application is prepared, you can deploy it as a real-time service and call the service for prediction.

## Constraints

A maximum of 20 real-time services can be deployed by a user.

## Prerequisites

• Data has been prepared. Specifically, you have created an AI application in the **Normal** state in ModelArts.

## Note

Real-time services deployed using the public resource pool also occupy quota resources when the services are **Abnormal** or **Stopped**. If the quota is insufficient and no more services can be deployed, delete some abnormal services to release resources.

Quota calculation:

- If a dedicated resource pool is used to deploy real-time services, the quota is not decreased. The quota is increased or decreased only when the dedicated pool is created, modified, or deleted.
- When a shared resource pool is used to deploy a real-time service, the quota will be increased or decreased when you create, change the number of, or delete instances.

Metering calculation:

- If a real-time service is deployed using a dedicated pool, only the data of the dedicated pool to which the service belongs is metered.
- When a shared pool is used to deploy a real-time service, the specifications used by the service will be metered.

## Procedure

- Log in to the ModelArts management console. In the left navigation pane, choose Service Deployment > Real-Time Services. The real-time service list is displayed by default.
- 2. In the real-time service list, click **Deploy** in the upper left corner. The **Deploy** page is displayed.
- 3. Set parameters for a real-time service.
  - a. Set basic information about model deployment. For details about the parameters, see **Table 8-13**.

Table	8-13	Basic	parameters
-------	------	-------	------------

Parameter	Description
Name	Name of the real-time service. Set this parameter as prompted.

Parameter	Description
Auto Stop	After this parameter is enabled and the auto stop time is set, a service automatically stops at the specified time. The auto stop function is enabled by default, and the default value is <b>1 hour later</b> .
	The options are <b>1 hour later</b> , <b>2 hours later</b> , <b>4 hours later</b> , <b>6 hours later</b> , and <b>Custom</b> . If you select <b>Custom</b> , you can enter any integer from 1 to 24 hours in the text box on the right.
Description	Brief description of the real-time service.

b. Enter key information including the resource pool and AI application configurations. For details, see **Table 8-14**.

## Table 8-14 Parameters

Param eter	Sub- Parame ter	Description
Resour ce Pool	Public Resourc e Pool	CPU/GPU computing resources are available for you to select.
	Dedicat ed Resourc e Pool	Select a specification from the dedicated resource pool specifications. The physical pools with logical subpools created are not supported temporarily. <b>NOTE</b>
		<ul> <li>The data of old-version dedicated resource pools will be gradually migrated to the new-version dedicated resource pools.</li> </ul>
		<ul> <li>For new users and the existing users who have migrated data from old-version dedicated resource pools to new ones, there is only one entry to new- version dedicated resource pools on the ModelArts management console.</li> </ul>
		• For the existing users who have not migrated data from old-version dedicated resource pools to new ones, there are two entries to dedicated resource pools on the ModelArts management console, where the entry marked with <b>New</b> is to the new version.
		For details about the new version of dedicated resource pools, see <b>Comprehensive Upgrades to ModelArts</b> <b>Resource Pool Management Functions</b> .
AI Applic ation and Config uration	AI Applicat ion Source	Select <b>My AI Applications</b> based on your requirements.

Param eter	Sub- Parame ter	Description
	AI Applicat ion and Version	Select the AI application and version that are in the <b>Normal</b> state.
	Traffic Ratio (%)	Set the traffic proportion of the current instance node. Service calling requests are allocated to the current version based on this proportion.
		If you deploy only one version of an AI application, set this parameter to <b>100%</b> . If you select multiple versions for gray release, ensure that the sum of the traffic ratios of these versions is <b>100%</b> .
	Specific ations	Select available specifications based on the list displayed on the console. The specifications in gray cannot be used in the current environment.
		If specifications in the public resource pools are unavailable, no public resource pool is available in the current environment. In this case, use a dedicated resource pool or contact the administrator to create a public resource pool. <b>NOTE</b>
		When the selected flavor is used to deploy the service, necessary system consumption is generated. Therefore, the resources actually occupied by the service are slightly greater than the selected flavor.
	Comput e Nodes	Set the number of instances for the current AI application version. If you set the number of nodes to <b>1</b> , the standalone computing mode is used. If you set the number of nodes to a value greater than 1, the distributed computing mode is used. Select a computing mode based on the actual requirements.
	Environ ment Variable	Set environment variables and inject them to the pod. To ensure data security, do not enter sensitive information in environment variables.
	Timeout	Timeout of a single model, including both the deployment and startup time. The default value is 20 minutes. The value must range from 3 to 120.
	Add Al Applicat ion Version and	If the selected AI application has multiple versions, you can add multiple versions and configure a traffic ratio. You can use gray launch to smoothly upgrade the AI application version. <b>NOTE</b>
	Configu ration	Free compute specifications do not support the gray launch of multiple versions.

Param eter	Sub- Parame ter	Description
	Mount Storage	This parameter is displayed when the resource pool is a dedicated resource pool. This function will mount a storage volume to compute nodes (compute instances) as a local directory when the service is running. It is recommended when the model or input data is large. Only OBS parallel file systems are supported.
		<ul> <li>Source Path: Select the storage path of the parallel file. A cross-region OBS parallel file system cannot be selected.</li> </ul>
		<ul> <li>Mount Path: Enter the mount path of the container, for example, /obs-mount/.</li> </ul>
		<ul> <li>Select a new directory. If you select an existing directory, existing files will be overwritten. OBS mounting allows you to add, view, and modify files in the mount directory but does not allow you to delete files in the mount directory. To delete files, manually delete them in the OBS parallel file system.</li> </ul>
		<ul> <li>It is a good practice to mount the container to an empty directory. If the directory is not empty, ensure that there are no files affecting container startup in the directory. Otherwise, such files will be replaced, resulting in failures to start the container and create the workload.</li> </ul>
		<ul> <li>The mount path must start with a slash (/) and can contain a maximum of 1,024 characters, including letters, digits, and the following special characters: \</li> </ul>
		<b>NOTE</b> Storage mounting can be used only by services deployed in a dedicated resource pool.
Traffic Limit	N/A	Maximum number of times a service can be accessed within a second. You can set this parameter as needed.

Param eter	Sub- Parame ter	Description	
WebSo cket	N/A	Whether to deploy a real-time service as a WebSocket service. For details about WebSocket real-time services, see Full-Process Development of WebSocket Real-Time Services.	
		NOTE	
		<ul> <li>This function is supported only if the AI application is WebSocket-compliant and comes from a container image.</li> </ul>	
		• After this function is enabled, <b>Traffic Limit</b> and <b>Data Collection</b> cannot be set.	
		<ul> <li>This parameter cannot be changed after the service is deployed.</li> </ul>	

4. After confirming the entered information, complete service deployment as prompted. Generally, service deployment jobs run for a period of time, which may be several minutes or tens of minutes depending on the amount of your selected data and resources.

#### D NOTE

After a real-time service is deployed, it is started immediately.

You can go to the real-time service list to check whether the deployment of the real-time service is complete. In the real-time service list, after the status of the newly deployed service changes from **Deploying** to **Running**, the service is deployed successfully.

## 8.3.1.2 Viewing Service Details

After an AI application is deployed as a real-time service, you can access the service page to view its details.

- Log in to the ModelArts management console and choose Service Deployment > Real-Time Services.
- 2. On the **Real-Time Services** page, click the name of the target service. The service details page is displayed.

You can view the service name, status, and other information. For details, see **Table 8-15**.

Parameter	Description
Name	Name of the real-time service.
Status	Status of the real-time service.
Source	AI application source of the real-time service.
Service ID	Real-time service ID

 Table 8-15 real-time service parameters

Parameter	Description	
Description	Service description, which can be edited after you click the edit button on the right side.	
Resource Pool	Resource pool specifications used by the service. If the public resource pool is used for deployment, this parameter is not displayed.	
Custom Settings	Customized configurations based on real-time service versions. This allows version-based traffic distribution policies and configurations. Enable this option and click <b>View Settings</b> to customize the settings. For details, see <b>Modifying Customized Settings</b> .	
Traffic Limit	Maximum number of times a service can be accessed within a second.	
WebSocket	Whether to upgrade to the WebSocket service.	

3. Switch between tabs on the details page of a real-time service to view more details. For details, see **Table 8-16**.

Table 8-16 Details	of a real-time	service
--------------------	----------------	---------

Parameter	Description	
Usage Guides	This page displays the API URL, AI application information, input parameters, and output parameters. You can click 🗖 to copy the API URL to call the service.	
Prediction	You can perform real-time prediction on this page. For details, see <b>Testing the Deployed Service</b> .	
Configuration Updates	This page displays <b>Current Configurations</b> and <b>Update History</b> .	
	• <b>Current Configurations</b> : Al application name, version, status, compute node specifications, traffic ratio, number of compute nodes, deployment timeout interval, environment variables, storage mounting, and resource pool information (for services deployed in a dedicated resource pool)	
	<ul> <li>Update History: historical AI application information.</li> </ul>	

Parameter	Description		
Monitoring	This page displays resource usage and AI application calls.		
	• <b>Resource Usage</b> : includes the used and available CPU, memory, GPU, and NPU resources.		
	• AI Application Calls: indicates the number of AI application calls. The statistics collection starts after the AI application status changes to <b>Ready</b> . (This parameter is not displayed for WebSocket services.)		
Event	This page displays key operations during service use, such as the service deployment progress, detailed causes of deployment exceptions, and time points wh a service is started, stopped, or modified.		
	Events are saved for one month and will be automatically cleared then.		
	For details about how to view events of a service, see <b>Viewing Service Events</b> .		
Logs	This page displays the log information about each AI application in the service. You can view logs generated in the latest 5 minutes, latest 30 minutes, latest 1 hour, and user-defined time segment.		
	You can select the start time and end time when defining the time segment.		
	Meet the following rules to search logs:		
	<ul> <li>Do not enter strings that contain any following delimiters: ,'";=()[]{}@&amp;&lt;&gt;/:\n\t\r.</li> </ul>		
	• Enter keywords for exact search. A keyword is a word between two adjacent delimiters.		
	• Enter keywords for fuzzy search. For example, you can enter <b>error</b> , <b>er?or</b> , <b>rro*</b> , or <b>er*r</b> .		
	• Enter phrases for exact search. For example, <b>Start to</b> refresh.		
	<ul> <li>Before enabling this function, you can combine keywords with AND (&amp;&amp;) or OR (  ). For example, query logs&amp;&amp;erro* or query logs  erro*. After enabling this function, you can combine keywords with AND or OR. For example, query logs AND erro* or query logs OR erro*.</li> </ul>		

## Modifying Customized Settings

A customized configuration rule consists of the configuration condition (**Setting**), access version (**Version**), and customized running parameters (including **Setting Name** and **Setting Value**).

You can configure different settings with customized running parameters for different versions of a real-time service.

The priorities of customized configuration rules are in descending order. You can change the priorities by dragging the sequence of customized configuration rules.

After a rule is matched, the system will no longer match subsequent rules. A maximum of 10 configuration rules can be configured.

Parameter	Man dator y	Description			
Setting	Yes	Expression of the Spring Expression Language (SPEL) rule. Only the equal, matches, and hashCode expressions of the character type are supported.			
Version	Yes	Access version for a customized service configuration rule. When a rule is matched, the real-time service of the version is requested.			
Setting Name	No	Key of a customized running parameter, consisting of a maximum of 128 characters. Configure this parameter if the HTTP message header is used to carry customized running parameters to a real-time service.			
Setting Value	No	Value of a customized running parameter, consisting of a maximum of 256 characters. Configure this parameter if the HTTP message header is used to carry customized running parameters to a real-time service.			

Table 8-17 Parameters for Custom Settings

Customized settings can be used in the following scenarios:

• If multiple versions of a real-time service are deployed for gray release, customized settings can be used to distribute traffic by user.

Table	8-18	Built-in	variables
-------	------	----------	-----------

Built-in Variable	Description
DOMAIN_NAME	Account name that is used to invoke the inference request
DOMAIN_ID	Account ID that is used to invoke the inference request
PROJECT_NAME	Project name that is used to call an inference request
PROJECT_ID	Project ID that invokes the inference request
USER_NAME	Username that is used to call an inference request

Built-in Variable	Description
USER_ID	User ID that is used to call an inference request

Pound key (#) indicates that a variable is referenced. The matched character string must be enclosed in single quotation marks.

#{Built-in variable} == 'Character string' #{Built-in variable} matches 'Regular expression'

Example 1:

If the account name in the inference request is **User A**, the specified version is matched.

#DOMAIN\_NAME == 'User A'

– Example 2:

If the account name in the inference request starts with **op**, the specified version is matched.

#DOMAIN\_NAME matches 'op.\*'

#### Table 8-19 Common regular expressions

Characte r	Description
	Match any single character except \n. To match any character including \n, use (. \n).
*	Match the subexpression that it follows for zero or multiple times. For example, <b>zo*</b> can match <b>z</b> and <b>zoo</b> .
+	Match the subexpression that it follows for once or multiple times. For example, <b>zo+</b> can match <b>zo</b> and <b>zoo</b> , but cannot match <b>z</b> .
?	Match the subexpression that it follows for zero or one time. For example, <b>do(es)?</b> can match <b>does</b> or <b>do</b> in <b>does</b> .
^	Match the start of the input string.
\$	Match the end of the input string.
{n}	<i>n</i> is a non-negative integer, which matches exactly <i>n</i> number of occurrences of an expression. For example, <b>o{2}</b> cannot match <b>o</b> in <b>Bob</b> , but can match two <b>o</b> s in <b>food</b> .
x y	Match x or y. For example, <b>z food</b> can match <b>z</b> or <b>food</b> , and <b>(z f)ood</b> can match <b>zood</b> or <b>food</b> .
[xyz]	Character set, where any single character in it can be matched. For example, <b>[abc]</b> can match <b>a</b> in <b>plain</b> .

#### Figure 8-3 Traffic distribution by user

custom Settings				
<b>D</b> You can drag configurations to rearrange ther	n.			
* Setting (?)	* Version (?)		Setting Name (Optional)	Setting Value (Optional)
#DOMAIN_NAME == 'User A'	0.0.2	•		
#DOMAIN NAME matches 'on *'	0.0.1	•		

• If multiple versions of a real-time service are deployed for gated launch, customized settings can be used to access different versions through the header.

Start with **#HEADER\_** to indicate that the header is referenced as a condition. #HEADER\_{key} == '{value}' #HEADER\_{key} matches '{value}'

- Example 1:

If the header of an inference HTTP request contains a version and the value is **0.0.1**, the condition is met. Otherwise, the condition is not met. #HEADER version == '0.0.1'

Example 2:

If the header of an inference HTTP request contains **testheader** and the value starts with **mock**, the rule is matched.

#HEADER\_testheader matches 'mock.\*'

– Example 3:

If the header of an inference HTTP request contains **uid** and the hash code value meets the conditions described in the following algorithm, the rule is matched.

#HEADER\_uid.hashCode() % 100 < 10

#### Figure 8-4 Using header to access different versions

Custom Settings			×
Setting ⑦	Version ⑦	Setting Name (Optional) ⑦ Setting Value (Optional) ⑦.	
#HEADER_version == '0.0.1'	0.0.1 •	Ū	
#HEADER_testheader matches 'mock.*'	0.0.1 💌	Ū	
#HEADER_uid.hashCode() % 100 < 10	0.0.1 💌	Ū	

If a real-time service version supports different runtime configurations, you can use Setting Name and Setting Value to specify customized runtime parameters so that different users can use different running configurations. Example:

When user A accesses the AI application, the user uses configuration A. When user B accesses the AI application, the user uses configuration B. When matching a running configuration, ModelArts adds a header to the request and also the customized running parameters specified by **Setting Name** and **Setting Value**.

# **Figure 8-5** Customized running parameters added for a customized configuration rule

Custom Settings					×
• You can drag configurations to rearrange them.					
* Setting ⑦	* Version (?)	Setting Name (Optional)	Setting Value (Optional)		
#DOMAIN_NAME == 'User A'	0.0.2 💌	testkey1	testvalue1	Ū	
#DOMAIN_NAME == 'User B'	0.0.2 💌	testkey2	testkey2	Ū	
(+) Add					

## 8.3.1.3 Testing the Deployed Service

After an AI application is deployed as a real-time service, you can debug code or add files for testing on the **Prediction** tab page. Based on the input request (JSON text or file) defined by the AI application, the service can be tested in either of the following ways:

- JSON Text Prediction: If the input type of the AI application of the deployed service is JSON text, that is, the input does not contain files, you can enter the JSON code on the **Prediction** tab page for service testing.
- File Prediction: If the input type of the AI application of the deployed service is file, including images, audios, and videos, you can add images on the **Prediction** tab page for service testing.

#### D NOTE

- If the input type is image, the size of a single image must be less than 8 MB.
- The maximum size of the request body for JSON text prediction is 8 MB.
- Due to the limitation of API Gateway, the duration of a single prediction cannot exceed 40s.
- The following image types are supported: png, psd, jpg, jpeg, bmp, gif, webp, psd, svg, and tiff.
- This function is used for commissioning. In actual production, you are advised to call APIs. You can select Access Authenticated Using a Token based on the authentication mode.

## **Input Parameters**

After a service is deployed, obtain the input parameters of the service on the **Usage Guides** tab page of the service details page.

The input parameters displayed on the **Usage Guides** tab page vary depending on the AI application source that you select.

- If your metamodel comes from ExeML or a built-in algorithm, the input and output parameters are defined by ModelArts. For details, see the Usage Guides tab page. On the Prediction tab page, enter the corresponding JSON text or file for service testing.
- If you use a custom meta model with the inference code and configuration file compiled by yourself (Specifications for Writing the Model Configuration File), ModelArts only visualizes your data on the Usage Guides tab page. The following figure shows the mapping between the input parameters displayed on the Usage Guides tab page and the configuration file.

1	(		
2	"model_type": "TensorFlow",		
3	"model_algorithm": "object_detection",		
4	"metrics": {	ter Configuration	
5	"fl": 0.345294,		
6	"accuracy": 0.462963,	DOGT /	
7	"precision": 0.338977,	POST	
8	"recall": 0.351852		
9	},	Input Parameter	
10	"apis": [{	input i ununeter	
11	"protocol": "https",	Nama	Turne
12	"url": "/",	Name	type
13	"method": "post",	images	file
14	"request": {		
15	"Content-type": "multipart/form-data",	Output Parameter	
16	"data": {	oupurrainneter	
17	"type": "object",	Mama	Time
18	"properties": {	Name	Type
19	"images": {		-
20	"type": "file"		
21	}		
22	}		
23	}		
24	},		
25	"response": {		

#### **Figure 8-6** Mapping between the configuration file and Usage Guides

## **JSON Text Prediction**

- Log in to the ModelArts management console and choose Service Deployment > Real-Time Services.
- 2. On the **Real-Time Services** page, click the name of the target service. The service details page is displayed. Enter the inference code on the **Prediction** tab, and click **Predict** to perform prediction.

## File Prediction

- Log in to the ModelArts management console and choose Service Deployment > Real-Time Services.
- 2. On the **Real-Time Services** page, click the name of the target service. The service details page is displayed. On the **Prediction** tab page, click **Upload** and select a test file. After the file is uploaded successfully, click **Predict** to perform a prediction test.

## 8.3.1.4 Accessing Real-Time Services

## 8.3.1.4.1 Accessing a Real-Time Service

If a real-time service is in the **Running** status, the real-time service has been deployed successfully. This service provides a standard RESTful API for you to call. Before integrating the API to the production environment, commission the API.

By default, APIs of real-time services are accessed using HTTPS. WebSocket-based access is also supported. If you select **WebSocket** during real-time service deployment, the API URL is a WebSocket address after the service is deployed. For details, see **Accessing a Real-Time Service Through WebSocket**.

ModelArts supports the following authentication methods for accessing real-time services (HTTPS requests are used as an example):

• Access Authenticated Using a Token

ModelArts allows you to call APIs to access real-time services in the following ways:

- Accessing a Real-Time Service (Public Network Channel)
- Accessing a Real-Time Service (VPC High-Speed Channel)

When you call an API to access a real-time service, the size of the prediction request body and the prediction time are subject to the following limitations:

- The size of a request body cannot exceed 12 MB. Otherwise, the request will fail.
- Due to the limitation of API Gateway, the prediction duration of each request does not exceed 40 seconds.

## 8.3.1.4.2 Authentication Mode

#### Access Authenticated Using a Token

If a real-time service is in the **Running** state, it has been deployed successfully. This service provides a standard RESTful API for users to call. Before integrating the API to the production environment, commission the API. You can use the following methods to send an inference request to the real-time service:

- Method 1: Use GUI-based Software for Inference (Postman). (Postman is recommended for Windows.)
- Method 2: Run the cURL Command to Send an Inference Request. (curl commands are recommended for Linux.)
- Method 3: Use Python to Send an Inference Request.

## Prerequisites

You have obtained a user token, local path to the inference file, URL of the realtime service, and input parameters of the real-time service.

- The local path to the inference file can be an absolute path (for example, D:/ test.png for Windows and /opt/data/test.png for Linux) or a relative path (for example, ./test.png).
- You can obtain the service URL and input parameters of a real-time service on the Usage Guides tab page of its service details page.

The API URL is the service URL of the real-time service. If a path is defined for **apis** in the model configuration file, the URL must be followed by the user-defined path, for example, *{URL of the real-time service}*/predictions/poetry.

## Method 1: Use GUI-based Software for Inference (Postman)

- 1. Download Postman and install it, or install the Postman Chrome extension. Alternatively, use other software that can send POST requests. Postman 7.24.0 is recommended.
- 2. Open Postman.
- 3. Set parameters on Postman. The following uses image classification as an example.
  - Select a POST task and copy the API URL to the POST text box. On the Headers tab page, set Key to X-Auth-Token and Value to the user token.
  - On the **Body** tab page, file input and text input are available.
    - File input

Select **form-data**. Set **KEY** to the input parameter of the AI application, which must be the same as the input parameter of the real-time service. In this example, the **KEY** is **images**. Set **VALUE** to an image to be inferred (only one image can be inferred).

Text input

Select **raw** and then **JSON(application/json)**. Enter the request body in the text box below. An example request body is as follows:

```
{
    "meta": {
    "uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
    },
    "data": {
        "req_data": [
            {
                "sepal_length": 3,
                "sepal_width": 1,
                "petal_length": 2.2,
                "petal_width": 4
            }
        ]
      }
}
```

**meta** can carry a universally unique identifier (UUID). When the inference result is returned after API calling, the UUID is returned to trace the request. If you do not need this function, leave **meta** blank. **data** contains a **req\_data** array for one or multiple pieces of input data. The parameters of each piece of data, such as **sepal\_length** and **sepal\_width** in this example are determined by the AI application.

- 4. After setting the parameters, click **send** to send the request. The result will be displayed in **Response**.
  - Inference result using file input: The field values in the return result vary with the AI application.
  - Inference result using text input: The request body contains meta and data. If the request contains uuid, uuid will be returned in the response. Otherwise, uuid is left blank. data contains a resp\_data array for the inference results of one or multiple pieces of input data. The parameters of each result are determined by the AI application, for example, sepal\_length and predictresult in this example.

## Method 2: Run the cURL Command to Send an Inference Request

The command for sending inference requests can be input as a file or text.

- File input
  - curl -kv -F 'images=@Image path' -H 'X-Auth-Token: Token value' -X POST Real-time service URL
    - -k indicates that SSL websites can be accessed without using a security certificate.
    - -F indicates file input. In this example, the parameter name is images, which can be changed as required. The image storage path follows @.
    - H indicates the header of a POST command. X-Auth-Token is the header key, which is fixed. *Token value* indicates the user token.
    - **POST** is followed by the API URL of the real-time service.

The following is an example of the cURL command for inference with file input:

curl -kv -F 'images=@/home/data/test.png' -H 'X-Auth-Token:MIISkAY\*\*\*80T9wHQ==' -X POST https:// modelarts-infers-1.xxx/v1/infers/eb3e0c54-3dfa-4750-af0c-95c45e5d3e83

Text input

```
curl -kv -d '{"data":{"req_data":
[{"sepal_length":3,"sepal_width":1,"petal_length":2.2,"petal_width":4}]}}' -H 'X-Auth-
Token:MIISkAY***80T9wHQ==' -H 'Content-type: application/json' -X POST https://modelarts-
infers-1.xxx/v1/infers/eb3e0c54-3dfa-4750-af0c-95c45e5d3e83
```

-d indicates the text input of the request body.

## Method 3: Use Python to Send an Inference Request

- 1. Download the Python SDK and configure it in the development tool. For details, see .
- 2. Create a request body for inference.

```
File input
# coding=utf-8
import requests
if __name__ == '__main__':
  # Config url, token and file path.
  url = "URL of the real-time service"
  token = "User token"
  file_path = "Local path to the inference file"
  # Send request.
  headers = {
     'X-Auth-Token': token
  files = {
     'images': open(file_path, 'rb')
  }
  resp = requests.post(url, headers=headers, files=files)
  # Print result.
```

print(resp.status\_code)
print(resp.text)

The **files** name is determined by the input parameter of the real-time service. The parameter name must be the same as that of the input parameter of the file type.

#### - Text input (JSON)

The following is an example of the request body for reading the local inference file and performing Base64 encoding:

```
# coding=utf-8
```

import base64 import requests

```
if __name__ == '__main__':
    # Config url, token and file path
    url = "URL of the real-time service"
    token = "User token"
    file_path = "Local path to the inference file"
    with open(file_path, "rb") as file:
        base64_data = base64.b64encode(file.read()).decode("utf-8")
    # Set body,then send request
    headers = {
```

```
'Content-Type': 'application/json',
```

'X-Auth-Token': token
}
body = {
 'image': base64\_data
}
resp = requests.post(url, headers=headers, json=body)
# Print result
print(resp.status\_code)
print(resp.text)

The **body** name is determined by the input parameter of the real-time service. The parameter name must be the same as that of the input parameter of the string type. The value of **base64\_data** in **body** is of the string type.

#### 8.3.1.4.3 Access Mode

## Accessing a Real-Time Service (Public Network Channel)

#### Context

By default, ModelArts inference uses the public network to access real-time services. After a real-time service is deployed, a standard RESTful API is provided for you to call. You can view the API URL on the **Usage Guides** tab page of the service details page.

#### Figure 8-7 API URL

Usage Guides	Prediction	Configuration Updates	Monitoring	Events	Logs
API URL https://	'inference.	i.cn/v1/infers/1b22941d-55	44-48de 🗇		

## Accessing a Real-Time Service

The following authentication modes are available for accessing real-time services from a public network:

• Access Authenticated Using a Token

#### Accessing a Real-Time Service (VPC Channel)

#### Context

To access a ModelArts real-time service from an internal VPC node of your account, you can use a VPC channel. By creating an endpoint in your VPC and connecting to the ModelArts VPC endpoint service, you can access the real-time service from your VPC endpoint.

#### Procedure

To access a real-time service through a VPC channel, perform the following steps:

- 1. Obtain the ModelArts VPC endpoint service address.
- 2. Buy and connect to a ModelArts endpoint.
- 3. Set a VPC access channel for real-time services.
- 4. Create a private DNS zone.
- 5. Access a real-time service through VPC.
- **Step 1** Obtain the ModelArts VPC endpoint service address.
  - Log in to the ModelArts management console and choose Service Deployment > Real-Time Services.
  - 2. Click **Access VPC**. In the displayed dialog box, view the VPC endpoint service address.

#### Figure 8-8 Viewing a VPC endpoint service address

Deplo	y	Delete	Authorize Access VPC		
Search	by na	ame by default.		~	
		Name/ID ↓Ξ	Access VPC	^	ated ↓Ξ
	~	service_for_upda 2a74f3e6-f4af-4a	To access a VPC, create an endpoint to connect to each endpoint service. A maximum of 1 endpoints can be created.		29, 2023 11:37:28 GMT+0
	~	service_for_upda d2ec9558-7873-4	VPC Domain Name Endpoint Service Endpoint ID Endpoint IP Address		29, 2023 11:37:14 GMT+0
	~	service_for_upda 9039218d-f70f-4	27506ed-496f-406a-620c-47		29, 2023 11:36:53 GMT+0
	~	service-1eea bb7a7865-f212-4	Close		29, 2023 10:50:29 GMT+0

**Step 2** Buy and connect to a ModelArts endpoint.

- Log in to the VPC management console. In the navigation pane, choose VPC Endpoint > VPC Endpoints.
- 2. Click **Buy VPC Endpoint** in the upper right corner.
  - **Region**: region where the VPC endpoint is located.

Resources in different regions cannot communicate with each other. The region must be the same as that of ModelArts.

- Service Category: Select Find a service by name.
- VPC Endpoint Service Name: Enter the endpoint service address obtained in 1. Click Verify on the right. The system automatically sets VPC, Subnet, and Private IP Address.
  - Create a Private Domain Name: Retain the default setting.
- 3. Confirm the specifications, and click **Next** and then **Submit**. The VPC endpoint list page is displayed.
- Step 3 Set a VPC access channel for real-time services.
  - Log in to the ModelArts management console. In the navigation pane, choose Service Deployment > Real-Time Services.
  - 2. Click **Access VPC**. In the displayed dialog box, select the VPC used in **2**. The endpoint ID and endpoint IP address are automatically displayed.

#### Figure 8-9 Selecting VPC

Acces	ss VPC		×
To acces	ss a VPC, create an endpoint to connect to each	endpoint service. A maximum of 1 endpoints can be o	created.
VPC:	vpc-zxy-test	▼ Domain Name:	
Endpo	int Service	Endpoint ID	Endpoint IP Address
	nnel1.1459ea75-34f6-400c-b1a2-c	I 5 c-cecf1b690d2a	
		Close	

- **Step 4** Create a private DNS zone.
  - 1. Log in to the DNS console. In the navigation pane on the left, choose **Private Zones**.
  - 2. Click Create Private Zone. Set the following parameters:
    - Domain Name: infer-modelarts-<*regionId*>.xxx.com. The current region ID without hyphens (-) is the value of *regionId*.
    - **VPC**: Select a VPC you want to associate with the private zone.
  - 3. Click **OK**.

**Step 5** Access a real-time service through VPC.

- 1. Use the following API to access a real-time service through VPC: https://{*Private DNS domain name*}/{*URL*}
  - Private DNS domain name: private domain name you set. You can also click Access VPC on the real-time service list page to view the domain name in the displayed dialog box.
  - URL: The URL for a real-time service is the part after the domain name of **API URL** in the **Usage Guides** tab of the service details page.

Figure 8-10 Obtaining the URL

API URL	/v1/infers/1b22941d-5544-48de	
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

2. Use GUI-based software, cURL command, or Python to access a real-time service. For details, see Access Authenticated Using a Token.

----End

## Accessing a Real-Time Service (VPC High-Speed Channel)

## Context

When accessing a real-time service, you may require:

- High throughput and low latency
- TCP or RPC requests

To meet these requirements, ModelArts enables high-speed access through VPC peering.

ð

In high-speed access through VPC peering, your service requests are directly sent to instances through VPC peering but not through the inference platform. This accelerates service access.

#### **NOTE**

The following functions that are available through the inference platform will be unavailable if you use high-speed access:

- Authentication
- Traffic distribution by configuration
- Load balancing
- Alarm, monitoring, and statistics

#### Figure 8-11 High-speed access through VPC peering



## Preparations

Deploy a real-time service in a dedicated resource pool and ensure the service is running.

#### NOTICE

- For details about how to deploy services in new-version dedicated resource pools, see Comprehensive Upgrades to ModelArts Resource Pool Management Functions.
- Only the services deployed in a dedicated resource pool support high-speed access through VPC peering.
- High-speed access through VPC peering is available only for real-time services.
- Due to traffic control, there is a limit on how often you can get the IP address and port number of a real-time service. The number of calls of each tenant account cannot exceed 2000 per minute, and that of each IAM user account cannot exceed 20 per minute.
- High-speed access through VPC peering is available only for the services deployed using the AI applications imported from custom images.

## Procedure

To enable high-speed access to a real-time service through VPC peering, perform the following operations:

- 1. Interconnect the dedicated resource pool to the VPC.
- 2. Create an ECS in the VPC.
- 3. Obtain the IP address and port number of the real-time service.
- 4. Access the service through the IP address and port number.

#### Step 1 Interconnect the dedicated resource pool to the VPC.

Log in to the ModelArts management console, choose **Dedicated Resource Pools** > **Elastic Cluster**, locate the dedicated resource pool used for service deployment, and click its name/ID to go to the resource pool details page. Obtain the network configuration. Switch back to the dedicated resource pool list, click the **Network** tab, locate the network associated with the dedicated resource pool, and interconnect it with the VPC. After the VPC is accessed, the VPC will be displayed on the network list and resource pool details pages. Click the VPC to go to the details page.

#### Figure 8-12 Locating the target dedicated resource pool

Elastic Cluster												
() We would much appre	ciate if you could complete	our questionnaire o	on Elastic Cluster. W	ur feedback will he	elp us provide a bett	er user experience.						×
Resource Pools N Create Records /	letwork A maximum of 15 resource p	ools can be created	d. You can create 1!	i more.						Enter a name.	QC	۲
Name/ID	Resource Pool Type 🏾 🏹	Status 🍞	Training Job	Inference Ser	DevEnviron	Accelerator Driver	Nodes (Available/Unavaila	Obtained At 🝦	Billing Mode	Description	Operation	
pool-Infer-autotest pool-Infer-autotest	Physical	Running	<ul> <li>Enabled</li> </ul>	Enabled	Enable Fal		1/0/1	Feb 10, 2023 16:47:00	Pay-per-use		Adjust Capacity   Mo	xe 🔹
test-auto poolb22ca38c	Physical	Running		Enabled		**	9/0/9	Feb 28, 2022 11:15:13	Pay-per-use		Adjust Capacity   Mo	xe 🔹
hg03-pool-video-infer pool47b9509b	Physical	<ul> <li>Running</li> <li>Restricted</li> </ul>		Enabled		460.32.03	1/0/1 ②	Apr 22, 2021 09:59:16			Adjust Capacity   Me	we 🕶

#### Figure 8-13 Obtaining the network configuration

Basic Information			
Name	test-auto	Resource Pool ID	
Resource Pool Type	Physical	Status	Running
DevEnviron	-	Training Job	
Inference Service	Enabled	Billing Mode	Pay-per-use
Metering ID		Description	
Network	net-managed-f6bs 1 resource pools associated	Interconnect VPC	
Obtained At	Feb 28, 2022 11:15:13 GMT+08:00		

#### Figure 8-14 Interconnecting the VPC

< test-auto -

Resource Pools Networ	rk					
Create A maximum of	15 network can be creat	ed. You can create 9 more.			Enter a nam	. Q C @
Network Name	Status 🖓	CIDR Block	Interconnect VPC	Associated sfsturbo	Obtained At JF	Operation
maos-network-fwx182425 maos-network-fwx182425	<ul> <li>Active</li> </ul>				Feb 08, 2023 18:13:19 GMT+08:00	Interconnect VPC   More +
network-c91e network-c91e-04f258c8478	<ul> <li>Active</li> </ul>		-	-	Feb 08, 2023 18:08:06 GMT+08:00	Interconnect VPC   More +
network-c342 network-c342-04f258c8478	<ul> <li>Active</li> </ul>			**	Jan 16, 2023 17:35:38 GMT+08:00	Interconnect VPC   More 💌
network-606b network-606b-04f258c8478	<ul> <li>Active</li> </ul>		**	**	Jan 16, 2023 17:12:15 GMT+08:00	Interconnect VPC   More +
net-managed-f6b9 net-managed-f6b9a371-53	Active ⑦		vpc-hpc-slurm / subnet-slurm		Jul 22, 2022 14:35:08 GMT+08:00	Interconnect VPC More +
network-37f5 network-37f5-04f258c8478	<ul> <li>Active</li> </ul>		💿 vpc-da8a / subnet-da9d		Apr 28, 2022 16:58:55 GMT+08:00	Interconnect VPC   More 💌
10 • Total Records: 6	< 1 > Go 1					

#### **Step 2** Create an ECS in the VPC.

Log in to the ECS management console and click **Buy ECS** in the upper right corner. On the **Buy ECS** page, configure basic settings and click **Next: Configure Network**. On the **Configure Network** page, select the VPC connected in 1, configure other parameters, confirm the settings, and click **Submit**. When the ECS status changes to **Running**, the ECS has been created. Click its name/ID to go to the server details page and view the VPC configuration.

<b>FIGURE OF IS</b> SELECTING A VEC WHEN PUTCHASING AN LC.	Figure 8-	15	Selecting	а	VPC	when	purchasing	an	ECS
--	-----------	----	-----------	---	-----	------	------------	----	-----

< Buy ECS						
(1) Configure Basic Settings	Configure Network	- (3) Configure Advanced Se	attings ④ Confi	m		
Network	vpc-hpc-slurm Create VPC	• C subnet-slurm	×	C Automatically assign IP address	▼ Available private IP a	addresses: 250 🕥
Extension NIC	Add NIC NICs you can still add: 1					
Security Group	default Similar to a freeval, a security group togical Ensure that the selected security group alow Security Group Rules A Inbound Rules Outbound Rules	v controls network access. Is access to port 22 (SSH-base	C Create Security Group	③ Iogin), and ICMP (ping operation). Configure	) Security Group Rules	
	Security Group Name	Priority	Action	Protocol & Port (?)	Туре	Source ⑦
		100	Permit	TCP: 3389	IPv4	0.0.0/0
		100	Permit	TCP: 22	IPv4	0.0.0/0
	detault	100	Permit	All	IPv4	default
		100	Permit	All	IPv6	default

## Figure 8-16 VPC

ECS Information			
ID			
Name	ecs-zxy 🖉		
Region	North-Ulangab203		
AZ	AZ1		
Specifications	General computing   2 vCPUs   16 GiB   m2.large.8		
Image	CentOS 8.0 64bit for Tenant 20210227   Public image		
VPC	vpc-hpc-slurm		
Billing Mode	Yearly/Monthly		
Order			
Obtained	Mar 02, 2023 16:40:41 GMT+08:00		
Launched	Mar 02, 2023 16:40:56 GMT+08:00		
Expires On	Apr 02, 2023 23:59:59 GMT+08:00		

**Step 3** Obtain the IP address and port number of the real-time service.

GUI software, for example, Postman can be used to obtain the IP address and port number. Alternatively, log in to the ECS, create a Python environment, and execute code to obtain the service IP address and port number.

#### API:

GET /v1/{project\_id}/services/{service\_id}/predict/endpoints?type=host\_endpoints

Method 1: Obtain the IP address and port number using GUI software.

#### Figure 8-17 Example response

GET + https://	/predict/endpoints?type=hast_endpoints		Send • Save •
Params  Authorization Headers (9) Body Pre-request Script Tests			Cookles Code Comments (0)
▼ Headers (2)			
KEY	VALUE	DESCRIPTION	••• Bulk Edit Presets 💌
Content-Type	application/json		
X-Auth-Token	{{TOKEN}}		
Key	Value	Description	
<ul> <li>Temporary Headers (7)</li> </ul>			
Body Cookies (2) Headers (13) Test Results		Status: 200 OK Time: 1518n	s Size: 667 B Save Response 👻
Pretty Raw Preview JSON *			Q
1 2 *****/calime** "service-tay-test", ************************************			I

Method 2: Obtain the IP address and port number using Python.

The following parameters in the Python code below need to be modified:

- project\_id: your project ID. To obtain it, see "Common Parameters" > "Obtaining a Project ID and Name" in *ModelArts API Reference*.
- **service\_id**: service ID, which can be viewed on the service details page.
- REGION\_ENDPOINT: service endpoint. To obtain it, see "Before You Start" > "Endpoints" in ModelArts API Reference.

```
def get_app_info(project_id, service_id):
  list_host_endpoints_url = "{}/v1/{}/services/{}/predict/endpoints?type=host_endpoints"
  url = list_host_endpoints_url.format(REGION_ENDPOINT, project_id, service_id)
  headers = {'X-Auth-Token': X_Auth_Token}
  response = requests.get(url, headers=headers)
  print(response.content)
```

**Step 4** Access the service through the IP address and port number.

Log in to the ECS and access the real-time service either by running Linux commands or by creating a Python environment and executing Python code. Obtain the values of **schema**, **ip**, and **port** from **3**.

Run the following command to access the real-time service: curl --location --request POST 'http://192.168.205.58:31997' \ --header 'Content-Type: application/json' \ --data-raw '{"a":"a"}'

Figure 8-18 Accessing a real-time service

```
root@ecs-zxy ~]# curl --location --request POST 'http://192.168.205.58:31997'
--header 'Content-Type: application/json' \
--data-raw '{"a":"a"}'
[root@ecs-zxy
call Post()[root@ecs-zxy ~]# _
```

Create a Python environment and execute Python code to access the realtime service.

def vpc\_infer(schema, ip, port, body): infer\_url = "{}://{}:{}"

```
url = infer_url.format(schema, ip, port)
response = requests.post(url, data=body)
print(response.content)
```

#### **NOTE**

High-speed access does not support load balancing. You need to customize load balancing policies when you deploy multiple instances.

----End

#### 8.3.1.4.4 Accessing a Real-Time Service Through WebSocket

#### Context

WebSocket is a network transmission protocol that supports full-duplex communication over a single TCP connection. It is located at the application layer in the OSI model. The WebSocket communication protocol was established by IETF as standard RFC 6455 in 2011 and supplemented by RFC 7936. The WebSocket API in the Web IDL is standardized by W3C.

WebSocket simplifies data exchange between the client and server and allows the server to proactively push data to the client. In the WebSocket API, if the initial handshake between the client and server is successful, a persistent connection can be established between them and bidirectional data transmission can be performed.

#### Prerequisites

- A real-time service has been deployed with **WebSocket** enabled.
- The image for importing the AI application is WebSocket-compliant.

## Constraints

- WebSocket supports only the deployment of real-time services.
- WebSocket supports only real-time services deployed using AI applications imported from custom images.

#### Calling a WebSocket Real-Time Service

WebSocket itself does not require additional authentication. ModelArts WebSocket is WebSocket Secure-compliant, regardless of whether WebSocket or WebSocket Secure is enabled in the custom image. WebSocket Secure supports only one-way authentication, from the client to the server.

You can use one of the following authentication methods provided by ModelArts:

Access Authenticated Using a Token

The following section uses GUI software Postman for prediction and token authentication as an example to describe how to call WebSocket.

- 1. Establish a WebSocket connection.
- 2. Exchange data between the WebSocket client and the server.

**Step 1** Establish a WebSocket connection.

in

1. Open Postman of a version later than 8.5, for example, 10.12.0. Click the upper left corner and choose **File** > **New**. In the displayed dialog box, select **WebSocket Request** (beta version currently).



Figure 8-19 WebSocket Request

2. Configure parameters for the WebSocket connection.

Select **Raw** in the upper left corner. Do not select **Socket.IO** (a type of WebSocket implementation, which requires that both the client and the server run on **Socket.IO**). In the address box, enter the **API Address** obtained on the **Usage Guides** tab on the service details page. If there is a finer-grained URL in the custom image, add the URL to the end of the address. If **queryString** is available, add this parameter in the **params** column. Add authentication information into the header. The header varies depending on the authentication mode, which is the same as that in the HTTPS-compliant inference service. Click **Connect** in the upper right corner to establish a WebSocket connection.

#### Figure 8-20 Obtaining the API address

Usage Guides	Prediction	Configuration Updates	Monitoring	Events	Logs	Tags
API Address	wss://inferen	/infers/90734aa0	)-3ad4-41f9 □			

#### **NOTE**

- If the information is correct, **CONNECTED** will be displayed in the lower right corner.
- If establishing the connection failed and the status code is 401, check the authentication.
- If a keyword such as WRONG\_VERSION\_NUMBER is displayed, check whether the port configured in the custom image is the same as that configured in WebSocket or WebSocket Secure.

The following shows an established WebSocket connection.

#### Raw v wss://inference.ulanqab.huawei.com/v1/infers/90734aa0-3ad4-41f9-9f16-27254d40548a Docs 7 Feedback 7 🖺 Save 🗸 Disconnect wss://inferen 5489 age Params Headers Settings Headers © 6 hidden Description Key Value ··· Bulk Edit X-Auth-Token Connected Search All Messages v 🗍 Clear Messages E E 3 19:10:00 v Connected to wss://im

#### Figure 8-21 Connection established

#### NOTICE

Preferentially check the WebSocket service provided by the custom image. The type of implementing WebSocket varies depending on the tool you used. Possible issues are as follows: A WebSocket connection can be established but cannot be maintained, or the connection is interrupted after one request and needs to be reconnected. ModelArts only ensures that it will not affect the WebSocket status in a custom image (the API address and authentication mode may be changed on ModelArts).

**Step 2** Exchange data between the WebSocket client and the server.

After the connection is established, WebSocket uses TCP for full-duplex communication. The WebSocket client sends data to the server. The implementation types vary depending on the client, and the lib package may also be different for the same language. Different implementation types are not considered here.

The format of the data sent by the client is not limited by the protocol. Postman supports text, JSON, XML, HTML, and Binary data. Take text as an example. Enter the text data in the text box and click **Send** on the right to send the request to the server. If the text is oversized, Postman may be suspended.

Figure 8-22 Sending data



----End

## 8.3.1.4.5 Server-Sent Events

#### Context

Server-Sent Events (SSE) is a server push technology enabling a server to push events to a client via an HTTP connection. This technology is usually used to enable a server to push real-time data to a client, for example, a chat application or a real-time news update.

SSE primarily facilitates unidirectional real-time communication from the server to the client, such as streaming ChatGPT responses. In contrast to WebSockets, which provide bidirectional real-time communication, SSE is designed to be more lightweight and simpler to implement.

## Prerequisites

The image for importing the AI application is SSE-compliant.

## Constraints

- SSE supports only the deployment of real-time services.
- It supports only real-time services deployed using AI applications imported from custom images.

## Calling an SSE Real-Time Service

The SSE protocol itself does not introduce new authentication mechanisms; it relies on the same methods as HTTP requests.

You can use one of the following authentication methods provided by ModelArts:

#### • Access Authenticated Using a Token

The following section uses GUI software Postman for prediction and token authentication as an example to describe how to call an SSE service.

	🖺 Save 🗸 🥖 [
Post v https:// //Viriders/	Send
Params Authorization Headers (11) Body • Pre-request Script Tests Settings	Cool
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ∨	Beauti
1 EB	

#### Figure 8-23 Calling an SSE service

tody Cookies (2) Headers (12) Test Results	Status: 200 DK Time: 10.22 s Size: 573 B Save as example
Search 📋 Clear Messages	
() Connection closed	09:35:05
↓ sse	09:35:05 🚿
↓ sse	09:35:05 🥆
↓ sse	09:35:04 🥆
↓ sse	09:35:04 🥆
↓ see	09:35:04 🥆
↓ sse	09:35:04 🥆
↓ sse	09:35:04 🥆
① Connected to	09:35:04

#### Figure 8-24 Response header Content-Type

I	Body	Cookies Headers (_)	Test Results	
l		KEY		VALUE
I				
		Content-Type (3)		text/event-stream;charset=UTF-8

#### **NOTE**

In normal cases, the value of **Content-Type** in the response header is **text/event-stream;charset=UTF-8**.

## 8.3.1.5 Cloud Shell

#### **Scenarios**

You can use Cloud Shell provided by the ModelArts console to log in to a running real-time service instance container.

## Constraints

- Cloud Shell can only access a container when the associated real-time service is deployed within a dedicated resource pool
- Cloud Shell can only access a container when the associated real-time service is running.

## **Using Cloud Shell**

- 1. Log in to the ModelArts console. In the navigation pane, choose **Service Deployment > Real-Time Services**.
- 2. On the real-time service list page, click the name or ID of the target service. The real-time service details page is displayed.
- 3. Click the **Cloud Shell** tab and select the AI application version and compute

node. When the connection status changes to  $\heartsuit$  , you have logged in to the instance container.

If the server disconnects due to an error or remains idle for 10 minutes, you can select **Reconnect** to regain access to the container instance.

**NOTE** 

An exception may occur when some users log in to the Cloud Shell page. Click **Enter** to rectify the fault.

Figure 8-25 Abnormal path

ind/model/1\$ @97c6-b87f-4410-9f74-18a8b1d0ff9d-59x451kz-6548f94565-1rjgs:/home/mi

# 8.3.2 Deploying AI Applications as Batch Services

## 8.3.2.1 Deploying as a Batch Service

After an AI application is prepared, you can deploy it as a batch service. The **Service Deployment > Batch Services** page lists all batch services.

## Prerequisites

- A ModelArts application in the **Normal** state is available.
- Data to be batch processed is ready and has been upload to an OBS directory.
- At least one empty folder has been created in OBS for storing the output.

## Context

- A maximum of 1,000 batch services can be created.
- Based on the input request (JSON or file) defined by the AI application, different parameters are entered. If the AI application input is a JSON file, a configuration file is required to generate a mapping file. If the AI application input is a file, no mapping file is required.
- Batch services can only be deployed in a public resource pool, but not a dedicated resource pool.

## Procedure

- Log in to the ModelArts management console. In the left navigation pane, choose Service Deployment > Batch Services. By default, the Batch Services page is displayed.
- 2. In the batch service list, click **Deploy** in the upper left corner. The **Deploy** page is displayed.

- 3. Set parameters for a batch service.
  - a. Set the basic information, including **Name** and **Description**. The name is generated by default, for example, **service-bc0d**. You can specify **Name** and **Description** according to actual requirements.
  - b. Set other parameters, including AI application configurations. For details, see **Table 8-20**.

Table 8-20         Parameters
-------------------------------

Parameter	Description	
AI Application Source	Select <b>My AI Applications</b> based on your requirements.	
AI Application and Version	Select an AI application and version that are running properly.	
Input Path	Select the OBS directory where the uploaded data is stored. Select a folder or a .manifest file. For details about the specifications of the .manifest file, see Manifest File Specifications.	
	NOTE	
	<ul> <li>If the input data is an image, ensure that the size of a single image is less than 10 MB.</li> </ul>	
	<ul> <li>If the input data is in CSV format, ensure that no Chinese character is included.</li> </ul>	
	<ul> <li>If the input data is in CSV format, ensure that the file size does not exceed 12 MB.</li> </ul>	
Request Path	URL used for calling the AI application API in a batch service, and also the request path of the AI application service. Its value is obtained from the <b>url</b> field of <b>apis</b> in the AI application configuration file.	

Parameter	Description
Mapping Relationship	If the AI application input is in JSON format, the system automatically generates the mapping based on the configuration file corresponding to the AI application. If the AI application input is other file, mapping is not required.
	Automatically generated mapping file. Enter the field index corresponding to each parameter in the CSV file. The index starts from 0.
	Mapping rule: The mapping rule comes from the input parameter ( <b>request</b> ) in the model configuration file <b>config.json</b> . When <b>type</b> is set to <b>string, number, integer, or boolean</b> , you are required to set the index parameter. For details about the mapping rule, see <b>Example Mapping</b> . The index must be a positive integer starting from 0. If the value of index does not comply with the rule, this parameter is ignored in the request. After the mapping rule is configured, the corresponding CSV data must be separated by commas (,).
Output Path	Select the path for saving the batch prediction result. You can select the empty folder that you create.
Specifications	Select available specifications based on the list displayed on the console. The specifications in gray cannot be used at the current region.
Compute Nodes	Set the number of instances for the current Al application version. If you set the number of nodes to <b>1</b> , the standalone computing mode is used. If you set the number of nodes to a value greater than 1, the distributed computing mode is used. Select a computing mode based on the actual requirements.
Environment Variable	Set environment variables and inject them to the pod. To ensure data security, do not enter sensitive information in environment variables.
Timeout	Timeout of a single model, including both the deployment and startup time. The default value is 20 minutes. The value must range from 3 to 120.

4. After setting the parameters, deploy the model as a batch service as prompted. Deploying a service generally requires a period of time, which may be several minutes or tens of minutes depending on the amount of your data and resources.

You can go to the batch service list to view the basic information about the batch service. In the batch service list, after the status of the newly deployed
service changes from **Deploying** to **Running**, the service is deployed successfully.

#### **Manifest File Specifications**

ModelArts batch services support manifest files, which describe data input and output.

#### Example input manifest file

- File name: **test.manifest**
- File content:
   {"source": "obs://test/data/1.jpg"}
   {"source": "s3://test/data/2.jpg"}
   {"source": "s3://test/data/2.jpg"}
   {"source": "https://infers-data.obs.xxx.com:443/xgboosterdata/data.csv?
   AccessKeyId=2Q0V0TQ461N26DDL18RB&Expires=1550611914&Signature=wZBttZj5QZrReDhz1uDzwve
   8GpY%3D&x-obs-security-token=gQpzb3V0aGNoaW5hixvY8V9a1SnsxmGoHYmB1SArYMyqnQT ZaMSxHvl68kKLAy5feYvLDM..."}
- Requirements on the file:
  - a. The file name extension must be .manifest.
  - b. The file content is in JSON format. Each row describes a piece of input data, which must be accurate to a file instead of a folder.
  - c. The value of **source** is the OBS file path in the format of *OBS path*{{Bucket name}}/{{Object name}}.

#### Example output manifest file

A manifest file will be generated in the output directory of the batch services.

• Assume that the output path is **//test-bucket/test/**. The result is stored in the following path:



#### • Content of the infer-result-0.manifest file:

{"source": "obs://obs-data-bucket/test/data/1.jpg","result":"SUCCESSFUL","inference-loc": "obs://test-bucket/test/infer-result/1.jpg\_result.txt"}

{"source": "s3://obs-data-bucket/test/data/2.jpg","result":"FAILED","error\_message": "Download file failed."}

{"source ": "https://infers-data.obs.xxx.com:443/xgboosterdata/2.jpg?

AccessKeyId=2Q0V0TQ461N26DDL18RB&Expires=1550611914&Signature=wZBttZj5QZrReDhz1uDzwve 8GpY%3D&x-obs-security-token=gQpzb3V0aGNoaW5hixvY8V9a1SnsxmGoHYmB1SArYMyqnQT-ZaMSxHvl68kKLAy5feYvLDMNZWxzhBZ6Q-3HcoZMh9gISwQOVBwm4ZytB\_m8sg1fL6isU7T3CnoL9jmv DGgT9VBC7dC1EyfSJrUcqfB\_N0ykCsfrA1Tt\_IQYZFDu\_HyqVk-

GunUcTVdDfWlCV3TrYcpmznZjliAnYUO89kAwCYGeRZsCsC0ePu4PHMsBvYV9gWmN9AUZIDn1sfRL4vo BpwQnp6tnAgHW49y5a6hP2hCAoQ-95SpUriJ434QlymoeKfTHVMKOeZxZea-

JxOvevOCGI5CcGehEJaz48sgH81UiHzl21zocNB\_hpPfus2jY6KPglEJxMv6Kwmro-

ZBXWuSJUDOnSYXI-3ciYjg9-

h10b8W3sW1mOTFCWNGoWsd74it7L5-7UUholeyPByO\_REwkur2FOJsuMpGlRaPyglZxXm\_jfdLFXobYtz Zhbul4yWXga6oxTOkfcwykTOYH0NPoPRt5MYGYweOXXxFs3d5w2rd0y7p0QYhyTzlkk5Clz7FlWNapFISL 7zdhsl8RfchTqESq94KgkeqatSF\_ilvnYMW2r8P8x2k\_eb6NJ7U\_q5ztMbO9oWEcfr0D2f7n7BLnb2HIB\_H9tj zKvqwngaimYhBbMRPfibvttW86GiwVP8vrC27FOn39Be9z2hSfJ\_8pHej0yMlyNqZ481FQ5vWT\_vFV3JHM-7I1ZB0\_hIdaHfItm-J69cTfHSEOzt7DGaMIES1o7U3w%3D%3D","result":"SUCCESSFUL","inference-loc": "obs://test-bucket/test/infer-result/2.jpg\_result.txt"}

- File format:
  - a. The file name is **infer-result-{{task\_id}}.manifest**, where **task\_id** is the batch task ID, which is unique for a batch service.

- b. If a large number of files need to be processed, multiple manifest files may be generated with the same suffix **.manifest** and are distinguished by suffix, for example, **infer-result-{{task\_id}}\_1.manifest**.
- c. The **infer-result**-*{{task\_id}}* directory is created in the manifest directory to store the file processing result.
- d. The file content is in JSON format. Each row describes the output result of a piece of input data.
- e. The file contains multiple fields:
  - i. **source**: input data description, which is the same as that of the input manifest file
  - ii. result: file processing result, which can be SUCCESSFUL or FAILED
  - iii. inference-loc: output result path. This field is available when result is **SUCCESSFUL**. The format is **obs://{{***Bucket name***}***}***/**{*Object name***}**.
  - iv. error\_message: error information. This field is available when the result is FAILED.

### **Example Mapping**

The following example shows the relationship between the configuration file, mapping rule, CSV data, and inference request.

The following uses a file for prediction as an example:



} } ]

The ModelArts management console automatically resolves the mapping relationship from the configuration file as shown below. When calling a ModelArts API, configure the mapping by following the rule.

```
{
   "type": "object",
   "properties": {
      "data": {
         "type": "object",
         "properties": {
             "req_data": {
                "type": "array",
"items": [
                   {
                       "type": "object",
                       "properties": {
                          "input_1": {
                             "type": "number",
"index": 0
                        },
"input_2": {
"type": "number",
"index": 1
                          "input_3": {
"type": "number",
                             "index": 2
                          },
"input_4": {
                              "type": "number",
                              "index": 3
                          }
   }
}
}
                      }
  }
}
```

Multiple pieces of CSV data for inference are separated by commas (,) The following shows an example:

5.1,3.5,1.4,0.2 4.9,3.0,1.4,0.2 4.7,3.2,1.3,0.2

Depending on the defined mapping relationship, the inference request is shown below, whose format is similar to that for real-time services.

```
{
    "data": {
        "req_data": [{
            "input_1": 5.1,
            "input_2": 3.5,
            "input_3": 1.4,
            "input_4": 0.2
        }]
    }
}
```

# 8.3.2.2 Viewing the Batch Service Prediction Result

When deploying a batch service, you can select the location of the output data directory. You can view the running result of the batch service that is in the **Completed** status.

#### Procedure

- Log in to the ModelArts management console and choose Service Deployment > Batch Services.
- 2. Click the name of the target service in the **Completed** status. The service details page is displayed.
  - You can view the service name, status, ID, input path, output path, and description.
  - You can click solution in the Description area to edit the description.
- 3. Obtain the detailed OBS path next to **Output Path**, switch to the path and obtain the batch service prediction results, including the prediction result file and the AI application prediction result.

If the prediction is successful, the directory contains the prediction result file and AI application prediction result. Otherwise, the directory contains only the prediction result file.

- Prediction result file: The file is in xxx.manifest format, which contains the file path and prediction result, and more.
- AI application prediction result:
  - If images are input, a result file is generated for each image in the Image name\_result.txt format, for example, IMG\_20180919\_115016.jpg\_result.txt.
  - If audio files are input, a result file is generated for each audio file in the *Audio file name*\_result.txt format, for example, 1-36929-A-47.wav\_result.txt.
  - If table data is input, the result file is generated in the *Table name\_result.txt* format, for example, *train.csv\_result.txt*.

# 8.3.3 Deploying AI Applications as Edge Services

### 8.3.3.1 Deploying an Edge Service

You can deploy an AI application as an edge service. The **Service Deployment** > **Edge Services** page lists all edge services.

#### Prerequisites

- A ModelArts application in the **Normal** state is available.
- An edge resource pool is available if you want to use one to deploy an edge service. For details, see **Creating an Edge Resource Pool**.

# Background

A maximum of 1,000 edge services can be deployed.

### Deploying an Edge Service (Synchronous Request)

- Log in to the ModelArts console. In the navigation pane, choose Service Deployment > Edge Services. By default, the Edge Services page is displayed.
- 2. In the edge service list, click **Deploy** in the upper left corner. The **Deploy** page is displayed.
- 3. Set edge service parameters.
  - a. Set the basic information, including **Name** and **Description**. The name is generated by default, for example, **service-bc0d**. You can specify **Name** and **Description** based on your actual needs.
  - b. Set other parameters, including the resource pool and AI application configurations. For details, see **Table 8-21**.

Parameter	Description
Deployment Mode	Only edge resource pool is supported.
Deployment Instances	Set the number of deployment instances.
Select Edge Resource Pool	Select an edge resource pool.
AI Application and Configuration	Select an AI application and set parameters. For details, see <b>Table 8-22</b> .

Table 8-21Parameters

Table 8-22	<b>AI Application</b>	and Configuration
------------	-----------------------	-------------------

Parameter	Description
AI Application Source	Select My AI Applications.
AI Application and Version	Select an AI application and version that are running properly.
Specifications	Select available specifications based on the list displayed on the console. The specifications in gray cannot be used.
Environment Variable	Set environment variables and inject them to the pod. To ensure data security, do not enter sensitive information in environment variables.

Parameter	Description
Network Configuration	This parameter is displayed when <b>Deployment</b> <b>Mode</b> is set to <b>Edge Resource Pool</b> and a pool of ModelArts edge nodes is selected. Only <b>Port</b> <b>Mapping</b> is supported. If you use this access method, you must configure the container port, host NIC address, and host port information. You can either choose a host port or let it be assigned automatically. If you use an assigned port, you need to define the minimum and maximum values for the host port.
Volumes	This parameter is displayed when <b>Deployment</b> <b>Mode</b> is set to <b>Edge Resource Pool</b> and a pool of ModelArts edge nodes is selected. You need to set volume type, volume name, disk source, mount path, storage media, and permission. The volume type can be host path, temporary path, or NFS. <b>NOTE</b> To use NFS, install the NFS service on related nodes beforehand. For details, see <b>Installing and Configuring</b> <b>NFS</b> .

4. Click **Create now**. Deploying a service generally requires a period of time, which may be several minutes or tens of minutes depending on the amount of your data and resources.

You can go to the edge service list to check the status of the edge service. In the edge service list, after the status of the service changes from **Deploying** to **Running**, the service is deployed. In the edge service list, you can view the request mode and deployment mode of the edge service.

# Deploying an Edge Service (Asynchronous Request)

- Log in to the ModelArts console. In the navigation pane, choose Service Deployment > Edge Services. By default, the Edge Services page is displayed.
- 2. In the edge service list, click **Deploy** in the upper left corner. The **Deploy** page is displayed.
- 3. Set edge service parameters.

a. Set the basic information, including **Name** and **Description**. The name is generated by default, for example, **service-bc0d**. You can enter **Name** and **Description** as needed.

b. Set other parameters, including the resource pool and AI application configurations. For details, see **Table 8-23**.

#### Table 8-23 Parameters

Parameter	Description
Select Edge Resource Pool	In the edge resource pool list, select a running resource pool. For details about how to create an edge resource pool, see <b>Creating an Edge Resource Pool</b> .
AI Application Source	Select My AI Applications.
AI Application and Version	Select an AI application and version that are running properly.
Streams	Number of video streams that can be concurrently processed.
Specifications	Select available specifications based on the list displayed on the console. The specifications in gray cannot be used.
Deployment Instances	Set the number of instances for the current Al application version. For example, <b>1</b> indicates one compute node is used. Enter a value based on actual requirements.
	<b>NOTICE</b> To avoid deployment failure caused by traffic limiting, do not exceed 10 instances when deploying or modifying an edge service.
Service Startup Parameters	This parameter is available when you select an AI application with service startup parameters configured during creation. Configure service startup parameters you added during AI application creation.

4. Click **Create now**. Deploying a service generally requires a period of time, which may be several minutes or tens of minutes depending on the amount of your data and resources.

You can go to the edge service list to check the status of the edge service. In the edge service list, after the status of the service changes from **Deploying** to **Running**, the service is deployed.

### 8.3.3.2 Accessing an Edge Service Deployed on IEF Edge Nodes

If the edge service and edge node are in the **Running** status, the edge service has been successfully deployed on the edge node.

The following methods describe how to perform predictions on the edge service deployed in an edge pool.

- Method 1: Use GUI-based Software for Inference (Postman)
- Method 2: Run the cURL Command to Send an Inference Request

The methods in this section apply only to edge services deployed on IEF edge nodes.

# Method 1: Use GUI-based Software for Inference (Postman)

- 1. Download Postman and install it, or add a Postman extension in Chrome. Alternatively, use other software that can send POST requests.
- 2. Open Postman. Figure 8-26 shows the Postman interface.

#### Figure 8-26 Postman software interface

NEW I Runner Impe	nt 다			Builder T				× (		lyalice 🗸	0	• •	۲
		Chrome ap	ps are being deprecated. <u>Downlo</u>	d our free native a	pps for continued si	upport and better p	performance. <u>Learn more</u>						×
Q Filter	New Tab								No Envir	onment		v ©	⇔
History Collections	GET $ \lor $	Enter request URL							Params	Sen	d ×	Save	~
	Authorization	Headers Body Pre-request Sc	ript Tests										Code
Nothing in your history yet. Requests that you send through Postman are automatically saved here.	Туре		No Auth		~								
	Response												
				Hit the	Send buttor	n to get a res	sponse.						
					Do more wi	th requests							
				Share	Mock	Monitor	Document						
				<		-Ay							

- 3. Set parameters on Postman. The following uses image classification as an example.
  - Select a POST task and copy the call address of an edge instance to the box next to POST. To obtain the call address, go to the instances list tab on the edge service details page to check the URL, click the node to access the node details page, and view the IP address on the dashboard. IPv4 and IPv6 addresses are supported.

#### Figure 8-27 POST parameter settings

PO	ST	https://	192.168.0.158:103	2			
Para	ims	Authorization	Headers (9)	Body ●	Pre-request Script	Tests	Settings
Hea	ders	9 hidden					
	KEY					VA	LUE

- On the **Body** tab page, input parameters are divided into file input and text input types.
  - File input

Select **form-data**. Set **KEY** to the input parameter of the AI application, for example, **images**. Set **VALUE** to an image to be inferred (only one image can be inferred).



#### Text input

Select **raw** and then select **JSON(application/json)**. Enter the request body in the text box below. An example request body is as follows:

```
{
"meta": {
"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
},
"data": {
"req_data": [
{
"sepal_length": 3,
"sepal_width": 1,
"petal_length": 2.2,
"petal_width": 4
}
]
}
```

**uuid** can be specified in **meta**. When the inference result is returned, this **uuid** is returned to trace the request. You can leave **meta** blank based on your needs. **data** contains **req\_data**, where you can input one or more pieces of request data. The parameters of each piece of data are determined by the AI application, such as **sepal\_length** and **sepal\_width** in this example.

- 4. After setting the parameters, click **Send** to send the request. The result is displayed in the response.
  - For a file input, the returned result varies depending on the AI application.
  - The request body contains meta and data. If the request contains uuid, uuid will be returned in the response. Otherwise, uuid is left blank. data contains the req\_data array. You can pass one or more pieces of request data. The parameters of each piece of data are determined by the model, such as sepal\_length and sepal\_width in this example.

#### Method 2: Run the cURL Command to Send an Inference Request

The command for sending inference requests can be input as a file or text.

1. File input

curl -F 'images=@Image path'-X POST Call address of the edge instance -k

 - -F indicates file input. In this example, the parameter name is images, which can be changed as required. The image storage path follows @. - **POST** is followed by the call address of the edge instance.

The following is an example of the cURL command for inference with file input:

curl -F 'images=@/home/data/cat.jpg' -X POST https://192.168.0.158:1032 -k

Figure 8-29 shows the inference result.

Figure 8-29 Inference result using the cURL command with file input

```
rootamodelarts006;/# curl -F 'images=0/home/data/cat.jpg' -X POST https://192.168.0.158:1032 -k

{*confidences": [[0.33620707154273887, 0.2228349424454652, 0.14882247352600098, 0.10647343059017279, 0.0678214889764785

8]], *logits": [[-0.06549632132053375, 0.6510115265846252, -0.17024759458137512, 0.25605931878089905, -0.175674393773078

92, 1.0341405988530273]], *labels*: [[5, 1, 3, 0, 2]]}rootamodelarts006:/#
```

```
2. Text input
```

```
curl -d '{
"meta": {
"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
},
"data": {
"req_data": [
{
"sepal_length": 3,
"sepal_width": 1,
"petal_length": 2.2,
"petal_width": 4
}
]
}/ -X POST <Call address of the edge instance> -k
```

 - d indicates the text input of the request body. If the AI application uses text input, this parameter is mandatory.

The following is an example of the cURL command for inference with text input:

```
curl -d '{
"meta": {
"uuid": "10eb0091-887f-4839-9929-cbc884f1e20e"
},
"data": {
"req_data": [
{
"sepal_length": 3,
"sepal_width": 1,
"petal_length": 2.2,
"petal_width": 4
}
]
}
' -X POST https://192.168.0.158:1033 -k
```

# 8.3.3.3 Accessing an Edge Service Deployed in a ModelArts Edge Resource Pool

If the edge service and edge node are in the **Running** status, the edge service has been successfully deployed on the edge node.

You can use either of the following methods to send a prediction request to an edge service deployed in an edge resource pool:

- Method 1: Use GUI-based Software for Inference (Postman)
- Method 2: Run the cURL Command to Send an Inference Request

#### **NOTE**

The methods in this section apply only to edge services deployed on ModelArts edge nodes.

### Method 1: Use GUI-based Software for Inference (Postman)

- 1. Download Postman and install it, or add a Postman extension in Chrome. Alternatively, use other software that can send POST requests.
- 2. Open Postman. Figure 8-30 shows the Postman interface.

#### Figure 8-30 Postman software interface

NEW C Runner Imp	art 📑			Builder To					🕱 🧿 conn				<b>L</b> V
		Chrome apps	are being deprecated. Download	our free native ap	pps for continued si	apport and better j	performance. <u>Learn mor</u>	1					×
Q. Filter	New Tab									No Environme	ent	~	⇒ \$
History Collections	GET 🗸	Enter request URL								Params	Send	Sav	e ~
	Authorization	Headers Body Pre-request Scrip	t Tests										Code
Nothing in your history yet. Requests that you send through Postman are automatically saved here.	Туре		No Auth		~								
	Response												
				Hit the	Send buttor	n to get a re	sponse.						
					Do more wi	th requests							
				Share	Mock	Monitor	Document						
				<		-Ay	đ						

- 3. Set parameters on Postman. The following uses intelligent Q&A as an example.
  - a. Select a POST task and copy the edge instance URL to the POST text box.

View the URL on the **Load Edge Balance** tab of edge services. Check the forwarding configuration of the resource pool and enter the URL based on the listener port and path matching rule. The IP address is that of the master node. You can log in to the Linux interface of the master node and run the **ifconfig** command to view the IP address.

#### Figure 8-31 POST parameter settings

POST		~ http:	s://100.85.	: <u>2035</u> 0/ <u>v1/</u> ;	3a1bb	4b	8492e/moc	lels/gpt/query			Send	~
Params	a Aut	Cont thorization	Headers (9)	Port Body ●	Pre-request Script	Tests	Settings	Path			Co	okies
Heade	ers ø	Hide auto-	generated headers	3								
	KEY			١	/ALUE			DESCRIPTIO	000	Bulk Edit	Presets	~

b. On the **Body** tab, enter the content to be predicted based on the usage of the AI application.

Select **raw** and then select **JSON(application/json)**. Enter the request body in the text box below. An example request body is as follows:

"data":["Who are you?"]
}

In the preceding request body, **data** contains an array, where you can input one or more pieces of request data.

4. After setting the parameters, click **Send** to send the request. The result is displayed in the response. The prediction result may vary according to AI applications.

### Method 2: Run the cURL Command to Send an Inference Request

The following is the **curl** command format: curl --location --request POST *<Edge instance URL>* --header 'Content-Type: application/json' --data-raw '{ "data":["Who are you?"] }' -k

In the preceding request body, enter the text input of the AI application in --**data**-**raw**.

The following is an example:

```
curl --location --request POST 'https://100.85.xxx.xxx:20350/v1/3a1bb61cc35e41bc9466a90164b8492e/
models/gpt/query'
--header 'Content-Type: application/json'
--data-raw '{
"data":["Who are you?"]
}' -k
```

Figure 8-32 shows the prediction result.

Figure 8-32 Inference result using the cURL command with text input

[root@ecs-edge-master-57f1-1746 ~]# curl --location --request POST 'https://100.85. ....:20350/v1/3albb61cc35e41bc9466a90164b
8492e/models/gpt/query' --header 'Content-Type: application/json' --data-raw '{
 "data':"who are you?"]
}' -k
{"answers":[{"avg\_log\_prob":0,"content":" Hello! I am an AI language model called Assistant. I am here to help you with any que
stions or tasks you might have. I have been trained on a wide range of topics and can provide information on many different sub
jects. Please let me know how I can assist you today.","index":0,"ppl":0,"tokens":56],"message\_id":"","model":{"decode\_strateg
y:("beams size":1,"do\_sample":true,"max\_output tokens":4095,"name":
"gpt","version":"8a0189b2-67dc-43b1-8d1b-1ddbb825302e"},"tokens":78}

### 8.3.3.4 Load Balancing

ModelArts allows you to enable load balancing for either an edge node or edge resource pool that you specify. ModelArts monitors the port numbers of the nodes' physical IP addresses and uses the round robin method to forward requests to the appropriate edge service and access port for processing, according to the forwarding configuration that you set up. This improves the application reliability and stability.

A ModelArts edge node or edge resource pool must be available for creating a load balancer. For details about how to create a ModelArts edge node, see **Creating an Edge Node**. For details about how to create an edge resource pool, see **Creating an Edge Resource Pool**.

### **Creating a Load Balancer**

- 1. Log in to the ModelArts console and choose **Edge Services** from the navigation pane.
- 2. Click the Load Edge Balance tab and click Create.
- 3. Set the parameters by referring to the following table.

Parameter	Description
Name	Load balancer name, which cannot be modified after creation.
Description	Brief description of the load balancer.
Config Target	Object of load balancing, which can be a single node or a specified resource pool.
	Node: Select a ModelArts edge node.
	Resource Pool: Select an edge resource pool.
Listener Port	Port number of the physical IP address of the node or resource pool that you want to listen to. The value ranges from <b>1</b> to <b>65,535</b> .
Lb Edge Policy	Load balancing scheduling algorithm for a resource pool. Round-Robin is a method of distributing requests evenly among a set of nodes, from the first node to the Nth node in a circular order.
Sticky Session	LB listening relies on IP address-based sticky session. When sticky session is enabled, access requests from the same IP address are always forwarded to the same edge service.
Protocol	Protocol used by the load balancer. The value can be <b>HTTP</b> or <b>HTTPS</b> . Currently, only <b>HTTP</b> is supported.

 Table 8-24 Parameters for creating a load balancer

Parameter	Description
Forward Config	Set the forwarding configuration. When the access address of a request matches the forwarding configuration, the request will be forwarded to the target for processing. You can <b>Add</b> or <b>Delete</b> a forwarding configuration.
	<ul> <li>If Config Target is set to Node, set the following parameters:</li> </ul>
	<ul> <li>Path Match Rule: Select a forwarding matching rule. Currently, prefix match, exact match, and regular expression match are supported.</li> </ul>
	<ul> <li>Path: Set the forwarding matching path. The path cannot be empty and must start with a slash (/). It contains 1 to 255 characters, including letters, digits, asterisks (*), slashes (/), and hyphens (-).</li> </ul>
	<ul> <li>Target Service Name: Select the target edge service.</li> </ul>
	<ul> <li>Container Port: Enter a container port.</li> </ul>
	• If <b>Config Target</b> is set to <b>Pool</b> , set the following parameters:
	<ul> <li>Path Match Rule: Select a forwarding matching rule. Currently, prefix match, exact match, and regular expression match are supported.</li> </ul>
	<ul> <li>Path: Set the forwarding matching path. The path cannot be empty and must start with a slash (/). It contains 1 to 255 characters, including letters, digits, asterisks (*), slashes (/), and hyphens (-).</li> </ul>
	<ul> <li>Target Access Port: Select the target access port.</li> </ul>
	<ul> <li>Access Port: Select the port for accessing the service.</li> </ul>

4. Confirm the configuration and click **OK**. On the displayed load balancer list page, wait until the status of the load balancer changes to **Running**. The created load balancer can be modified and deleted.

# **Creating an Access Port**

Before creating a load balancer, ensure that a ModelArts resource pool has been created and an edge service has been deployed in the ModelArts resource pool.

- 1. Log in to the ModelArts console and choose **Edge Services** in the navigation pane on the left.
- Choose Load Edge Balance > Access Edge Port and click Create in the displayed tab.
- 3. Set the parameters by referring to the following table.

Parameter	Description
Name	Name of the access port.
Access Type	Access type of the access port. Only ClusterIP is supported.
Associated Edge Service	Select the edge service associated with the access port.
Port Config	Set the port configuration, including the protocol, access port, and container port parameters. You can <b>Add</b> or <b>Delete</b> the port configuration. At least one record must be added.
	• <b>Protocol</b> : Only TCP is supported.
	• Access Port: Select the port for accessing the service.
	• <b>Container Port</b> : Select the port used by the service to access the target container. This port is related to the applications running in the container.

Table 8-25 Parameters for creating an access port

4. Confirm the configuration and click **OK**. On the displayed access port list page, wait until the status of the access port changes to **Running**. The created access port can be modified and deleted.

#### Accessing a Load Balancer

Perform this call based on the created access ports and load balancers.

```
curl -X POST \
https://100.85.220.207:13458/v1/models/gpt/query \
-d`{
"data": ["Hello"]
}`
```

In the above example, **100.85.220.207** is the IP address of the master node, **13458** is the specified listening port when the load balancer is created, and **/v1/models/gpt/query** must meet the route matching rules.

### 8.3.3.5 Installing and Configuring NFS

NFS is a data storage volume service provided by ModelArts edge resource pools. When deploying a service, you can use NFS mounting to access shared data, such as model files stored in OBS.

Configure NFS in the following scenarios:

- During AI application creation, Meta Model Source is set to OBS and AI Engine is set to Custom.
- During service deployment, NFS storage volumes are used.

#### **Installing NFS**

1. Log in to a storage node.

In the edge resource pool, select a node as the storage node. The node provides the NFS web disk service for storing files shared by the cluster. Use a node with a storage space large enough for storing large model files. Use PuTTY to log in to the storage node.

ssh <Username>@<Node IP address>

- Username: Enter the username for logging in to the server.
- Node IP address. Enter the IP address of the server you want to log in to.
   If the node is an ECS, you can obtain its IP address on the ECS console.
- 2. Install NFS.

Connect to the Internet and download dependent software packages.

– Ubuntu

Online installation: sudo apt install nfs-kernel-server

sudo apt instatt fils-kernet-se

- Euler OS

Online installation: sudo yum install nfs-utils

3. Create a model directory.

The storage space of this path must be large enough for storing large model files.

mkdir -p /var/docker/hilens

4. Add the access permissions.

Configure the nfs-server access whitelist and file storage path.

vim /etc/exports

Add the following configurations:

/var/docker/hilens 192.168.0.0/24(rw,no\_all\_squash,anonuid=1000,anongid=100,fsid=0)

**192.168.0.0/24** is the IP address range of the cluster intranet. To obtain the IP address, log in to the master node and run the **ifconfig** command.

- 5. Load the configuration. exportfs -rv
- 6. Start NFS and rpcbind. systemctl enable nfs-server && systemctl enable rpcbind && systemctl start rpcbind nfs-server
- Run this command to check whether the preceding configuration is correct. If the following information is displayed, the configuration is correct, that is, the NFS service is installed.
   showmount -e localhost



### **Configuring ModelArts Node Information**

- 1. Log in to the Linux host of the master node. ssh <*Username*>@<*Node IP address*>
  - Username: Enter the username for logging in to the server.
  - Node IP address. Enter the IP address of the server you want to log in to.
     If the node is an ECS, you can obtain its IP address on the ECS console.

2. Configure firmware boot parameters. vim /etc/hilens/hda.conf

Add the following configurations. Replace **192.168**.*xxx*.*xxx* with the actual private IP address of the NFS storage node.

hilens.nfs.server.ip=192.168.xxx.xxx hilens.nfs.mount.dir=/home/mind/model hilens.nfs.source.dir=/var/docker/hilens

Parameter description:

- hilens.nfs.server.ip: private IP address of the NFS storage node
- **hilens.nfs.mount.dir**: default mount path of a large model, that is, the access path in the container, which is determined by the image.
- hilens.nfs.source.dir: path for downloading a large model, that is, the shared directory of the storage node. Configure the share permission for the directory in /etc/exports. Otherwise, you do not have the permission to mount the directory.
- 3. Restart the firmware. systemctl restart hdad

# 8.3.4 Upgrading a Service

For a deployed service, you can modify its basic information to match service changes and change the AI application version to upgrade it.

You can modify the basic information about a service in either of the following ways:

Method 1: Modify Service Information on the Service Management Page

Method 2: Modify Service Information on the Service Details Page

#### Prerequisites

The service has been deployed. The service in the **Deploying** state cannot be upgraded by modifying the service information.

### Constraints

- Improper upgrade operations will interrupt service running during the upgrade. Therefore, exercise caution when performing this operation.
- ModelArts supports hitless rolling upgrade of real-time services in some scenarios. Before upgrade, prepare for it and confirm the prerequisites.

Meta Model Source for Creating an Al Application	Using a Public Resource Pool	Using a Dedicated Resource Pool
Training job	Not supported	Not supported
Template	Not supported	Not supported

 Table 8-26
 Scenarios for hitless rolling upgrade

Meta Model Source for Creating an Al Application	Using a Public Resource Pool	Using a Dedicated Resource Pool
Container image	Not supported	Supported. The custom image for creating an AI application must meet Custom Image Specifications for Creating AI Applications.
OBS	Not supported	Not supported

### Method 1: Modify Service Information on the Service Management Page

- 1. Log in to the ModelArts management console and choose **Service Deployment** from the left navigation pane. Go to the service management page of the target service.
- 2. In the service list, click **Modify** in the **Operation** column of the target service, modify basic service information, and submit the modification task as prompted.

When some parameters are modified, the system automatically restarts the service for the modification to take effect. When you submit a service modification task, if a restart is required, a dialog box will be displayed.

- For details about the real-time service parameters, see Deploying as a Real-Time Service. To modify a real-time service, you also need to set Max. Invalid Instances to set the maximum number of nodes that can be concurrently upgraded, during which time these nodes are invalid.
- For details about the batch service parameters, see Deploying as a Batch Service.

### Method 2: Modify Service Information on the Service Details Page

- Log in to the ModelArts management console and choose Service Deployment from the left navigation pane. Go to the service management page of the target service.
- 2. Click the name of the target service. The service details page is displayed.
- 3. Click **Modify** in the upper right corner of the page, modify the service details, and submit the modification task as prompted.

When some parameters are modified, the system automatically restarts the service for the modification to take effect. When you submit a service modification task, if a restart is required, a dialog box will be displayed.

- For details about the real-time service parameters, see Deploying as a Real-Time Service. To modify a real-time service, you also need to set Max. Invalid Instances to set the maximum number of nodes that can be concurrently upgraded, during which time these nodes are invalid.
- For details about the batch service parameters, see Deploying as a Batch Service.

# 8.3.5 Starting, Stopping, Deleting, or Restarting a Service

## **Starting a Service**

You can start services in the **Successful**, **Abnormal**, or **Stopped** status. Services in the **Deploying** state cannot be started. You can start a service in the following ways:

- Log in to the ModelArts management console and choose **Service Deployment** from the left navigation pane. Go to the service management page of the target service. Click **Start** in the **Operation** column to start the target service.
- Log in to the ModelArts management console and choose **Service Deployment** from the left navigation pane. Go to the service management page of the target service. Click the name of the target service. The service details page is displayed. Click **Start** in the upper right corner of the page to start the service.

### **Stopping a Service**

Stop a service in either of the following ways:

- Log in to the ModelArts management console and choose Service
   Deployment from the left navigation pane. Go to the service management page of the target service. Click Stop in the Operation column to stop a service. (For a real-time service, choose More > Stop in the Operation column.)
- Log in to the ModelArts management console and choose Service
   Deployment from the left navigation pane. Go to the service management page of the target service. Click the name of the target service. The service details page is displayed. Click Stop in the upper right corner of the page to stop the service.

### **Deleting a Service**

If a service is no longer in use, delete it to release resources.

Log in to the ModelArts management console and choose **Service Deployment** from the left navigation pane. Go to the service management page of the target service.

- Real-time services
  - In the real-time service list, choose More > Delete in the Operation column of the target service to delete it.
  - Select services in the real-time service list and click **Delete** above the list to delete services in batches.
  - Click the name of the target service. On the displayed service details page, click **Delete** in the upper right corner to delete the service.
- Batch services
  - In the batch service list, click **Delete** in the **Operation** column of the target service to delete it.

- Select services in the batch service list and click **Delete** above the list to delete services in batches.
- Click the name of the target service. On the displayed service details page, click **Delete** in the upper right corner to delete the service.

#### 

- A deleted service cannot be recovered.
- A service cannot be deleted without agency authorization.

### **Restarting a Service**

You can restart a real-time service only when the service is in the **Running** or **Alarm** state. Batch services and edge services cannot be restarted. You can restart a real-time service in either of the following ways:

- Log in to the ModelArts management console and choose Service
   Deployment from the navigation pane. Go to the real-time service list page.
   Click More > Restart in the Operation column to restart the target service.
- Log in to the ModelArts management console and choose **Service Deployment** from the navigation pane. Go to the real-time service list page. Click the name of the target service. The service details page is displayed. Click **Restart** in the upper right corner of the page to restart the service.

# 8.3.6 Viewing Service Events

During the whole lifecycle of a service, every key event is automatically recorded. You can view the events on the details page of the service at any time.

This helps you better understand the process of deploying a service and locate faults more accurately when a task exception occurs. The following table lists the available events.

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Normal	The service starts to deploy. Start to deploy service.	-
Abnormal	Insufficient resources. Wait until idle resources are sufficient. Lack of resources, transform state to waiting.	Wait until the resources are released and try again.

Table 8-27 Events

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Abnormal	Insufficient <i>xxx</i> . The scheduling failed. Supplementary information: <i>xxx</i> %s %s Schedule failed due to insufficient resources. Retry later. %s nodes are available: %s Insufficient memory.	Learn about resource insufficiency details based on the supplementary information. For details, see FAQs.
Normal	The image starts to create. Start to build image.	-
Abnormal	Failed to create model image <b>xxx</b> . For details, see logs :  <i>nxxx</i> . Failed to build image for model (%s %s), docker build log:\n%s.	Locate and rectify the fault based on the build logs.
Abnormal	Failed to create the image. Failed to build image.	Contact technical support.
Normal	The image created. Image built successfully.	-
Abnormal	Service xxx failed. Error: xxx Failed to %s service, retry later. Error message: %s	Locate and rectify the fault based on the error information.
Abnormal	Failed to update the service. Perform a rollback. Failed to update service, rollback it.	Contact technical support.
Normal	The service is being updated. Updating service.	-
Normal	The service is being started. Starting service.	-
Normal	The service is being stopped. Stopping service.	-
Normal	The service has been stopped. Service stopped.	-
Normal	Auto stop has been disabled. Auto-stop switched off.	-

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Normal	Auto stop has been enabled. The service will stop after <i>x</i> s.	-
	Auto-stop switched on, service will be stopped in %d %s.	
Normal	The service stops when the auto stop time expires.	-
	Service stopped automatically because due time is reached.	
Abnormal	The service is stopped because the quota exceeds the upper limit.	Contact technical
	Service stopped automatically because over quota.	support.
Abnormal	Failed to automatically stop the service. Error: <i>xxx</i>	Locate and rectify the fault
	Failed to stop service automatically, error message: %s	based on the error information.
Normal	Service instances deleted from resource pool <i>xxx</i> .	-
	Model in node(%s) deleted.	
Normal	Service instances stopped in resource pool <i>xxx</i> .	-
	Model in node(%s) stopped.	
Abnormal	The batch service failed. Try again later. Error: <i>xxx</i> Failed to %s batch service, retry later. Error message: %s.	Locate and rectify the fault based on the error information
Normal	The service has been executed	
Normat	Service stopped automatically after running.	
Abnormal	Failed to stop the service. Error: xxx Failed to stopped service, error message: %s	Locate and rectify the fault based on the error information.
Normal	The subscription license <i>xxx</i> is to expire. Impending expiration notice: %s	-
Normal	Service xxx started. Service %s started successfully.	-

Туре	Event ( <i>xxx</i> should be replaced with the actual value.)	Solution
Abnormal	Failed to start service xxx. Service %s started failed.	For details about how to locate and rectify the fault, see FAQs.
Abnormal	Service deployment timed out. Error: <i>xxx</i> Deploying timeout, details: %s	Locate and rectify the fault based on the error information.
Normal	Failed to update the service. The update has been rolled back. Failed to update service, rollback succeeded.	-
Abnormal	Failed to update the service. The rollback failed. Failed to update service, rollback failed.	Contact technical support.

During service deployment and running, key events can both be manually and automatically refreshed.

### **Viewing Events**

- In the left navigation pane of the ModelArts management console, choose Service Deployment > Real-Time Services or Batch Services or Edge Services. In the service list, click the name or ID of the target service to go to its details page.
- 2. View the events on the **Events** tab page.

# 8.4 Edge Resource Pool

# 8.4.1 Overview

An edge dedicated resource pool is a collection of tenant edge nodes for edge service deployment. Inference services run in the edge pool. After you create an asynchronous service or synchronous edge service, the edge service selects a proper node to process this asynchronous service or synchronous edge service in an asynchronous algorithm container.

Node

ModelArts edge nodes are devices provided by ModelArts for deploying edge services. Before creating an edge resource pool, you must create a ModelArts edge node and activate the node.

#### • Resource pool

An edge resource pool is dedicated for deploying edge services. When creating an edge resource pool, you can add ModelArts edge nodes or edge nodes managed by IEF.

#### Figure 8-33 Process of creating a ModelArts edge resource pool



Figure 8-34 Process of creating an IEF edge resource pool



# 8.4.2 Node

ModelArts edge nodes are devices provided by ModelArts for deploying edge services. Before creating an edge resource pool, you must create a ModelArts edge node. After a node is created, download the certificate and edge agent firmware, copy the firmware to the node, and run the registration command to register the device.

You can activate, modify, and delete the edge node, and view details about it. You can also use it to create edge services.

### **Specifications Requirements**

An edge node can be a physical machine or a virtual machine. Edge nodes must meet the specifications listed in the following table.

ltem	Specifications
OS	<ul> <li>x86_64 architecture</li> <li>Ubuntu LTS (Xenial Xerus), Ubuntu LTS (Bionic Beaver) , CentOS, EulerOS, and openEuler</li> </ul>
	<ul> <li>AArch64 (Arm64) architecture Ubuntu LTS (Bionic Beaver), CentOS, EulerOS, and openEuler</li> </ul>
Memory	The basic memory overhead of an edge node is about 64 MB, which varies among services. Set the memory to more than 256 MB.
CPU	The basic overhead is at least one core. For Docker+K3S/K8S scenarios, you need at least four cores.
Hard disk	≥ 512 MB

Item	Specifications
GPU (optional)	The GPU models of the same edge node or edge pool must be the same. When registering an edge node, you can choose whether to use GPUs. If an edge node needs to use GPUs, you must install the GPU driver before registering the edge node. <b>NOTE</b>
NPU	Ascend Al processors. When registering an edge node, you can
(optional)	choose whether to use NPU acceleration.
	• If an edge node needs to use NPUs, ensure that it has a driver installed. For details about the installation, contact the device vendor.
	<ul> <li>To use an NPU accelerator card for an edge resource pool (cluster), you must also install cluster-related plug-ins. For details about the installation, contact the device vendor.</li> </ul>
	NOTE Snt9, Snt9B, Snt3P, and Snt3 series NPU accelerator cards are supported.
Container engine	The Docker version must be 19.0.0 or later. Install it using these commands:
	curl -fsSL get.docker.com -o get-docker.sh sh get-docker.sh sudo systemctl daemon-reload
	sudo systemctl restart docker
K3s engine	v1.21.12+k3s1
Port	Edge nodes must use ports <b>5066</b> and <b>5067</b> . For edge pools (clusters), K8s and K3s ports must be reserved, including <b>2379</b> , <b>2380</b> , and <b>6443</b> .
	Port <b>5066</b> is used to provide HTTP and HTTPS REST services. <b>5067</b> functions as the Kubernetes client proxy of worker nodes.
Time synchroniz ation	The time of an edge node must be the same as the UTC time. Otherwise, the monitoring data of the edge node may be inaccurate. You can use a proper NTP server for time synchronization. You can also keep the connection with the cloud and synchronize the time with the cloud.

# Agency Authorization

When using edge nodes, you need to authorize users to use the log service and enable LTS by referring to **Enabling LTS**. To add authorization, perform the following steps:

- Step 1 Create the hilens\_admin\_trust agency.
  - 1. Log in to the ModelArts console. In the navigation pane, choose **Settings**.
  - 2. Click Add Authorization. Set Agency Name to hilens\_admin\_trust, set **Permissions** to **Custom**, and select **OBS Administrator** and **SWR** Administrator.

3. Click **Create**.

Step 2 Add the LTS Administrator permission for the hilens\_admin\_trust agency.

- 1. On the top menu bar of the console, click **System**.
- 2. Choose **Permissions** > **Agencies**. In the agency list, click **Modify** in the **Operation** column of **hilens\_admin\_trust**.
- 3. Switch to the **Agency Permissions** tab, click **Assign Permissions**, and select **Resource spaces**. In the search box in the upper right corner of the permission list, search for and select **LTS Administrator** and **AOM FullAccess**.
- 4. Click **OK**. In the confirm dialog box, click **OK**.

----End

# Creating an Edge Node

- 1. Log in to the ModelArts console and choose **Edge Resource Pool** from the left navigation pane.
- 2. On the **Nodes** page, click **Create**. You will see the **Create Edge Node** page.
- 3. Configure parameters by referring to the following table.

Parameter	Description
Name	Name of a node. The value contains 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-).
Description	Brief description of a node. The value contains 0 to 255 characters and cannot contain the following characters: #~^\$%&*<>[]\ / and ASCII characters (0-31).
AI accelerator card	You can choose whether to use an accelerator card for a node.
	• <b>Do not use</b> : No accelerator card is used.
	• <b>GPU</b> : A GPU accelerator card is used.
	<ul> <li>Ascend: An Ascend accelerator card is used. Ascend accelerator cards include snt3, snt3p, and snt9.</li> </ul>
Batch Registration	A certificate will be used for registration of multiple nodes. This function is disabled by default. After this function is enabled, you must set <b>Batch</b> <b>Registration Quantity</b> to the number of nodes you want to register.
Log Storage Duration	Period of time that the system keeps the log data before automatically deleting it. The unit is days and the value can range from 1 to 30.

Table 8-28 Parameters for creating an edge node

Parameter	Description	
System Log Settings	You can configure the following system log parameters.	
	• Log level: log level, which defaults to Debug. The options are Error, Warning, Info, and Debug.	
	• <b>Size Limit</b> : maximum size of a node's logs. The default value is 50 MB and it cannot be changed.	
	• Log Rotate Count: The system scrolls local logs based on the number of logs every <i>N</i> days and deletes the earliest log file. Each log file has a fixed size of 10 MB. The default value of this parameter is 5 days.	
	• <b>Uploading Logs</b> : This function is disabled by default. By default, system logs and application logs are stored locally. After you enable <b>Uploading Logs</b> and select a log level, logs of the corresponding level are uploaded to LTS.	
Application Log Settings	You can configure the following application log	
	• Log level: log level, which defaults to Debug. The options are Error, Warning, Info, and Debug.	
	• <b>Size Limit</b> : maximum size of a node's logs. The default value is 50 MB and it cannot be changed.	
	• Log Rotate Count: The system scrolls local logs based on the number of logs every <i>N</i> days and deletes the earliest log file. Each log file has a fixed size of 10 MB. The default value of this parameter is 5 days.	
	<ul> <li>Uploading Logs: This function is disabled by default. By default, system logs and application logs are stored locally. After you enable</li> <li>Uploading Logs and select a log level, logs of the corresponding level are uploaded to LTS.</li> </ul>	

- 4. After you confirm the configuration, click **OK** to create a node. You will see the **Node creation completed, download cert and agent** page.
- 5. Click **Download Now** next to the certificate name to download the certificate file. The certificate is valid within 24 hours after being downloaded. You must complete the registration within the validity period of the certificate. The certificate file can be downloaded only after the basic registration information is configured. After the page is closed, the certificate file cannot be downloaded again.
- 6. Download the edge Agent firmware. The following table lists the parameters.

Parameter	Description
CPU Architecture	Select a CPU architecture type. The options are <b>x86</b> , <b>ARM32</b> , and <b>ARM64</b> .
Operating system	Select an operating system. The options are <b>Linux</b> and <b>Windows</b> .
Agent Name	Select the name of the Agent you want to download.
Version	Select an Agent version. Currently, the available version is 2.0.26. <b>NOTE</b> After ModelArts is installed and deployed, you must upload the Agent firmware. After that, the firmware version is displayed on the console. For details about how to upload the firmware, see "Importing New Firmware (ModelArts Edge Agent) to ModelArts Edge Nodes" in <i>ModelArts 6.5.0.1 Maintenance Guide (for HCS Online</i> 24.3.0) 01 > ModelArts Configuration Guide 01.

 Table 8-29 Parameters for downloading an edge Agent firmware

#### **NOTE**

Download the firmware based on the node type, copy the firmware to the node, and run the registration command based on **Registering a Certificate**.

7. After the certificate and firmware are downloaded, select **I've finished downloading certificates and firmware** and click **OK**.

### **Registering a Certificate**

After an edge node is created, register the node certificate. The certificate registration method varies according to the operating system of the firmware.

Windows

Prerequisites: The PC must run Windows 10 or later.

- a. Decompress the firmware package downloaded in 6 and run the program.
- b. Copy the certificate downloaded in **5** to the decompressed file directory.
- c. Open the CMD command line program and switch to the decompressed file directory.
- d. Run the registration command. hdad.exe hdactl bind -p {*Certificate name*}
- Linux
  - a. Log in to the Linux PC and copy the firmware package downloaded in 6 to any directory on the PC.
  - Decompress the Agent firmware package and install the Agent. To do so, run this command: tar -xvf {*Firmware package name*}

- c. Install the Agent firmware. To do so, run this command: sh {*Running file*}
- d. Copy the certificate downloaded in **5** to the directory where the decompressed firmware package is stored.
- e. Run the registration command. hdactl bind -p {*Certificate name*}

### Activating an Edge Node

After a node is registered, you need to activate it. Nodes in the **UNCONNECTED** or **ACTIVATED** state cannot be activated.

1. On the **Edge Resource Pool** > **Nodes** page, click **Activate** in the **Operation** column of the target node or batch select nodes and click **Activate** above the list. Then, configure parameters by referring to the following table.

Parameter	Description
Name	Name of a node you want to activate.
Quantity	Number of nodes you want to activate. Only 50 nodes can be activated.
Effective Time	Time when the node activation takes effect. Only <b>Immediately</b> is supported.
Trial Duration	Trial duration of an edge node. <b>1 month</b> and <b>3 months</b> are supported.

**Table 8-30** Parameters for activating an edge node

2. After you confirm the configuration, click **Confirm** to activate the node.

### Modifying an Edge Node

On the **Edge Resource Pool** > **Nodes** page, click **Modify** in the **Operation** column of the node you want to modify. **Table 8-28** describes the parameters.

#### **NOTE**

- The IAM user, workspace, and batch registration parameters cannot be modified.
- The node name, log size limit, and log rotate count cannot be modified.

### Deleting an Edge Node

On the **Edge Resource Pool** > **Nodes** page, choose **More** > **Delete** in the **Operation** column of the node you want to remove. Before deleting an edge node, delete the resource pool bound to the node. Deleted edge nodes cannot be restored. You can also click **Force Delete** to delete a worker node.

#### D NOTE

- Master nodes of an edge resource pool cannot be deleted.
- You can use **Force Delete** to delete faulty worker nodes when the edge resource pool is in the **Running** state.
- You can use **Force Delete** to delete a node that has been used to create services or load balancers.

# Viewing Details About an Edge Node

On the **Edge Resource Pool** > **Nodes** page, click the name of the target node. On the node details page that appears, view the edge node details.

Table 8-31 Basic parameters

Parameter	Description
Name	Node name
ID	Node ID
Node Specifications	Node specifications
Description	Node description
Operating system	Node OS
Architecture	Node architecture

 Table 8-32
 Management information

Parameter	Description
Status	Node status. The possible values are UNCONNECTED, RUNNING, FAULTY, UPGRADING, and FREEZE.
Agent Version	Version of the firmware bound to the node.
Agent Name	Name of the firmware bound to the node.
Certificate/Profile	Certificate and configuration file of the node.
Inference Acceleration Card	Inference accelerator card mounted to the node.
IP	IP address of the node.
Activation Status	Activation status of the node, which can be <b>Unactivated</b> or <b>Activated</b> .
Activation Expiration Time	Expiration time of node activation.
Obtained At	Time when the node is created.

Parameter	Description
Updated	Last update time of the node.
IAM User	IAM user to which the node belongs.
Resource Pool	Resource pool to which the node belongs.

You can switch between tabs on the node details page to view more details.

Table	8-33	Node	details
iubic	0 33	1 VOUC	actures

Parameter	Description
Edge Services	Edge services deployed on an edge node. You can:
	• Obtain edge service information, such as the name, status, request mode, creation time, and description.
	Create edge services.
	Modify or delete edge services.
Log Setting	You can view the edge node's log settings, including log storage duration, system log settings, and application log settings. You can also edit the log settings.
Monitoring	You can view the edge node's monitoring information, such as CPU, memory, and disk. If the node uses NPUs or GPUs, select an NPU or GPU from the drop-down list box in the upper right corner of the NPU/GPU icon. The information collection period is 5 minutes.

# 8.4.3 Resource Pool

A ModelArts edge node must be activated or an IEF edge node must have been managed for creating a resource pool. After an edge resource pool is created, you can modify or delete it and view its details.

#### **Creating an Edge Resource Pool**

- 1. Log in to the ModelArts console and choose **Edge Resource Pool** from the navigation pane.
- 2. Click the **Resource Pools** tab. The resource pool list page is displayed.
- 3. Click **Create**. On the **Create Edge Resource Pool** page that is displayed, set parameters by referring to the following table.

Parameter	Description
Name	Name of an edge resource pool.
Description	Brief description of an edge resource pool.
Node Type	Edge node type, which can be ModelArts edge node or IEF edge node.
	• ModelArts edge node: edge node created in Creating an Edge Node.
	<ul> <li>Master Nodes: A master node manages and controls the entire resource pool. You can add no more than three master nodes.</li> </ul>
	<ul> <li>Max. Worker Nodes: The upper limit of worker nodes that can belong to a resource pool. The value ranges from 1 to 64.</li> </ul>
	<ul> <li>Worker Nodes: A worker node in a resource pool executes jobs assigned by master nodes.</li> </ul>
	NOTE A resource pool cannot have the same node as both a master node and a worker node.
	• IEF edge node: edge nodes managed in IEF.
	<ul> <li>Resource Instance: Select Platinum Service Instance.</li> </ul>
	<ul> <li>Edge Nodes: Select edge nodes to run edge applications, process your data, and collaborate with cloud applications securely and conveniently.</li> </ul>

4. Click **Create Now**.

# Modifying an Edge Resource Pool

1. On the **Edge Resource Pool** > **Resource Pools** page, click **Modify** in the **Operation** column of the resource pool you want to modify. The following lists the parameters.

Table 8-35	Parameters	for mo	difying	an edge	resource pool
------------	------------	--------	---------	---------	---------------

Parameter	Description
Name	Name of a resource pool, which cannot be changed.
Description	Brief description of a resource pool.

Parameter	Description
Node Type	The edge node type cannot be modified.
	If the edge node type is ModelArts edge node, the following parameters are displayed:
	• <b>Master Nodes</b> : A master node manages and controls the entire resource pool. which cannot be changed.
	• Max. Worker Nodes: The upper limit of worker nodes that can belong to an edge resource pool. The value ranges from 1 to 64. This parameter cannot be modified.
	<ul> <li>Worker Nodes: You can add or delete edge nodes.</li> </ul>
	NOTE A resource pool cannot have the same node as both a master node and a worker node.
	If the edge node type is IEF edge node, you can modify the following parameters:
	• <b>Resource Instance</b> : You can select a professional or platinum service instance.
	• Edge Nodes: Edge nodes are your edge computing devices used to run edge applications, process your data, and collaborate with cloud applications securely and conveniently.

2. Click **Submit** to finish the modification.

# Deleting an Edge Resource Pool

On the **Edge Resource Pool** > **Resource Pools** page, locate the target edge resource pool and click **Delete** in the **Operation** column. Before deleting an edge resource pool, you need to delete the associated edge service, load balancer, and access port. Deleted edge resource pools cannot be restored.

### **Obtaining Details About an Edge Resource Pool**

On the **Edge Resource Pool** > **Resource Pools** page, click the name of the target resource pool to go to the edge resource pool details page.

Parameter	Description
Name	Name of a resource pool.
Created	Time when a resource pool is created.
Updated	Time when a resource pool was last updated.

Table 8-36 Basic parameters

Parameter	Description
Description	Description of a resource pool.
Status	Status of a resource pool.
ID	ID of a resource pool. This parameter is displayed for edge resource pools created using ModelArts edge nodes.
Instance ID	ID of a resource instance. This parameter is displayed for edge resource pools created using IEF edge nodes.
Node Type	Edge node type of a resource pool. The value can be <b>ModelArts edge node</b> or <b>IEF edge node</b> .

Switch between tabs on the resource pool details page to view more details. Example:

Table 8-37 Resource pool details						
-						

Parameter	Description
Edge Services	Displays edge services associated with a resource pool. You can create, modify, and delete edge services.
Node	Displays information about nodes bound to a resource pool. You can add or delete nodes.

# 8.4.4 Enabling LTS

You can enable LTS for your edge nodes. This section describes how to enable LTS.

- Step 1 Create a node.
  - 1. In the navigation pane on the left, choose **Edge Resource Pool**. Click the **Nodes** tab.
  - 2. Click **Create** to create an edge node. Enter basic information. In the log settings area, enable uploading logs. Click **OK** to download the certificate and firmware.
  - 3. Click **OK**.
- **Step 2** Log in to the edge node and upload the device certificate and firmware.
  - 1. Use PuTTY to log in to the VM.

ssh <Username>@<VM IP address>

- Username: Enter the username for logging in to the server.
- *VM IP address*. On the cloud server console, select a cloud service from the cloud service list and copy its EIP.
- 2. Use a tool to upload the device certificate and firmware.

#### Figure 8-35 Uploading the certificate and firmware

- rw	1	opsadmin	admingroup	2811	Dec	10	18:22	edgeNode-lts.tar.gz
- rwx	1	opsadmin	admingroup	60092932	Dec	11	19:33	hdad
drwxr-x	5	opsadmin	admingroup	4096	Dec	11	19:40	hilens
- rw	1	opsadmin	admingroup	18061714	Dec	11	19:39	hilens-agent x86 64 2.0.26 20231211193033.tar.gz
- rwx	1	opsadmin	admingroup	34	Dec	11	19:33	install_manual.sh
- rw	1	opsadmin	admingroup	254	Dec	11	19:33	launch.sh
- rw	1	opsadmin	admingroup	105	Dec	11	19:33	readme.txt
[opsadmin@host-172-16-0-180~]\$								

#### **Step 3** Install the edge agent and register and bind the node.

1. Decompress the agent firmware package. tar -xvf *hilens-agent\_x86\_64\_2.0.26\_20231211193033.tar.gz* 

In the preceding command, *hilensagent\_x86\_64\_2.0.26\_20231211193033.tar.gz* indicates the firmware package. Replace it with your own firmware package.

- 2. Install the agent. sh install\_manual.sh
- 3. Modify the agent configuration. vi /etc/hilens/hda.conf

#### Add the following configurations:

hilens.lts.upload.url=https://*8.28.30.68*.8102 hilens.request.ctx.lts=v2 hilens.lts.url=https://lts.*ei-a3-1.externala3.com* 

- To obtain the IP address of hilens.lts.upload.url, log in to the LTS console, choose Host Management from the navigation pane, and click Install ICAgent in the upper right corner. In the Copy the ICAgent installation command area, obtain the IP address following https://, which is the IP address of hilens.lts.upload.url.
- To construct the value of hilens.lts.url, use the following format: https:// lts.{region}.{external\_global\_domain\_name}. You can obtain the value of external\_global\_domain\_name by searching for external\_global\_domain\_name on the LLD "Basic Parameters" sheet.
- 4. Restart the agent. systemctl restart hdad
- 5. Register and bind the edge node. hdactl bind -p edgeNode-lts.tar.gz
- **Step 4** Activate the edge node.
  - 1. Log in to the ModelArts console. In the navigation pane on the left, choose **Edge Resource Pool**.
  - 2. In the **Nodes** tab, locate the bound node and click **Activate** in the **Operation** column to activate the node.
- **Step 5** Log in to the LTS console to view the uploaded logs.

In the navigation pane on the left, choose **Log Management**. In **Log Groups**, select the log group with the node name and select the log stream corresponding to the node ID.

----End

# **8.5 Inference Specifications**

# 8.5.1 Model Package Specifications

# 8.5.1.1 Introduction to Model Package Specifications

When creating an AI application on the AI application management page, make sure that any meta model imported from OBS complies with certain specifications.

#### **NOTE**

- The model package specifications are used when you import one model. If you import multiple models, for example, there are multiple model files, use custom images.
- If you want to use an AI engine that is not supported by ModelArts, use a custom image.
- For details about how to create a custom image, see Custom Image Specifications for Creating AI Applications and Creating a Custom Image and Using It to Create an AI Application.
- For more examples of custom scripts, see Examples of Custom Scripts.

The model package must contain the **model** directory. The **model** directory stores the model file, model configuration file, and model inference code file.

- **Model files:** The requirements for model files vary according to the model package structure. For details, see **Model Package Example**.
- Model configuration file: The model configuration file must be available and its name is consistently to be config.json. There must be only one model configuration file. For details about how to edit a model configuration file, see Specifications for Editing a Model Configuration File.
- Model inference code file: It is mandatory. The file name is consistently to be customize\_service.py. There must be only one model inference code file. For details about how to edit model inference code, see Specifications for Writing Model Inference Code.
  - The .py file on which customize\_service.py depends can be directly stored in the model directory. Use a relative import mode to import the custom package.
  - The other files on which customize\_service.py depends can be stored in the model directory. You must use absolute paths to access these files.
     For more details, see Obtaining an Absolute Path.

ModelArts also provides custom script examples of common AI engines. For details, see **Examples of Custom Scripts**.

If you encounter any problem when importing a meta model, .

### Model Package Example

• Structure of the TensorFlow-based model package

When publishing the model, you only need to specify the **ocr** directory.

OBS bucket/directory name

| | — saved\_model.pb (Mandatory) Protocol buffer file, which contains the diagram description of the model
<ul> <li>variables Name of a fixed sub-directory, which contains the weight and deviation rate of the model. It is mandatory for the main file of the *.pb model.</li> <li>variables.index Mandatory</li> <li>variables.data-00000-of-00001 Mandatory</li> <li>config.json (Mandatory) Model configuration file. The file name is fixed to config.json.</li> <li>Only one model configuration file is supported.</li> <li>customize_service.py (Mandatory) Model inference code. The file name is fixed to customize_service.py (Mandatory) Model inference code file exists.</li> <li>The files on which customize service my depends can be directly stored in the model directory.</li> </ul>
Structure of the MindSpore-based model package
OBS bucket/directory name
resnet          resnet         model (Mandatory) Name of a fixed subdirectory, which is used to store model-related files
Structure of the PyTorch-based model package
When publishing the model, you only need to specify the <b>resnet</b> directory.
OBS bucket/directory name resnet model (Mandatory) Name of a fixed subdirectory, which is used to store model-related files Custom Python package>> (Optional) User's Python package, which can be directly referenced in model inference code The resnet50.pth (Mandatory) PyTorch model file, which contains variable and weight information and is saved as state_dict more fine ison (Mandatory) Model configuration file. The file name is fixed to configuration

Only one model configuration file is supported. Control one model configuration file is supported. Customize\_service.py (Mandatory) Model inference code. The file name is fixed to customize service.py. Only one model inference code file exists. The files on which

customize\_service.py depends can be directly stored in the model directory.

• Structure of a custom model package depends on the AI engine in your custom image. For example, if the AI engine in your custom image is TensorFlow, the model package uses the TensorFlow structure.

#### 8.5.1.2 Specifications for Editing a Model Configuration File

A model developer needs to edit a configuration file **config.json** when publishing a model. The model configuration file describes the model usage, computing framework, precision, inference code dependency package, and model API.

#### **Configuration File Format**

The configuration file is in JSON format. Table 8-38 describes the parameters.

Paramete r	Mand atory	Data Type	Description	
model_alg orithm	Yes	String	Model algorithm, which is set by the model developer to help model users understand the usage of the model. The value must start with a letter and contain no more than 36 characters. Chinese characters and special characters (&!'\"<>=) are not allowed. Common model algorithms include <b>image_classification</b> (image classification), <b>object_detection</b> (object detection), and <b>predict_analysis</b> (prediction analysis).	
model_typ e	Yes	String	<ul> <li>Model AI engine, which indicates the computing framework used by a model. Common AI engines and Image are supported.</li> <li>For details about supported AI engines, see Supported AI Engines for ModelArts Inference.</li> <li>If model_type is set to Image, the AI application is created using a custom image. In this case, parameter swr_location is mandatory. For details about specifications for custom images, see Custom Image</li> </ul>	
runtime	No	String	Model runtime environment. Python3.6 is used by default The value of <b>runtime</b> depends on the value of <b>model_type</b> . If <b>model_type</b> is set to <b>Image</b> , you do not need to set <b>runtime</b> . If <b>model_type</b> is set to another mainstream framework, select the engine and runtime environment.	
metrics	No	Objec t	Model precision information, including the average value, recall rate, precision, and accuracy For details about the <b>metrics</b> object structure, see <b>Table 8-39</b> . The result is displayed in the model precision area on the AI application details page.	

Table 8-38 Parameters

Paramete r	Mand atory	Data Type	Description	
apis	No	api array	<ul> <li>Format of the requests received and returned by a model. The value is structure data.</li> <li>It is the RESTful API array provided by a model.</li> <li>For details about the API data structure, see</li> <li>Table 8-40. For details about the code example, see Code Example of apis Parameters.</li> <li>If model_type is set to Image, the AI application is created using a custom image.</li> <li>When model_type is not Image, only one API whose request path is / can be declared in apis because the preconfigured AI engine exposes only one inference API whose request path is /.</li> </ul>	
dependen cies	No	depen dency array	Package on which the model inference code depends, which is structure data. Model developers need to provide the package name, installation mode, and version constraints. Only the pip installation mode is supported. <b>Table 8-43</b> describes the dependency array. If the model package does not contain the <b>customize_service.py</b> file, you do not need to set this parameter. Dependency packages cannot be installed for custom image models	
health	No	healt h data struct ure	Configuration of an image health interface. This parameter is mandatory only when <b>model_type</b> is set to <b>Image</b> . If services cannot be interrupted during a rolling upgrade, a health check API must be provided for ModelArts to call. For details about the health data structure, see <b>Table 8-45</b> .	

Table 8-39 metrics object description

Paramete r	Mand atory	Data Type	Description
f1	No	Numb er	F1 score. The value is rounded to 17 decimal places.
recall	No	Numb er	Recall rate. The value is rounded to 17 decimal places.
precision	No	Numb er	Precision. The value is rounded to 17 decimal places.

Paramete	Mand	Data	Description
r	atory	Type	
accuracy	No	Numb er	Accuracy. The value is rounded to 17 decimal places.

#### Table 8-40 api array

Paramet er	Manda tory	Data Type	Description
url	No	String	Request path. The default value is a slash (/). For a custom image model ( <b>model_type</b> is <b>Image</b> ), set this parameter to the actual request path exposed in the image. For a non-custom image model ( <b>model_type</b> is not <b>Image</b> ), the URL can only be /.
method	No	String	Request method. The default value is <b>POST</b> .
request	No	Object	Request body. For details, see Table 8-41.
response	No	Object	Response body. For details, see Table 8-42.

#### Table 8-41 request description

Paramet er	Mandat ory	Data Type	Description	
Content- type	No for real-time services Yes for batch services	String	<ul> <li>Data is sent in a specified content format. The default value is application/json.</li> <li>The options are as follows:</li> <li>application/json: JSON data is uploaded.</li> <li>multipart/form-data: A file is uploaded.</li> <li>NOTE For machine learning models, only application/json is supported.</li></ul>	
data	No for real-time services Yes for batch services	String	The request body is described in JSON schema. For details about the parameter description, see the <b>official guide</b> .	

Paramet er	Mandat ory	Data Type	Description
Content- type	No for real-time services Yes for batch services	String	Data is sent in a specified content format. The default value is <b>application/json</b> . <b>NOTE</b> For machine learning models, only <b>application/json</b> is supported.
data	No for real-time services Yes for batch services	String	The response body is described in JSON schema. For details about the parameter description, see the <b>official guide</b> .

#### Table 8-42 response description

#### Table 8-43 dependency array

Parameter	Mandatory	Data Type	Description
installer	Yes	String	Installation method. Only <b>pip</b> is supported.
packages	Yes	package array	Dependency package collection. For details about the package structure array, see <b>Table 8-44</b> .

#### Table 8-44 package array

Parameter	Mandatory	Туре	Description
package_na me	Yes	String	Dependency package name. Chinese characters and special characters (&!'''<>=) are not allowed.
package_ver sion	No	String	Dependency package version. If the dependency package does not rely on package versions, leave this field blank. Chinese characters and special characters (&!'''<>=) are not allowed.

Parameter	Mandatory	Туре	Description	
restraint	No	String	Version restriction. This parameter is mandatory only when <b>package_version</b> is configured. Possible values are <b>EXACT</b> , <b>ATLEAST</b> , and <b>ATMOST</b> .	
			• <b>EXACT</b> indicates that a specified version is installed.	
			• ATLEAST indicates that the version of the installation package is not earlier than the specified version.	
			• ATMOST indicates that the version of the installation package is not later than the specified version.	
			NOTE	
			<ul> <li>If there are specific requirements on the version, preferentially use EXACT. If EXACT conflicts with the system installation packages, you can select ATLEAST.</li> </ul>	
			<ul> <li>If there is no specific requirement on the version, retain only the package_name parameter and leave restraint and package_version blank.</li> </ul>	

Table	8-45	health	data	structure	description
-------	------	--------	------	-----------	-------------

Parameter	Mandatory	Туре	Description
check_meth od	Yes	String	Health check method. The value can be <b>HTTP</b> or <b>EXEC</b> .
			• <b>HTTP</b> : Use an HTTP request.
			• <b>EXEC</b> : Execute a command.
command	No	String	Health check command. This parameter is mandatory when <b>check_method</b> is set to <b>EXEC</b> .
url	No	String	Request URL of a health check API. This parameter is mandatory when <b>check_method</b> is set to <b>HTTP</b> .

Parameter	Mandatory	Туре	Description
protocol	No	String	Request protocol of a health check API. The default value is <b>http</b> . This parameter is mandatory when <b>check_method</b> is set to <b>HTTP</b> .
initial_delay _seconds	No	String	Delay for initializing the health check.
timeout_sec onds	No	String	Health check timeout.
period_seco nds	Yes	String	Health check period, in seconds. Enter an integer greater than 0 and no more than 2147483647.
failure_thres hold	Yes	String	Maximum number of health check failures. Enter an integer greater than 0 and no more than 2147483647.

#### **Code Example of apis Parameters**



#### Example of the Object Detection Model Configuration File

The following code uses the TensorFlow engine as an example. You can modify the **model\_type** parameter based on the actual engine type.

- Model input
   Key: images
   Value: image files
- Model output

```
{
  "detection_classes": [
      "face",
"arm"
  ],
   "detection_boxes": [
      [
         33.6,
         42.6,
         104.5,
         203.4
      ],
      [
         103.1,
         92.8,
         765.6,
         945.7
     ]
  ],
   "detection_scores": [0.99, 0.73]
}
```

#### • Configuration file

```
{
   "model_type": "TensorFlow",
   "model_algorithm": "object_detection",
   "metrics": {
      "f1": 0.345294,
      "accuracy": 0.462963,
"precision": 0.338977,
      "recall": 0.351852
   },
   "apis": [{
"url": "/",
      "method": "post",
      "request": {
          "Content-type": "multipart/form-data",
         "data": {
"type": "object",
             "properties": {
                "images": {
"type": "file"
                }
            }
         }
     },
"response": {
          "Content-type": "application/json",
         "data": {
             "type": "object",
             "properties": {
                "detection_classes": {
                   "type": "array",
                   "items": [{
"type": "string"
                   }]
                },
"detection_boxes": {
                   "type": "array",
                   "items": [{
"type": "array",
"minItems": 4,
                       "maxItems": 4,
                       "items": [{
```

```
"type": "number"
                    }]
                }]
             },
"detection_scores": {
                 "type": "array",
                 "items": [{
                    "type": "number"
                }]
             }
         }
      }
   }
}],
"dependencies": [{
    "installer": "pip",
   "packages": [{
          "restraint": "EXACT",
"package_version": "1.15.0",
"package_name": "numpy"
      },
       {
          "restraint": "EXACT",
          "package_version": "5.2.0",
          "package_name": "Pillow"
      }
   ]
}]
```

#### Example of the Image Classification Model Configuration File

The following code uses the TensorFlow engine as an example. You can modify the **model\_type** parameter based on the actual engine type.

Model input

}

Key: images

Value: image files

• Model output

```
{
   "predicted_label": "flower",
   "scores": [
     ["rose", 0.99],
     ["begonia", 0.01]
  ]
}
Configuration file
Ł
   "model_type": "TensorFlow",
"model_algorithm": "image_classification",
   "metrics": {
      "f1": 0.345294,
      "accuracy": 0.462963,
      "precision": 0.338977,
      "recall": 0.351852
   },
   "apis": [{
"url": "/",
"method": "post",
      "request": {
          "Content-type": "multipart/form-data",
         "data": {
            "type": "object",
            "properties": {
                "images": {
```

```
"type": "file"
              }
           }
        }
      },
      "response": {
         "Content-type": "application/json",
         "data": {
            "type": "object",
            "properties": {
               "predicted_label": {
"type": "string"
               },
               "scores": {
                  "type": "array",
"items": [{
                     "type": "array",
                     "minItems": 2,
                     "maxItems": 2,
                     "items": [
                        {
                            "type": "string"
                        },
                        {
                            "type": "number"
                        }
         }]
}
                    ]
        }
      }
   }],
   "dependencies": [{
      "installer": "pip",
"packages": [{
            "restraint": "ATLEAST",
            "package_version": "1.15.0",
            "package_name": "numpy"
         },
         {
            "restraint": "",
            "package_version": "",
            "package_name": "Pillow"
         }
     ]
  }]
}
```

The following code uses the MindSpore engine as an example. You can modify the **model\_type** parameter based on the type of the engine you use.

Model input

Key: images

Value: image files

- Model output
   "[[-2.404526 -3.0476532 -1.9888215 0.45013925 -1.7018927 0.40332815\n -7.1861157 11.290332 -1.5861531 5.7887416 ]]"
- Configuration file

```
{
    "model_algorithm": "image_classification",
    "model_type": "MindSpore",
    "metrics": {
        "f1": 0.124555,
        "recall": 0.171875,
        "precision": 0.0023493892851938493,
```

```
"accuracy": 0.00746268656716417
  },
"apis": [{
"url
        "url": "/",
        "method": "post",
        "request": {
            "Content-type": "multipart/form-data",
           "data": {
              "type": "object",
              "properties": {
                 "images": {
                    "type": "file"
                 }
              }
           }
        },
         "response": {
            "Content-type": "applicaton/json",
            "data": {
              "type": "object",
              "properties": {
                 "mnist_result": {
                    "type": "array",
                    "item": [{
                       "type": "string"
                    }]
                }
             }
           }
        }
     }
  1,
  "dependencies": []
}
```

#### **Example of the Predictive Analytics Model Configuration File**

The following code uses the TensorFlow engine as an example. You can modify the **model\_type** parameter based on the actual engine type.

• Model input



```
"resp_data": [
        {
            "predict_result": "unacc"
        },
        {
            "predict_result": "unacc"
        }
        ]
        }
}
```

• Configuration file

**NOTE** 

In the code, the **data** parameter in the request and response structures is described in JSON Schema. The content in **data** and **properties** corresponds to the model input and output.

```
{
   "model_type": "TensorFlow",
   "model_algorithm": "predict_analysis",
   "metrics": {
      "f1": 0.345294,
      "accuracy": 0.462963,
"precision": 0.338977,
      "recall": 0.351852
  },
   "apis": [
      {
        "url": "/",
        "method": "post",
         "request": {
            "Content-type": "application/json",
            "data": {
              "type": "object",
              "properties": {
                 "data": {
                     "type": "object",
                     "properties": {
                       "req_data": {
                           "items": [
                             {
                                "type": "object",
                                 "properties": {}
                             }
                           ],
"type": "array"
                       }
                    }
                 }
              }
           }
        },
"response": {
            "Content-type": "application/json",
            "data": {
              "type": "object",
               "properties": {
                 "data": {
                    "type": "object",
                    "properties": {
                       "resp_data": {
                          "type": "array",
                          "items": [
                             {
                                 "type": "object",
                                 "properties": {}
                             }
                          1
```



#### Example of the Custom Image Model Configuration File

The model input and output are similar to those in **Example of the Object Detection Model Configuration File**.

• If the input is an image, the request example is as follows.

In the example, a model prediction request containing the parameter **images** with the parameter type of **file** is received. For this example, the file upload button is displayed on the inference page, and the inference is performed in file format.

```
"Content-type": "multipart/form-data",
"data": {
    "type": "object",
    "properties": {
        "images": {
            "type": "file"
        }
    }
}
```

{

}

• If the input is JSON data, the request example is as follows.

In this example, the model prediction JSON request body is received. In the request, there is only one prediction request containing the parameter **input** with the parameter type of string. On the inference page, a text box is displayed for you to enter the prediction request.

```
{
    "Content-type": "application/json",
    "data": {
        "type": "object",
        "properties": {
            "input": {
               "type": "string"
            }
        }
}
```

```
A complete request example is as follows:
```

```
"model_algorithm": "image_classification",
   "model_type": "Image",
   "metrics": {
      "f1": 0.345294,
      "accuracy": 0.462963,
"precision": 0.338977,
      .
"recall": 0.351852
   },
   "apis": [{
"url": "/"
      "method": "post",
      "request": {
         "Content-type": "multipart/form-data",
         "data": {
            "type": "object",
             "properties": {
               "images": {
                  "type": "file"
               }
            }
         }
     },
"response": {
         "Content-type": "application/json",
         "data": {
            "type": "object",
            "required": [
               "predicted_label",
               "scores"
           ],
"properties": {
                "predicted_label": {
                  "type": "string"
               },
               "scores": {
"type": "array",
                  "items": [{
"type": "array",
                     "minItems": 2,
                      "maxItems": 2,
                      "items": [{
                            "type": "string"
                         },
                        {
                            "type": "number"
                        }
                     ]
       }
}
}
                 }]
     }
  }]
}
```

#### Example of the Machine Learning Model Configuration File

The following uses XGBoost as an example:

• Model input

{

```
"req_data": [
{
"sepal_length": 5,
"sepal_width": 3.3,
```

{

{

```
"petal_length": 1.4,
         "petal_width": 0.2
      },
{
         "sepal_length": 5,
         "sepal_width": 2,
         "petal_length": 3.5,
         "petal_width": 1
      },
{
         "sepal_length": 6,
         "sepal_width": 2.2,
         "petal_length": 5,
         "petal_width": 1.5
      }
  ]
}
      Model output
•
   "resp_data": [
      {
         "predict_result": "Iris-setosa"
      },
      {
         "predict_result": "Iris-versicolor"
      }
  ]
}
      Configuration file
•
   "model_type": "XGBoost",
   "model_algorithm": "xgboost_iris_test",
  "runtime": "python2.7",
"metrics": {
      "f1": 0.345294,
     "accuracy": 0.462963,
"precision": 0.338977,
"recall": 0.351852
  },
"apis": [
r
         "url": "/",
         "method": "post",
         "request": {
            "Content-type": "application/json",
            "data": {
               "type": "object",
               "properties": {
                  "req_data": {
                     "items": [
                        {
                           "type": "object",
                           "properties": {}
                        }
                     1,
                     "type": "array"
                  }
          }
        },
"response": {
            "Content-type": "applicaton/json",
            "data": {
               "type": "object",
               "properties": {
                  "resp_data": {
                     "type": "array",
                     "items": [
```



#### Example of a Model Configuration File Using a Custom Dependency Package

The following example defines the NumPy 1.16.4 dependency environment.

```
ł
  "model_algorithm": "image_classification",
  "model_type": "TensorFlow",
  "runtime": "python3.6",
  "apis": [
     {
        "url": "/",
        "method": "post",
"request": {
           "Content-type": "multipart/form-data",
           "data": {
              "type": "object",
              "properties": {
                 "images": {
                    "type": "file"
                }
             }
          }
        },
        "response": {
           "Content-type": "applicaton/json",
           "data": {
              "type": "object",
              "properties": {
                 "mnist_result": {
                    "type": "array",
"item": [
                      {
                          "type": "string"
                      }
                   ]
                }
         }
       }
     }
  ],
  "metrics": {
     "f1": 0.124555,
     "recall": 0.171875,
     "precision": 0.00234938928519385,
     "accuracy": 0.00746268656716417
  },
  "dependencies": [
     {
        "installer": "pip",
        "packages": [
           {
              "restraint": "EXACT",
```

}

```
"package_version": "1.16.4",
"package_name": "numpy"
}
]
}
]
```

#### 8.5.1.3 Specifications for Writing Model Inference Code

This section describes the general method of editing model inference code in ModelArts. This section also provides an inference code example for the TensorFlow engine and an example of customizing the inference logic in the inference script.

Due to the limitation of API Gateway, the duration of a single prediction in ModelArts cannot exceed 40s. The model inference code must be logically clear and concise for satisfactory inference performance.

#### **Specifications for Compiling Inference Code**

1. In the model inference code file **customize\_service.py**, add a child model class. This child model class inherits properties from its parent model class. For details about the import statements of different types of parent model classes, see **Table 8-46**.

Model Type	Parent Class	Import Statement
TensorFlow	TfServingBaseService	from model_service.tfserving_model_service import TfServingBaseService
PyTorch	PTServingBaseService	from model_service.pytorch_model_service import PTServingBaseService
MindSpore	SingleNodeService	from model_service.model_service import SingleNodeService

**Table 8-46** Import statements of different types of parent model classes

2. The following methods can be rewritten:

#### Table 8-47 Methods to be rewritten

Method	Description
init(self, model_name, model_path)	Initialization method, which is suitable for models created based on deep learning frameworks. Models and labels are loaded using this method. This method must be rewritten for models based on PyTorch and Caffe to implement the model loading logic.

Method	Description
init(self, model_path)	Initialization method, which is suitable for models created based on machine learning frameworks. The model path ( <b>self.model_path</b> ) is initialized using this method. In Spark_MLlib, this method also initializes SparkSession ( <b>self.spark</b> ).
_preprocess(self, data)	Preprocess method, which is called before an inference request and is used to convert the original request data of an API into the expected input data of a model
_inference (self, data)	Inference request method. You are advised not to rewrite the method because once the method is rewritten, the built-in inference process of ModelArts will be overwritten and the custom inference logic will run.
_postprocess(self, data)	Postprocess method, which is called after an inference request is complete and is used to convert the model output to the API output

#### D NOTE

- You can choose to rewrite the preprocess and postprocess methods to implement preprocessing of the API input and postprocessing of the inference output.
- Rewriting the init method of the parent model class may cause an AI application to run abnormally.
- 3. The attribute that can be used is the local path where the model resides. The attribute name is **self.model\_path**. In addition, PySpark-based models can use **self.spark** to obtain the SparkSession object in **customize\_service.py**.

#### **NOTE**

An absolute path is required for reading files in the inference code. You can obtain the local path of the model from the **self.model\_path** attribute.

- When TensorFlow, Caffe, or MXNet is used, self.model\_path indicates the path of the model file. See the following example:
   # Store the label.json file in the model directory. The following information is read: with open(os.path.join(self.model\_path, 'label.json')) as f: self.label = json.load(f)
- When PyTorch, Scikit\_Learn, or PySpark is used, self.model\_path indicates the path of the model file. See the following example:
   # Store the label.json file in the model directory. The following information is read: dir\_path = os.path.dirname(os.path.realpath(self.model\_path)) with open(os.path.join(dir\_path, 'label.json')) as f: self.label = json.load(f)
- 4. **data** imported through the API for pre-processing, actual inference request, and post-processing can be **multipart/form-data** or **application/json**.

#### - multipart/form-data request

curl -X POST \ <modelarts-inference-endpoint> \ -F image1=@cat.jpg \ -F images2=@horse.jpg

The corresponding input data is as follows:

```
Γ
 {
   "image1":{
     "cat.jpg":"<cat.jpg file io>"
   }
 },
 {
   "image2":{
     "horse.jpg":"<horse.jpg file io>"
   }
 }
1
application/json request
curl -X POST \
 <modelarts-inference-endpoint> \
 -d '{
  "images":"base64 encode image"
  }'
The corresponding input data is python dict.
```

# { "images":"base64 encode image"

#### **TensorFlow Inference Script Example**

}

The following is an example of TensorFlow MnistService.

```
Inference code
from PIL import Image
import numpy as np
from model_service.tfserving_model_service import TfServingBaseService
class MnistService(TfServingBaseService):
  def _preprocess(self, data):
     preprocessed_data = {}
     for k, v in data.items():
        for file_name, file_content in v.items():
           image1 = Image.open(file_content)
          image1 = np.array(image1, dtype=np.float32)
image1.resize((1, 784))
           preprocessed_data[k] = image1
     return preprocessed_data
  def _postprocess(self, data):
     infer_output = {}
     for output_name, result in data.items():
        infer_output["mnist_result"] = result[0].index(max(result[0]))
     return infer_output
Request
curl -X POST \ Real-time service address \ -F images=@test.jpg
```

Response
{"mnist\_result": 7}

The preceding code example resizes images imported to the user's form to adapt to the model input shape. The **32×32** image is read from the Pillow library and resized to **1×784** to match the model input. In subsequent processing, convert the model output into a list for the RESTful API to display.

#### Inference Script Example of the Custom Inference Logic

Customize a dependency package in the configuration file by referring to **Example** of a Model Configuration File Using a Custom Dependency Package. Then, use the following code example to load the model in **saved\_model** format for inference.

#### **NOTE**

The logging module of Python used by the base inference image uses the default log level Warning. Only warning logs can be queried by default. To query INFO logs, set the log level to INFO in the code.

```
# -*- coding: utf-8 -*-
import json
import os
import threading
import numpy as np
import tensorflow as tf
from PIL import Image
from model_service.tfserving_model_service import TfServingBaseService
import logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(name)s - %(levelname)s - %(message)s')
logger = logging.getLogger(__name__)
class MnistService(TfServingBaseService):
  def __init__(self, model_name, model_path):
     self.model name = model name
     self.model_path = model_path
     self.model_inputs = {}
     self.model_outputs = {}
    # The label file can be loaded here and used in the post-processing function.
     # Directories for storing the label.txt file on OBS and in the model package
     # with open(os.path.join(self.model_path, 'label.txt')) as f:
         self.label = json.load(f)
     #
     # Load the model in saved_model format in non-blocking mode to prevent blocking timeout.
     thread = threading.Thread(target=self.get_tf_sess)
     thread.start()
  def get_tf_sess(self):
     # Load the model in saved model format.
    # The session will be reused. Do not use the with statement.
     sess = tf.Session(graph=tf.Graph())
     meta_graph_def = tf.saved_model.loader.load(sess, [tf.saved_model.tag_constants.SERVING],
self.model_path)
     signature_defs = meta_graph_def.signature_def
     self.sess = sess
     signature = []
     # only one signature allowed
     for signature def in signature defs:
        signature.append(signature_def)
     if len(signature) == 1:
        model_signature = signature[0]
     else:
        logger.warning("signatures more than one, use serving_default signature")
        model_signature = tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY
     logger.info("model signature: %s", model_signature)
     for signature_name in meta_graph_def.signature_def[model_signature].inputs:
        tensorinfo = meta_graph_def.signature_def[model_signature].inputs[signature_name]
        name = tensorinfo.name
        op = self.sess.graph.get_tensor_by_name(name)
        self.model_inputs[signature_name] = op
```

```
logger.info("model inputs: %s", self.model_inputs)
    for signature_name in meta_graph_def.signature_def[model_signature].outputs:
       tensorinfo = meta_graph_def.signature_def[model_signature].outputs[signature_name]
       name = tensorinfo.name
       op = self.sess.graph.get_tensor_by_name(name)
       self.model_outputs[signature_name] = op
    logger.info("model outputs: %s", self.model_outputs)
  def _preprocess(self, data):
    # Two request modes using HTTPS
    # 1. The request in form-data file format is as follows: data = {"Request key value":{"File
name":<File io>}}
    # 2. Request in JSON format is as follows: data = json.loads("JSON body transferred by the API")
    preprocessed_data = {}
     for k, v in data.items():
       for file name, file content in v.items():
          image1 = Image.open(file_content)
          image1 = np.array(image1, dtype=np.float32)
          image1.resize((1, 28, 28))
          preprocessed_data[k] = image1
    return preprocessed_data
  def _inference(self, data):
     feed_dict = {}
     for k, v in data.items():
       if k not in self.model_inputs.keys():
          logger.error("input key %s is not in model inputs %s", k, list(self.model_inputs.keys()))
          raise Exception("input key %s is not in model inputs %s" % (k, list(self.model_inputs.keys())))
       feed_dict[self.model_inputs[k]] = v
     result = self.sess.run(self.model_outputs, feed_dict=feed_dict)
    logger.info('predict result : ' + str(result))
     return result
  def _postprocess(self, data):
    infer_output = {"mnist_result": []}
    for output_name, results in data.items():
       for result in results:
          infer_output["mnist_result"].append(np.argmax(result))
    return infer_output
  def __del__(self):
     self.sess.close()
To load models that are not supported by ModelArts or multiple models, specify the
    loading path using the __init__ method. Example code:
    # -*- coding: utf-8 -*-
    import os
    from model_service.tfserving_model_service import TfServingBaseService
    class MnistService(TfServingBaseService):
       def __init__(self, model_name, model_path):
         # Obtain the path to the model folder.
          root = os.path.dirname(os.path.abspath(__file__))
          # test.onnx is the name of the model file to be loaded and must be stored in the model folder.
          self.model_path = os.path.join(root, test.onnx)
          # Loading multiple models, for example, test2.onnx
```

```
# self.model_path2 = os.path.join(root, test2.onnx)
```

#### MindSpore Inference Script Example

```
The inference script is as follows:
import threading
import mindspore
import mindspore.nn as nn
import numpy as np
import logging
from mindspore import Tensor, context
from mindspore.common.initializer import Normal
from mindspore.train.serialization import load_checkpoint, load_param_into_net
from model_service.model_service import SingleNodeService
from PIL import Image
logger = logging.getLogger(__name__)
logger.setLevel(logging.INFO)
context.set_context(mode=context.GRAPH_MODE, device_target="Ascend")
class LeNet5(nn.Cell):
  """Lenet network structure."""
  # define the operator required
  def __init__(self, num_class=10, num_channel=1):
     super(LeNet5, self).__init__()
     self.conv1 = nn.Conv2d(num_channel, 6, 5, pad_mode='valid')
     self.conv2 = nn.Conv2d(6, 16, 5, pad_mode='valid')
     self.fc1 = nn.Dense(16 * 5 * 5, 120, weight_init=Normal(0.02))
     self.fc2 = nn.Dense(120, 84, weight_init=Normal(0.02))
     self.fc3 = nn.Dense(84, num_class, weight_init=Normal(0.02))
     self.relu = nn.ReLU()
     self.max_pool2d = nn.MaxPool2d(kernel_size=2, stride=2)
     self.flatten = nn.Flatten()
  # use the preceding operators to construct networks
  def construct(self, x):
     x = self.max_pool2d(self.relu(self.conv1(x)))
     x = self.max_pool2d(self.relu(self.conv2(x)))
     x = self.flatten(x)
     x = self.relu(self.fc1(x))
     x = self.relu(self.fc2(x))
     x = self.fc3(x)
     return x
class MnistService(SingleNodeService):
  def __init__(self, model_name, model_path):
     self.model_name = model_name
     self.model_path = model_path
     logger.info("self.model_name:%s self.model_path: %s", self.model_name,
             self.model_path)
     self.network = None
     # Load the model in non-blocking mode to prevent blocking timeout.
     thread = threading.Thread(target=self.load_model)
     thread.start()
  def load_model(self):
     logger.info("load network ... \n")
     self.network = LeNet5()
     ckpt_file = self.model_path + "/checkpoint_lenet_1-1_1875.ckpt"
     logger.info("ckpt_file: %s", ckpt_file)
     param_dict = load_checkpoint(ckpt_file)
     load_param_into_net(self.network, param_dict)
     # Inference warm-up. Otherwise, the initial inference will take a long time.
     self.network_warmup()
     logger.info("load network successfully ! \n")
```

```
def network_warmup(self):
  # Inference warm-up. Otherwise, the initial inference will take a long time.
  logger.info("warmup network ... \n")
  images = np.array(np.random.randn(1, 1, 32, 32), dtype=np.float32)
  inputs = Tensor(images, mindspore.float32)
  inference_result = self.network(inputs)
  logger.info("warmup network successfully ! \n")
def _preprocess(self, input_data):
  preprocessed_result = {}
  images = []
  for k, v in input data.items():
     for file_name, file_content in v.items():
        image1 = Image.open(file_content)
        image1 = image1.resize((1, 32 * 32))
        image1 = np.array(image1, dtype=np.float32)
        images.append(image1)
  images = np.array(images, dtype=np.float32)
  logger.info(images.shape)
  images.resize([len(input_data), 1, 32, 32])
  logger.info("images shape: %s", images.shape)
  inputs = Tensor(images, mindspore.float32)
  preprocessed_result['images'] = inputs
  return preprocessed_result
def _inference(self, preprocessed_result):
  inference_result = self.network(preprocessed_result['images'])
  return inference_result
def _postprocess(self, inference_result):
```

```
return str(inference_result)
```

# **8.5.2 Examples of Custom Scripts**

#### 8.5.2.1 TensorFlow

There are two types of TensorFlow APIs, Keras and tf. They use different code for training and saving models, but the same code for inference.

#### Training a Model (Keras API)

from keras.models import Sequential model = Sequential() from keras.layers import Dense import tensorflow as tf

# Import a training dataset. mnist = tf.keras.datasets.mnist (x\_train, y\_train),(x\_test, y\_test) = mnist.load\_data() x\_train, x\_test = x\_train / 255.0, x\_test / 255.0

print(x\_train.shape)

from keras.layers import Dense from keras.models import Sequential import keras from keras.layers import Dense, Activation, Flatten, Dropout

# Define a model network. model = Sequential() model.add(Flatten(input\_shape=(28,28))) model.add(Dense(units=5120,activation='relu')) model.add(Dropout(0.2))

# Train the model. model.fit(x\_train, y\_train, epochs=2) # Evaluate the model. model.evaluate(x\_test, y\_test)

#### Saving a Model (Keras API)

```
from keras import backend as K
# K.get_session().run(tf.global_variables_initializer())
# Define the inputs and outputs of the prediction API.
# The key values of the inputs and outputs dictionaries are used as the index keys for the input and output
tensors of the model.
# The input and output definitions of the model must match the custom inference script.
predict_signature = tf.saved_model.signature_def_utils.predict_signature_def(
  inputs={"images" : model.input},
  outputs={"scores" : model.output}
# Define a save path.
builder = tf.saved_model.builder.SavedModelBuilder('./mnist_keras/')
builder.add_meta_graph_and_variables(
  sess = K.get_session(),
  # The tf.saved_model.tag_constants.SERVING tag needs to be defined for inference and deployment.
  tags=[tf.saved_model.tag_constants.SERVING],
  signature_def_map: Only single items can exist, or the corresponding key needs to be defined as follows:
  tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY
  signature_def_map={
     tf.saved_model.signature_constants.DEFAULT_SERVING_SIGNATURE_DEF_KEY:
        predict_signature
  }
builder.save()
```

#### Training a Model (tf API)

from \_\_future\_\_ import print\_function

import gzip import os import urllib

import numpy import tensorflow as tf from six.moves import urllib

# Training data is obtained from the Yann LeCun official website http://yann.lecun.com/exdb/mnist/. SOURCE\_URL = 'http://yann.lecun.com/exdb/mnist/' TRAIN\_IMAGES = 'train-images-idx3-ubyte.gz' TRAIN\_LABELS = 'train-labels-idx1-ubyte.gz' TEST\_IMAGES = 't10k-images-idx3-ubyte.gz' TEST\_LABELS = 't10k-labels-idx1-ubyte.gz' VALIDATION\_SIZE = 5000

```
def maybe_download(filename, work_directory):
   """Download the data from Yann's website, unless it's already here."""
  if not os.path.exists(work_directory):
     os.mkdir(work_directory)
  filepath = os.path.join(work_directory, filename)
  if not os.path.exists(filepath):
     filepath, _ = urllib.request.urlretrieve(SOURCE_URL + filename, filepath)
     statinfo = os.stat(filepath)
     print('Successfully downloaded %s %d bytes.' % (filename, statinfo.st_size))
  return filepath
def _read32(bytestream):
  dt = numpy.dtype(numpy.uint32).newbyteorder('>')
  return numpy.frombuffer(bytestream.read(4), dtype=dt)[0]
def extract_images(filename):
   """Extract the images into a 4D uint8 numpy array [index, y, x, depth]."""
  print('Extracting %s' % filename)
  with gzip.open(filename) as bytestream:
     magic = _read32(bytestream)
     if magic != 2051:
       raise ValueError(
           'Invalid magic number %d in MNIST image file: %s' %
          (magic, filename))
     num_images = _read32(bytestream)
     rows = _read32(bytestream)
     cols = _read32(bytestream)
     buf = bytestream.read(rows * cols * num_images)
     data = numpy.frombuffer(buf, dtype=numpy.uint8)
     data = data.reshape(num_images, rows, cols, 1)
     return data
def dense to one hot(labels dense, num classes=10):
   """Convert class labels from scalars to one-hot vectors."""
  num_labels = labels_dense.shape[0]
  index_offset = numpy.arange(num_labels) * num_classes
  labels_one_hot = numpy.zeros((num_labels, num_classes))
  labels_one_hot.flat[index_offset + labels_dense.ravel()] = 1
  return labels_one_hot
def extract_labels(filename, one_hot=False):
  """Extract the labels into a 1D uint8 numpy array [index]."""
  print('Extracting %s' % filename)
  with gzip.open(filename) as bytestream:
     magic = _read32(bytestream)
     if magic != 2049:
        raise ValueError(
           'Invalid magic number %d in MNIST label file: %s' %
          (magic, filename))
     num_items = _read32(bytestream)
     buf = bytestream.read(num_items)
     labels = numpy.frombuffer(buf, dtype=numpy.uint8)
     if one hot:
        return dense_to_one_hot(labels)
     return labels
class DataSet(object):
  """Class encompassing test, validation and training MNIST data set."""
  def __init__(self, images, labels, fake_data=False, one_hot=False):
      ""Construct a DataSet. one_hot arg is used only if fake_data is true."""
     if fake_data:
```

```
self._num_examples = 10000
       self.one_hot = one_hot
     else:
       assert images.shape[0] == labels.shape[0], (
             'images.shape: %s labels.shape: %s' % (images.shape,
                                        labels.shape))
       self._num_examples = images.shape[0]
       # Convert shape from [num examples, rows, columns, depth]
        # to [num examples, rows*columns] (assuming depth == 1)
       assert images.shape[3] == 1
       images = images.reshape(images.shape[0],
                        images.shape[1] * images.shape[2])
       # Convert from [0, 255] -> [0.0, 1.0].
       images = images.astype(numpy.float32)
       images = numpy.multiply(images, 1.0 / 255.0)
     self._images = images
     self._labels = labels
     self._epochs_completed = 0
     self._index_in_epoch = 0
  @property
  def images(self):
     return self._images
  @property
  def labels(self):
     return self._labels
  @property
  def num_examples(self):
     return self._num_examples
  @property
  def epochs_completed(self):
     return self._epochs_completed
  def next_batch(self, batch_size, fake_data=False):
      """Return the next `batch_size` examples from this data set."""
     if fake_data:
        fake_image = [1] * 784
       if self.one_hot:
          fake_label = [1] + [0] * 9
       else:
          fake_label = 0
       return [fake_image for _ in range(batch_size)], [
          fake_label for _ in range(batch_size)
       ]
     start = self._index_in_epoch
     self._index_in_epoch += batch_size
     if self._index_in_epoch > self._num_examples:
        # Finished epoch
       self. epochs completed += 1
       # Shuffle the data
       perm = numpy.arange(self._num_examples)
       numpy.random.shuffle(perm)
       self._images = self._images[perm]
       self._labels = self._labels[perm]
        # Start next epoch
       start = 0
       self._index_in_epoch = batch_size
       assert batch_size <= self._num_examples
     end = self. index in epoch
     return self._images[start:end], self._labels[start:end]
def read_data_sets(train_dir, fake_data=False, one_hot=False):
```

```
"""Return training, validation and testing data sets."""
```

```
class DataSets(object):
     pass
  data_sets = DataSets()
  if fake_data:
     data_sets.train = DataSet([], [], fake_data=True, one_hot=one_hot)
     data_sets.validation = DataSet([], [], fake_data=True, one_hot=one_hot)
     data_sets.test = DataSet([], [], fake_data=True, one_hot=one_hot)
     return data sets
  local_file = maybe_download(TRAIN_IMAGES, train_dir)
  train images = extract images(local file)
  local_file = maybe_download(TRAIN_LABELS, train_dir)
  train_labels = extract_labels(local_file, one_hot=one_hot)
  local_file = maybe_download(TEST_IMAGES, train_dir)
  test_images = extract_images(local_file)
  local_file = maybe_download(TEST_LABELS, train_dir)
  test_labels = extract_labels(local_file, one_hot=one_hot)
  validation_images = train_images[:VALIDATION_SIZE]
  validation_labels = train_labels[:VALIDATION_SIZE]
  train_images = train_images[VALIDATION_SIZE:]
  train labels = train labels[VALIDATION SIZE:]
  data_sets.train = DataSet(train_images, train_labels)
  data_sets.validation = DataSet(validation_images, validation_labels)
  data_sets.test = DataSet(test_images, test_labels)
  return data_sets
training_iteration = 1000
modelarts_example_path = './modelarts-mnist-train-save-deploy-example'
export_path = modelarts_example_path + '/model/'
data_path = './'
print('Training model...')
mnist = read_data_sets(data_path, one_hot=True)
sess = tf.InteractiveSession()
serialized_tf_example = tf.placeholder(tf.string, name='tf_example')
feature_configs = {'x': tf.FixedLenFeature(shape=[784], dtype=tf.float32), }
tf example = tf.parse example(serialized tf example, feature configs)
x = tf.identity(tf_example['x'], name='x') # use tf.identity() to assign name
y_ = tf.placeholder('float', shape=[None, 10])
w = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
sess.run(tf.global_variables_initializer())
y = tf.nn.softmax(tf.matmul(x, w) + b, name='y')
cross_entropy = -tf.reduce_sum(y_ * tf.log(y))
train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)
values, indices = tf.nn.top_k(y, 10)
table = tf.contrib.lookup.index_to_string_table_from_tensor(
  tf.constant([str(i) for i in range(10)]))
prediction_classes = table.lookup(tf.to_int64(indices))
for _ in range(training_iteration):
  batch = mnist.train.next_batch(50)
  train_step.run(feed_dict={x: batch[0], y_: batch[1]})
correct_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(y_, 1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, 'float'))
print('training accuracy %g' % sess.run(
  accuracy, feed_dict={
     x: mnist.test.images,
     y_: mnist.test.labels
  }))
print('Done training!')
```

#### Saving a Model (tf API)

```
# Export the model.
# The model needs to be saved using the saved_model API.
print('Exporting trained model to', export_path)
builder = tf.saved_model.builder.SavedModelBuilder(export_path)
tensor_info_x = tf.saved_model.utils.build_tensor_info(x)
tensor_info_y = tf.saved_model.utils.build_tensor_info(y)
# Define the inputs and outputs of the prediction API.
# The key values of the inputs and outputs dictionaries are used as the index keys for the input and output
tensors of the model.
# The input and output definitions of the model must match the custom inference script.
prediction_signature = (
  tf.saved_model.signature_def_utils.build_signature_def(
     inputs={'images': tensor_info_x},
     outputs={'scores': tensor_info_y},
     method_name=tf.saved_model.signature_constants.PREDICT_METHOD_NAME))
legacy_init_op = tf.group(tf.tables_initializer(), name='legacy_init_op')
builder.add meta graph and variables(
  # Set tag to serve/tf.saved_model.tag_constants.SERVING.
  sess, [tf.saved_model.tag_constants.SERVING],
  signature_def_map={
     'predict_images':
       prediction_signature,
  legacy_init_op=legacy_init_op)
builder.save()
print('Done exporting!')
```

#### Inference Code (Keras and tf APIs)

In the model inference code file **customize\_service.py**, add a child model class which inherits properties from its parent model class. For details about the import statements of different types of parent model classes, see **Table 8-46**.

```
from PIL import Image
import numpy as np
from model_service.tfserving_model_service import TfServingBaseService
class MnistService(TfServingBaseService):
  # Match the model input with the user's HTTPS API input during preprocessing.
  # The model input corresponding to the preceding training part is {"images":<array>}.
  def _preprocess(self, data):
     preprocessed_data = {}
     images = []
     # Iterate the input data.
     for k, v in data.items():
        for file_name, file_content in v.items():
          image1 = Image.open(file_content)
          image1 = np.array(image1, dtype=np.float32)
          image1.resize((1,784))
          images.append(image1)
     # Return the numpy array.
     images = np.array(images,dtype=np.float32)
     # Perform batch processing on multiple input samples and ensure that the shape is the same as that
inputted during training.
     images.resize((len(data), 784))
     preprocessed_data['images'] = images
     return preprocessed_data
```

# Processing logic of the inference for invoking the parent class.

# The output corresponding to model saving in the preceding training part is {"scores":<array>}.
# Postprocess the HTTPS output.
def \_postprocess(self, data):
 infer\_output = {"mnist\_result": []}
 # Iterate the model output.
 for output\_name, results in data.items():
 for result in results:
 infer\_output["mnist\_result"].append(result.index(max(result)))
 return infer\_output

# 8.6 ModelArts Monitoring on Cloud Eye

### **8.6.1 ModelArts Metrics**

#### Description

The cloud service platform provides Cloud Eye to help you better understand the status of your ModelArts real-time services and models. You can use Cloud Eye to automatically monitor your ModelArts real-time services and model loads in real time and manage alarms and notifications so that you can obtain the performance metrics of ModelArts and models.

#### Namespace

SYS.ModelArts

#### **Monitoring Metrics**

Fable 8-4	<b>48</b> Mod	lelArts	metrics
-----------	---------------	---------	---------

Metric ID	Metric Name	Description	Value Range	Monitored Entity	Monitorin g Interval
cpu_usag e	CPU Usage	CPU usage of ModelArts Unit: %	≥ 0%	ModelArts model loads	1 minute
mem_usa ge	Memory Usage	Memory usage of ModelArts Unit: %	≥ 0%	ModelArts model loads	1 minute
gpu_util	GPU Usage	GPU usage of ModelArts Unit: %	≥ 0%	ModelArts model loads	1 minute
gpu_mem _usage	GPU Memory Usage	GPU memory usage of ModelArts Unit: %	≥ 0%	ModelArts model loads	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Entity	Monitorin g Interval
npu_util	NPU Usage	NPU usage of ModelArts Unit: %	≥ 0%	ModelArts model loads	1 minute
npu_mem _usage	NPU Memory Usage	NPU memory usage of ModelArts Unit: %	≥ 0%	ModelArts model loads	1 minute
successful ly_called_t imes	Number of Successfu I Calls	Times that ModelArts has been successfully called Unit: times/ minute	≥ counts/ minute	ModelArts models ModelArts real-time services	1 minute
failed_call ed_times	Number of Failed Calls	Times that ModelArts failed to be called Unit: times/ minute	≥ counts/ minute	ModelArts models ModelArts real-time services	1 minute
total_calle d_times	Total Calls	Times that ModelArts is called Unit: times/ minute	≥ counts/ minute	ModelArts model loads ModelArts real-time services	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Entity	Monitorin g Interval	
lf a measur measureme monitoring	If a measurement object has multiple measurement dimensions, all the measurement dimensions are mandatory when you use an API to query monitoring metrics.					
<ul> <li>The follo query a dim.0=se 3773b05</li> </ul>	<ul> <li>The following provides an example of using the multi-dimensional dim to query a single monitoring metric: dim.0=service_id,530cd6b0-86d7-4818-837f-935f6a27414d&amp;dim.1="model_id, 3773b058-5b4f-4366-9035-9bbd9964714a</li> </ul>					
<ul> <li>The follo query m "dimensi</li> </ul>	<ul> <li>The following provides an example of using the multi-dimensional <b>dim</b> to query monitoring metrics in batches: "dimensions": [</li> </ul>					
{	{					
"name":	"name": "service_id",					
"value":	"value": "530cd6b0-86d7-4818-837f-935f6a27414d"					
}						
{						
"name":	"name": "model_id",					
"value":	"value": "3773b058-5b4f-4366-9035-9bbd9964714a"					
}						
]						

#### Dimensions

#### Table 8-49 Dimension description

Кеу	Value
service_id	Real-time service ID
model_id	Model ID

# 8.6.2 Setting Alarm Rules

#### Scenario

Setting alarm rules allows you to customize the monitored objects and notification policies so that you can know the status of ModelArts real-time services and models in a timely manner.

An alarm rule includes the alarm rule name, monitored object, metric, threshold, monitoring interval, and whether to send a notification. This section describes how to set alarm rules for ModelArts services and models.

#### **NOTE**

Only real-time services in the **Running** status can be interconnected with CES.

#### Prerequisites

- A ModelArts real-time service has been created.
- ModelArts monitoring has been enabled on Cloud Eye. To do so, log in to the Cloud Eye console. On the Cloud Eye page, click **Custom Monitoring**. Then, enable ModelArts monitoring as prompted.

#### Procedure

Set an alarm rule in any of the following ways:

- Set an alarm rule for all ModelArts services.
- Set an alarm rule for a ModelArts service.
- Set an alarm rule for a model version.
- Set an alarm rule for a metric of a service or model version.

#### Method 1: Setting an Alarm Rule for All ModelArts Services

- 1. Log in to the management console.
- 2. On the Service List, click Cloud Eye under Management & Governance.
- 3. In the navigation pane on the left, choose **Alarm Management** > **Alarm Rules** and click **Create Alarm Rule**.
- 4. On the **Create Alarm Rule** page, set **Resource Type** to **ModelArts**, **Dimension** to **Service**, and **Method** to **Configure manually**, and set alarm policies. Then, confirm settings and click **Create**.

#### Method 2: Setting an Alarm Rule for a Single Service

- 1. Log in to the management console.
- 2. On the Service List, click Cloud Eye under Management & Governance.
- 3. In the left navigation pane, choose **Cloud Service Monitoring > ModelArts**.
- 4. Select a real-time service for which you want to create an alarm rule and click **Create Alarm Rule** in the **Operation** column.
- 5. On the **Create Alarm Rule** page, create an alarm rule for ModelArts real-time services and models as prompted.

#### Method 3: Setting an Alarm Rule for a Model Version

- 1. Log in to the management console.
- 2. On the Service List, click Cloud Eye under Management & Governance.
- 3. In the left navigation pane, choose **Cloud Service Monitoring > ModelArts**.
- 4. Click the down arrow next to the target real-time service name. Then, click **Create Alarm Rule** in the **Operation** column of the target version.
- 5. On the **Create Alarm Rule** page, create an alarm rule for model loads as prompted.

#### Method 4: Setting an Alarm Rule for a Metric of a Service or Model Version

- 1. Log in to the management console.
- 2. On the Service List, click Cloud Eye under Management & Governance.
- 3. In the left navigation pane, choose **Cloud Service Monitoring > ModelArts**.
- 4. Click the down arrow next to the target real-time service name. Then, click the target version and view alarm rule details.
- 5. On the alarm rule details page, click the plus sign (+) in the upper right corner of a metric and set an alarm rule for the metric.

# **8.6.3 Viewing Monitoring Metrics**

#### Scenario

Cloud Eye on the cloud service platform monitors the status of ModelArts realtime services and model loads. You can obtain the monitoring metrics of each ModelArts real-time service and model loads on the management console. Monitored data requires a period of time for transmission and display. The status of ModelArts displayed on the Cloud Eye console is usually the status obtained 5 to 10 minutes before. You can view the monitored data of a newly created realtime service 5 to 10 minutes later.

#### Prerequisites

- The ModelArts real-time service is running properly.
- Alarm rules have been configured on the Cloud Eye page. For details, see **Setting Alarm Rules**.
- The real-time service has been properly running for at least 10 minutes.
- The monitored data and graphics are available for a new real-time service after the service runs for at least 10 minutes.
- Cloud Eye does not display the metrics of a faulty or deleted real-time service. The monitoring metrics can be viewed after the real-time service starts or recovers.

Monitoring data is unavailable without alarm rules configured on Cloud Eye. For details, see **Setting Alarm Rules**.

#### Procedure

- 1. Log in to the management console.
- 2. In the Service List, click Cloud Eye under Management & Governance.
- 3. In the left navigation pane, choose **Cloud Service Monitoring > ModelArts**.
- 4. View monitoring graphs.
  - Viewing monitoring graphs of a real-time service: Click **View Metric** in the **Operation** column.
  - Viewing monitoring graphs of the model loads: Click v next to the target real-time service, and click View Metric in the Operation column of the target model.
- 5. In the monitoring area, you can select a duration to view the monitoring data.

You can view the monitoring data in the recent 1 hour, 3 hours, or 12 hours.

To view the monitoring curve of a longer time range, click  $\sum$  to enlarge the graph.

# **9** Resource Management

# 9.1 Resource Pool

#### **ModelArts Resource Pools**

When using ModelArts for AI development, you can use either of the following resource pools:

- **Dedicated resource pool**: It delivers more controllable resources and cannot be shared with other users. Create a dedicated resource pool and select it during AI development. The dedicated resource pool can be an elastic cluster or an elastic BMS.
  - Elastic cluster: It can be Standard or Lite.
    - In a Standard elastic cluster, exclusive computing resources are provided, with which you can deliver instances during job training, model deployment, and environment development on ModelArts.
    - A Lite elastic cluster provides hosted Kubernetes clusters with mainstream AI development plug-ins for Kubernetes resource users. You can operate the nodes and Kubernetes clusters in the resource pool with provided AI Native resources and tasks.
  - Elastic BMS: It provides different models of *x*PU BMSs. You can access an elastic BMS through an EIP and install GPU- and NPU-related drivers and software on a specified OS image. To meet the routine training requirements of algorithm engineers, SFS and OBS can be used to store and read data.
- **Public Resource Pool**: provides large-scale public computing clusters, which are allocated based on job parameter settings. Resources are isolated by job. You can use ModelArts public resource pools to deliver training jobs, deploy models, or run DevEnviron instances and will be billed on a pay-per-use basis.

#### **Differences Between Dedicated Resource Pools and Public Resource Pools**

• Dedicated resource pools provide dedicated computing clusters and network resources for users. The dedicated resource pools of different users are

physically isolated, while public resource pools are only logically isolated. Compared with public resource pools, dedicated resource pools feature better performance in isolation and security.

- When a dedicated resource pool is used for creating jobs and the resources are sufficient, the jobs will not be queued. When a public resource pool is used for creating jobs, there is a high probability that the jobs will be queued.
- A dedicated resource pool is accessible to your network. All running jobs in the pool can access storage and resources in your network. For example, if you select a dedicated resource pool with an accessible network when creating a training job, you can access SFS data after the training job is created.
- Dedicated resource pools allow you to customize the runtime environment of physical nodes, for example, you can upgrade GPU or Ascend drivers. This function is not supported by public resource pools.

# 9.2 Elastic Cluster

# 9.2.1 Comprehensive Upgrades to ModelArts Resource Pool Management Functions

ModelArts dedicated resource pools have been upgraded. In the new system, there are only unified ModelArts dedicated resource pools, which are no longer classified as the pools dedicated for development/training and the pools dedicated for service deployment. The new-version dedicated resource pools support flexible configuration of job types, and allow you to manage networks and interconnect VPCs with the networks.

The new dedicated resource pool management page provides more comprehensive functions and displays more information about the resource pools. More details about how to use and manage dedicated resource pools are provided in subsequent sections of this document. If you are new to ModelArts dedicated resource pools, try new-version dedicated resource pools. If you have used ModelArts dedicated resource pools, the old-version pools will be smoothly switched to new-version pools.

Read the following contents to learn about new-version dedicated resource pools.

#### Features of New-Version Dedicated Resource Pools

The new-version dedicated resource pool management is a comprehensive technology and product improvement. The main improvements are as follows:

- Single dedicated resource pool type for diverse purposes: Dedicated resource pools are no longer classified into those for development/training and those for service deployment. You can run both training and inference workloads in a dedicated resource pool. You can also set the job types supported by a dedicated resource pool based on your needs.
- Dedicated resource pool network connection: You can create and manage dedicated resource pool networks on the ModelArts management console. If you need to access resources in your VPC for jobs running in a dedicated resource pool, interconnect the VPC with the dedicated resource pool network.
- **More cluster details**: The new-version dedicated resource pool details page provides more cluster details, such as jobs, nodes, and resource monitoring, helping you learn about the cluster status and better plan and use resources.
- Cluster GPU/NPU driver management: On the new-version dedicated resource pool details page, you can select an accelerator card driver and perform change upon submission or smooth upgrade of the driver based on service requirements.
- **Fine-grained resource allocation (coming soon)**: You can divide your dedicated resource pool into multiple small pools and assign different quotas and permissions to each small pool for flexible and refined resource allocation and management.

More features will be provided in later versions for a better user experience.

# Can I Continue to Use the Existing Dedicated Resource Pools After the Upgrade Takes Effect?

If you have created dedicated resource pools, you can still access the old-version dedicated resource pool (elastic cluster) management page on the ModelArts management console and use the created resource pools, but you cannot create dedicated resource pool on that page. ModelArts allows you to migrate existing dedicated resource pools to the new management page. You will be contacted to complete the migration and this does not require you to perform any operations. In addition, the migration does not affect the workloads running in the dedicated resource pools. Pay attention to the easy-to-use new management functions of dedicated resource pools. There is no change in creating training jobs or inference services.

### **Differences Between New and Old Dedicated Resource Pools**

- In the old version, the dedicated resource pools dedicated for development/ training are separated from those dedicated for service deployment. In addition, the pools of the two types offer different functions and their user experience varies. In the new version, the dedicated resource pools of the two types are unified. You only need to configure one or multiple job types. Then, the dedicated resource pool automatically supports the configured job type.
- New dedicated resource pools inherit all functions of the old ones and have greatly improved user experience in key functions such as purchasing and resizing a resource pool. Use new dedicated resource pools for smooth, transparent experience.
- Additionally, the new dedicated resource pools offer enhanced functions, for example, allowing you to upgrade GPU or Ascend drivers, view details about job queuing, and use one network for multiple pools. More new functions of the new dedicated resource pools are coming soon.

# How Can I Get Help or Provide Feedback if I Encounter Problems During Use?

Similar to other ModelArts functions, you can report problems or obtain help in the sidebar of the console. In addition, you are advised to read the subsequent sections of this document to further understand how to use ModelArts dedicated resource pools.

# **Instructions of Dedicated Resource Pools**

- If you use dedicated resource pools for the first time, get started by reading **Resource Pool**.
- Create a dedicated resource pool by referring to Creating a Resource Pool.
- View the details about a created dedicated resource pool by referring to Viewing Details About a Resource Pool.
- If the specifications of a dedicated resource pool do not meet your service requirements, adjust the specifications by referring to **Resizing a Resource Pool**.
- Set or change job types supported by a dedicated resource pool by referring to **Changing Job Types Supported by a Resource Pool**.
- Upgrade the GPU/Ascend driver of your dedicated resource pools by referring to Upgrading a Resource Pool Driver.
- If a dedicated resource pool is no longer needed, delete it by referring to **Deleting a Resource Pool**.
- If any exception occurs when you use a dedicated resource pool, handle the exception by referring to **Abnormal Status of a Dedicated Resource Pool**.
- Manage dedicated resource pool networks or interconnect VPCs with the networks by referring to ModelArts Network.

# 9.2.2 Creating a Resource Pool

This section describes how to create a dedicated resource pool.

### Procedure

1. Log in to the ModelArts console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

### **NOTE**

For new users, only new-version elastic clusters are available on the ModelArts console. For users who have used old-version dedicated resource pools, they can access both old-version and new-version elastic clusters.

2. On the **Resource Pools** tab, click **Create** and configure parameters.

Table 9-1 Dedicated resource	e pool parameters
------------------------------	-------------------

Para met er	Sub- Para met er	Description
Na me	N/A	Name of a dedicated resource pool. Only lowercase letters, digits, and hyphens (-) are allowed. The value must start with a lowercase letter and cannot end with a hyphen (-).
Desc ripti on	N/A	Brief description of a dedicated resource pool.

Para met er	Sub- Para met er	Description		
Billi ng Mod e	N/A	You can select <b>Pay-per-use</b> .		
Reso urce Pool Type	N/A	You can select <b>Physical</b> or <b>Logical</b> . If there is no logical specification, <b>Logical</b> is not displayed.		
Job Type	N/A	Select job types supported by the resource pool based on service requirements.		
		<ul> <li>Physical: DevEnviron, Training Job, and Inference Service are supported.</li> </ul>		
		• Logical: Only Training Job is supported.		
Net wor k	N/A	Network in which the target service instance is deployed. The instance can exchange data with other cloud service resources in the same network. The network needs to be set only for physical resource pools.		
		Select a network from the drop-down list box. If no network is available, click <b>Create</b> on the right to create a network. For details about how to create a network, see <b>Creating a</b> <b>Network</b> .		
Spec ifica tion Man age men t	Spec ificat ions	Select required specifications. Due to system loss, the actual available resources are less than those specified in the specifications. After a dedicated resource pool is created, you can view the actual available resources on the <b>Nodes</b> tab page of the dedicated resource pool details page.		
	AZ	You can select <b>Automatically allocated</b> or <b>Specifies AZ</b> . An AZ is a physical region where resources use independent power supplies and networks. AZs are physically isolated but interconnected over an intranet.		
		• Automatically allocated: AZs are automatically allocated.		
		• <b>Specifies AZ</b> : Specify AZs for resource pool nodes. To ensure system disaster recovery, deploy all nodes in the same AZ. You can set the number of nodes in an AZ.		
	Nod es	Select the number of nodes in a dedicated resource pool. More nodes mean higher computing performance.		
		If <b>AZ</b> is set to <b>Specifies AZ</b> , you do not need to configure <b>Nodes</b> .		
		NOTE It is a good practice to create no more than 30 nodes at a time. Otherwise, the creation may fail due to traffic limiting.		

Para met er	Sub- Para met er	Description
	Adva nced Conf igura tion	This allows you to set the container engine space. You must enter an integer for the container engine space. It cannot be less than 50 GB, which is the default and minimum value. The maximum value depends on the specifications. To see the valid values, check the console prompt. Customizing the container engine space does not increase costs.
Cust om Driv er	N/A	This parameter is available only when a GPU or Ascend flavor is selected. Enable this function and select a driver.
GPU Driv er	N/A	This parameter is available only when custom driver is enabled. Select a GPU accelerator driver. <b>NOTE</b> You should choose NVIDIA driver 535.129.03 or later for hnt8 series specifications.
Adv ance d Opti ons	N/A	Select <b>Configure Now</b> to set the tag information, CIDR block, and controller node distribution.
CID R bloc k	N/A	<ul> <li>You can select Default or Custom.</li> <li>Default: The system randomly allocates an available CIDR block to you, which cannot be modified after the resource pool is created. For commercial use, customize your CIDR block.</li> <li>Custom: You need to customize K8S container and K8S service CIDR blocks.</li> <li>K8S Container Network: used by the container in a cluster, which determines how many containers there can be in a cluster. The value cannot be changed after the resource pool is created.</li> <li>K8S Service Network: used when the containers in the same cluster access each other, which determines how many Services there can be. The value cannot be changed after the resource pool is created.</li> </ul>
Clus ter Spec ifica tion s	N/A	<b>Cluster Scale</b> : maximum number of nodes that can be managed by the cluster. After the creation, the cluster can be scaled out but cannot be scaled in. You can select <b>Default</b> or <b>Custom</b> .

Para met er	Sub- Para met er	Description
Mas ter Distr ibuti on	N/A	Distribution locations of controller nodes. You can select <b>Random</b> or <b>Custom</b> .
		• <b>Random</b> : Use the AZs randomly allocated by the system.
		• <b>Custom</b> : Select AZs for controller nodes.
		Distribute controller nodes in different AZs for disaster recovery.

- 3. Click **Next** and confirm the settings. Then, click **Submit** to create the dedicated resource pool.
  - After a resource pool is created, its status changes to **Running**. Only when the number of available nodes is greater than 0, tasks can be delivered to this resource pool.
  - Hover the cursor over Creating to view the details about the creation process. Click View Details. The operation record page is displayed.
  - You can view the task records of the resource pool by clicking **Records** in the upper left corner of the resource pool list.

# FAQs

### Q: Why cannot I use all the CPU resources on a node in a resource pool?

Resource pool nodes have systems and plug-ins installed on them. These take up some CPU resources. For example, if a node has 8 vCPUs, but some of them are used by system components, the available resources will be fewer than 8 vCPUs.

You can check the available CPU resources by clicking the **Nodes** tab on the resource pool details page, before you start a task.

# 9.2.3 Viewing Details About a Resource Pool

### **Resource Pool Details Page**

- Log in to the ModelArts console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- Click V next to the resource pool type or status in the table header. In the top right corner of the list, select **Name** or **Resource ID** to filter resource pools. To obtain the resource ID, go to the **Billing Center** > **Orders** > **My Orders** page and click **Details** in the **Operation** column of the target order.
- In the resource pool list, click a resource pool to go to its details page and view its information.
  - If there are multiple resource pools, click 
     in the top left corner of the details page of one resource pool to switch between resource pools. Click
     More in the top right corner to perform operations such as resize or
     delete the resource pool. The available operations vary depending on the
     resource pool.

- In the **Network** area of **Basic Information**, you can click the number of resource pools associated to view associated resource pools.
- In the extended information area, you can view the monitoring information, jobs, nodes, specifications, and events. For details, see the following section.

### Viewing Jobs in a Resource Pool

On the resource pool details page, click **Jobs**. You can view all jobs running in the resource pool. If a job is queuing, you can view its queuing position.

#### **NOTE**

Only training jobs can be viewed.

### **Viewing Resource Pool Events**

On the resource pool details page, click **Events**. You can view all events of the resource pool. The cause of an event is **PoolStatusChange** or **PoolResourcesStatusChange**.

In the event list, click  $\overline{V}$  on the right of **Event Type** to filter events.

- When a resource pool starts to be created or becomes abnormal, the resource pool status changes and the change will be recorded as an event.
- When the number of nodes that are available or abnormal or in the process of being created or deleted changes, the resource pool node status changes and the change will be recorded as an event.

Figure 9-1 Viewing Resource Pool Events

			Start Date - Eac Date	
EventType 🏆	Cause	Details	Occurred At @	
O Abromal	PeoDistus Change	Poel status changed, from Abnormal Is Error.	Nev 14, 2023 19:27:03 GMT+00:00	
O Almontal	PeoFlesourcesStatusChange	Pool resources status changed, available/admortmal/covering/billeting count from 15/0/0 to 0110/0, timestamp: 1000050073.	Nev 13, 2023 15:17:53 GMT+00:00	
O Abnormal	PeofileAcchange	Poet stables changed, their Running to Abnormal.	Nev 13, 2023 09:58:00 GMT+08:00	
Normal	ProPersonesStateChange	Post recourses status charged, available/automotive/contributiong count from DIII/10 to 100010, timestamp: 1080147122.	Oct 86, 2023 14/38/42 OMT+88/08	
Normal	Pooldstuschange	Posi status changed, how Creating to Running.	Oct 88, 2023 18:38:82 GMT+88:08	
<ul> <li>Normal</li> </ul>	PeoPersourcesStatusChange	Post recourses status changed, available/approximationating/detecting count from 0/0/0/3 to 0/0/118, Smeetamp: 10/07/20008.	Oct BR 2023 14:38:09 GMT+88:00	
Normal	PaolitatusChange	Start creating pool	Oct 80, 2023 14:38:26 GWT+82:08	

### **Viewing Resource Pool Nodes**

On the resource pool details page, click **Nodes**. You can view all nodes in the resource pool and the resource usage of each node.

Some resources are reserved for cluster components. Therefore, **CPUs (Available/Total)** does not indicate the number of physical resources on the node. It only displays the number of resources that can be used by services. CPU cores are metered in milicores, and 1000 milicores equal 1 physical core.

• Replacing a node

In the **Nodes** tab, locate the node to be replaced. In the **Operation** column, click **Replace**. No fee is charged for this operation.

Check the node replacement records on the **Records** page. **Running** indicates that the node is being replaced. After the replacement, you can check the new node in the node list.

The replacement can last no longer than 24 hours. If no suitable resource is found after the replacement times out, the status changes to **Failed**. Hover

over 🕐 to check the failure cause.

**NOTE** 

- The number of replacements per day cannot exceed 20% of the total nodes in the resource pool. The number of nodes to be replaced cannot exceed 5% of the total nodes in the resource pool.
- Ensure that there are idle node resources. Otherwise, the replacement may fail.
- If there are any nodes in the resetting status in the operation records, nodes in this resource pool cannot be replaced.
- Resetting a node

In the **Nodes** tab, locate the node you want to reset. Click **Reset** in the **Operation** column to reset a node. You can also select multiple nodes, and click **Reset** to reset multiple nodes.

Configure the parameters described in the table below.

Table 9-2 Parameters

Parameter	Description			
Operating system	Select an OS from the drop-down list box.			
Configurati on Mode	<ul> <li>Select a configuration mode for resetting the node.</li> <li>By node percentage: the maximum ratio of nodes that can be reset if there are multiple nodes in the reset task</li> <li>By node quantity: the maximum number of nodes that can be reset if there are multiple nodes in the reset task</li> </ul>			

Check the node reset records on the **Records** page. If the node is being reset, its status is **Resetting**. After the reset is complete, the node status changes to **Available**. Resetting a node will not be charged.

**NOTE** 

- Resetting a node will affect running services.
- Only nodes in the Available state can be reset.
- A single node can be in only one reset task at a time. Multiple reset tasks cannot be delivered to the same node at a time.
- If there are any nodes in the replacing status in the operation records, nodes in this resource pool cannot be reset.
- When the driver of a resource pool is being upgraded, nodes in this resource pool cannot be reset.
- For GPU and NPU specifications, after the node is reset, the driver of the node may be upgraded. Wait patiently.
- Deleting, unsubscribing from, or releasing a node

For a pay-per-use resource pool, click **Delete** in the **Operation** column.

To delete nodes in batches, select the check boxes next to the node names, and click **Delete**.

#### **NOTE**

- Before deleting, unsubscribing from, or releasing a node, ensure that there are no running jobs on this node. Otherwise, the jobs will be interrupted.
- Delete, unsubscribe from, or release abnormal nodes in a resource pool and add new ones for substitution.
- If there is only one node, it cannot be deleted, unsubscribed from, or released.

### **Viewing Resource Pool Specifications**

On the resource pool details page, click **Specifications**. You can view the specifications used by the resource pool and the number of each specification.

**Figure 9-2** View resource pool specifications (The container engine size is displayed as the default value if it is not set.)

Jobs Events Nodes	Specifications Monitoring Subpools					
						C
Specifications	Metaring ID	Container Engine Space Size	CPU Cores CPU Architecture	Memory	Al Accelerator	Quantity Disk Capacity

### Viewing Resource Pool Monitoring Information

On the resource pool details page, click **Monitoring**. The resource usage including used CPUs, memory usage, and available disk capacity of the resource pool is displayed. If AI accelerators are used in the resource pool, the GPU and NPU monitoring information is also displayed.

# 9.2.4 Resizing a Resource Pool

### Description

The demand for resources in a dedicated resource pool may change due to the changes of AI development services. In this case, you can resize your dedicated resource pool in ModelArts.

- You can add nodes for existing flavors in the resource pool.
- You can delete nodes for existing flavors in the resource pool.

#### **NOTE**

Before scaling in a resource pool, ensure that there are no services running in the pool. Alternatively, go to the resource pool details page, delete the nodes where no services are running to scale in the pool.

### Constraints

- Only dedicated resource pools in the **Running** status can be resized.
- When scaling in a dedicated resource pool, the number of flavors or nodes of a flavor cannot be decreased to 0.

### **Resizing a Dedicated Resource Pool**

You can resize a resource pool in any of the following ways:

- Adjusting the number of nodes of existing specifications
- Resizing the container engine space
- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.

**NOTE** 

A resource pool is suspended when it is migrated from the old version to the new version. You cannot adjust the capacity of such a resource pool or unsubscribe from it.

2. Add or delete nodes.

Click **Adjust Capacity** in the **Operation** column of the target resource pool. In the **Resource Configurations** area, set **AZ** to **Automatically allocated** or **Specifies AZ**. Click **Submit** and then **OK** to save the changes.

- If **AZ** is set to **Automatically allocated**, you can increase or decrease the number of nodes to scale out or in the resource pool. After the scaling, nodes are automatically allocated to AZs.
- If you select **Specifies AZ**, you can allocate nodes to different AZs.

**Figure 9-3** Resource configuration (The container engine size is displayed as the default value if it is not set.)

 Specifications
 Specifications

 AZ
 Automatically allocated

 Nodes in Total
 Metering ID

 Container Engine
 Target Nodes

 Space Size
 L

3. Resizing the container engine space

If you need larger container engine size, perform any of the following operations:

- For new resources, you can specify the container engine space when creating a resource pool. For details, see advanced configurations of Specification Management in Creating a Resource Pool.
- For existing resources, the container engine space can be modified.
  - Method 1: Click the target resource pool to view its details. Click the Specifications tab, locate the target specifications, and click Change the container engine space size in the Operation column.
  - Method 2: Locate the target resource pool and click Adjust Capacity in the Operation column.

### NOTICE

Resizing the container engine space is only applicable to new nodes. Furthermore, dockerBaseSize may vary across nodes of this flavor within the resource pool. Consequently, this can lead to discrepancies in the status of tasks distributed among different nodes.

# 9.2.5 Migrating the Workspace

# Context

The workspace of a dedicated resource pool is associated with an enterprise project, which involves bill collection. ModelArts provides workspaces to isolate resource operation permissions of different IAM users. Workspace migration includes resource pool migration and network migration. For details, see the following sections.

# Migrating the Workspace for a Resource Pool

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- 2. In the resource pool list, choose **More** > **Migrate Workspace** in the **Operation** column of the target resource pool.
- 3. In the **Migrate Dedicated Resource Pool** dialog box that appears, select the target workspace and click **OK**.

### Migrating the Workspace for a Network

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**. Then, click the **Network** tab.
- 2. In the network list, choose **More** > **Migrate Workspace** in the **Operation** column of the target network.
- 3. In the dialog box that appears, select the target workspace and click **OK**.

# 9.2.6 Changing Job Types Supported by a Resource Pool

### Description

ModelArts supports many types of jobs. Some of them can run in dedicated resource pools, including training jobs, inference services, and notebook development environments.

You can change job types supported by a dedicated resource pool. Available options for **Job Type** are **Training Job**, **Inference Service**, and **DevEnviron**.

Only selected types of jobs can be delivered to the corresponding dedicated resource pool.

### 

To support different job types, different operations are performed in the backend, such as installing plug-ins and setting the network environment. Some operations use resources in the resource pool. As a result, available resources for you decrease. Therefore, select only the job types you need to avoid resource waste.

# Constraints

The target dedicated resource pool must be running.

# Procedure

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- 2. In the **Operation** column of a resource pool, choose **More** > **Set Job Type**.
- 3. In the Set Job Type dialog box, select job types.
- 4. Click OK.

# 9.2.7 Upgrading a Resource Pool Driver

# Description

If GPUs or Ascend resources are used in a dedicated resource pool, you may need to customize GPU or Ascend drivers. ModelArts allows you to upgrade GPU or Ascend drivers of your dedicated resource pools.

There are two driver upgrade modes: secure upgrade and forcible upgrade.

### **NOTE**

- Secure upgrade: Running services are not affected. After the upgrade starts, the nodes are isolated (new jobs cannot be delivered). After the existing jobs on the nodes are complete, the upgrade is performed. The secure upgrade may take a long time because the jobs must be completed first.
- Forcible upgrade: The drivers are directly upgraded, regardless of whether there are running jobs.

### Constraints

- The target dedicated resource pool must be running, and the resource pool contains GPU or Ascend resources.
- For a logical resource pool, the driver can be upgraded only after node binding is enabled. To enable node binding, submit a service ticket to contact engineers.

### **Upgrading the Driver**

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- 2. In the **Operation** column of the target resource pool, choose **More** > **Upgrade Driver**.
- 3. In the **Upgrade Driver** dialog box, the driver type, number of nodes, current version, target version, and upgrade mode of the dedicated resource pool are displayed.
  - **Target Version**: Select a target driver version from the drop-down list.
  - Upgrade Mode: Select Secure upgrade or Forcible upgrade.
- 4. Click **OK** to start the driver upgrade.

# 9.2.8 Deleting a Resource Pool

If a dedicated resource pool is no longer needed for AI service development, you can delete the resource pool to release resources.

### **NOTE**

After a dedicated resource pool is deleted, the development environments, training jobs, and inference services that depend on the resource pool are unavailable. A dedicated resource pool cannot be restored after being deleted.

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- 2. Locate the row that contains the target resource pool, choose **More** > **Delete** in the **Operation** column.
- 3. In the **Delete Dedicated Resource Pool** dialog box, enter **DELETE** in the text box and click **OK**.

You can switch between tabs on the details page to view the training jobs and notebook instances created using the resource pool, and inference services deployed in the resource pool.

# 9.2.9 Abnormal Status of a Dedicated Resource Pool

### **Resource Quota Limit**

When you use a dedicated resource pool (for example, scaling resources, creating a VPC, creating a VPC and subnet, or interconnecting a VPC), if the system displays a message indicating that the resource quota is limited, .

### **Creation Failed/Change Failed**

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- 2. Click **Records** on the right of **Create**. On the **Records** dialog box, view failed task records.

Figure 9-4 Creating a resource pool failed

Records					>
You can view your order records (excluding logical	sub-pools) below. Each record can be re	ained for a maximum of 90 day	'S.		×
				Enter a name.	Q Q
Name/ID	Operation Status	Operation Status	Billing Mode	Obtained At	
~	Successful	Create	Pay-per-use	Mar 05, 2024 10:28:23 GMT+08:00	
~	Successful	Delete	Yearly/Monthly	Mar 05, 2024 10:20:42 GMT+08:00	
~	• Failed 🧿	Create	Pay-per-use	Mar 04, 2024 15:22:47 GMT+08:00	

3. Hover the cursor over 2, view the cause of task failures.

### **NOTE**

By default, failed task records are sorted by application time. A maximum of 500 failed task records can be displayed and retained for three days.

### Locating Faulty Node

ModelArts will add a taint on a detected K8S faulty node so that jobs will not be affected or scheduled to the tainted node. The following table lists the faults can

be detected. You can locate the fault by referring to the isolation code and detection method.

lsol atio n Cod e	Cate gory	Sub- Categ ory	Description	Detection Method
A05 0101	GPU	GPU memo ry	GPU ECC error exists.	<ul> <li>Run the nvidia-smi -a command and check whether Pending Page Blacklist is Yes or the value of multi-bit Register File is greater than 0. For Ampere GPUs, check whether the following content exists:</li> <li>Uncorrectable SRAM error</li> <li>Remapping Failure records</li> <li>Xid 95 events in dmsg (For details, see NVIDIA GPU Memory Error Management.)</li> <li>The Ampere architecture has the following levels of GPU memory errors:</li> <li>L1: These are single-bit ECC errors that can be corrected. They do not affect the running services. To check for these errors, run the nvidia-smi -a command and look for Volatile Correctable.</li> <li>L2: These are multi-bit ECC errors that cannot be corrected. They cause the running services to fail and require a process restart to recover. To check for these errors, run the nvidia-smi -a command and look for Volatile Uncorrectable.</li> <li>L3: These are unsuppressed errors and may affect other services. They require a card reset or a node reboot to clear. To check for these errors, look for the Xid events that contain the number 95. (The Remapped Pending records are only for reference. You need to reset the cards when the</li> </ul>
				<ul> <li>service is idle to trigger the remapping process.)</li> <li>L4: These are errors that require a card replacement. To check for these errors, look for the SRAM</li> </ul>

Isol atio n Cod e	Cate gory	Sub- Categ ory	Description	Detection Method
				<b>Uncorrectable</b> field that is greater than 4 or the <b>Remapped Failed</b> field that is not zero.
A05 0102	GPU	Other	The <b>nvidia-smi</b> output contains ERR.	Run <b>nvidia-smi -a</b> and check whether the output contains ERR. Normally, the hardware, such as the power supply or the fan, is faulty.
A05 0103	GPU	Other	The execution of <b>nvidia-smi</b> times out or does not exist.	Check that exit code of <b>nvidia-smi</b> is not <b>0</b> .
A05 0104	GPU	GPU Memo ry	ECC error occurred 64 times.	Run the <b>nvidia-smi -a</b> command, locate <b>Retired Pages</b> , and check whether the sum of <b>Single Bit</b> and <b>Double Bit</b> is greater than 64.
A05 0148	GPU	Other	An infoROM alarm occurs.	Run the <b>nvidia-smi</b> command and check whether the output contains the alarm "infoROM is corrupted".
A05 0109	GPU	Other	Other GPU errors	Check whether other GPU error exists. Normally, there is a faulty hardware. Contact the technical engineer.
A05 0147	IB	Link	The IB NIC is abnormal.	Run the <b>ibstat</b> command and check whether the NIC is not in active state.
A05 0121	NPU	Other	A driver exception is detected by NPU DCMI.	The NPU driver environment is abnormal.
A05 0122	NPU	Other	The NPU DCMI device is abnormal.	The NPU device is abnormal. The Ascend DCMI interface returns a major or urgent alarm.
A05 0123	NPU	Link	The NPU DCMI net is abnormal.	The NPU network connection is abnormal.
A05 0129	NPU	Other	Other NPU errors	Check whether other NPU error exists. You cannot rectify the fault. Contact the technical engineer.

Isol atio n Cod e	Cate gory	Sub- Categ ory	Description	Detection Method
A05 0149	NPU	Link	Check whether the network port of the hccn tool is intermittently disconnected.	The NPU network is unstable and intermittently disconnected. Run the hccn_tool-i \${device_id} -link_stat -g command and the network is disconnected more than five times within 24 hours.
A05 0951	NPU	GPU memo ry	The number of NPU ECCs reaches the maintenance threshold.	The NPU's HBM Double Bit Isolated Pages Count value is greater than or equal to 64.
A05 0146	Runti me	Other	The NTP is abnormal.	The ntpd or chronyd service is abnormal.
A05 0202	Runti me	Other	The node is not ready.	<ul> <li>The node is unavailable. The K8S node contains one of the following taints:</li> <li>node.kubernetes.io/unreachable</li> <li>node.kubernetes.io/not-ready</li> </ul>
A05 0203	Runti me	Discon nectio n	The number of normal AI cards does not match the actual capacity.	The GPU or NPU is disconnected.
A05 0206	Runti me	Other	The Kubelet hard disk is read-only.	The <b>/mnt/paas/kubernetes/kubelet</b> directory is read-only.
A05 0801	Node man age ment	Node O&M	Resource is reserved.	The node is marked as the standby node and contains a taint.
A05 0802	Node man age ment	Node O&M	An unknown error occurs.	The node is marked with an unknown taint.
A20 0001	Node man age ment	Driver upgra de	The GPU is being upgraded.	The GPU is being upgraded.

lsol atio n Cod e	Cate gory	Sub- Categ ory	Description	Detection Method
A20 0002	Node man age ment	Driver upgra de	The NPU is being upgraded.	The NPU is being upgraded.
A20 0008	Node man age ment	Node admiss ion	The admission is being examined.	The admission is being examined, including basic node configuration check and simple service verification.
A05 0933	Node man age ment	Fault tolera nce Failov er	The Failover service on the tainted node will be migrated.	The Failover service on the tainted node will be migrated.
A05 0931	Traini ng toolk it	Pre- check contai ner	A GPU error is detected in the pre-check container.	A GPU error is detected in the pre- check container.
A05 0932	Traini ng toolk it	Pre- check contai ner	An IB error is detected in the pre-check container.	An IB error is detected in the pre- check container.

# 9.2.10 ModelArts Network

# ModelArts Network and VPC

ModelArts networks are used for interconnecting nodes in a ModelArts resource pool. You can only configure the name and CIDR block for a network. To ensure that there is no IP address segment in the CIDR block overlapped with that of the VPC to be accessed, multiple CIDR blocks are available for you to select.

A VPC provides a logically isolated virtual network for your instances. You can configure and manage the network as required. VPC provides logically isolated, configurable, and manageable virtual networks for cloud servers, cloud containers, and cloud databases. It helps you improve cloud service security and simplify network deployment.

# Prerequisites

- A VPC is available.
- A subnet is available.

## **Creating a Network**

- 1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools > Elastic Cluster**.
- 2. Click **Network** and then **Create**.
- 3. In the **Create Network** dialog box, set parameters.
  - **Network Name**: customizable name
  - CIDR Block: You can select Preset or Custom.

#### 

- Each user can create a maximum of 15 networks.
- Ensure there is no IP address segment in the CIDR block overlaps that of the VPC to be accessed. The CIDR block cannot be changed after the network is created. Possible conflict CIDR blocks are as follows:
  - Your VPC CIDR block
  - Container CIDR block (consistently to be 172.16.0.0/16)
  - Service CIDR block (consistently to be 10.247.0.0/16)
- 4. Confirm the settings and click **OK**.

### (Optional) Interconnecting a VPC with a ModelArts Network

VPC interconnection allows you to use resources across VPCs, improving resource utilization.

1. On the **Network** page, click **Interconnect VPC** in the **Operation** column of the target network.

#### Figure 9-5 Interconnect VPC



2. In the displayed dialog box, click the button on the right of **Interconnect VPC**, and select an available VPC and subnet from the drop-down lists.

#### **NOTE**

The peer network to be interconnected cannot overlap with the current CIDR block.

- If no VPC is available, click **Create VPC** on the right to create a VPC.
- If no subnet is available, click Create Subnet on the right to create a subnet.
- Multiple subnets in a VPC can be interconnected. You can click + to add up to 10 subnets.

### Enabling a Dedicated Resource Pool to Access the Internet

To enable a dedicated resource pool to access the Internet, follow these steps:

Step 1 Interconnect a VPC. For details, see (Optional) Interconnecting a VPC with a ModelArts Network .

Step 2 For details about how to configure an SNAT server for a VPC, see section "Configuring an SNAT Server" in *Virtual Private Cloud (VPC) Usage Guide > User Guide*.

----End

### Deleting a Network

If a network is no longer needed for AI service development, you can delete the network.

- 1. Go to the **Network** tab page and click **Delete** in the **Operation** column of a network.
- 2. Confirm the information and click **OK**.

# 9.3 Elastic Server

# 9.3.1 Overview

Elastic Server provides you with dedicated physical servers on the cloud. You can log in to a BMS as user **root** and install and deploy third-party software, such as AI frameworks and applications. To create an elastic server, you only need to specify the flavor, image, required network configuration, and key pair.

Term	Description
BMS	A Bare Metal Server (BMS) features both the scalability of VMs and high performance of physical servers. It provides dedicated servers on the cloud, delivering the performance and security required by core databases, critical applications, high- performance computing (HPC), and big data.
Image	Elastic Server provides public images. The image version is EulerOS 2.8 and the driver version is C78.
Disk	Elastic Server provides local disks based on BMS. Local disks include NVMe SSDs, SATA disks, and others. They provide low latency, high throughput, and high cost-effectiveness product and are applicable to scenarios that have large volumes of data and require high storage I/O performance and real-time performance.
SSH key pair	You can log in to an elastic server only using an SSH key pair. Therefore, you do not need to worry about password interception, cracking, and leakage.
	As an alternative to the traditional username+password authentication method, key pairs allow you to remotely log in to Linux ECSs.

Table	9-4	Terms
-------	-----	-------

Term	Description
Network	<ul> <li>Virtual Private Cloud (VPC)         A VPC is a logically isolated, configurable, and manageable virtual network. It helps improve the security of cloud resources and simplifies network deployment. Within your own VPC, you can create security groups and VPNs, configure IP address ranges, specify bandwidth sizes, by customizing security groups, VPNs, IP address ranges, and bandwidth. This simplifies network management. You can also customize access rules to control BMS access within a security group and across different security groups to enhance BMS security.     </li> </ul>
	<ul> <li>RoCE network         Elastic Server supports RoCE networks, facilitating large-scale distributed computing. RoCE is a network protocol that leverages Remote Direct Memory Access (RDMA) capabilities, and is commonly used in distributed storage networks. By using related hardware and network technologies, the NICs of server 1 can directly read and write the memory of server 2, achieving high bandwidth, low latency, and low resource utilization.     </li> </ul>

# 9.3.2 Preparations

### Step 1: Enable the Feature

To use Elastic Server, contact your region owner to enable it.

### **Step 2: Enable Basic Permissions**

To enable basic permissions, log in to the management console as the administrator account and assign the basic permissions (such as ModelArts FullAccess, BMS FullAccess, ECS FullAccess, VPC FullAccess, and VPC Administrator) required by Elastic Server to IAM users.

- **Step 1** Log in to the IAM console.
- **Step 2** Click **User Groups** and then click **Create User Group**.
- Step 3 Enter the user group name and click OK.
- **Step 4** Click **Manage User** in the **Operation** column and add the users for which you want to assign permissions to the user group.
- **Step 5** Click the name of the user group to go to the group details page.
- **Step 6** On the **Permissions** tab page, click **Assign Permissions**.

#### Figure 9-6 Assign Permissions

Permissions	Users
Assign Permis	sions

**Step 7** Set **Scope** to **Region-specific projects** and select **All resources (including future projects in all regions)** from the drop-down list.

Figure 9-7 Scope

Sco	e	
_	0	Global service project Select the option to assign permissions for dobal services, such as OBS based on the dobal service project. Users who are granted these permissions do not need to switch regions when accessing these services.
ſ	۲	Region-specific projects Select this option to assign parmissions for project-level services, such as ECS, based on region-specific projects. Users who are granted these permissions can access these services only in the selected projects. To assign parmissions for all projects, select "All resources".
L		All resources (including there projects in all regions)

Step 8 Enter ModelArts FullAccess in the search box and select ModelArts FullAccess.

Figure 9-8 ModelArts FullAccess



Step 9 Use the same method to select BMS FullAccess, ECS FullAccess, VPC FullAccess, and VPC Administrator. (Server Administrator is the dependency of VPC Administrator and is automatically selected.)

Figure 9-9 Basic permissions

-eme	ssors							
	Selected permissions: 6, which include the dependency permissions of VPC Administrator. Click View Selected or expand the details area	to view the dependency permissions.						
	Verv Selected (5) Copy Permissions from Another Project			All policies/toles	All services	•	Enter a policy name, role name, or description	im Q
	Policy/Rale Name	Description	Type					
	V 🗹 Modelifus Fullicosos	All permissions of MedelAds service.	System-defined policy					
	V 💟 Bild Full-Costs	All permissions at ERRS service.	Bystem-defined policy					
	V 💟 BCS Fulkcoss	All permissions of ECS service.	Dystem-defined policy					
	V 💟 WC Falkcom	All permissions of VPC service.	System-defined policy					
	V 💟 VPC Administrator	VPC Administrator	System-defined role					
	V 💟 Server Administrator	Server Administrator	System-defined role					

Step 10 Click OK.

----End

### Step 3: Create a VPC

To create a VPC, you need to log in to the management console as the administrator account.

- **Step 1** Log in to the management console.
- **Step 2** In the service list on the left, choose **Networking** > **Virtual Private Cloud**.
- **Step 3** On the displayed page, click **Create VPC** in the upper right corner and click **Create Now**.

Basic Information	
Region	×
	Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.
Name	Vpc-
CIDR Block	192 · 168 · 0 · 0 / 16 •
	Recommended: 10.0.0/8-24 (Select) 172.16.0.0/12-24 (Select) 192.168.0.0/16-24 (Select)
	A The CIDR block 192.168.0.0/16 overlaps with a CIDR block of another VPC in the current region. If you intend to enable communication between VPCs or between a VPC and an on-premises data center, change the CIDR block. View VPC CIDR blocks in current region
Enterprise Project	Select    C Create Enterprise Project
Advanced Settings 🔻	Tag   Description
Default Subnet	
AZ	AZ1 • ⑦
Name	subnet
CIDR Block	192 • 168 • 0 • 0 / 24 • ⑦ Available IP Addresses: 251
	The CIDR block cannot be modified after the subnet has been created.
Associated Route Table	Default 💮
Advanced Settings 🔻	Gateway   DNS Server Address   Tag   Description

#### Figure 9-10 Create VPC

----End

### Step 4: Create a Key Pair

- **Step 1** Log in to the ModelArts management console.
- **Step 2** In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Server**.
- Step 3 Click Create.
- Step 4 Click Create Key Pair.
- **Step 5** On the displayed page, click **Create Key Pair** in the upper right corner and click **OK** to save the key pair to your local PC.

Figure 9-11 Creating a key pair

Key Pair 💿	Create Ke	ey Pair	Import Key P	air
Coulte fog Poir				×
Here: xopts				

----End

# 9.3.3 Getting Started

This section describes how to create an elastic server, log in to it through SSH, and release it.

# Prerequisites

You have enabled Elastic Server and created a VPC and a key pair. For details, see **Preparations**.

## Procedure

- **Step 1** Log in to the ModelArts management console.
- **Step 2** In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Server**.
- **Step 3** Click **Create**. On the **Create DevServer Instance** page that is displayed, set parameters.
  - **Resource Type**: Select **BMS**.
  - CPU Architecture: Select Arm.
  - **Network**: Select your VPC and subnet.
  - **Key Pair**: Select your key pair.
  - Retain default settings for other parameters.

### Figure 9-12 Parameters

* Name	devserver-dcd6					
* Billing Mode	Pay-per-use Ye	arly/Monthly				
* Resource Type	ECS BMS					
* CPU Architecture	x86 Arm					
* AZ	AZ1					
* Flavor	Flavor	GPU		vCPUs	Memory (GB)	Description
	physical.kat1.6xlarge			64	8*HUAWEI Asce	
* Image ⑦	ModelArts-Euler2.8_Aarch64	_D910_C78-202 ▼				
* Network ⑦	vpc-	•	Create VP	c		
	subnet	•	Create Sub	net		
RoCE Network ⑦						
* Security Group	default	•	C Create See	curity Group		
* Key Pair	KeyPair	•	Create Ke	y Pair		
Enterprise Project	default	•	C Create En	terprise Project		

- Step 4 Click Next.
- **Step 5** Contact the administrator to **configure the network** and obtain the EIP address and public port number.
- **Step 6** Start the SSH tool and set parameters (MobaXterm is used as an example).

Click **SSH** and type the EIP, username, and port.

ession settings										
										)
SSH Telnet Rsh	Xdmcp R	DP VNC	🌏 FTP	<pre> SFTP </pre>	کی Serial	9 File	> Shell	<b>R</b> rowser	📡 Mosh	🚏 Aws S3
Sasic SSH settings										
Remote host *		🗌 Sp	becify use	rname		2	Po	rt	•	
Advanced SSH settings	Terminal s	ettings 🔅 N	etwork setti	ngs 🌟 I	Bookmark s	settings				
									0	ı
	Secure S	hell (SSH)	) sessio	n					<u> </u>	
					Connel					
		<b>O</b> K			Jancer					
gure 9-14 Settir	ng param	neters								
ssion settings										
sion settings	X =	<b>V</b> E	2 4		•	>		(m))	<b>4</b> 0	
sion settings	Xdmcp RDP	VNC F	TP SFT	P Seria	I File	Shell	<b>o</b> Browse	Mosh	ere Si Aws Si Aw	WSL
sion settings SSH Telnet Rsh Varning: you have reached the m ou can start a new session but SBasic SSH settings	Xdmcp RDP naximum numbe it will not be auto	VNC F r of saved sessio matically saved.	TP SFT	P Seria ersonal editi	I File	Shell Xterm.	<b>(</b> Browse	Mosh	ws S3	WSL
Assion settings	Xdmcp RDP naximum numbe it will not be auto	VNC F rof saved session matically saved.	TP SFT ons for the pe username	P Seria ersonal edit	I File	Shell Xterm.	Browse	Mosh	💖 Aws S3	WSL
Sion settings SSH Telnet Rsh Varning: you have reached the m You can start a new session but Basic SSH settings Remote host * 100.	Xdmcp RDP naximum numbe it will not be auto	VNC F of saved session matically saved.	TP SFT ons for the pe username	P Seria ersonal edit	I File	Shell Xterm.	Browser	Mosh	ws S3	WSL
ssion settings SSH Telnet Rsh 2 SSH Telnet Rsh 2 Warning: you have reached the m You can start a new session but Basic SSH settings Remote host * 100. Please support MobaXterm by Advanced SSH settings	Xdmcp RDP naximum numbe it will not be auto	VNC F r of saved sessif matically saved.	TP SFT ons for the pe username edition here	P Seria prsonal edition	I File ion of Mobal	Shell Xterm. 2 Pc nobatek.ne kmark sett	Browser browser	Mosh	ws S3	WSL
ssion settings SSH Telnet Rsh 2 SSH Telnet Rsh 2 Warning: you have reached the m You can start a new session but Basic SSH settings Remote host * 100. Please support MobaXterm by Advanced SSH settings	Xdmcp RDP naximum numbe it will not be auto	VNC F r of saved sessif matically saved.	TP SFT ons for the pe username edition here	P Seria arsonal editi root : https://mu settings	I File Ion of Mobal	Shell Xterm.	Browsei ort t	Mosh	ws S3	WSL
ssion settings SSH Telnet Rsh Varning: you have reached the m You can start a new session but Basic SSH settings Remote host * 100. Please support MobaXterm by Advanced SSH settings	Xdmcp RDP naximum numbe it will not be auto	VNC F r of saved sessif matically saved.	TP SFT ons for the pe username edition here	P Seria ersonal editi root : https://mr settings	I File I File I File I File I I I I I I I I I I I I I I I I I I I	Shell Xterm.	Browser ortt t	Mosh	ws S3	WSL
ssion settings SSH Telnet Rsh Warning: you have reached the m You can start a new session but Basic SSH settings Remote host * 100. Please support MobaXterm by Advanced SSH settings	Xdmcp RDP naximum numbe it will not be auto subscribing to t	VNC F rof saved sessic matically saved. Specify the Professional al settings	SSH) Sest	P Seria ersonal edit	I File I File I File I File I I I I I I I I I I I I I I I I I I I	Shell Xterm.	Browser ortt t	Mosh	eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee	WSL
ssion settings	Xdmcp RDP naximum numbe it will not be auto subscribing to t	VNC F r of saved session matically saved.	SSH) sess	P Seria ersonal edit	I File I File File I File I File File File File File File File File	Shell Xterm.	Browsee	Mosh	Aws S3	WSL
ssion settings SSH Telnet Rsh Varning: you have reached the n you can start a new session but Basic SSH settings Remote host * 100. Please support MobaXterm by Advanced SSH settings	Xdmcp RDP naximum numbe it will not be auto	VNC F r of saved session matically saved.	SSH) sess	P Seria ersonal edit	I File I File Ioon of Mobal	Shell Xterm.	Browsee	Mosh	Aws S3	WSL

0 12 Ectablishin **-:** o cti c

Click **Advanced SSH settings**, select the key pair used for creating the instance, and click **OK**.

			X	-	V.C				8	D	0	and D	0.0	-
SH	Telnet Rs	h Xol	mcp	RDP	VNC	FTP	SFTP	Serial	File	Shell	Browser	Mosh	Aws S3	WSL
ning: yo	u have reache	the max	dmum	number o	fsaveds	essions fo	r the perso	nal edition	of MobaX	term.				
an sta	rt a new sessio	n but it v	vill not l	be autom:	atically sa	ved.								
Das	IC SSH setting	s												
Re	emote host * 1	0.05.1	-		Spe	ecify user	name ro	oot	2	P	ort 👘			
se sur	port MobaXte	m by su	bscribi	ina to the	Professi	onal editio	on here: ht	tos://moba	pterm m	obatek ne	t			
	portinopulation									o b at o t t. t t				
	10011		_						· · · · · · · · · · · · · · · · · · ·					
Adva	anced SSH se	tings	1	Terminal	settings	🔆 N	etwork se	ttings	🛨 Book	mark sett	ings			
Adva	anced SSH se	ttings	191	Terminal	settings	🔆 N	etwork se	ttings	📩 Book	m <mark>ark s</mark> eti	ings			
Adva	anced SSH se	tings	191	Terminal	settings	1 N	etwork se	ttings	📩 Book	mark set	ings			
Adva Adva	anced SSH se	itings warding		Terminal :	settings ompressio	Shirt N	etwork set Remote et	ttings	🛨 Book	mark sett	tings			
Adva Adva	anced SSH se	itings warding nmand:		Terminal :	settings ompressio	🔆 N	etwork set Remote e	ttings	★ Book : Interac xit after c	mark sett ctive shell	ends			
Adva Adva	anced SSH se ⊠X11-For Execute cor SSH-brows	warding nmand: er type:	SFTF	Terminal : Co P protoco	settings ompressio	Stranger	etwork set	nvironment Do not e	★ Book : Interac xit after c :SH path	mark sett ctive shell command (experim	ings v ends ental)		¢	
Adva Adva	⊠X11-For Execute cor SSH-brows	ttings warding nmand: er type:	SFTF	Terminal : Co	settings ompressio	X N	Remote e	ttings	★ Book : Interac xit after c :SH path	mark sett ctive shell command (experim	ends ental)		•	
Adva Adva	⊠X11-For Execute cor SSH-brows ⊠Use priv	ttings warding nmand: er type: ate key	SFTF	Terminal : Co	settings ompressio	in KeyPa	Remote e	nvironment Do not e Follow S Adapt Io	★ Book : Interac xit after c :SH path cales on	mark seti ctive shell command (experim remote s	ings v ends ental) erver		•	
Adva Adva	anced SSH se ⊠X11-For Execute cor SSH-brows ⊠Use priv	ttings warding nmand: er type: ate key xecute	SFTF	Terminal Co P protoco at sessio	settings ompressio I	KeyPa	Remote el	nvironment Do not e Follow S Adapt Io	★ Book Interact xit after c SH path cales on	mark sett ctive shell command (experim remote s	ends ental)		¢	

#### Figure 9-15 Configuring a key pair

Figure 9-16 Successful login



**Step 7** (Optional) To release an instance, in the elastic server list on the console, click **Delete** in the **Operation** column of the target instance. In the displayed dialog box, confirm the operation.

Figure 9-17 Deleting an instance

Elastic Server						
Create						
Name/ID J=	Monitoring	Status	Flavor	VPC	Remote Access	Operation
devserve Be743287-6ac6-4a5c	8	<ul> <li>Running</li> </ul>	physical.ki1ne.6xlarge)64cores(8*HUAWEI Ascend 910	vpc-		Start   Stop Delete   Synchronize

----End

# 9.3.4 Managing an Elastic Server

# 9.3.4.1 Creating an Elastic Server

# Prerequisites

You have enabled basic permissions, created a VPC, and created a key pair. For details, see **Preparations**.

# Procedure

- **Step 1** Log in to the ModelArts management console.
- **Step 2** In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Server**.
- **Step 3** Click **Create**. On the **Create DevServer Instance** page that is displayed, set parameters.

Table 9-5 Basic information

Parameter	Description
Name	Name of an elastic server. Enter 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Billing Mode	Currently, only pay-per-use is supported.

Table 9-6 Detailed specification
----------------------------------

Parameter	Description
Resource Type	ECS and BMS are supported.
	• ECS provides scalable, on-demand cloud servers.
	• <b>BMS</b> features both the scalability of VMs and high performance of physical servers. It provides dedicated servers on the cloud.
CPU Architecture	CPU architecture of the resource type, which can be x86 or Arm
AZ	A standalone data center with an independent network and power supply. When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.
	<ul> <li>For high DR capability, deploy resources in different AZs within the same region.</li> </ul>
	• For lower network latency, deploy resources in the same AZ.
Flavor	General-computing and GPU-accelerated flavors are supported. The flavors vary by region. The actual flavors are displayed on the console.

Parameter	Description
Image	VM image provided by ModelArts

 Table 9-7 Network resource parameters

Parameter	Description
Network	Network environment of the elastic server
RoCE Network	Configure the RoCE network for the elastic server if distributed scenarios are involved. If no RoCE network is available, contact the region personnel.
Name	RoCE network name. Set this parameter when the RoCE network is enabled.
Security Group	A security group implements access control for elastic servers that have the same security protection requirements in a VPC.
System Disk	This parameter is displayed only when you select a flavor that supports mounting. After an ECS is created, you can mount a data disk to the ECS or expand the capacity of the system disk on the ECS. The recommended value is not smaller than 100 GB.
Key Pair	This key pair is the only way to access the elastic server through SSH.
Enterprise Project	(Optional) You can select an enterprise project.

### Step 4 Click Next.

----End

# 9.3.4.2 Viewing Instance Details

- **Step 1** Log in to the ModelArts management console.
- **Step 2** In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Server**.
- **Step 3** In the **Elastic Server** list, click the target instance name to go to the instance details page and view details about the instance.

### Table 9-8 Parameters

Parameter	Description
Name	Name of an instance
ID	Unique instance ID

Parameter	Description
VPC	VPC of the instance
Image	Image used by the instance
Access Key	Key pair for logging in to the instance
Flavor	Specifications of the instance
Status	Status of the instance
BMS	BMS of an instance
Created At	Time when the instance is created
Updated At	Time when an instance is modified

----End

# 9.3.4.3 Using SSH to Remotely Log In to an Instance

# Prerequisites

- An elastic server is in the **Running** state.
- The key pair is available.

A key pair is automatically downloaded after you create it. Securely store your key pair. If an existing key pair is lost, create a new one.

• Operations in **Configuring the Network as an Administrator** have been performed.

# Procedure

- **Step 1** Contact the administrator to obtain the EIP and public port number.
- **Step 2** Start the SSH tool and set parameters (MobaXterm is used as an example).

Click **SSH** and type the EIP, username, and port.

										>
SSH Telnet F	🛃 🔯 Rsh Xdmcp	I VNC	S FTP	<pre> SFTP </pre>	ي Serial	<b>Q</b> File	Shell	<b>R</b> rowser	📡 Mosh	🚏 Aws S3
🛯 Basic SSH setti	ngs									
Remote host *			Specify user	name		2,	Pa	rt	•	
Advanced SSH set	tings 🔐 Termir	nal settings	Network settin	as 🔶 B	ookmark s	ettings				
-						ge				
	Coours								•	
	Secure	e Shell (SSF	i) sessior	1						
		Ok	(	8 C	ancel					
<b>aure 9-19</b> Se	etting par	ameters								
<b>gure 9-19</b> Se	etting para	ameters								
gure 9-19 Se sion settings	etting para	ameters		A.W.	Ţ	2	۲	M,	ŵ	=
gure 9-19 Se sion settings SSH Telnet Re Jarning: you have reache	etting para	ameters	FTP SFTF	P Serial rsonal editio	File File	≥ Shell Cterm.	Browse	Mosh	eee Aws S3	WSL
gure 9-19 Se sion settings SSH Telnet Re Jarning: you have reache ou can start a new sessi S Basic SSH setting	etting para Etting para Sh Xdmcp F d the maximum nui on but it will not be gs	Ameters	FTP SFTF sions for the per	P Serial rsonal editio	File File	> Shell Clerm.	Browse	Mosh	😵 Aws S3	WSL
gure 9-19 Se sion settings SSH Telnet Re Varning: you have reache ou can start a new sessi Basic SSH setting Remote host * [	etting para Markov ( Sh Xdmcp F dthe maximum num on but it will not be gs 100.	Ameters	FTP SFTF sions for the per fy username	P Serial rsonal editio	File File	Shell Cterm.	Browse rtt	Mosh	ese Aws S3	WSL
gure 9-19 Se ision settings SSH Telnet Re Varning: you have reache You can start a new sessi Basic SSH setting Remote host * [	etting para Market Stranger Sh Xdmcp F d the maximum num on but it will not be gs 100. rm by subscribing	Ameters	FTP SFTF sions for the per d. fy username	P Serial rsonal editio	File File on of Moba)	Shell Cterm.	Browse nt	Mosh	éé Aws S3	WSL
gure 9-19 Se ision settings SSH Telnet Re Varning: you have reache You can start a new sessi SB Basic SSH setting Remote host * [- Please support MobaXte	etting para h Xdmcp F d the maximum nui on but it will not be gs 100. m by subscribing ettings I Ter	Ameters	FTP SFTF sions for the period fy username al edition here:	P Serial rsonal editio root https://mol settings	File File on of Moba)	Shell Clerm.	Browser ntt	Mosh	eee Aws S3	WSL
gure 9-19 Se ssion settings SSH Telnet R: Varning: you have reache You can start a new sessi Basic SSH setting Remote host * [ Please support MobaXte	etting para h Xdmcp F d the maximum nu on but it will not be gs 100. rm by subscribing ettings I Ter	Ameters	FTP SFTF sions for the per fy username al edition here:	P Serial rsonal editio root https://mol settings	File File on of Moba)	Shell Clerm. Po obatek.net mark setti	Browser ntt	Mosh	ee Aws S3	WSL
gure 9-19 Se ssion settings SSH Telnet R: Varning: you have reache You can start a new sessi Basic SSH setting Remote host * [ Please support MobaXte	etting para h Xdmcp F d the maximum nu on but it will not be- gs 100. m by subscribing ettings I Ter	Ameters	FTP SFTF stons for the per fy username al edition here:	P Serial rsonal editio root https://mol settings	File File on of Moba)	Shell Clerm.	Browser ntt	Mosh	eee Aws S3	WSL
gure 9-19 Se ssion settings SSH Telnet Rt Naming: you have reache fou can start a new sessi Basic SSH setting Remote host * [ Please support MobaXte	etting para	Ameters	FTP SFTF stons for the per fy username al edition here: X Network and SSH) sess	P Serial rsonal editio root https://mol settings	File File File File File File File File	Shell Cterm.	Browsei nt ngs	Mosh	Aws S3	WSL
gure 9-19 Se ssion settings SSH Telnet Rt Varning: you have reache You can start a new sessi Basic SSH setting Remote host * [ Please support MobaXte	etting para	Ameters	FTP SFTF stons for the per fy username al edition here: Yetwork :	P Serial rsonal editio root https://mol settings	File File on of Moba)	Shell Cterm. Po obat ek. net armark setti	Browse nt ngs	Mosh	Aws S3	WSL
gure 9-19 Se ision settings SSH Telnet Rs Varning: you have reache ou can start a new sessi Basic SSH setting Remote host * [ * Please support MobaXte	etting para	Ameters	FTP SFTF stons for the per fy username al edition here: Network : (SSH) sess	P Serial rsonal editio root https://mol settings	File File on of Moba	Shell Cterm.	Rrowsee	Mosh	Aws S3	WSL

Figure 9-18 Establishing an SSH connection

Click **Advanced SSH settings**, select the key pair used for creating the instance, and click **OK**.

	<b>(</b>	e Contra	X		v C	۲	۲	AN	3	3	3	a la		
H	Telnet	Rsh	Xdmcp	RDP	VNC	FTP	SFTP	Serial	File	Shell	Browser	Mosh	Aws S3	WSL
ing: yo an sta	ou have rea art a new se	ached the ession bu	maximum it it will not	number of be automa	f saved se atically say	ssions for red.	r the perso	nal edition	of MobaX	term.				
Bas	sic SSH se	ettings												
-				_										
Re	ernote hos	st * 1			≤Spe	cify userr	name ro	ot	2	P	ort			
se suj	pport Moba	aXterm b	y subscrib	ing to the	Professio	nal editio	n here: ht	tps://moba	oxterm.mo	obatek.ne	et			
se suj	pport Mobi	aXterm b	y subscrib	ing to the	Professio	onal editio	n here: ht	tps://moba	ixterm.mo	obatek.ne	et			
e suj Adv	pport Moba	aXterm by H setting	y subscrib s	ing to the Terminal s	Professio settings	onal editio	on here: ht etwork set	tps://moba	ixterm.mo	obatek.ne mark seti	et tings			
se suj Adv	pport Moba	aXterm b H setting	s s	ing to the Terminal s	Professio settings	onal editio	on here: ht etwork set	tps://moba tings	ixterm.mo	obatek.ne mark seti	et tings			
se suj	pport Moba	aXterm by	y subscrib s	ing to the Terminal s	Professio settings	onal editio	on here: ht	tps://moba	ixterm.mo	obatek.ne mark seti	et tings			
Adv	pport Moba ranced SSI	aXterm b H setting 1-Forward	y subscrib s s	ing to the Terminal s ⊡Co	Professio settings mpressio	nal editio	en here: ht etwork set Remote er	tings	ixterm.mo transformer.mo Booki Interac	obatek.ne mark sett	et tings			
Adv	pport Moba vanced SSI	aXterm b H setting 1-Forward e comma	y subscrib s IS ding	ing to the Terminal s ⊡Co	Professio settings mpressio	nal editio	n here: ht etwork set Remote er	tings	xterm.mo Booki Interaci xit after c	batek.ne mark seti ctive shell	tings			
Adv	pport Mobi ranced SSI	aXterm b H setting 1-Forward e comma	y subscrib s s ding	ing to the Terminal s Co	Professio settings mpressio	nal editio	etwork set	tings tings nvironment	xterm.mo	mark sett	tings			
ae suj	Pport Mobi ranced SSI 2X1 <sup>-</sup> Execute SSH-b	aXterm b H setting 1-Forward e comma rowser ty	y subscrib s 191 ding nd: rpe: SFT	ing to the Terminal s I Co P protocol	Professio settings mpressio	nal editio	Remote er	tings tings nvironment Do not e Follow S	xterm.mo Booki Interac xit after c SH path	obatek.ne mark sett tive shell command (experim	et tings		ſ	
Adv	pport Moba ranced SSI ⊠X1 <sup>-</sup> Execute SSH-b ⊠Use	aXterm by H setting 1-Forward e comma rowser ty e private	y subscrib s 19 ding und: rpe: SFTT key .	ing to the Terminal s Co P protocol	Professio settings mpressio	n f	etwork set	tings wironment Do not e Follow S Adapt Io	xterm.mo	mark sett crive shell command (experim remote s	ental) enver		٩	
Adv	pport Mobi ranced SSI ⊠X1 <sup>-</sup> Execut SSH-bi ⊠Use	aXterm by H setting 1-Forward e comma rowser ty e private	y subscrib s 5 ling ind: rpe: SFTT key	ing to the Terminal s ☑ Co P protocol	Professic settings mpressio	n F	Remote er	tps://moba tings nvironment: Do not e Follow S Adapt Io	term.mo Booki Interaci SH path cales on	mark sett tive shell command (experim remote s	ettings		¢	

### Figure 9-20 Configuring a key pair

**Step 3** Check whether the login is successful as shown in the following figure.



Figure 9-21 Successful login

----End

### 9.3.4.4 Starting or Stopping an Instance

Stop instances that are not needed and restart them when they are needed again.

- Log in to the ModelArts management console. In the navigation pane, choose Dedicated Resource Pools > Elastic Server.
- 2. Perform the following operations to start or stop an elastic server:
  - To start an elastic server, click **Start** in the **Operation** column. Only stopped instances can be started.

- To stop an elastic server, click **Stop** in the **Operation** column. Only running instances can be stopped.

#### **NOTE**

Please note that instances are stopped in forcible shutdown mode, which may interrupt your services. Make sure you have saved the files on them before stopping.

### 9.3.4.5 Synchronizing the Status of an Elastic Server

After you change the status of a BMS on the Cloud Server Console, synchronize the change to the elastic server on ModelArts.

- Step 1 Log in to the ModelArts management console.
- Step 2 In the navigation pane, choose **Dedicated Resource Pools** > Elastic Server.
- **Step 3** In the **Elastic Server** list, click **Synchronize** in the **Operation** column of the target instance. In the dialog box that is displayed, click **OK**.

----End

### 9.3.4.6 Deleting an Instance

Delete the elastic servers that are no longer used.

- **Step 1** Log in to the ModelArts management console.
- **Step 2** In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Server**.
- **Step 3** In the **Elastic Server** list, click **Delete** in the **Operation** column of the target instance. In the dialog box that is displayed, click **OK**.

----End

# 9.3.5 Configuring the Network as an Administrator

### Context

After an elastic server is created, you need to contact the administrator to configure the network before accessing the elastic server using SSH. This section describes how to configure the network as the administrator. The following steps must be performed using the administrator account.

### Prerequisites

An elastic server has been created.

### Step 1: Create an EIP and a NAT Gateway

- **Step 1** Log in to the management console.
- Step 2 In the service list on the left, choose Networking > Elastic IP.
- Step 3 Click Buy EIP.
- **Step 4** Retain the default settings and click **Next**.

#### **Step 5** Choose **NAT Gateway** > **Public NAT Gateway**.

#### Step 6 Click Buy Public NAT Gateway.

**Step 7** Select the VPC and subnet for the elastic server, retain the default settings for other parameters, and click **Next**.

----End

### **Step 2: Configure SNAT and DNAT Rules**

- Step 1 On the Public NAT Gateways page, click the name of the created NAT gateway.
- Step 2 On the SNAT Rules tab, click Add SNAT Rule.
- Step 3 Add an SNAT rule.

Scenario: VPC

Subnet: Use an existing subnet.

EIP: Select the created EIP.

- Step 4 Click OK.
- Step 5 On the DNAT Rules tab, click Add DNAT Rule.
- **Step 6** Configure a DNAT rule.

Scenario: VPC

Port Type: Specific port

Protocol: TCP

EIP: Select the created EIP.

**Outside Port**: You are advised to set this parameter to a value ranging from 20000 to 30000 to ensure that the port number is unique.

**Private IP Address**: Enter the IP address of the elastic server. You can click **View ECS IP Address** and then click **Bare Metal Server** to view the IP address.

Inside Port: 22

Step 7 Click OK.

----End

# 9.4 Monitoring Resources

# 9.4.1 Overview

All metrics reported by ModelArts are stored in AOM, which enables you to consume metrics. You can view metric threshold alarms and reported alarms on the AOM console or use visualization tools such as Grafana to view and analyze the alarms. Grafana provides different views and templates for monitoring, which allow you to see the real-time resource usage on dashboards clearly.

# 9.4.2 Using Grafana to View AOM Monitoring Metrics

### 9.4.2.1 Procedure

Grafana supports various monitoring views and templates, meeting your diverse requirements. After adding the data source in Grafana, you can view all ModelArts monitoring metrics stored in AOM using Grafana.

To view AOM monitoring metrics using Grafana plugins, perform the following steps:

1. Installing and Configuring Grafana

#### **NOTE**

You can install and configure Grafana using any of the following ways: **Installing and Configuring Grafana on Windows**, **Installing and Configuring Grafana on Linux**, and **Installing and Configuring Grafana on a Notebook Instance**.

- 2. Configuring a Grafana Data Source
- 3. Using Grafana to Configure Dashboards and View Metric Data

# 9.4.2.2 Installing and Configuring Grafana

### 9.4.2.2.1 Installing and Configuring Grafana on Windows

### **Application Scenario**

This section describes how to install and configure Grafana on a Windows operating system.

### Procedure

- Download the Grafana installation package.
   Go to the download link, click Download the installer, and wait until the download is successful.
- 2. Install Grafana.

Double-click the installation package and install Grafana as instructed.

- 3. In Windows Services Manager, enable Grafana.
- 4. Log in to Grafana.

Grafana runs on port 3000 by default. After you open http://localhost:3000, the Grafana login page is displayed. The default username and password for the first login are **admin**. After the login is successful, change the password as prompted.

<image/> Contraction   Defense to contraction   Defense to contraction   Defense to contraction   Desend   Desend   Desend   Desend   Desend		
Welcome to Grafana Email or username email or username Password password Log in Forgot your password?	<b>L</b> O	
Email or username email or username Password password Log in Forgot your password?	Welcome to Grafana	
Password password Log in Forgot your password?	Email or username email or username	
password (©) Log in Forgot your password?	Password	
Log in Forgot your password?	password	۲
Forgot your password?	Log in	
	Forgot your pas	sword?

### 9.4.2.2.2 Installing and Configuring Grafana on Linux

### Prerequisites

- An Ubuntu server that is accessible to the Internet is available. If no, the following conditions must be met:
- You have obtained an ECS. (You are advised to select 8 vCPUs or higher, Ubuntu image of 22.04 version, and 100 GB local storage.) For details, see "Getting Started" > "Purchasing an ECS with Customized Configurations" in the *"Elastic Cloud Server User Guide*.
- You have purchased an EIP and bound it to the ECS. For details, see "Getting Started" > "Assigning an EIP and binding it to an ECS" in the *Elastic IP User Guide*.

### Procedure

- Log in to the ECS. Select a login method. For details, see "ECS Instances" > "Logging In to a Linux ECS" > "Linux ECS Login Overview" in the *Elastic Cloud Server User Guide*.
- 2. Run the following command to install libfontconfig1: sudo apt-get install -y adduser libfontconfig1

The operation is successful if the following information is displayed:



3. Run the following command to download the Grafana installation package: wget https://dl.grafana.com/oss/release/grafana\_9.3.6\_amd64.deb --no-check-certificate

#### Download completed:



4. Run the following command to install Grafana: sudo dpkg -i grafana\_9.3.6\_amd64.deb



- 5. Run the following command to start Grafana: sudo /bin/systemctl start grafana-server
- 6. Access Grafana configurations on your local PC.

Ensure that an EIP has been bound to the ECS and the security group configuration is correct (the inbound traffic from TCP port 3000 and all outbound traffic are allowed). Configuration process:

a. Click the ECS name to go to the ECS details page. Then, click the **Security Groups** tab, and click **Manage Rule**.

< ecs-5d50						
Sum	nmary	Disks	Network	Interfaces	Security Groups	
[	192	(primary	')		•	
	All (1)	C	)rganize	Change S	ecurity Group	
	1 sg-56	961		N	lanage Rule	

b. Click **Inbound Rules** and allow inbound traffic from TCP port 3000. By default, all outbound traffic is allowed.

< sp.set	C Feedback D import Rule					
Summary Inbound Rules Outbound Rules Associated Instances						
Some sexity group rules will not take effect for ECBs with exitan specifications. Learn more						
Add Rule Pash-Add Rule Dokles Addre Common Parts Housed Risks: 2 Learn more about security group configuration.						
T specify filter enteria.						
Priority ③ Action ③ Protocol & Pert ③ Type Source ③ Description Last Modified	I Operation					
Protocols/TCP (Custon V						
1 Allow Job 0.0.00 Contains April. 2023 C	39:20:06 GMT+08:00 Confirm   Cancel					

7. Access http://{*EIP*}:3000 in a browser. The default username and password for the first login are **admin**. After the login is successful, change the password as prompted.
| Welcome to Grafana    |
|-----------------------|
|                       |
| Email or username     |
| email or username     |
| Password              |
| password O            |
| Log in                |
| Forgot your password? |
|                       |
|                       |
|                       |

## 9.4.2.2.3 Installing and Configuring Grafana on a Notebook Instance

## Prerequisites

- A running CPU- or GPU-based notebook instance is available.
- A terminal is opened.



## Procedure

1. Run the following commands in sequence in your terminal to download and install Grafana:

```
mkdir -p /home/ma-user/work/grf

<u>cd /home/ma-user/work/grf</u>

wget https://dl.grafana.com/oss/release/grafana-9.1.6.linux-amd64.tar.gz

tar -zxvf grafana-9.1.6.linux-amd64.tar.gz

(%forch:18) [m-user work]&ddr -p /hom/ma-user/work/grf

(%forch:18) [m-user work]&ddr -p /hom/ma-user/work]

(%forch:18) [m-user work]&ddr -p
```

- 2. Register Grafana with jupyter-server-proxy.
  - a. Run the following commands in your terminal: mkdir -p /home/ma-user/.local/etc/jupyter vi /home/ma-user/.local/etc/jupyter\_jupyter\_notebook\_config.py

```
(PyTorch-1.8) [ma-user grf]$vi /home/ma-user/.local/etc/jupyter/jupyter_notebook_config.py
In jupyter_notebook_config.py, add the following code, press Esc to exit,
and type :wq to save the changes:
c.ServerProxy.servers = {
    'grafana': {
        'command': ['/home/ma-user/work/grf/grafana-9.1.6/bin/grafana-server', '--
homepath', '/home/ma-user/work/grf/grafana-9.1.6', 'web'],
    'timeout': 1800,
    'port': 3000
    }
}
```

(PyTorch-1.8) [ma-user grf]\$mkdir -p /home/ma-user/.local/etc/jupyter

#### **NOTE**

b.

If jupyter\_notebook\_config.py (path: /home/ma-user/.local/etc/jupyter/ jupyter\_notebook\_config.py) contains the c.ServerProxy.servers field, add the corresponding key-value pair.

- 3. Modify the URL for accessing Grafana in JupyterLab.
  - a. In the navigation pane on the left, open the vi /home/ma-user/ work/grf/grafana-9.1.6/conf/defaults.ini file.
  - b. Change the root\_url and serve\_from\_sub\_path fields in [server].

Figure 9-22 Modifying the defaults.ini file



In the file:

The value of root\_url is in the format of https:{Jupyterlab domain name}/{Instance ID}/grafana. You can obtain the domain name and instance ID from the address box of the JupyterLab page.

C 2 (auto-W) - JupyterLab × + ← → C P authoring-modelarts J.huaweicloud.com/9b1070cd-29f1-4346-9bc6 //ab/workspaces/auto-W

- Set Serve\_from\_sub\_path to true.
- 4. Save the image of the notebook instance.
  - a. Log in to the ModelArts console and choose **DevEnviron** > **Notebook**. In the notebook instance list, choose **More** > **Save Image** in the **Operation** column of the target instance.

Operation		
Open	Start   Stop	More 🔺
Open	Delete	
open	Change In	nage
Open	Save Imag	e
Open	Access VS	Code

b. In the **Save Image** dialog box, configure parameters. Click **OK** to save the image.

#### Figure 9-23 Saving an image

• Orgnization	Please select an or	gnization		• C (	Ireate
Image Name	Select or enter an	image name.			
Image Version	Enter an image ver	sion.			
Description					
			0/	256	
<ol> <li>snapshot will</li> <li>it will take 3</li> </ol>	not include mount pa	th(/home/ma-user/v	vork)		
2. It will turke D-					

c. The image will be saved as a snapshot, and it will take about 5 minutes. During this period of time, do not perform any operations on the instance.

#### Figure 9-24 Snapshotting



d. After the image is saved, the instance status changes to **Running**. Then, restart the notebook instance.

#### Figure 9-25 Image saved

Name ↓Ξ	Status ↓Ξ
notebc 295de0	Running 59 minute

5. Open the Grafana page.

Open a browser window and type the value of **root\_url** configured in **3** in the address box. If the Grafana login page is displayed, Grafana is installed and configured in the notebook instance. The default username and password for the first login are **admin**. After the login is successful, change the password as prompted.



## 9.4.2.3 Configuring a Grafana Data Source

Before viewing ModelArts monitoring data on Grafana, configure the data source.

## Prerequisites

• Grafana has been installed.

## Procedure

1. Add an access code.

a. Log in to the AOM console.

Service List	AOM
Elastic Cloud Server	Management & Governance
Bare Metal Server	Application Operations Management

b. In the navigation pane on the left, choose **Configuration Management** > **Agent Access**, and click **Add Access Code** to generate an access code.

Figure 9-26 Generating an access code

Agent Access ⑦		
AccessCode Access code is the identity credential for ca Tips: You can create up to 2 AccessCode	alling an API.	
	Add Access (	Code
Add Access Code	Generation Mode	Automatically generated
112d05eb624e3f78527610b7ad5bd130		OK Cancel

c. Click  $\bigcirc$  to view the generated access code.

Figure 9-27 Viewing the access code

ID	AccessCode
2a406bi	⊙ X****b
624ce2ff	💿 e****G

2. Obtain the data source URL.

The URL is in the format of https://{Endpoint}/v1/{project\_id}.

 Endpoints vary depending on services and regions. To obtain the regions and endpoints, contact the enterprise administrator.

A service endpoint consists of the service name, region ID, and external domain name in the format of "{service\_name}.{region\_id}. {external\_domain\_name}". Table 9-9 describes these parameters.

#### Table 9-9 Endpoint parameters

Parameter	Description	How to Obtain
service_name	Abbreviation of a case-insensitive service name	<b>aom</b> for AOM by default.
region_id	Region ID	Obtain the value from the system administrator.
external_domain_ name	External domain name suffix	Obtain the value from the system administrator.

- Set **project\_id** to the project ID of the corresponding region. You can obtain the project ID from **My Credentials**.

#### Figure 9-28 My Credentials



#### Figure 9-29 Obtaining the project ID

My Credentials	API Credentials ⑦
API Credentials Access Keys	IAM User Name
	Projects Project ID ↓≡

- 3. Add a data source to Grafana.
  - a. Log in to Grafana. The default username and password for the first login are **admin**. After the login is successful, change the password as prompted.

b. In the navigation pane, choose **Configuration** > **Data Sources**. Then, click **Add data source**.



c. Click **Prometheus** to access the configuration page.

Figure 9-31	Entering	the	Prometheus	configuration	page

	dd data source type
Q. Filter by r	iante or type cancel
	Prometheus Open source time series database & alerting
	Graphite Open source time series database
~~~	OpenTSDB Open source time series database
$\bigcirc$	InfluxD8 Open source time series database
Logging & do	cument databases
<u>   _</u>	Loki Like Prometheus but for logs. OSS logging solution from Grafana Labs
-	Elasticsearch Open source logging & analytics divisibase

d. Configure parameters as shown in the following figure.

Configure years of the end of the	our Promethe effort and get	u <b>s data source below</b> Prometheus (and Loki) as fi	ılly-man	aged, sca	lable, and host
Name 🕕 Prometheus	1			Default	
нттр					
URL	O https://www.com/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/action/ac	e.A.			
Access	Serv	er (default)			Help >
Allowed cookies	③ New	tag (enter key to add			
Timeout	① Time	out in seconds			
Auth					
Basic auth		With Credentials			
TLS Client Auth		With CA Cert			
Skip TLS Verify					
Forward OAuth Identity	0				
Basic Auth Details					
User	aom_access	_code			
Password					

Figure 9-32 Configuring a Grafana data source

#### **NOTE**

The actual Grafana version varies depending on the installation method. **Figure 9-32** is only an example.

Table 9-10 Parameters

Parameter	Description
Name	Customizable name
URL	URL https://{ <i>Endpoint</i> }/v1/ { <i>project_id</i> } combined in Obtain the data source URL.
Basic auth	Enabled
Skip TLS Verify	Enabled
User	aom_access_code

Parameter	Description
Password	Access code generated in Add an access code.

e. After the configuration, click **Save & test**. If the message **Data source is working** is displayed, the data source is configured.

Figure 9-33 Data source added

Exemplars + Add	
✓ Data source is working	
Back Explore Delete	Save & test

## 9.4.2.4 Using Grafana to Configure Dashboards and View Metric Data

In Grafana, you can customize dashboards for various views. ModelArts also provides configuration templates for clusters. This section describes how to configure a dashboard by using a ModelArts template or creating a dashboard. For more usage, see **Grafana tutorials**.

## **Preparations**

ModelArts provides templates for cluster view, node view, user view, task view, and task details view. These templates can be downloaded from Grafana official documents. You can import and use them on Dashboards.

Template Name	Download URL
Cluster view	https://grafana.com/grafana/dashboards/18582- modelarts-cluster-view/
Node view	https://grafana.com/grafana/dashboards/18583- modelarts-node-view/
User view	https://grafana.com/grafana/dashboards/18588- modelarts-user-view/
Task view	https://grafana.com/grafana/dashboards/18604- modelarts-task-view/
Task details view	https://grafana.com/grafana/dashboards/18590- modelarts-task-detail-view/

Table 9-11 Template download URLs

## Using a ModelArts Template to View Metrics

1. (Optional) Select the template you want to use. **Preparations** displays the download addresses of all templates. Open the target address and click **Download JSON**.

Figure 9-34 Downloading the template for the task details view



2. Open Dashboards and choose New > Import.

<mark>ہ</mark> م	Dashboards Manage dashboards an					
☆	<b>용 Browse</b> 포 Playlists					
88						New ~
0		Starred				Z) New Dashboard
÷						New Folder Import

- 3. Import a dashboard template in either of the following ways:
  - Method 1: Upload the JSON file downloaded in 1, as shown in Figure 9-35.
  - Method 2: Copy the template download address provided in **Preparations** and click **Load**, as shown in **Figure 9-36**.

-			
	Dashboards Manage dashboards and folders		
	器 Browse 👳 Playlists 💿 Snapshots 🔡 Library panels		
	Upload JSON file		
	Grafana.com dashboard URL or ID	Load	
	Import via panel json		
	Load		

Figure 9-35 Uploading a JSON file to import a dashboard template

Figure 9-36 Copying the template address and importing the dashboard template

	ashb anage da	Oards ishboards and	l folders					
器 Brows	ie 및	Playlists	Snaps	hots	문급 Library	y panels		
Upload JSON file								
https://gr	afana.cor	m/grafana/da	shboards/18	3582-moo	delarts-clus	ter-view/		Load
Import via p	anel json							
Load								

4. Change the view name and click **Import**.

ModelArts-User-View-te	st		
Folder			
General			
dashboard between multiple for accessing dashboards so	a dashboard can be used Grafana installs. The UID a changing the title of a das	for uniquely identify a Ilows having consistent URLs hboard will not break any	
dashboard between multiple for accessing dashboards so bookmarked links to that das	a dashboard can be used Grafana installs. The UID a changing the title of a das hboard.	for uniquely identify a Ilows having consistent URLs hboard will not break any	
dashboard between multiple for accessing dashboards so bookmarked links to that das 8QcY6pLVk	a dashboard can be used f Grafana installs. The UID a changing the title of a das hboard.	for uniquely identify a Ilows having consistent URLs hboard will not break any	Change uid
dashboard between multiple for accessing dashboards so bookmarked links to that das 8QcY6pLVk	a dashboard can be used i Grafana installs. The UID a changing the title of a das hboard. delArts-User-View <sup>*</sup> in folde	for uniquely identify a llows having consistent URLs hboard will not break any <b>'General' has the same UID</b>	Change uid
dashboard between multiple for accessing dashboards so bookmarked links to that das 8QcY6pLVk	a dashboard can be used f Grafana installs. The UID a changing the title of a das hboard. delArts-User-View' in folde	for uniquely identify a llows having consistent URLs hboard will not break any <b>r 'General' has the same UID</b>	Change uid

#### Figure 9-37 Changing the view name

Note: If a message is displayed, indicating that the UID is duplicate, change the UID in the JSON file and click **Import**.

#### Figure 9-38 Changing the UID



5. After the import, view the imported views in **Dashboards**. Then, click a view to open the monitoring page.

0 2	Create and manage dashboards to visualize your data		
☆	Browse •		
88	Search for dashboards		New ~
ø 4	♥ Filter by tag	t≡ Sort (Default A-Z)	
	C General		
	C) General		
@ 0	C General		

6. Use the template.

After the import is successful, you can click the template to view its details. This section introduces some common functions.

- Changing the data source and resource pool

Figure 9-39 Changing the data source and resource pool



Click the area marked by the red box. A drop-down list will appear. From there, you can change the data source and the resource pool.

Refreshing data



Click the refresh button in the upper right corner to refresh all data on the dashboard. The data on each panel is also updated.

- Changing the automatic refresh time

Figure 9-40 Changing the automatic refresh time

88 General	I / ModelArts-Us	ser-View2 ☆ ≪									Ð
data_source											
i	任务数	17	GPU使用情况			NPU使用情况		i CPU8内	存使用情况		
	1	<sub>占用卡数</sub> 0	GPU平均利用率 <b>0%</b>	显存平均利用率 <b>0%</b>	<sup>占用卡数</sup>	NPU平均利用率 <b>0%</b>	NPU内存平均利用率 <b>0%</b>	CPU平均利用率 <b>2%</b>	内存	30s 1m 5m 15m	蘨
1				165	524小时任劳概要					30m 1h	Pours
	<b>⊞\$10</b> ⊽		検型 ▽			医行时长 🍸	CPU平均利用率 💎				
			notebook	2023/05/11 18:54:00		56 min					

The default refresh interval of a template is 15 minutes. If you need to update the interval, change the value from the drop-down list box in the upper right corner.

- Changing the time range for obtaining dashboard data

Figure 9-41 Changing the time range for obtaining data

88 General / ModelArts-User-View2 😭 🗳

Click the button in the upper right corner to change time range for obtaining data. This time range affects all panels except those with a fixed time.

Adding a panel

#### Figure 9-42 Adding a panel



Click the + icon in the upper right corner to add a panel.

After a panel is added, you can obtain the data in the panel. Configure the data source and resource pool as follows to use the current dashboard settings.



ModelArts-Cluster-View / Edit Panel		Oiscard Save Apply
dala_secreta		Time series
Panel Title		Q. Search options
15		All Overrides
		<ul> <li>Panel options</li> </ul>
5 added Norther Head Norther Deal Norther Norther Deal State Deal Deal Norther Deal Deal Deal Deal Deal Deal Deal Deal	. Nan den den den den den den den den den de	Tite Panel Title
0 0015 09.00 09.45 10.00 10.15 10.20 10.45 11.00 11.15 11.00 11.45 12.00 12.15 12.20 12.45 13.00	12:15 13:20 13:45 14:60 14:15 14:30 14:45 15:00	Description
		Transparent background
😫 Query 👔 🔅 Trensform 🕲 🔒 Net 🔘		<ul> <li>Panel links</li> </ul>
Data searce S(datasource) ~ 0 > Query options MD - auto = 1616 Interval = 156		+ Add link
		<ul> <li>Recent cations</li> </ul>
Guery patients 🤟 Explain 🌒 Raw query 🂽 🖾 Bive feedback	Run queries Builder Behin Code	Repeat by variable
Metric Labors		This is not visible while in edit mode. You need to go back to dashboard and then update the variable or reload the dashboard.
		Choose ~
+ Oprations		
ma_contsiner_cpu_util(pool_id="\$ptol_id")		<ul> <li>Tooltip</li> <li>Tooltin mote</li> </ul>
> Options Learned Auto Economit Time emine State auto. Time Renne Economical failes		(White invec

#### **Creating a Dashboard to View Metrics**

- 1. Open **Dashboards**, click **New**, and choose **New Dashboard**.
- 2. Click Add a new panel.
- On the New dashboard / Edit Panel page, set the following parameters:
   Data source: Configured Grafana data source

**Metric**: Metric name. You can obtain the metric to be queried by referring to **Table 9-12**, **Table 9-13**, and **Table 9-14**.

Labels: Used for filtering the metric. For details, see Table 9-15 and .



Figure 9-44 Creating a dashboard to view metrics

# 9.4.3 Viewing All ModelArts Monitoring Metrics on the AOM Console

ModelArts periodically collects the usage of key metrics (such as GPUs, NPUs, CPUs, and memory) of each node in a resource pool as well as the usage of key metrics of the development environment, training jobs, and inference services, and reports the data to AOM. You can view the information on AOM.

- 1. Log in to the console and search for **AOM** to go to the AOM console.
- 2. Choose **Metric Monitoring**. On the **Metric Monitoring** page that is displayed, click **Add Metric**.
- 3. Add metrics and click Add to Metric List.
  - Add By: Select All Metrics.
  - Metric Name: Select the desired ones for query. For details, see Table 9-12, Table 9-13, and Table 9-14.
  - Scope: Enter the tag for filtering the metric. For details, see Table 9-15.
     The following shows an example.
- 4. View the metrics.

Metric Browsing ①	
Metric Sources	
Statistic Avg    Statistical Period 1 minute	C Last 30 minutes •
••ma_container_step_unt]         Container_step_unt]         Container_step_unt]         Container_step_unt]           13	< 1/1 >
L' Line ⑧ Digit t≩ Top N  ☐ Table	
Metric List db Graph Settings	
Ø Metric Query	Enter a Metrics name.
Metric Name     Dimension     Group Key     ma_container_cou_util     account_name: '	Operations 民

## Table 9-12 Container metrics

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
CPU	CPU Usage	ma_container_c pu_util	CPU usage of a measured object	%	0%–100%
	Used CPU Cores	ma_container_c pu_used_core	Number of CPU cores used by a measured object	Cores	≥ 0
	Total CPU Cores	ma_container_c pu_limit_core	Total number of CPU cores that have been applied for a measured object	Cores	≥ 1
Memo ry	Total Physical Memory	ma_container_ memory_capaci ty_megabytes	Total physical memory that has been applied for a measured object	МВ	≥ 0

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	Physical Memory Usage	ma_container_ memory_util	Percentage of the used physical memory to the total physical memory	%	0%–100%
	Used Physical Memory	ma_container_ memory_used_ megabytes	Physical memory that has been used by a measured object (container _memory_ working_s et_bytes in the current working set) (Memory usage in a working set = Active anonymou s page and cache, and file-baked page ≤ container_ memory_u sage_bytes )	MB	≥ 0
Storag e	Disk Read Rate	ma_container_ disk_read_kilob ytes	Volume of data read from a disk per second	KB/s	≥ 0
	Disk Write Rate	ma_container_ disk_write_kilo bytes	Volume of data written into a disk per second	KB/s	≥ 0

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
GPU memo ry	Total GPU Memory	ma_container_ gpu_mem_total _megabytes	Total GPU memory of a training job	МВ	> 0
	GPU Memory Usage	ma_container_ gpu_mem_util	Percentage of the used GPU memory to the total GPU memory	%	0%–100%
	Used GPU Memory	ma_container_ gpu_mem_used _megabytes	GPU memory used by a measured object	МВ	≥ 0
GPU	GPU Usage	ma_container_ gpu_util	GPU usage of a measured object	%	0%–100%
	GPU Memory Bandwidth Usage	ma_container_ gpu_mem_copy _util	GPU memory bandwidth usage of a measured object For example, the maximum memory bandwidth of NVIDIA GPU V100 is 900 GB/s. If the current memory bandwidth is 450 GB/s, the memory bandwidth usage is 50%.	%	0%-100%

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	GPU Encoder Usage	ma_container_ gpu_enc_util	GPU encoder usage of a measured object	%	%
	GPU Decoder Usage	ma_container_ gpu_dec_util	GPU decoder usage of a measured object	%	%
	GPU Temperatur e	DCGM_FI_DEV_ GPU_TEMP	GPU temperatur e	°C	Natural number
	GPU Power	DCGM_FI_DEV_ POWER_USAGE	GPU power	Watt (W)	> 0
	GPU Memory Temperatur e	DCGM_FI_DEV_ MEMORY_TEM P	GPU memory temperatur e	°C	Natural number
Netwo rk I/O	Downlink rate	ma_container_ network_receiv e_bytes	Inbound traffic rate of a measured object	Bytes/s	≥ 0
	Packet receive rate	ma_container_ network_receiv e_packets	Number of data packets received by an NIC per second	Packets/s	≥ 0
	Downlink Error Rate	ma_container_ network_receiv e_error_packets	Number of error packets received by an NIC per second	Packets/s	≥ 0
	Uplink rate	ma_container_ network_trans mit_bytes	Outbound traffic rate of a measured object	Bytes/s	≥ 0

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	Uplink Error Rate	ma_container_ network_trans mit_error_pack ets	Number of error packets sent by an NIC per second	Packets/s	≥ 0
	Packet send rate	ma_container_ network_trans mit_packets	Number of data packets sent by an NIC per second	Packets/s	≥ 0
NPU	NPU Usage	ma_container_ npu_util	NPU usage of a measured object (To be replaced by ma_contai ner_npu_ai _core_util)	%	0%-100%
	NPU Memory Usage	ma_container_ npu_memory_u til	Percentage of the used NPU memory to the total NPU memory (To be replaced by ma_contai ner_npu_d dr_memor y_util for Snt3 series, and ma_contai ner_npu_h bm_util for Snt9 series)	%	0%-100%

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	Used NPU Memory	ma_container_ npu_memory_u sed_megabytes	NPU memory used by a measured object (To be replaced by ma_contai ner_npu_d dr_memor y_usage_b ytes for Snt3 series, and ma_contai ner_npu_h bm_usage _bytes for Snt9 series)	≥ 0	MB
	Total NPU Memory	ma_container_ npu_memory_t otal_megabyte s	Total NPU memory of a measured object (To be replaced by ma_contai ner_npu_d dr_memor y_bytes for Snt3 series, and ma_contai ner_npu_h bm_bytes for Snt9 series)	> 0	MB
	Al Processor Error Codes	ma_container_ npu_ai_core_err or_code	Error codes of Ascend AI processors	N/A	N/A

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	Al Processor Health Status	ma_container_ npu_ai_core_he alth_status	Health status of Ascend AI processors	N/A	<ul> <li>1: healthy</li> <li>0: unhealt hy</li> </ul>
	Al Processor Power Consumptio n	ma_container_ npu_ai_core_po wer_usage_wat ts	Power consumpti on of Ascend Al processors	Watt (W)	> 0
	Al Processor Temperatur e	ma_container_ npu_ai_core_te mperature_celsi us	Temperatur e of Ascend Al processors	°C	Natural number
	AI Core Usage	ma_container_ npu_ai_core_uti l	Al core usage of Ascend Al processors	%	0%-100%
	Al Core Clock Frequency	ma_container_ npu_ai_core_fre quency_hertz	AI core clock frequency of Ascend AI processors	Hertz (Hz)	> 0
	AI Processor Voltage	ma_container_ npu_ai_core_vo ltage_volts	Voltage of Ascend AI processors	Volt (V)	Natural number
	Al Processor DDR Memory	ma_container_ npu_ddr_memo ry_bytes	Total DDR memory capacity of Ascend AI processors	Byte	> 0
	AI Processor DDR Usage	ma_container_ npu_ddr_memo ry_usage_bytes	DDR memory usage of Ascend AI processors	Byte	> 0
	AI Processor DDR Memory Utilization	ma_container_ npu_ddr_memo ry_util	DDR memory utilization of Ascend AI processors	%	0%–100%

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	Al Processor HBM Memory	ma_container_ npu_hbm_bytes	Total HBM memory of Ascend AI processors (dedicated for Snt9 processors)	Byte	> 0
	Al Processor HBM Memory Usage	ma_container_ npu_hbm_usag e_bytes	HBM memory usage of Ascend AI processors (dedicated for Snt9 processors)	Byte	> 0
	AI Processor HBM Memory Utilization	ma_container_ npu_hbm_util	HBM memory utilization of Ascend AI processors (dedicated for Snt9 processors)	%	0%–100%
	AI Processor HBM Memory Bandwidth Utilization	ma_container_ npu_hbm_band width_util	HBM memory bandwidth utilization of Ascend Al processors (dedicated for Snt9 processors)	%	0%-100%
	Al Processor HBM Memory Clock Frequency	ma_container_ npu_hbm_frequ ency_hertz	HBM memory clock frequency of Ascend AI processors (dedicated for Snt9 processors)	Hertz (Hz)	> 0

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
	Al Processor HBM Memory Temperatur e	ma_container_ npu_hbm_temp erature_celsius	HBM memory temperatur e of Ascend Al processors (dedicated for Snt9 processors)	°C	Natural number
	AI CPU Utilization	ma_container_ npu_ai_cpu_util	AI CPU utilization of Ascend AI processors	%	0%-100%
	AI Processor Control CPU Utilization	ma_container_ npu_ctrl_cpu_u til	Control CPU utilization of Ascend AI processors	%	0%–100%
NPU RoCE netwo rk	NPU RoCE Network Uplink Rate	ma_container_ npu_roce_tx_ra te_bytes_per_se cond	Uplink rate of the NPU network module used by the container	Bytes/s	≥ 0
	NPU RoCE Network Downlink Rate	ma_container_ npu_roce_rx_ra te_bytes_per_se cond	Downlink rate of the NPU network module used by the container	Bytes/s	≥ 0

Classif icatio n	Name	Metric	Descriptio n	Unit	Value Range
Noteb ook servic e metric s	Notebook Cache Directory Size	ma_container_ notebook_cach e_dir_size_byte s	A high- speed local disk is attached to the <b>/cache</b> directory for GPU and NPU notebook instances. This metric indicates the total size of the directory.	Bytes	≥ 0
	Notebook Cache Directory Utilization	ma_container_ notebook_cach e_dir_util	A high- speed local disk is attached to the <b>/cache</b> directory for GPU and NPU notebook instances. This metric indicates the utilization of the directory.	%	0%-100%

Table 9-13 Node metrics	(collected only	y in dedicated	resource pools)
-------------------------	-----------------	----------------	-----------------

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
CPU	Total CPU Cores	ma_node_c pu_limit_co re	Total number of CPU cores that have been applied for a measured object	Cores	≥ 1

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	Used CPU Cores	ma_node_c pu_used_co re	Number of CPU cores used by a measured object	Cores	≥ 0
	CPU Usage	ma_node_c pu_util	CPU usage of a measured object	%	0%–100%
	CPU I/O Wait Time	ma_node_c pu_iowait_ counter	Disk I/O wait time accumulate d since system startup	jiffies	≥ 0
Memory	Physical Memory Usage	ma_node_ memory_ut il	Percentage of the used physical memory to the total physical memory	%	0%–100%
	Total Physical Memory	ma_node_ memory_to tal_megab ytes	Total physical memory that has been applied for a measured object	МВ	≥ 0
Network I/O	Downlink rate	ma_node_n etwork_rec eive_rate_b ytes_secon ds	Inbound traffic rate of a measured object	Bytes/s	≥ 0
	Uplink rate	ma_node_n etwork_tra nsmit_rate_ bytes_seco nds	Outbound traffic rate of a measured object	Bytes/s	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
Storage	Disk Read Rate	ma_node_d isk_read_ra te_kilobyte s_seconds	Volume of data read from a disk per second (Only data disks used by containers are collected.)	KB/s	≥ 0
	Disk Write Rate	ma_node_d isk_write_r ate_kilobyt es_seconds	Volume of data written into a disk per second (Only data disks used by containers are collected.)	KB/s	≥ 0
	Total Cache	ma_node_c ache_space _capacity_ megabytes	Total cache of the Kubernetes space	МВ	≥ 0
	Used Cache	ma_node_c ache_space _used_capa city_megab ytes	Used cache of the Kubernetes space	МВ	≥ 0
	Total Container Space	ma_node_c ontainer_sp ace_capacit y_megabyt es	Total container space	МВ	≥ 0
	Used Container Space	ma_node_c ontainer_sp ace_used_c apacity_me gabytes	Used container space	МВ	≥ 0
	Disk Informatio n	ma_node_d isk_info	Basic disk informatio n	N/A	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	Total Reads	ma_node_d isk_reads_c ompleted_t otal	Total number of successful reads	N/A	≥ 0
	Merged Reads	ma_node_d isk_reads_ merged_tot al	Number of merged reads	N/A	≥ 0
	Bytes Read	ma_node_d isk_read_by tes_total	Total number of bytes that are successfully read	Bytes	≥ 0
	Read Time Spent	ma_node_d isk_read_ti me_second s_total	Time spent on all reads	Seconds	≥ 0
	Total Writes	ma_node_d isk_writes_ completed_ total	Total number of successful writes	N/A	≥ 0
	Merged Writes	ma_node_d isk_writes_ merged_tot al	Number of merged writes	N/A	≥ 0
	Written Bytes	ma_node_d isk_written _bytes_tota l	Total number of bytes that are successfully written	Bytes	≥ 0
	Write Time Spent	ma_node_d isk_write_ti me_second s_total	Time spent on all write operations	Seconds	≥ 0
	Ongoing I/Os	ma_node_d isk_io_now	Number of ongoing I/Os	N/A	≥ 0
	I/O Execution Duration	ma_node_d isk_io_time _seconds_t otal	Time spent on executing I/Os	Seconds	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	I/O Execution Weighted Time	ma_node_d isk_io_time _weighted_ seconds_to ta	Weighted time spent on executing I/Os	Seconds	≥ 0
GPU	GPU Usage	ma_node_g pu_util	GPU usage of a measured object	%	0%–100%
	Total GPU Memory	ma_node_g pu_mem_t otal_mega bytes	Total GPU memory of a measured object	МВ	> 0
	GPU Memory Usage	ma_node_g pu_mem_u til	Percentage of the used GPU memory to the total GPU memory	%	0%–100%
	Used GPU Memory	ma_node_g pu_mem_u sed_megab ytes	GPU memory used by a measured object	МВ	≥ 0
	Tasks on a Shared GPU	node_gpu_ share_job_c ount	Number of tasks running on a shared GPU	Number	≥ 0
	GPU Temperatur e	DCGM_FI_ DEV_GPU_ TEMP	GPU temperatur e	°C	Natural number
	GPU Power	DCGM_FI_ DEV_POWE R_USAGE	GPU power	Watt (W)	> 0
	GPU Memory Temperatur e	DCGM_FI_ DEV_MEM ORY_TEMP	GPU memory temperatur e	°C	Natural number

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
NPU	NPU Usage	ma_node_n pu_util	NPU usage of a measured object (To be replaced by ma_node_ npu_ai_cor e_util)	%	0%–100%
	NPU Memory Usage	ma_node_n pu_memor y_util	Percentage of the used NPU memory to the total NPU memory (To be replaced by ma_node_ npu_ddr_ memory_u til for Snt3 series, and ma_node_ npu_hbm_ util for Snt9 series)	%	0%-100%
	Used NPU Memory	ma_node_n pu_memor y_used_me gabytes	NPU memory used by a measured object (To be replaced by ma_node_ npu_ddr_ memory_u sage_bytes for Snt3 series, and ma_node_ npu_hbm_ usage_byt es for Snt9 series)	≥ 0	MB

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	Total NPU Memory	ma_node_n pu_memor y_total_me gabytes	Total NPU memory of a measured object (To be replaced by ma_node_ npu_ddr_ memory_b ytes for Snt3 series, and ma_node_ npu_hbm_ bytes for Snt9 series)	> 0	MB
	Al Processor Error Codes	ma_node_n pu_ai_core_ error_code	Error codes of Ascend Al processors	N/A	N/A
	Al Processor Health Status	ma_node_n pu_ai_core_ health_stat us	Health status of Ascend AI processors	N/A	<ul> <li>1: healthy</li> <li>0: unhealt hy</li> </ul>
	AI Processor Power Consumpti on	ma_node_n pu_ai_core_ power_usa ge_watts	Power consumpti on of Ascend Al processors	Watt (W)	> 0
	Al Processor Temperatur e	ma_node_n pu_ai_core_ temperatur e_celsius	Temperatur e of Ascend Al processors	°C	Natural number
	Al Core Usage	ma_node_n pu_ai_core_ util	Al core usage of Ascend Al processors	%	0%-100%

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	Al Core Clock Frequency	ma_node_n pu_ai_core_ frequency_ hertz	Al core clock frequency of Ascend Al processors	Hertz (Hz)	> 0
	Al Processor Voltage	ma_node_n pu_ai_core_ voltage_vol ts	Voltage of Ascend Al processors	Volt (V)	Natural number
	AI Processor DDR Memory	ma_node_n pu_ddr_me mory_bytes	Total DDR memory capacity of Ascend AI processors	Byte	> 0
	AI Processor DDR Usage	ma_node_n pu_ddr_me mory_usag e_bytes	DDR memory usage of Ascend AI processors	Byte	> 0
	AI Processor DDR Memory Utilization	ma_node_n pu_ddr_me mory_util	DDR memory utilization of Ascend AI processors	%	0%–100%
	Al Processor HBM Memory	ma_node_n pu_hbm_by tes	Total HBM memory of Ascend AI processors (dedicated for Snt9 processors)	Byte	> 0
	Al Processor HBM Memory Usage	ma_node_n pu_hbm_us age_bytes	HBM memory usage of Ascend Al processors (dedicated for Snt9 processors)	Byte	> 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	AI Processor HBM Memory Utilization	ma_node_n pu_hbm_ut il	HBM memory utilization of Ascend AI processors (dedicated for Snt9 processors)	%	0%–100%
	AI Processor HBM Memory Bandwidth Utilization	ma_node_n pu_hbm_ba ndwidth_ut il	HBM memory bandwidth utilization of Ascend AI processors (dedicated for Snt9 processors)	%	0%–100%
	AI Processor HBM Memory Clock Frequency	ma_node_n pu_hbm_fr equency_h ertz	HBM memory clock frequency of Ascend AI processors (dedicated for Snt9 processors)	Hertz (Hz)	> 0
	Al Processor HBM Memory Temperatur e	ma_node_n pu_hbm_te mperature_ celsius	HBM memory temperatur e of Ascend AI processors (dedicated for Snt9 processors)	°C	Natural number
	AI CPU Utilization	ma_node_n pu_ai_cpu_ util	AI CPU utilization of Ascend AI processors	%	0%–100%

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	AI Processor Control CPU Utilization	ma_node_n pu_ctrl_cpu _util	Control CPU utilization of Ascend AI processors	%	0%–100%
NPU RoCE network	NPU RoCE Network Uplink Rate	ma_node_n pu_roce_tx _rate_bytes _per_secon d	NPU RoCE network uplink rate	Bytes/s	≥ 0
	NPU RoCE Network Downlink Rate	ma_node_n pu_roce_rx _rate_bytes _per_secon d	NPU RoCE network downlink rate	Bytes/s	≥ 0
	MAC Uplink Pause Frames	ma_node_n pu_roce_m ac_tx_paus e_packets_t otal	Total number of pause frame packets sent by NPU RoCE network MAC	Number	≥ 0
	MAC Downlink Pause Frames	ma_node_n pu_roce_m ac_rx_paus e_packets_t otal	Total number of pause frame packets received by NPU RoCE network MAC	Number	≥ 0
	MAC Uplink PFC Frames	ma_node_n pu_roce_m ac_tx_pfc_p ackets_tota l	Total number of PFC frame packets sent by NPU RoCE network MAC	Number	≥ 0
Classificati on	Name	Metric	Descriptio n	Unit	Value Range
--------------------	------------------------------------	-----------------------------------------------------------	-----------------------------------------------------------------------------------------	--------	----------------
	MAC Downlink PFC Frames	ma_node_n pu_roce_m ac_rx_pfc_p ackets_tota l	Total number of PFC frame packets received by NPU RoCE network MAC	Number	≥ 0
	MAC Uplink Bad Packets	ma_node_n pu_roce_m ac_tx_bad_ packets_tot al	Total number of bad packets sent by NPU RoCE network MAC	Number	≥ 0
	MAC Downlink Bad Packets	ma_node_n pu_roce_m ac_rx_bad_ packets_tot al	Total number of bad packets received by NPU RoCE network MAC	Number	≥ 0
	RoCE Uplink Bad Packets	ma_node_n pu_roce_tx _err_packet s_total	Total number of bad packets sent by NPU RoCE	Number	≥ 0
	RoCE Downlink Bad Packets	ma_node_n pu_roce_rx _err_packet s_total	Total number of bad packets received by NPU RoCE	Number	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
InfiniBand or RoCE network	Total Amount of Data Received by an NIC	ma_node_i nfiniband_ port_receiv ed_data_by tes_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), received on all VLs from the port.	Double words (32 bits)	≥ 0
	Total Amount of Data Sent by an NIC	ma_node_i nfiniband_ port_trans mitted_dat a_bytes_tot al	The total number of data octets, divided by 4, (counting in double words, 32 bits), transmitted on all VLs from the port.	Double words (32 bits)	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
NFS mounting status	NFS Getattr Congestion Time	ma_node_ mountstats _getattr_ba cklog_wait	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permission s, owner, etc. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performanc e and slow system response times.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Getattr Round Trip Time	ma_node_ mountstats _getattr_rtt	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permission s, owner, etc. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server	ms	≥ 0
			issues.		

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Access Congestion Time	ma_node_ mountstats _access_ba cklog_wait	Access is an NFS operation that checks the access permission s of a file or directory for a given user. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performanc e and slow system response times.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Access Round Trip Time	ma_node_ mountstats _access_rtt	Access is an NFS operation that checks the access permission s of a file or directory for a given user. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Lookup Congestion Time	ma_node_ mountstats _lookup_ba cklog_wait	Lookup is an NFS operation that resolves a file name in a directory to a file handle. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performanc e and slow system response times.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Lookup Round Trip Time	ma_node_ mountstats _lookup_rtt	Lookup is an NFS operation that resolves a file name in a directory to a file handle. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Read Congestion Time	ma_node_ mountstats _read_back log_wait	Read is an NFS operation that reads data from a file. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performanc e and slow system response times.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Read Round Trip Time	ma_node_ mountstats _read_rtt	Read is an NFS operation that reads data from a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Write Congestion Time	ma_node_ mountstats _write_bac klog_wait	Write is an NFS operation that writes data to a file. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performanc e and slow system response times.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Write Round Trip Time	ma_node_ mountstats _write_rtt	Write is an NFS operation that writes data to a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0

Classif icatio n	Name	Metric	Description	Uni t	Value Rang e
InfiniB and or RoCE netwo rk	PortXmitData	infiniband_po rt_xmit_data_ total	The total number of data octets, divided by 4, (counting in double words, 32 bits), transmitted on all VLs from the port.	Tota l cou nt	Natur al numb er
	PortRcvData	infiniband_po rt_rcv_data_to tal	The total number of data octets, divided by 4, (counting in double words, 32 bits), received on all VLs from the port.	Tota l cou nt	Natur al numb er
	SymbolErrorC ounter	infiniband_sy mbol_error_c ounter_total	Total number of minor link errors detected on one or more physical lanes.	Tota l cou nt	Natur al numb er
	LinkErrorRec overyCounter	infiniband_lin k_error_recov ery_counter_t otal	Total number of times the Port Training state machine has successfully completed the link error recovery process.	Tota l cou nt	Natur al numb er
	PortRcvErrors	infiniband_po rt_rcv_errors_t otal	Total number of packets containing errors that were received on the port including: Local physical errors (ICRC, VCRC, LPCRC, and all physical errors that cause entry into the BAD PACKET or BAD PACKET DISCARD states of the packet receiver state machine) Malformed data packet errors (LVer, length, VL) Malformed link packet errors (operand, length, VL) Packets discarded due	Tota l cou nt	Natur al numb er
			Packets discarded due to buffer overrun (overflow)		

**Table 9-14** Diagnosis (IB, collected only in dedicated resource pools)

Classif icatio n	Name	Metric	Description	Uni t	Value Rang e
	LocalLinkInte grityErrors	infiniband_loc al_link_integri ty_errors_tota l	This counter indicates the number of retries initiated by a link transfer layer receiver.	Tota l cou nt	Natur al numb er
	PortRcvRemo tePhysicalErr ors	infiniband_po rt_rcv_remote _physical_erro rs_total	Total number of packets marked with the EBP delimiter received on the port.	Tota l cou nt	Natur al numb er
	PortRcvSwitc hRelayErrors	infiniband_po rt_rcv_switch_ relay_errors_t otal	Total number of packets received on the port that were discarded when they could not be forwarded by the switch relay for the following reasons: DLID mapping VL mapping Looping (output port = input port)	Tota l cou nt	Natur al numb er
	PortXmitWait	infiniband_po rt_transmit_w ait_total	The number of ticks during which the port had data to transmit but no data was sent during the entire tick (either because of insufficient credits or because of lack of arbitration).	Tota l cou nt	Natur al numb er
	PortXmitDisc ards	infiniband_po rt_xmit_discar ds_total	Total number of outbound packets discarded by the port because the port is down or congested.	Tota l cou nt	Natur al numb er

For details about the metrics of an InfiniBand or RoCE network, see **NVIDIA Mellanox documents**.

Table 9-15 Metric names

Classification	Metric	Description
Container metrics	modelarts_service	Service to which a container belongs, which can be <b>notebook</b> , <b>train</b> , or <b>infer</b>

Classification	Metric	Description
	instance_name	Name of the pod to which the container belongs
	service_id	Instance or job ID displayed on the page, for example, cf55829e-9bd3-48fa-8071-7ae870dae9 3a for a development environment 9f322d5a- b1d2-4370-94df-5a87de27d36e for a training job
	node_ip	IP address of the node to which the container belongs
	container_id	Container ID
	cid	Cluster ID
	container_name	Name of the container
	project_id	Project ID of the account to which the user belongs
	user_id	User ID of the account to which the user who submits the job belongs
	npu_id	Ascend card ID, for example, <b>davinci0</b> (to be discarded)
	device_id	Physical ID of Ascend AI processors
	device_type	Type of Ascend AI processors
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	pool_name	Name of a resource pool corresponding to a physical dedicated resource pool
	logical_pool_id	ID of a logical subpool
	logical_pool_name	Name of a logical subpool
	gpu_uuid	UUID of the GPU used by the container
	gpu_index	Index of the GPU used by the container
	gpu_type	Type of the GPU used by the container
	account_name	Account name of the creator of a training, inference, or development environment task
	user_name	Username of the creator of a training, inference, or development environment task

Classification	Metric	Description
	task_creation_time	Time when a training, inference, or development environment task is created
	task_name	Name of a training, inference, or development environment task
	task_spec_code	Specifications of a training, inference, or development environment task
	cluster_name	CCE cluster name
Node metrics	cid	ID of the CCE cluster to which the node belongs
	node_ip	IP address of the node
	host_name	Hostname of a node
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	project_id	Project ID of the user in a physical dedicated resource pool
	npu_id	Ascend card ID, for example, <b>davinci0</b> (to be discarded)
	device_id	Physical ID of Ascend AI processors
	device_type	Type of Ascend AI processors
	gpu_uuid	UUID of a node GPU
	gpu_index	Index of a node GPU
	gpu_type	Type of a node GPU
	device_name	Device name of an InfiniBand or RoCE network NIC
	port	Port number of the IB NIC
	physical_state	Status of each port on the IB NIC
	firmware_version	Firmware version of the IB NIC
	filesystem	NFS-mounted file system
	mount_point	NFS mount point
Diagnos	cid	ID of the CCE cluster to which the node where the GPU resides belongs
	node_ip	IP address of the node where the GPU resides

Classification	Metric	Description
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	project_id	Project ID of the user in a physical dedicated resource pool
	gpu_uuid	GPU UUID
	gpu_index	Index of a node GPU
	gpu_type	Type of a node GPU
	device_name	Name of a network device or disk device
	port	Port number of the IB NIC
	physical_state	Status of each port on the IB NIC
	firmware_version	Firmware version of the IB NIC

# **10** AI Hub

# 10.1 Al Hub

AI Hub is a ModelArts-empowered developer ecosystem community. In this community, scientific research institutions, AI application developers, solution integrators, enterprises, and individual developers can share and purchase AI assets such as algorithms, models, datasets, and workflows. This accelerates the development and implementation of AI assets and enables every participant in the AI development ecosystem to achieve business success.

AI Hub provides , where you can share AI assets such as algorithms, models, datasets, and workflows.

• Data: Datasets are shared.

Datasets on the **Data** page can be shared and downloaded. You can search for and download datasets meeting your service requirements on the **Data** page. You can also publish your local datasets in AI Hub and share them with other users.

• Algorithms: Algorithms are shared.

You can share and subscribe to algorithms on the **Algorithms** page. You are allowed to search for algorithms, subscribe to assets, and use them on the ModelArts management console. Additionally, you can publish self-developed algorithms to AI Hub and share them with other users.

• Asset Hub > Models: ModelArts models are shared.

You can publish and subscribe to shared models on the **Models** page. You are allowed to search for ModelArts models, subscribe to assets, and use them on the ModelArts management console. Additionally, you can publish selfdeveloped ModelArts models to AI Hub and share them with other users.

• Workflows: Workflows are shared.

You can share and subscribe to workflows on the **Workflows** page. You are allowed to search for workflows, subscribe to assets, and use them on the ModelArts management console. Additionally, you can publish self-developed workflows to AI Hub and share them with other users.

## AI Hub Constraints

- Subscribing to an AI asset is to purchase the usage quota of it. This AI asset can be used based on the quota.
- If you do not want to display a published AI asset in the asset list, discontinue it. After an AI asset is discontinued, it is visible only to the publisher. Even if a subscribed AI asset has been discontinued, the subscribers can still use it based on the quota. This ensures the subscribers' rights and prevents problems caused by discontinue operations.

# 10.2 Registering with AI Hub

Before sharing AI assets in AI Hub, register with AI Hub.

- 1. Click **Publish** on the **Algorithms** or **Models** page. The **Welcome to AI Hub** page is displayed.
- 2. On the **Welcome to AI Hub** page, enter your nickname. Then, click **OK**.
- 3. After the registration is complete, publish AI assets such as datasets and models in AI Hub.

# **10.3 Management Center**

View your information and the AI assets you have published and subscribed to in **Management Center**.

Option	Description
My Algorithms	Displays your published and algorithms that have been subscribed to.
	• <b>Published Algorithms</b> : View your published algorithms, such as the number of views, favorites, and subscriptions. Click <b>Release</b> or <b>Discontinue</b> on the right to manage published algorithms. After an asset is discontinued, users who have subscribed to the asset can continue using it, but other users cannot view or subscribe to it. Discontinued assets can be released again.
	• <b>My Subscriptions</b> : View your subscribed algorithms, such as the publisher, application console, and remaining quota. Click <b>Cancel Subscription</b> or <b>Retrieve</b> <b>Subscription</b> on the right to manage subscribed algorithms. After a subscription is canceled, the algorithm will not be available in <b>Subscription</b> on the ModelArts console. You can retrieve the subscription of an algorithm that has been unsubscribed from and continue to use the algorithm under the original quota constraints.

<b>Fable 10-1</b> Management	Center
------------------------------	--------

Option	Description
My Models	Displays the models you have published and subscribed to, including ModelArts models and HiLens skills.
	• <b>Published Models</b> : View your published models, such as the number of views, favorites, and subscriptions. Click <b>Release</b> or <b>Discontinue</b> on the right to manage published models. After an asset is discontinued, users who have subscribed to the asset can continue using it, but other users cannot view or subscribe to it. Discontinued assets can be released again.
	<ul> <li>My Subscriptions: View your subscribed models, such as the publisher, application console, and remaining quota. Click Cancel Subscription or Retrieve Subscription on the right to manage subscribed ModelArts models. After a subscription is canceled, the model will not be available in AI Application Management &gt; AI Applications &gt; My Subscriptions on the ModelArts management console. You can retrieve the subscription of a model that has been unsubscribed from and continue to use the model under the original quota constraints.</li> </ul>
My Data	Displays the datasets you have released and downloaded.
	• <b>Published Data</b> : View your published datasets, such as the file size and number of files. Click <b>Retry</b> or <b>Delete</b> on the right to manage published datasets.
	• <b>My Downloads</b> : View your downloaded datasets. Click the drop-down arrow to view the dataset information, including the dataset ID, download mode, and destination region.
My Workflows	Displays the workflows you have published and subscribed to.
	• <b>Published Workflows</b> : View your published workflows, such as the number of views, favorites, and subscriptions. Click <b>Release</b> , <b>Discontinue</b> , or <b>Delete</b> on the right to manage published workflows.
	<ul> <li>My Subscriptions: View the workflows you have subscribed to. Click Cancel Subscription or Retrieve Subscription on the right to manage subscribed workflows.</li> </ul>
My Information	View your information, including the account, profile photo, nickname, email, and description.
	Click Edit to edit the nickname and description.
	Click <b>Change</b> to change the profile photo.

# 10.4 Subscription & Use

# **10.4.1 Searching for and Adding an Asset to Favorites**

In AI Hub, various AI assets such as algorithms, models, datasets, and workflows are shared. To quickly search for assets, AI Gallery provides multiple quick search methods and the function of adding assets to favorites.

#### Searching for an Asset

On the asset page, use the following search methods to quickly find the assets you want:

#### Figure 10-1 Searching for an asset



#### Table 10-2 Quick search

No.	Туре	Search Mode	Supported AI Assets
1	Official assets	Click <b>Official</b> . All official assets are displayed and can be used for free.	Algorithms
2	Featured assets	Click <b>Featured</b> . All featured assets are displayed.	Algorithms, datasets, and workflows
3	Assets by category	Click <b>All categories</b> , select a category, and click <b>OK</b> .	Algorithms, datasets, and workflows
4	Assets by ranking	Click <b>Ranking</b> to change a ranking mode.	Algorithms, datasets, and workflows

#### Adding a Free Asset to Favorites

If you find a free asset that you are interested in, you can add it to **My Favorites** for quick search.

- 1. Click an asset. The asset details page is displayed.
- 2. In the upper right corner of the details page, click 2 0 to add the asset to favorites.

After the asset is added to favorites, you can quickly view it on the **My Favorites** tab page of the module the asset belongs to.

3. (Optional) To remove it from favorites, click again.

# 10.4.2 Subscribing to an Algorithm

In AI Hub, you can search for and subscribe to free algorithms that meet service requirements, and use them to create training jobs.

#### Procedure

- 1. Log in to Al Hub.
- 2. Choose **Asset Market > Algorithms**. The **Algorithm** page is displayed, showing all shared algorithms.
- 3. Search for your required algorithm. For details, see **Searching for an Asset**.
- 4. Click the target algorithm to go to the details page.
  - On the details page, you can view the information of the algorithm, including its description, restrictions, and versions.
  - Before using an algorithm, view the constraints for the target version on the Version tab page, and then prepare data and resources based on the constraints.
  - For an algorithm with open source code, you can preview or download the code on the details page.

On the **Code** tab page, click **Download** on the right to download the complete code to the local PC. Alternatively, click the file name in the list below to preview the code.

The following file types support code preview: .txt, .py, .h, .xml, .html, .c, .properties, .yml, .cmake, .sh, .css, .js, .c pp, .json, .md, .sql, .bat, and .conf.

#### Figure 10-2 Downloading or previewing code

Code	Copy Download Link	Download
Open Source Code Yes		
License Type 🕐 Unknown		
Version 3.0.0 +		
test-lxm / hard_example_mining.py		Total Size 205MB
Name		
deep_moxing-1.0.2-py3-none-any.whl     The file cannot be previewed.		
A hard_example_mining.py		
huaweicloud_sdk_python_modelarts_dataset=0.1.5-py2.py3-none-any.whl     f     The file cannot be previewed.		
moxing_tensorflow-1.17.2-py3-none-any.whl     f The file cannot be previewed.		
P resnet_v1_50		
hard_example_mining.py		
70 lines ( 58 sloc )		
1 # Copyright 2019 ModelArts Authors from Mussei Cloud. All Rights Reserved. 2 # https://www.hawweicloud.com/aroduct/modelarts.html		
4 Incensed under the Apache Lorense, Version 2.0 (the License ); 5 # you may not use this file except in compliance with the License.		
6 # You may obtain a copy of the License at		
# http://www.apache.org/licenses/LICENSE-2.0		
9 # 10 * Melana summinal bu and india las as amand to in amining software		
or on the strength ready applicance law or agrees to in writing, so there are a strength or and a strength or an a "AST" MAXIS.		
12 # WITHOUT WARRANTIES OR COMDITIONS OF ANY KIND, either express or inplied.		

- On the details page, click **Subscribe**.
  - If there are constraints on the algorithm, the **Constraints** page will be displayed. Confirm the information and click **Continue**.
  - If there are no constraints on the algorithm, the subscription is successful.

5.

After the algorithm is subscribed to, **Subscribe** on the details page is displayed as **Subscribed**. The subscribed asset is also displayed in **Management Center** > **My Algorithms** > **My Subscriptions**.

#### Using an Algorithm

1. Subscribed algorithms can be used on the ModelArts management console, for example, to create a training job.

# Method 1: Access the ModelArts management console from the algorithm details page.

- a. On the algorithm details page, click **Access Console**.
- b. In the **Select Service Region** dialog box, select the service region where ModelArts is located and click **OK**. The **Subscription** page in **Algorithm Management** of the ModelArts management console is displayed.

urce ct Detection-YOLOv5	Latest Version 2.2.2	Versions 2	Available In (Days) 1799	Ma
ct Detection-YOLOv5	222	2	1799	Ma
Reterence ID Subscribe ID Publisher	238ebe1e-1560-4d95- cf9cf24a-a5f7-4a41- ModelArts			
	Publisher	Publisher ModelArts Fixed a compatibility issue in the evaluation of	Publisher ModelArts Fixed a compatibility issue in the evaluation of the new training algorithm	Publisher ModelArts Fixed a compatibility issue in the evaluation of the new training algorithm.

Figure 10-3 My subscriptions

#### Method 2: Access the ModelArts management console from AI Hub.

- a. In Al Hub, choose **Management Center** > **My Algorithms**. The **My Algorithms** page is displayed.
- b. Click the **My Subscriptions** tab.
- c. Select the target algorithm from the list and click **ModelArts** on the right of **Application Console**.
- d. In the **Select Service Region** dialog box, select the service region where ModelArts is located and click **OK**. The **Subscription** page in **Algorithm Management** of the ModelArts management console is displayed.

#### Figure 10-4 My subscriptions

Algorithms					
My Algorithms	Subscription				
Q Subscribe to	o Algorithm lesource	Latest Version	Versions	Available In (Days)	
<u>^</u> 0	bject Detection-YOLOv5	2.2.2	2	1799	Mar
Basic Informat	Reference ID Subscribe ID Publisher	238ebe1e-1560-4d95- cf9cf24a-a5f7-4a41- ModelArts			
Versions 2.2.2	Fixed	a compatibility issue in the evaluation c	f the new training algorithr	n.	Create training job
1.0.2	Add A	lgorithm Constraints			Create training job

2. On the **Subscription** page in **Algorithm Management**, click the downward arrow on the left of the algorithm to unfold the algorithm details.

#### Canceling Subscription or Retrieving a Subscribed Algorithm

Cancel the subscription of an algorithm if it not required. After the subscription is canceled, the algorithm will not be available on the **Subscription** page in **Algorithm Management** of the ModelArts management console. To use an unsubscribed algorithm, retrieve the subscription, and the algorithm will be available on the **Subscription** page again on the ModelArts management console.

- In Al Hub, choose Management Center > My Algorithms. The My Algorithms page is displayed.
- 2. Click the **My Subscriptions** tab.
  - Cancel Subscription: This option is available only for subscribed assets.
     Click Cancel Subscription on the right of the target asset. In the dialog box that is displayed, confirm the information and click OK.
  - Retrieve Subscription: This option is available only for subscriptions that have been canceled.

Click **Retrieve Subscription** on the right of the target asset.

#### Figure 10-5 Canceling or retrieving a subscription

My Algorithr	ns		
Published Algor	ithms My Subscriptions		
All	▼ Q Enter a keyword.		
Q	Subscribed At 2022-03-21 20:02 Publisher	Console <u>ModelArts</u>	Cancel Subscription Remaining Quota 71271 days
	Subscription canceled Subscribed At 2022-03-21 19:52 Publisher	Console ModelArts	Retrieve Subscription Remaining Quota 0 day

# 10.4.3 Subscribing to a Model

In AI Hub, search for and subscribe to ModelArts models. Subscribed models can be used for deployment on ModelArts.

#### Procedure

- 1. Log in to Al Hub.
- 2. Go to the model page, which displays all shared models.
- 3. Search for a model by referring to **Searching for and Adding an Asset to Favorites**.
- Click the target model to go to the details page.
   On the details page, you can view the information of the model.
- 5. Click Subscribe.

After the model is subscribed to, **Subscribe** on the details page is displayed as **Subscribed**. The subscribed asset is also displayed in **Management Center** > **My Models** > **My Subscriptions**.

#### Using a Model

Subscribed models can be deployed or installed on ModelArts.

1. Push the subscribed model to the application console.

Method 1: Access the ModelArts management console from the model details page.

On the model details page, click **Access Console**. The **My Subscriptions** page in **AI Application Management** > **AI Applications** of the ModelArts management console is displayed.

If the status of the model version is **Ready**, the model can be used.

#### Figure 10-6 Pushing a model

y AI Applications	My Subscriptions			
Find AI Application				
Item Name	La	test Version	Versions	Available In (Days)
∧ test-model	1.	0.0	1	
Basic Information	Reference ID 59601 Subscription ID 65e Publisher	0387-eb4a-4c4c-8167-ca 2b678d-4207-4541-b31d- p_svc_elwizard_1	b0023c420d •ddf111a5bbc7	
Versions 1.0.0	Ready	Oct 09	, 2022 09:44:59 GMT+0	08:00

Method 2: Access the ModelArts management console from AI Hub.

- a. On the AI Hub page, choose **Management Center** > **My Models** in the upper right corner.
- b. Click the **My Subscriptions** tab.
- c. Select the target model from the list, click ModelArts on the right of Console. The My Subscriptions page in AI Application Management > AI Applications of the ModelArts management console is displayed.

#### Figure 10-7 Console

My Models		
My Publishes	My Subscriptions	
All	✓ Enter a keyword.	
0	test-model Subscribed Subscribed At 2022-10-09 09.47 Publisher 🛐 op_svc_elwiz Console ModelArts Remaining Quota 35635 days	Unsubscribe

If the status of the model version is **Ready**, the model can be used.

2. Use the subscribed model on the console.

On the **My Subscriptions** page in **AI Application Management > AI Applications**, click the downward arrow next to the target model name. Then, choose **Deploy > Real-time services** or **Batch Services** in the version list. For details, see **Deploying as a Real-Time Service**.

#### Canceling Subscription or Retrieving a Subscribed Model

Cancel the subscription of a model if it not required. After the subscription is canceled, the model will not be available on the **My Subscriptions** page in **AI Application Management** > **AI Applications** of the ModelArts management console. To use an unsubscribed model, retrieve the subscription, and the model will be available on the **My Subscriptions** page in **AI Application Management** > **AI Applications** again on the ModelArts management console.

- 1. On the AI Hub page, choose **Management Center** > **My Models** in the upper right corner.
- 2. Click the My Subscriptions tab.

- Cancel Subscription: This option is available only for subscribed assets.
   Click Cancel Subscription on the right of the target asset. In the dialog box that is displayed, confirm the information and click OK.
- **Retrieve Subscription**: This option is available only for subscriptions that have been canceled.

Click Retrieve Subscription on the right of the target asset.

# **10.4.4 Downloading Datasets**

You can search for and download datasets meeting your service requirements in AI Hub.

#### **Downloading Datasets**

- 1. Log in to Al Hub.
- 2. Choose **Data** to enter the data page, which displays all shared datasets.
- 3. For details about how to search for a dataset, see **Searching for and Adding an Asset to Favorites**.
- 4. Click the target dataset to go to the details page.

On the details page, view the dataset information.

- 5. Click **Download**. The parameters to be configured vary based on the method of downloading a dataset.
  - Downloading a dataset to OBS
    - Download Through: Select OBS.
    - Target Region: Select a region to which a dataset is to be downloaded.
    - Target Location: Select an OBS path. If a file or folder with the same name already exists in the bucket, it will be overwritten by the newly downloaded file or folder.

#### Figure 10-8 Parameters for downloading a dataset to OBS

	P	auto_Al <sup>i</sup>	• 1 day ago	( version 1.0.0 )	
Download Mode	OBS	ModelArts dataset			
Destination Region				•	
* Target Path	Note that if a file o	or folder with the same name ex	ists in the bucket, it will b	e overwritten by the	newly downloaded file or folde
	Select an OBS pa	th.		Ð	

- Downloading a dataset to ModelArts

- Download Through: Select ModelArts dataset.
- Target Region: Select a region to which a dataset is to be downloaded.
- Data Type: Select the type of the file to be processed.
- Output Dataset Path: OBS path where your labeled data is stored. The path cannot be the same as the file path in the OBS data source or subdirectories of the file path.
- Input Dataset Path: OBS path to which AI Hub datasets are downloaded. This path is used as the data storage path of the dataset. Ensure that output dataset path is different from the input dataset path.
- Name: A dataset name in the format of "data-xxxx" is automatically generated by default. The dataset will be synchronized to the ModelArts dataset list.
- **Description**: describes the dataset.

9	auto_Al
	el-modetarts
Download Mode	OBS ModelArts dataset
Destination Region	v
* Data Type	Images Audio Text Video Free format
	Supported formats: .jpg, .png, .jpeg, .bmp
★ Output Dataset Path	Select an OBS path.
	Path for storing output files such as labeled files. The path cannot be the same as the input path or subdirectory of the input path.
★ Input Dataset Path	Select an OBS path.
	The dataset input path cannot be the same as the output path.
* Name	dataset-2b16
Description	
	0/128

**Figure 10-9** Parameters for downloading a dataset to ModelArts

6. Click **OK**. Then, you will be redirected to **Data > My Data > My Downloads**. You can view the file size in the dataset list.

#### Using a Downloaded Dataset

1. In AI Hub, choose **Management Center > My Data**. The **My Data** page is displayed.

- 2. Click My Downloads to view all downloaded datasets.
- 3. Expand the target dataset to view its details.
  - If the dataset is downloaded to OBS, obtain the target location and import data to the dataset by performing operations described in Importing Data from an OBS Path. Then, use the dataset on ModelArts.

 Downloaded On 2022-08-30 10:59
 Publisher
 File Size 5MB
 Files 200

 Dataset ID
 0a286863-5cee-477b-adc7-ab27e979e8b2

 Download Job ID
 744501dc-23c7-4d9a-9775-816d7a930d0d

 Downloaded Through
 OBS

 Target Region
 cn-north-213

 Target Location
 /5ac491d49b7b4b8ba97{ :/ceshi/

Figure 10-10 Obtaining the target location of a dataset

 If the dataset is downloaded to ModelArts, click the target dataset to go to the dataset details page on the ModelArts management console.

#### Figure 10-11 Obtaining the target dataset

download-test Downloaded On 2022-08-30	11:21 Publisher	File Size 5MB
Dataset ID	0a286863-5cee-477b-adc7-ab	27e979e8b2
Download Job ID	dd9919cd-73af-49b8-89db-8ee	2739155771
Downloaded Through	ModelArts dataset	
Target Region	cn-north-213	
Target Location	/chang-test/models/	
Target Dataset	dataset-4956	
Description		

## 10.4.5 Subscribing to a Workflow

In AI Hub, search for and subscribe to workflows. After a subscribed workflow is imported to ModelArts through AI Hub, the workflow can be used on the ModelArts management console.

#### Procedure

- 1. Log in to Al Hub.
- 2. Switch to the workflow page, on which all shared workflows are displayed.
- 3. Search for your desired workflow. For details, see **Searching for and Adding an Asset to Favorites**.
- Click the target workflow to go to the details page.
   On the details page, you can view the information of the workflow.

#### 5. Click Subscribe.

After the workflow is subscribed to, **Subscribe** on the details page is displayed as **Subscribed**. The subscribed asset is also displayed in **Management Center** > **My Workflows** > **My Subscriptions**.

#### Using a Workflow

Subscribed workflows can be imported to and used on the ModelArts management console.

1. Import a subscribed workflow to the ModelArts management console.

# Method 1: Access the ModelArts management console from the workflow details page.

On the workflow details page, click **Run**. In the dialog box that is displayed, select the asset version, service region, and workspace of the workflow, and click **Import** to go to the workflow details page on the ModelArts management console.

Figure 10-12 Importing a workflow

Asset Name	autotest_workflow_9r9k1q
Asset Version	1.1.1
Region	•
Workspace	default 🔻
Impo	No

#### Import Workflow from AI Hub

	<		1.1.1.00 (0.000 AN 101) (TECHN			itis itis C
Index_date		18: 遠行世話 ④	0. 🖬 🗵			
Birdat         III           Birdat         Birdat           Vordensity         C         Birdat         <		e labeling_step,		- O MELER	(2) 5555	
Bit West         Bit West						
2010年         1000           Workdow,SW         第000000           Workdow,SW         第010000           Workdow,SW         第010000           B0128580         -		127-14-16				
Windswall         EIG711         000000           Windswall         EIG712         0           Windswall         -         0           Birdin         -         0           Windswall         -         0           Birdin         -         0           Birdin         -         0           Birdin         -         0           Birdin         -         0		运行总流				
Workson D         前介部の         6月の回         0           周介回路の          米名         ●素銀行           前介目の          周常世            開設町          月間でき            開設町          月間でき            開設町          月間でき            開設         He is a demo workflow		运行总流 运行状况 (	1	111		
副子(1885年		道行总章 运行状况 【 Workflow名称	un .	 2679916	00.0000	
a所改動の ··		运行总路 运行状况 和 Workflow名称 Workflow ID	99	111 道行时长 运行次数	000500 0	
Reltifi 合意人 REI 的k is a demo wolflow		<mark>通行总応</mark> 运行状況 4 WorkflowS称 Workflow ID 运行记录名称	n 	111 运行时长 运行效取 秋志	000000 0 ● 未退行	
III.5 this is a demo workflow		通行总算 通行状况 和 Workflow名称 Workflow名称 运行记录名称 运行记录口	ия «	III	00.0000 0 - 来迎行 	
		运行状况 和 运行状况 和 Workflow名称 Workflow名称 运行记录名称 运行记录口 品 被打阅	14 14 14 14 14 14 14 14 14 14 14 14 14 1	111 二百行时长 运行时秋 快志 当前节点 交肌人	05.0500 0 ● 東道行 	
		<b>运行状况</b> 适行状况 《 Workflow名称 Workflow名称 证行记录名称 运行记录名称 运行记录名称 运行记录句 周动时间 周述	19 с 	111 二 二 二 行 初数 大 志 二 行 次数 大 志 二 行 次数 大 志 二 行 次数 大 数 二 二 二 元 次数 大 数 二 二 二 元 次数 大 数 二 二 二 二 二 二 二 二 二 二 二 二 二	000560 0 ● 東助行 	
	·	<mark>通行总数</mark> 連行状況 単 Workflow活動 Workflow活動 Workflow活動 通行記念句 通行記念句 周辺打問 周辺 調送	195 C 	111 道行町長 道行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行町長 地行 地方 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た 地 た た 地 た た 地 た た 地 た た 地 た た 地 た た た た た た た た た た た た た	080500 0 ● #305 	

Figure 10-13 Workflow details page on the ModelArts management console

#### Method 2: Access the ModelArts management console from AI Hub.

- a. In Al Hub, choose **Management Center > My Workflows**. The **My Workflows** page is displayed.
- b. Click **My Subscriptions**. The workflows you have subscribed to are displayed.
- c. Select the workflow to be imported from the list and click **Workflow** next to **Application Console**.

#### Figure 10-14 Application Console

My Workflov	N	
My Publishes	My Subscriptions	
All	Enter a keyword.	
	workflow_test Subscribed Subscribed At 2022-10-09 10:22 Publisher b op_svc_ekwiz Console Workflow	Unsubscribe

d. In the dialog box that is displayed, select the asset version, service region, and workspace of the workflow, and click **Import** to go to the workflow details page on the ModelArts management console.

## Import Workflow from AI Hub

Asset Name	autotest_workflow_9r9k1q	
Asset Version	1.1.1	•
Region		•
Workspace	default	•
In	No	

2. On the ModelArts management console, use the workflow imported from AI Hub.

In the navigation pane of the ModelArts management console, choose **Workflow**. In the workflow list, locate the workflow imported from AI Hub and click **Configure** in the **Operation** column to access the workflow.

#### Canceling or Retrieving the Subscription of a Workflow

Cancel the subscription of a workflow if it not required. To use an unsubscribed workflow, click **Retrieve Subscription** to restore the canceled subscription.

# 10.5 Publish & Share

# 10.5.1 Publishing an Algorithm

In AI Hub, you can share your algorithms with others.

#### Prerequisites

- You have registered with AI Hub.
- You have created an algorithm in ModelArts Algorithm Management. For details about how to create an algorithm, see **Creating an Algorithm**.

#### **NOTE**

When creating an algorithm, ensure that the names of files and folders in the OBS bucket where the algorithm code is stored are unique. Otherwise, the algorithm may fail to be published. If the algorithm is published, the code fails to be **opened**.

#### **Publishing an Algorithm**

- 1. On the Al Hub home page, choose **Asset Market** > **Algorithms**. The **Algorithm** page is displayed.
- 2. Click **Publish**. The **Algorithm Management** > **My Algorithms** page of the ModelArts management console is displayed.
- 3. Click the target algorithm. The algorithm details page is displayed.
- 4. Click **Publish** in the upper right corner of the algorithm details page. The **Publish Asset Version** page is displayed.
- 5. Set the parameters and click **Publish**.
  - Publishing a new version
    - i. Click **Create Asset** on the right of **Item Name**. In the dialog box that is displayed, enter an asset name and description, and click **OK**.

#### **NOTE**

By default, the asset is published as a private asset. If you want to add it to the whitelist or publicize it, go to AI Hub for settings.

- ii. Enter the asset version and version description.
- iii. Click Publish.
- Updating an existing version
  - i. Select an existing asset name from the **Item Name** drop-down list box.
  - ii. Enter a new version for **Offering Version**. You can click **View Asset Version** on the right to view the historical version information.
  - iii. Enter the version description.
  - iv. Click Publish.
- 6. Go to AI Hub to view or edit the asset details.

#### **Editing Asset Details**

After an asset is published, you can modify the title, cover, and description of the asset on the details page to attract attentions.

Modifying the cover and subtitle

- 1. On the details page of the published asset, click **Edit** on the right, upload a new cover, and edit the title and subtitle.
- 2. After the editing is complete, click **Save**.

Figure 10-15 Modifying the cover and subtitle

Upload a 360 x 200 image as the asset cover. Upload Cover Image		Describe th	Describe the asset.   4 minutes ago			☐ 0 Lower Permission to Private Recommended Featured		
Description	Publish	Restrictions	Version	Paper	Code			Save Cancel

#### **Modifying categories**

- 1. Click  $\swarrow$  on the right of **Category**. In the displayed edit box, select categories from the drop-down list box.
- 2. Click the check mark on the right of the edit box.

The saved categories will be used as filter criteria on the asset search page.

#### Figure 10-16 Adding categories

Description	Publish	Restrictions	Version	Paper	Code			
Category	Deployi Country	ment Environment * //Region *	Industry 🏽 G	Soverment 🏽 S	icenario 🕷	^	~ ×	
Assets Id	Alg	go and Model >	Deploy	yment Environ	ment >	Sn	nart Home	>
			🗌 Engine	9	>	Ro	bot	>
Description	r i		🗹 Indust	ry	>	Ci	ty Area	>
			Specia	I Subject	>	H	uman Identity	>
			MoreT	ag	>	🗌 In	telligent Transportation	>
			Scenar	rio	>	lo	r	>
			Count	ry/Region	>	Re	mote Sensing	>
			Model	Arts	>	Re	mote Sensing	>
			D Poord	o Din	× .	Fi	ahtina	>

#### Editing the description

- 1. Click **Edit** on the right and enter the asset description in the text box, including but not limited to the background, introduction, and usage method. You can edit content in Markdown mode.
- 2. After the editing is complete, click **Save**.

#### **Editing restrictions**

You can modify the access policy and maximum duration of your published assets.

- 1. Click the **Restrictions** tab and click **Edit** in the upper right corner.
  - Select the access policy from the drop-down list box on the right of **Visible To**.
    - Public: indicates that all users who use AI Hub can view and use this asset.
    - Specific users: indicates that only specified users can view and use this asset.
    - Just me: indicates that only the current account can view and use this asset.

#### **NOTE**

Only permissions higher than the one set when you published your asset are allowed. Therefore, if a public algorithm is created, **Visible To** cannot be modified.

- Maximum Duration can be disabled or enabled. When this function is enabled, set the duration of the asset and whether to renew the subscription after the asset expires.
- 2. Click Save.

#### Figure 10-17 Editing restrictions

Restrictions								
Visible To 🕜	Public							
Billing Mode	Free							
Maximum Duration	Maximum Duration	Free Duration	Renew					
	Enable 👻	1 year 👻	✓ Auto-renew					

#### **Editing versions**

- 1. Click the Version tab and click Edit in the upper right corner.
- 2. On this page, modify the version description or click **Discontinue** in the **Operation** column of the target version to discontinue it. You can only discontinue a released asset that has two or more versions.
- Click Add Version on the right of the Version area to go to the Algorithm Management > My Algorithms page of the ModelArts management console. Publish a new version of the target algorithm by referring to Publishing an Algorithm.
- 4. After the editing is complete, click **Save** in the upper right corner.

#### Figure 10-18 Editing versions

Description	Publish	Restrictions	Version	Paper	Code			Save Cancel
Version								Add Version
Version		Published At		Status		Description	Constraints	Operation
1.0.0		2022-03-08 16:15		Normal		Initial release.	View	Discontinue d

#### Editing the paper

- 1. On the **Paper** tab, click **Edit** in the upper right corner, and enter the paper name and URL in the text box. You can click the URL to view the paper details.
- 2. After the modification is complete, click **Save**.

#### **Editing code**

1. On the **Code** tab, click **Edit** in the upper right corner to determine whether to open the code.

#### **NOTE**

If you discontinue an algorithm whose code is not open, the subscribers can continue using the algorithm before the subscription expires. After the algorithm is released again with its code open, the algorithm that has been discontinued is not displayed on the home page. You can choose **Management Center** > **My Algorithms** > **My Subscriptions** and click the algorithm name to preview the code.

2. If the code is open, set the license type.

You can click the exclamation mark (!) next to a license type to view license details.

3. After the modification is complete, click **Save**.

#### Figure 10-19 Editing the code

Description	Publish	Restrictions	Version	Paper	Code	Save Cancel
Paper						
Title						
Link						
Code						
Open Source Co	ode No	Ves				
License Type 🕜	Unkno	wn			-	

#### **Discontinuing an Algorithm**

To discontinue a shared asset from AI Hub, perform the following operations:

- In Al Hub, choose Management Center > My Algorithms. The My Algorithms page is displayed.
- On the My Algorithms > Published Algorithms page, click Discontinue on the right of the target asset. In the dialog box that is displayed, confirm the asset information and click OK.

**NOTE** 

After an asset is discontinued, users who have subscribed to the asset can continue using it within the duration, but other users cannot view or subscribe to it.

#### Figure 10-20 Discontinuing an asset

0	Release Published At 2022-03-01 10:34	Views 10	Favorites 0	Subscriptions 2	Discontinue
0	Discontinue Published At 2022-03-01 10:13	Views 2	Favorites 0	Subscriptions 1	Release

After the asset is discontinued, **Discontinue** in the **Operation** column changes to **Release**. You can click **Release** to share a discontinued asset to AI Hub.

## 10.5.2 Publishing a Model

In AI Hub, you can share your models developed on ModelArts with others.
### Prerequisites

• You have created a model in ModelArts AI Application Management. For details, see AI Application Management. Both models imported from container images and trained models can be published to AI Hub.

### Procedure

- 1. On the AI Hub home page, go to the model page.
- 2. Click **Publish**. On the displayed page, enter information.
  - Figure 10-21 shows parameters for creating an asset.
    - i. Set **Publish Mode** to **Create Asset**.
    - ii. Enter an item title which is displayed in AI Hub.
    - iii. Source defaults to ModelArts.
    - iv. Select a ModelArts region for using this asset.
    - v. Select an AI application from ModelArts AI Application Management.
       Both models imported from container images and trained models can be published to AI Hub.
    - vi. Enter the offering version in the *x.x.x* format.
    - vii. Configure Visible To.

Options:

**Public**: indicates that all users who use AI Hub can view and use this asset.

**Specific users**: indicates that only specified users can view and use this asset.

**Only to me**: indicates that only the current account can view and use this asset.

viii. Configure Duration Limit.

This function is disabled by default, that is, subscribers can use the asset without duration limit. When this function is enabled, set the duration limit of the asset and whether to renew the subscription after the asset expires.

- ix. Click Publish.
- Figure 10-22 shows parameters for adding an asset version.
  - i. Set Publish Mode to Add asset version.
  - ii. Select an existing asset name from the **Item Title** drop-down list box. You can search for asset names.
  - iii. Select a ModelArts region for using this asset.
  - iv. Select an AI application from ModelArts AI Application Management.
    - Both models imported from container images and trained models can be published to AI Hub.
  - v. Enter a new version number in the **Offering Version** text box.
  - vi. Click **Publish**.

Publish Mode	Create Asset	Add asset version		
Item Title	Enter an item title that	is easy to identify		
item nue		is easy to identify.		
	Ensure that you do not u	ise sensitive words like polit	ics, pornography, or advertisement.	4
Source	ModelArts			
ModelArts Region	cn-north-213			
AI Application Name	Select AI Application	l.		
Offering Version	100			
Offering version	1.0.0			
Visible To	Public			
Duration Limit				
		Publish	No	

#### Figure 10-21 Create Asset

### Figure 10-22 Add asset version

Publish Mode	Create Asset	Add asset version		
Item Title	Select			v
ModelArts Region	cn-north-213			•
AI Application Name	Select AI Applicatio	on.		Select
Offering Version	Enter the version.			
		Publish	No	

3. View the asset details page.

### **Editing Asset Details**

After an asset is published, you can modify the title, cover, and description of the asset on the details page to attract attentions.

#### Modifying the cover and subtitle

- 1. On the details page of the published asset, click **Edit** on the right, upload a new cover, and edit the title and subtitle.
- 2. After the editing is complete, click **Save**.

Figure 10-23 Modifying the cover and subtitle

Upload a 360 x asse Upload C	200 image as the t cover.			🖻 495KB 🕞 3 💭 1	Lower Permission to Private Recommended
Description	Publish	Restrictions	Version		Save

### Selecting categories

- 1. Click  $\checkmark$  on the right of **Category**. In the displayed edit box, select categories from the drop-down list box.
- 2. Click the check mark on the right of the edit box.

The saved categories will be used as filter criteria on the asset search page.

### Figure 10-24 Selecting categories

	Text classification 8 Named Entity Re	cognitio	n 8 CPU Inference 8		
Category	Public Cloud 🗴 TensorFlow 🕺			√ X	
	🗌 Data Type	$\rangle$	Computer Vision	$\rangle$	Text classification
Descriptio	Technical Branch	>	Natural Language Processing	>	Named Entity Recognitio
Descriptio	ResourceType	>	Speech Recognition	$\rangle$	
	Deployment Environment	>	Knowledge Graph		
BIS	Engine	>	Search Recommendation		
1	Industry	$\rangle$	Time Series Prediction		

### Editing the description

- Click Edit on the right and enter the asset description in the text box, including but not limited to the background, introduction, and usage method. You can edit content in Markdown mode.
- 2. After the editing is complete, click **Save**.

#### **Editing restrictions**

You can modify the access policy and duration limit of your published assets.

1. Click the **Restrictions** tab and click **Edit** in the upper right corner.

- Select the access policy from the drop-down list box on the right of Visible To.
  - Public: indicates that all users who use AI Hub can view and use this asset.
  - **Specific users**: indicates that only specified users can view and use this asset.
  - Just me: indicates that only the current account can view and use this asset.

#### **NOTE**

- Only permissions higher than the one set when you published your asset are allowed. Therefore, if a public algorithm or model is created, **Visible To** cannot be modified.
- You can manually enable **Duration Limit**. When this function is enabled, set the duration limit of the asset and whether to renew the subscription after the asset expires.
- 2. Click Save.

#### Figure 10-25 Editing restrictions

#### Restrictions

Visible To?	Public			
Maximum	Maximum Duration		Duration	Renew
Duration	Disable	•		

#### **Editing versions**

- 1. Click the Version tab and click Edit in the upper right corner.
- 2. On this page, modify the version description or click **Discontinue** in the **Operation** column of the target version to discontinue it. You can only discontinue a released asset that has two or more versions.
- Click Add Version on the right of the Version area. Add an asset version in Publish in AI Hub by referring to parameters for adding an asset version.
- 4. After the editing is complete, click **Save** in the upper right corner.

#### Figure 10-26 Editing versions

Version				Add Version
Version	Published At	Status	Description	Operation
1.0.0	2022-10-11 10:21	Done	Init/2	Discontinue

### **Discontinuing a Model**

To discontinue a shared asset from AI Hub, perform the following operations:

- 1. On the AI Hub page, choose **Management Center** > **My Models**.
- On the My Models > My Publishes page, click Discontinue on the right of the target asset. In the dialog box that is displayed, confirm the asset information and click OK.

### **NOTE**

After an asset is discontinued, users who have subscribed to the asset can continue using it within the duration, but other users cannot view or subscribe to it.

Figure 10-27 Discontinuing an asset

6	model-test Release					
9	Published At 2022-10-11 10:21	Views 4	Favorites 0	Subscriptions 2	Discontinue	

After the asset is discontinued, **Discontinue** in the **Operation** column changes to **Release**. You can click **Release** to share a discontinued asset to AI Hub.

### 10.5.3 Publishing Data

In Al Hub, you can share your datasets with others.

### Prerequisites

- You have registered with AI Hub.
- A dataset is available in ModelArts dataset list or OBS. For details about how to create or upload a dataset, see **Creating a Dataset**.

### **Publishing a Dataset**

- 1. Go to the **Datasets** page on the Al Hub home page.
- 2. Click **Publish**. On the displayed page, enter information.
  - If you publish a ModelArts dataset, configure parameters by referring to Table 10-3.

Parameter	Description
Category	Select <b>data</b> .
Item Title	Asset name displayed in AI Hub
Source	Select <b>ModelArts</b> . A dataset supports a maximum of 20,000 files, and the total size cannot exceed 30 GB.
ModelArts Region	Select the region where the dataset is located.
Select Data	Select the target dataset in the current region from the drop-down list box.
	Only image datasets of image classification or object detection type and datasets in free format can be selected.

Table 10-3 Parameters for publishing a ModelArts dataset

Parameter	Description
Version	Select the version you want to publish for the target dataset.
Data Type	Select at least one data type label. Options: <b>Image, Audio, Video, Text, Table</b> , and <b>Other</b>
License Type	Select a proper license type based on the service requirements and dataset type. Click next to a license type to view the license details.
Visible To	<ul> <li>Set the access policy for the dataset. Available options:</li> <li>Public: indicates that all users who use Al Hub can view and use this asset.</li> <li>Specific users: indicates that only specified users can view and use this asset.</li> <li>Only to me: indicates that only the current account can view and use this asset.</li> </ul>

 If you publish a dataset in OBS, configure parameters by referring to Table 10-4.

Parameter	Description
Category	<b>data</b> is used by default.
Item Title	Asset name displayed in AI Hub
Source	Select <b>OBS</b> . A dataset supports a maximum of 20,000 files, and the total size cannot exceed 30 GB.
OBS Region	Select the storage region of the OBS bucket where the data is stored.
Storage Path	Select the OBS path of the dataset you want to publish.
Data Type	Select at least one data type label. Options: Image, Audio, Video, Text, Table, and Other

Table 10-4 Parameters for	publishing an	<b>OBS</b> dataset
---------------------------	---------------	--------------------

Parameter	Description
License Type	Select a proper license type based on the service requirements and dataset type.
	Click <sup>9</sup> next to a license type to view the license details.
Visible To	Set the access policy for the dataset. Available options:
	• <b>Public</b> : indicates that all users who use AI Hub can view and use this asset.
	• <b>Specific users</b> : indicates that only specified users can view and use this asset.
	• <b>Only to me</b> : indicates that only the current account can view and use this asset.

3. Click **Publish**. The dataset details page is displayed.

### **Editing Asset Details**

After a dataset is published, you can modify the dataset information on the details page.

#### Modifying the cover and subtitle

- 1. On the details page of the published asset, click **Edit** on the right, upload a new cover, and edit the title and subtitle.
- 2. After the editing is complete, click **Save**.

Figure 10-28 Modifying the cover and subtitle

Upload a 360 x 200 image as the asset cover.	Describe the accet	↓ 0 Lower Permission to Private
Upload Cover Image	4 minutes ago 😑 175MB ⊕ 43 🛓 5	Recommended Featured
Description Publish	Restrictions Version Paper Code	Save Cancel

#### Editing the license type

- 1. On the details page of the published asset, click **Edit** on the right.
- 2. Select the license you want to update from the drop-down list on the right of **License Type**, and then click **Save**.

You can click the exclamation mark (!) next to a license type to view license details.

### Figure 10-29 Editing the license type

Description Version Restrictions

Save Cancel

### Selecting categories

- 1. Click  $\swarrow$  on the right of **Category**. The edit box is displayed.
- 2. Select categories for the asset from the drop-down list box and click the check mark on the right of the edit box.

Other users can search for your assets by category.

### Figure 10-30 Selecting categories

Category	Select			^	✓ ×
	DataType	>	Image		
			Audio		
			Video		
			Text		
-			Table		
_			Other		

### Editing the description

- 1. Click **Edit** on the right and enter the asset description in the text box, including but not limited to the background, introduction, and usage method. You can edit content in Markdown mode.
- 2. After the editing is complete, click **Save**.

#### **Editing versions**

- 1. Click the **Version** tab and click **Edit** in the upper right corner.
- 2. Click  $\checkmark$  in the **Description** column, add version description, and click  $\checkmark$ . The dataset version description is used to distinguish the dataset from other ones.

### Editing restrictions

- 1. Click the **Restrictions** tab and click **Edit** in the upper right corner.
- 2. Select the access policy from the drop-down list box on the right of **Visible To**.
  - **Public**: indicates that all users who use AI Hub can view and use this asset.
  - Specific users: indicates that only specified users can view and use this asset.
  - Just me: indicates that only the current account can view and use this asset.
- 3. Click Save.

### Figure 10-31 Editing restrictions

	Public
Restrictio	Specific users
	Just me
Visible To	Public 🔺

### **Republishing a Dataset**

If a dataset failed to be published, you can publish it again.

- 1. In AI Hub, choose **Management Center > My Data**. The **My Data** page is displayed.
- 2. On the **My Publishes** tab page, view the datasets that failed to be published.

Figure 10-32 Viewing datasets that failed to be published

My Data			
My Publishes	My Downloads		
All	Q Enter a keyword.		Publish
<b>S</b>	Created On 2022-07-20 17:29 File Size 0KB	Files 0	Delete

3. Modify the source data based on the error message and click **Retry** on the right of the target dataset to publish the dataset again.

### **Deleting a Published Dataset**

To delete a dataset published in AI Hub, perform the following steps:

- In AI Hub, choose Management Center > My Data. The My Data page is displayed.
- 2. On the **My Publishes** tab page, click **Delete** on the right of the target dataset. In the displayed dialog box, confirm the deletion.

### **NOTE**

Because datasets are downloaded to OBS for use, deleting published datasets has no impact on their users.

# **11** Custom Images

### 11.1 Image Management

### Overview

During the development and runtime of AI services, complex environment dependencies need to be debugged for containerization. In the best practices of AI development in ModelArts, container images are used to provide fixed runtime environments. In this way, dependencies can be managed and the runtime environments can be easily switched. The container resources provided by ModelArts enable quick and efficient AI development and model experiment iteration.

The preset images provided by ModelArts by default have the following features:

- Out-of-the-box and scenario-specific: Typical dependent environments for AI development are preset in these images to provide optimal software, OS, and network configurations. They have been fully tested on hardware to ensure optimal compatibility and performance.
- Configuration customizable: Preset images are stored in the SWR repository for you to customize and register them as your own images.
- Secure and reliable: Access policies, user permissions control, vulnerability scanning for development software, and OS are configured based on best practices for security hardening to ensure the security of images.

If you have special requirements on the deep learning engine and development library, you can use ModelArts custom images to customize runtime engines.

Based on the container technology, you can customize container images and run them on ModelArts. Custom images support CLI parameters and environment variables in free text format, featuring high flexibility for a wide range of compute engines.

### **Application Scenarios of Preset Images**

ModelArts provides a group of preset images. You can use a preset image to create a notebook instance. After installing and configuring dependencies on the

instance, create a custom image. Then, you can directly use the image in ModelArts for training jobs without any adaptation. You can also use preset images to submit training jobs and create AI applications.

We recommend the preset image version based on your development requirements and stability of the version. If your development can be carried out using versions preset in ModelArts, for example, MindSpore 1.*X*, use the preset images. They have been fully verified and have many commonly-used installation packages, relieving you from configuring the environment.

### **Application Scenarios of Custom Images**

### • Using custom images on notebook instances

If the preset images of notebook instances cannot meet requirements, you can create a custom image by installing and configuring the software and other data required by the environment in a preset image. Then, use the custom image to create new notebook instances.

### • Using a custom image to create training jobs

If you have developed a model or training script locally but the AI engine you used is not supported by ModelArts, create a custom image and upload it to SWR. Then, use this image to create a training job on ModelArts and use the resources provided by ModelArts to train models.

### • Using a custom image to create AI applications

If you have developed a model using an AI engine that is not supported by ModelArts, to use this model to create AI applications, do as follows: Create a custom image, import the image to ModelArts, and use it to create AI applications. The AI applications created in this way can be centrally managed and deployed as services.

### **NOTE**

The rules for creating a custom image vary according to the application scenario. The details are as follows:

- General rule: SWR images can be shared with others only when the image type is **Private**. This rule applies to development environments, training jobs, and AI applications.
- Development environment: Other users can register and use SWR images on the ModelArts image management page only when the SWR image type is **Public**.
- Training job: To create a training job using a **Public** SWR image, enter the **organization or the image name, and the corresponding version name** in the **Image** text box, for example, you can enter **test-image/tensorflow2\_1\_1:1.1.1** in the **Image** text box to use a public image whose address is **xxx.com/test-image/tensorflow2\_1\_1:1.1.1**.

### **Custom Image Services**

When you use a custom image, the following services may be involved:

• SWR

Software Repository for Container (SWR) provides easy, secure, and reliable management over container images throughout their lifecycle, facilitating the deployment of containerized applications. You can upload, download, and manage container images through the SWR console, SWR APIs, or community CLI.

Your custom images must be uploaded to SWR. The custom images used by ModelArts for training or creating AI applications are obtained from the SWR service management list.





• OBS

Object Storage Service (OBS) is a cloud storage service optimized for storing massive amounts of data. It provides unlimited, secure, and highly reliable storage capabilities at a relatively low cost.

ModelArts exchanges data with OBS. You can store data in OBS.

• ECS

An Elastic Cloud Server (ECS) is a basic computing unit that consists of vCPUs, memory, OS, and Elastic Volume Service (EVS) disks. After an ECS is created, you can use it similarly to how you would use your local PC or physical server.

You can create a custom image on premises or on an ECS.

D NOTE

When you use a custom image, ModelArts may need to access dependent services, such as SWR and OBS. The custom image can be used only after the access is authorized. It is a good practice to use an agency for authorization. After the agency is configured, the permissions to access dependent services are delegated to ModelArts so that ModelArts can use the dependent services and perform operations on resources on your behalf. For details, see **Configuring Access Authorization (Global Configuration)**.

## 11.2 Introduction to Preset Images (Mainstream Images)

### 11.2.1 Preset Images

### Preset Images of the Arm + Ascend Architecture

Preset Image	Applicable Scope
mindspore_2.0.0-cann_6.3.0-py_3.7- euler_2.8.3	Notebook, training, and inference deployment
mindspore1.8.0-cann5.1.2-py3.7- euler2.8.3	Notebook

### Table 11-1 MindSpore

Preset Image	Applicable Scope
mindspore1.7.0-cann5.1.0-py3.7- euler2.8.3	Notebook

### Table 11-2 TensorFlow

Preset Image	Applicable Scope
tensorflow_1.15.0-cann_6.3.0-py_3.7- euler_2.8.3	Notebook, training, and inference deployment
tensorflow1.15.0-cann5.1.2-py3.7- euler2.8.3	Notebook
tensorflow1.15-cann5.1.0-py3.7- euler2.8.3	Notebook

### Table 11-3 PyTorch

Preset Image	Applicable Scope	
pytorch_1.11.0-cann_6.3.0-py_3.7- euler_2.8.3	Notebook, training, and inference deployment	

### 11.2.2 Preset MindSpore Images on Arm

### Image 1: mindspore\_2.2.0-cann\_7.0.1-py\_3.9-euler\_2.10.7-aarch64-snt9b

Al Engin e	URL	Dependency	
minds	swr. <region>.myhuaweicloud.co</region>	PyPI package	YUM package
pore 2.2.0 + minds pore- lite 2.2.0 + Ascend CANN Toolkit 7.0.RC 1	m/atelier/ mindspore_2_2_ascend:mindspor e_2.2.0-cann_7.0.1-py_3.9- euler_2.10.7-aarch64- snt9b-20231107190844-50a1a83	mindspore 2.2.0 ipykernel 6.7.0 ipython 8.17.2 jupyter-client 7.4.9 ma-cau 1.1.7 ma-cau-adapter 1.1.3 ma-cli 1.2.3 matplotlib 3.5.1 modelarts 1.4.20 moxing-framework 2.2.3.2c7f2141 numpy 1.22.0 pandas 1.2.5 pillow 10.0.1 pip 21.0.1 psutil 5.9.5 PyYAML 6.0.1 scipy 1.10.1 scikit-learn 1.0.2 tornado 6.3.3 mindinsight 2.2.0	cmake cpp curl ffmpeg g++ gcc git grep python3 rpm tar unzip wget zip

### Image 2: mindspore\_2.1.0-cann\_6.3.2-py\_3.7-euler\_2.10.7-aarch64-snt9b

Al Engin e	URL	Dependency	
minds pore 2.1.0 + minds pore- lite 2.1.0 + Ascend CANN Toolkit 6.3.RC 2	swr.{Region ID}.{Site domain name}/atelier/ mindspore_2_0_ascend:mindspo re_2.1.0-cann_6.3.2-py_3.7- euler_2.10.7-aarch64- snt9b-20231009152946- e7b7e70	PyPI package mindspore 2.1.0 ipykernel 6.7.0 ipython 7.34.0 jupyter-client 7.4.9 ma-cau 1.1.6 ma-cau-adapter 1.1.3 ma-cli 1.2.2 matplotlib 3.5.1 modelarts 1.4.20 moxing-framework 2.2.3.2c7f2141 numpy 1.21.6 pandas 1.3.5 pillow 9.5.0 pip 21.0.1 psutil 5.9.5 PyYAML 6.0.1 scipy 1.7.3 scikit-learn 1.0.2 tornado 6.2 mindinsight 2.1.0	YUM package cmake cpp curl ffmpeg g++ gcc git grep python3 rpm tar unzip wget zip

 Table 11-5 Introduction to mindspore\_2.1.0-cann\_6.3.2-py\_3.7-euler\_2.10.7-aarch64-snt9b images

### Image 3: mindspore1.8.0-cann5.1.2-py3.7-euler2.8.3

Table 11-6	Information	about the	image
------------	-------------	-----------	-------

Al Engi ne	Whethe r to Use Ascend (CANN Version)	URL	Dependency	
mind spore 1.8.0	(CANN Version) Yes cann5.1. 2	swr.{Region ID}.{Region domain name}./atelier/ mindspore_1_8_ascend:mind spore_1.8.0-cann_5.1.2- py_3.7-euler_2.8.3-aarch64- d910-20221009094203-402 5e09	PyPI package mindspore- ascend 1.8.0 mindinsight 1.8.0 ipykernel 6.7.0 ipython 7.34.0 jupyter-client 7.3.4 ma-cau 1.1.3 ma-cau-adapter 1.1.3 ma-cli 1.2.2 matplotlib 3.5.1 modelarts 1.4.18 moxing- framework 2.0.1.rc0.ffd1c0c 8 numpy 1.21.2 pandas 1.1.3 pillow 9.2.0	YUM package ca- certificates. noarch cmake cpp curl gcc-c++ gcc gdb grep nginx python3 rpm tar unzip vim vim wget zip
			pip 22.1.2 psutil 5.7.0 PyYAML 5.3.1 scipy 1.5.4 scikit-learn 0.24.0 tornado 6.2	

### Image 4: mindspore1.7.0-cann5.1.0-py3.7-euler2.8.3

Table	11-7	Information	about the	image
-------	------	-------------	-----------	-------

Al Engi ne	Whethe r to Use Ascend (CANN Version)	URL	Dependency	
Mind Spore 1.7.0	Yes (CANN 5.1)	swr.{Region ID}.{Region domain name}./atelier/ mindspore_1_7_0:mindspore _1.7.0-cann_5.1.0-py_3.7- euler_2.8.3-aarch64- d910-20220906 For example: swr.cn- central-231.xckpjs.com/ atelier/ mindspore_1_7_0:mindspore _1.7.0-cann_5.1.0-py_3.7- euler_2.8.3-aarch64- d910-20220906 swr.cn- southwest-228.cdzs.cn/ atelier/ mindspore_1_7_0:mindspore _1.7.0-cann_5.1.0-py_3.7- euler_2.8.3-aarch64- d910-20220906	PyPI package mindspore- ascend 1.7.0 mindinsight 1.7.0 ipykernel 5.3.4 ipython 7.34.0 jupyter-client 7.3.4 ma-cau 1.1.2 ma-cau-adapter 1.1.2 ma-cli 1.1.3 matplotlib 3.5.1 modelarts 1.4.7 moxing- framework 2.0.1.rc0.ffd1c0c 8 numpy 1.21.2 pandas 1.1.3 pillow 9.2.0 pip 22.1.2 psutil 5.7.0 PyYAML 5.3.1 scipy 1.5.4 scikit-learn 0.24.0 tensorboard 1.15.0	YUM package ca- certificates. noarch cmake cpp curl gcc-c++ gcc gdb grep nginx python3 rpm tar unzip vim wget zip
			tornado 6.2	

### Image 5: mindspore\_2.2.10-cann\_8.0.rc1-py\_3.9-hce\_2.0.2312-aarch64-snt9c

Al URL Engin e		Dependency	
minds swr.<{Region pore name }/ateli 2.2.10 mindspore_2 + re_2.2.10-car minds hce_2.0.2312 pore- snt9c-202403 lite d5e7cea 2.2.10 + Ascend CANN Toolkit 8.0.rc1	n ID}.{Site domain er/ 2_2_ascend:mindspo nn_8.0.rc1-py_3.9- 2-aarch64- 301174404-	PyPI package ipykernel 6.7.0 ipython 8.18.1 jupyter-client 7.4.9 ma-cau 1.1.7 ma-cau-adapter 1.1.3 ma-cli 1.2.3 matplotlib 3.5.1 modelarts 1.4.20 moxing-framework 2.2.3.2c7f2141 numpy 1.22.0 pandas 1.3.5 pillow 10.0.1 pip 21.0.1 psutil 5.9.5 PyYAML 6.0.1 scipy 1.10.1 scikit-learn 1.0.2 tornado 6.4	YUM package cmake cpp curl ffmpeg g++ gcc git grep python3 rpm tar unzip wget zip

 Table 11-8 Introduction to mindspore\_2.2.10-cann\_8.0.rc1-py\_3.9-hce\_2.0.2312-aarch64-snt9c images

### 11.2.3 Preset TensorFlow Images on Arm

### Image 1: tensorflow\_1.15.0-cann\_6.3.0-py\_3.7-euler\_2.8.3

Al Engi ne	Whethe r to Use Ascend (CANN Version)	URL	Dependency	
tenso rflow _1.15. 0	Yes cann6.3. 0	<pre>swr.{Region ID}.{Region domain name}./atelier/ tensorflow_1_15_ascend:ten sorflow_1.15.0-cann_6.3.0- py_3.7-euler_2.8.3-aarch64- d910-20230425164623-343 00db</pre>	PyPI package tensorflow 1.15.0 tensorboard 1.15.0 ipykernel 6.7.0 ipython 7.34.0 jupyter-client 7.4.9 ma-cau 1.1.3 ma-cau-adapter 1.1.3 ma-cli 1.2.2 matplotlib 3.5.1 modelarts 1.4.18 moxing- framework 2.0.1.rc0.ffd1c0c 8 numpy 1.17.5 pandas 0.24.2 pillow 9.5.0 pip 21.0.1 psutil 5.7.0 PyYAML 5.3.1 scipy 1.3.3 scikit-learn 0.20.0 tornado 6.2	YUM package

Table 11-9 Information abo	ut the image
----------------------------	--------------

### Image 2: tensorflow1.15-cann5.1.0-py3.7-euler2.8.3

Al Engi ne	Whethe r to Use Ascend (CANN Version)	URL	Dependency	
Tenso rFlow 1.15	Yes (CANN 5.1)	<pre>swr.{Region ID}.{Region domain name}./atelier/ tensorflow_1_15_ascend:ten sorflow_1.15-cann_5.1.0- py_3.7-euler_2.8.3-aarch64- d910-20220906 For example: swr.cn- central-231.xckpjs.com/ atelier/ tensorflow_1_15_ascend:ten sorflow_1.15-cann_5.1.0- py_3.7-euler_2.8.3-aarch64- d910-20220906 swr.cn- southwest-228.cdzs.cn/ atelier/ tensorflow_1_15_ascend:ten sorflow_1.15-cann_5.1.0- py_3.7-euler_2.8.3-aarch64- d910-20220906</pre>	PyPI package tensorflow 1.15.0 tensorboard 1.15.0 ipykernel 5.3.4 ipython 7.34.0 jupyter-client 7.3.4 ma-cau 1.1.2 ma-cau-adapter 1.1.2 ma-cli 1.1.3 matplotlib 3.5.1 modelarts 1.4.7 moxing- framework 2.0.1.rc0.ffd1c0c 8 numpy 1.17.5 pandas 0.24.2 pillow 9.2.0 pip 22.1.2 psutil 5.7.0 PyYAML 5.3.1 scipy 1.3.3 scikit-learn 0.20.0 tornado 6.2	YUM package ca- certificates. noarch cmake cpp curl gcc-c++ gcc gdb grep nginx python3 rpm tar unzip vim wget zip

Table 11-10 Information about the image

### Image 3: tensorflow1.15.0-cann5.1.2-py3.7-euler2.8.3

Al Engi ne	Whethe r to Use Ascend (CANN Version)	URL	Dependency	
tenso rflow 1.15. 0	Yes cann5.1. 2	swr.{Region ID}.{Region domain name}./atelier/ tensorflow_1_15_ascend:ten sorflow_1.15.0-cann_5.1.2- py_3.7-euler_2.8.3-aarch64- d910-20221009094203-402 5e09	PyPI package tensorflow 1.15.0 tensorboard 1.15.0 ipykernel 6.7.0 ipython 7.34.0 jupyter-client 7.3.4 ma-cau 1.1.3 ma-cau adapter 1.1.3 ma-cli 1.2.2 matplotlib 3.5.1 modelarts 1.4.18 moxing- framework 2.0.1.rc0.ffd1c0c 8 numpy 1.17.5 pandas 0.24.2 pillow 9.2.0 pip 22.1.2 psutil 5.7.0 PyYAML 5.3.1 scipy 1.7.3 scikit-learn 0.20.0	YUM package ca- certificates. noarch cmake cpp curl gcc-c++ gcc gdb grep nginx python3 rpm tar unzip vim wget zip

### Table 11-11 Information about the image

### 11.2.4 Preset PyTorch Images on Arm

### Image 1: pytorch\_1.11.0-cann\_6.3.0-py\_3.7-euler\_2.8.3

Al Engi ne	Whethe r to Use Ascend (CANN Version)	URL	Dependency	
pytor ch_1. 11.0	Yes cann_6.3 .0	swr.{Region ID}.{Region domain name}./atelier/ pytorch_1_11_ascend:pytorc h_1.11.0-cann_6.3.0-py_3.7- euler_2.8.3-aarch64- d910-20230425164623-343 00db	PyPI package torch 1.11.0 torch-npu 1.11.0.dev20230 417 ipykernel 6.7.0 ipython 7.34.0 jupyter-client 7.4.9 ma-cau 1.1.3 ma-cau-adapter 1.1.3 ma-cli 1.2.2 matplotlib 3.5.1 modelarts 1.4.18 moxing- framework 2.0.1.rc0.ffd1c0c 8 numpy 1.21.2 pandas 0.24.2 pillow 9.5.0 pip 21.0.1 psutil 5.7.0 PyYAML 5.3.1 scipy 1.3.3 scikit-learn 0.20.0 tornado 6.2	YUM package ca- certificates. noarch cmake cpp curl gcc-c++ gcc gdb grep nginx python3 rpm tar unzip vim wget zip

Table 11-12 Information about the image

### **11.3 Using Custom Images in Notebook Instances**

### 11.3.1 Registering an Image in ModelArts

After a custom image is created, register it on the ModelArts **Image Management** page before using it in notebook.

### **NOTE**

Only the sub-users (IAM users) of the account can register and use the SWR images if the image type is **Private**.

Other users can register and use SWR images only when the image type is **Public**.

- 1. Log in to the ModelArts management console and choose Image Management. Then, click Register.
- 2. Configure parameters and click **Register**.
  - SWR Source: Select a built image as the image source. You can copy the
    - complete SWR address or click **D** to select the target image for registration.
  - Architecture and Type: Configure them based on the actual framework of the custom image.
- 3. View the registered image on the **Image Management** page.

#### Figure 11-2 Image list

Image Management			D PyCharm Toolkit	M ModelArts SDK Register
$\overline{\boldsymbol{\mathbb{V}}}$ Search or filter by keyword.				Q C 🕲
Name 0	Organization 0	Total Versions 单	Updated 0	Operation
msite-2.0.0-cann-6.3.1	dev-custom	1	Jul 28, 2023 14:25:59 GMT+08:00	Details
tensorflow2_1_test1		1	Sep 26, 2023 17:15:34 GMT+08:00	Details

### **Creating a Notebook Instance**

Click the image name. On the image details page that appears, click **Create Notebook**. The page for creating a notebook instance using this image is displayed.

Figure 11-3 Image details page

< pytorch_1_8							
Image Version/ID	Status	Resource Type	Size	SWR Address	Description	Created At $\protect$	Operation
v1	Available	CPU,GPU	2.15 G8	SWT.Ch-	-	Sep 07, 2023 17:21:39 GMT+08:00	Create Notebook Sync   Delete

### Synchronizing an Image

After the image fault is rectified, go to the image details page. Click **Sync** in the **Operation** column to refresh the image status.

### 11.3.2 Creating a Custom Image

To create a custom image, perform the following steps:

- Method 1: Use a preset image of notebook instances to create a development environment instance. Then, install and configure dependencies in the environment. After the configuration, use the image saving function provided by the development environment to save the running instance as a custom container image. For details, see Saving a Notebook Instance as a Custom Image.
- Method 2: Use ModelArts base images or third-party images to write a Dockerfile on an ECS, and reconstruct the ModelArts base images or thirdparty images with the file. This allows you to customize Docker images push the images to SWR. For details, see Creating and Using a Custom Image in Notebook.

### 11.3.3 Saving a Notebook Instance as a Custom Image

### 11.3.3.1 Saving a Notebook Environment Image

To save a notebook environment image, do as follows: Create a notebook instance using a preset image, install custom software and dependencies on the base image, and save the running instance as a container image.

In the saved image, the installed dependencies are retained. The data stored in **home/ma-user/work** for persistent storage will not be stored. When you use VS Code for remote development, the plug-ins installed on the Server are retained.

### **NOTE**

Images stored in a notebook instance cannot be larger than 35 GB and there cannot be more than 125 image layers. Otherwise, the image cannot be saved.

If error "The container size (xx) is greater than the threshold (25G)" is reported when an image is saved, handle the error by referring to What Do I Do If Error "The container size (xG) is greater than the threshold (25G)" Is Displayed When I Save an Image?.

### Prerequisites

The notebook instance is in **Running** state.

### Saving an Image

- Log in to the ModelArts management console and choose **DevEnviron** > **Notebook** in the navigation pane on the left to switch to notebook of the new version.
- 2. In the notebook instance list, select the target notebook instance and choose **Save Image** from the **More** drop-down list in the **Operation** column. The **Save Image** dialog box is displayed.
- 3. In the **Save Image** dialog box, configure parameters. Click **OK** to save the image.

Choose an organization from the **Organization** drop-down list. If no organization is available, click **Create** on the right to create one.

Users in an organization can share all images in the organization.

4. The image will be saved as a snapshot, and it will take about 5 minutes. During this period of time, do not perform any operations on the instance.

Figure 11-4 Saving as a snapshot

notebook	C Snapshotting
NOTICE	

The time required for saving an image as a snapshot will be counted in the instance running duration. If the instance running duration expires before the snapshot is saved, saving the image will fail.

- 5. After the image is saved, the instance status changes to **Running**. View the image on the **Image Management** page.
- 6. Click the name of the image to view its details.

### 11.3.3.2 Using a Custom Image to Create a Notebook Instance

The images saved from a notebook instance can be viewed on the **Image Management** page. You can use these images to create new notebook instances, which inherit the software configurations of the original notebook instances.

You can use either of the following methods:

Method 1: On the **Create Notebook** page, click **Private Image** and select the saved image.

Figure 11-5 Selecting a custom image to create a notebook instance

<b>∗</b> Image	Pu	ublic image	Private image					
					E	nter an image name	Q	С
		Name		Тад		Description		
	٢		, i	1				

Method 2: On the **Image Management** page, click the target image to access its details page. Then, click **Create Notebook**.

### 11.3.4 Creating and Using a Custom Image in Notebook

### **11.3.4.1 Application Scenarios and Process**

If preset images cannot meet your service requirements, you can create container images based on the preset images for development and training.

Generally, you will need to reconstruct the ModelArts development environment, for example, by installing, upgrading, or uninstalling some packages. However, the root permission is required when certain packages are installed or upgraded. The running notebook instance does not have the root permission. As a result, you need to install the software that requires the root permission in the notebook instance, which is currently unavailable in the preset development environment.

You need to write a Dockerfile based on a preset public image to customize your image. Then, debug the image so that it can be used in ModelArts. At last, register the image with ModelArts so that it can be used to create development environments to meet your service requirements.

This example shows how to use **ma-cli** commands in ModelArts CLI to create and register a custom image for AI development with a MindSpore base image. For details, see **ma-cli Image Building Command**. The following figure shows the whole process.

Figure 11-6 Creating an image



### 11.3.4.2 Step 1 Creating a Custom Image

This section shows you how to create an image by loading an image creation template and writing a Dockerfile. Ensure that you have created the development environment and opened a terminal on the **Notebook** page. For details about Dockerfiles, see **Dockerfile reference**.

Step 1 Configure authentication information, specify a profile, and enter the account information as prompted. For more information about authentication, see ma-cli Authentication.

ma-cli configure --auth PWD -P xxx



**Step 2** Run **env|grep -i CURRENT\_IMAGE\_NAME** to query the image used by the current instance.

**Step 3** Create an image.

1. Obtain the SWR address of the base image.

CURRENT\_IMAGE\_NAME=swr.cn-south-222.ai.pcl.cn/atelier/ mindspore\_1\_7\_0:mindspore\_1.7.0-cann\_5.1.0-py\_3.7-euler\_2.8.3-aarch64d910-20220906 2. Load an image creation template.

Run the ma-cli image get-template command to query the image template.

Run the **ma-cli image add-template** command to load the image template to the specified folder. The default path is where the current command is located. For example, load the

**upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_5.1.RC2** image creation template.

ma-cli image add-template upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_5.1.RC2

d\$pore) [ma-user work]Sma-cli image add-template upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_S.1.RC2 --force (f) Successfully add configuration template [upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_S.1.RC2] under folder [/home/ma-user/work/.ma/upgrade\_ascend\_mindspore\_1.8.1\_and\_

3. Modify a Dockerfile.

Use the Dockerfile to upgrade the base image **mindspore1.7.0-cann5.1.0py3.7-euler2.8.3** to adapt to CANN 5.1.RC2 and MindSpore 1.8.1 and create an image for AI development.

After the image template is loaded, the Dockerfile will be automatically loaded in **.ma/upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_5.1.RC2**. The content is as follows and you can modify it based on your needs.

#The following uses Mindspore-1.7 as an example, which can be replaced with the image used by the notebook instance. FROM swr.cn-south-222.ai.pcl.cn/atelier/mindspore\_1\_7\_0:mindspore\_1.7.0-cann\_5.1.0-py\_3.7-euler\_2.8.3-aarch64-d910-20220715093657-9446c6a

ARG CANN=Ascend-cann-toolkit\_5.1.RC2\_linux-aarch64.run

# Modify the notebook proxy based on the actual needs. ENV HTTP\_PROXY=http://proxy.modelarts.com:80 \ http\_proxy=http://proxy.modelarts.com:80 \ HTTPS\_PROXY=http://proxy.modelarts.com:80 \ https\_proxy=http://proxy.modelarts.com:80

USER root

# Download CANN-5.1.RC2 and install CANN package, which is a dependency package for mindspore-1.8.1.

# For details about the mapping between Mindpore and CANN and the download address of CANN, see the official website of Mindpore.

RUN wget https://ascend-repo.obs.cn-east-2.myhuaweicloud.com/CANN/CANN%205.1.RC2/\${CANN} - P /tmp && \

chmod +x /tmp/\${CANN} && \ sh -x /tmp/\${CANN} --quiet --full && \ rm -f /tmp/\${CANN}

ENV PYTHONPATH=/usr/local/Ascend/tfplugin/latest/python/site-packages:/usr/local/Ascend/ascend-toolkit/latest/python/site-packages:/usr/local/Ascend/ascend-toolkit/latest/opp/op\_impl/built-in/ai\_core/tbe:/usr/local/seccomponent/lib

USER ma-user

echo "successfully install mindspore 1.8.1"

## [Optional] Uncomment to set default conda env #ENV DEFAULT\_CONDA\_ENV\_NAME=/home/ma-user/anaconda3/envs/MindSpore 4. Build an image.

Run the **ma-cli image build** command to build an image with the Dockerfile. For more information, see **Creating an Image in ModelArts Notebook**.

ma-cli image build .ma/upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_5.1.RC2/Dockerfile -swr notebooktest/my\_image:0.0.1 -P XXX

The Dockerfile is stored in .ma/

**upgrade\_ascend\_mindspore\_1.8.1\_and\_cann\_5.1.RC2/Dockerfile** and the new image is stored in **notebook-test/my\_image:0.0.1** in SWR. **XXX** indicates the profile specified for authentication.

(Mi	ndSpore)	[ma-user	work]\$ma-cli image build .ma/upgrade_ascend_mindspore_1.8.1_and_cann_5.1.RC2/Dockerfile -swr notebook-test/my_image:0.0.1 -P yuan
[+]	Building	1121.25	(8/8) FINISHED
			.dockerignore
			context: 2B
			build definition from Dockerfile
			dockerfile: 1.71kB
			metadata for swr.cn-north-4.myhuaweicloud.com/atelier/mindspore_1_7_0:mindspore_1.7.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64-d910-2022090
			n-north-4.myhuaweicloud.com/atelier/mindspore_1_7_0:mindspore_1.7.0-cann_5.1.0-py_3.7-euler_2.8.3-aarch64-d910-20220906@sha256:c1ee08554
			n-north-4.myhuaweicloud.com/atelier/mindspore 1 7 0:mindspore 1.7.0-cann 5.1.0-py 3.7-euler 2.8.3-aarch64-d910-20220906@sha256:c1ee08554

----End

### 11.3.4.3 Step 2 Registering a New Image

After an image is debugged, register it with ModelArts image management so that the image can be used in ModelArts.

Use either of the following methods to register the image with ModelArts:

 Method 1: Run the ma-cli image register command to register an image. Then, the information of the registered image is returned, including image ID and name, as shown in the following figure. For more information, see
 Registering SWR Images with ModelArts Image Management. ma-cli image register --swr-path=swr.cn-south-222.ai.pcl.cn/cloud-test/mindspore\_1\_8:v1 -a AARCH64 -rs ASCEND -P XXX

-a indicates that the image supports the Arm architecture, -rs indicates that the image supports the Ascend chip, and XXX indicates the profile specified during authentication.



Figure 11-7 Registered image

• **Method 2**: Register the image on the ModelArts management console.

Log in to the ModelArts management console. In the navigation pane on the left, select **Image Management**. The **Image Management** page is displayed.

Click **Register**. Paste the complete SWR address, or click **D** to select a private image from SWR for registration, as shown in **Figure 11-8**.

Select the architecture and type based on the site requirements. The architecture and type must be the same as those of the image source.

#### Figure 11-8 Selecting an image

Select Im	nage				×
My Images	/ swr.cn-				С
My Images					Q
Ima	age Version	Updated On 🔶	Template Address	Size	
<b>€</b> E	Back				
<ul> <li>Image: Image: Ima</li></ul>	v3	Sep 15, 2023 14:16:29 GMT+08:00		2.15GB	
00	v2	Sep 14, 2023 18:57:47 GMT+08:00			
		OK Cancel			

### 11.3.4.4 Step 3 Using a New Image to Create a Development Environment

### Procedure

1. After an image is registered, it is available for development environment creation. You can log in to the ModelArts management console, choose **DevEnviron** > **Notebook**, and select the image during creation.

#### Figure 11-9 Creating a development environment

★ Image	Public image	Private image				
					Enter an image name	QC
	Name		Тад	Organization	Description	
	۲		1.0	deep-learning1		

2. Access the development environment.

3 Launcher		
	Notebook	c
	Μ	Р
	MindSpore	python-3.7.10
	>_ Console	
	MindSpore	P python-3.7.10

Figure 11-10 Accessing a development environment

3. Click the MindSpore icon to create an IPYNV file and import MindSpore to the file. Then, the installed MindSpore 1.8.1 can be used.

### Figure 11-11 Creating an IPYNB file

🖪 Untitled.ipynb					٠								
	+	Ж		Ċ	۲	-	C	**	Code	~	()	git	
124 ms	[4]	] <pre>import mindspore print(mindsporeversion)</pre>											
	1.8.1												

4. Open a new terminal and check the CANN version. The version is the same as that installed in Dockerfile.

Figure 11-12 Checking the CANN version

(MindSpore)	[	na-use	er wor	rk]\$19	5 -al	l /ı	usr/loo	cal/Ascend/ascend-toolkit
total 24								
drwxr-xr-x	1	root	root	4096	Jul	12	16:03	
drwxr-xr-x	1	root	root	4096	Jul	12	15:59	
1rwxrwxrwx	1	root	root	9	Jul	12	16:03	5.1 -> ./5.1.RC2
drwxr-xr-x	1	root	root	4096	Jul	12	16:02	5.1.RC1.1
drwxr-xr-x	16	root	root	4096	Jul	12	16:03	5.1.RC2
drwxr-xr-x	1	root	root	4096	Jul	12	16:03	latest
-r-xr-xr-x	1	root	root	651	Jul	12	16:03	set env.sh

# 11.4 Using a Custom Image to Train Models (Model Training)

### 11.4.1 Overview

The subscribed algorithms and preset images can be used in most training scenarios. In certain scenarios, ModelArts allows you to create custom images to train models.

Customizing an image requires a deep understanding of containers. Use this method only if the subscribed algorithms and preset images cannot meet your requirements. Custom images can be used to train models in ModelArts only after they are uploaded to the Software Repository for Container (SWR).

You can use custom images for training on ModelArts in either of the following ways:

• Using a preset image with customization

If you use a preset image to create a training job and you need to modify or add some software dependencies based on the preset image, you can customize the preset image. In this case, select a preset image and choose **Customize** from the framework version drop-down list box.

• Using a custom image

You can create an image based on the ModelArts image specifications, select your own image and configure the code directory (optional) and boot command to create a training job.

### **NOTE**

When you use a custom image to create a training job, the boot command must be executed in the **/home/ma-user** directory. Otherwise, the training job may run abnormally.

### Using a Preset Image with Customization

The only difference between this method and creating a training job totally based on a preset image is that you must select an image. You can create a custom image based on a preset image. For details about how to create a custom image based on a preset framework, see **Using a Base Image to Create a Training Image**.

★ Boot Mode	1 Preset image	Custom image	
		Customize	•
* Image			Select
★ Code Directory ⑦			Select
* Boot File			Select

Figure 11-13 Creating an algorithm using a preset image with customization

The process of this method is the same as that of creating a training job based on a preset image. For example:

- The system automatically injects environment variables.
  - PATH=\${MA\_HOME}/anaconda/bin:\${PATH}
  - LD\_LIBRARY\_PATH=\${MA\_HOME}/anaconda/lib:\${LD\_LIBRARY\_PATH}
  - PYTHONPATH=\${MA\_JOB\_DIR}:\${PYTHONPATH}
- The selected boot file will be automatically started using Python commands. Ensure that the Python environment is correct. The PATH environment variable is automatically injected. Run the following commands to check the Python version for the training job:
  - export MA\_HOME=/home/ma-user; docker run --rm {image} \$ {MA\_HOME}/anaconda/bin/python -V
  - docker run --rm {image} \$(which python) -V
- The system automatically adds hyperparameters associated with the preset image.

### Using a Custom Image

Figure 11-14 Creating an algorithm using a custom image

★ Boot Mode	Preset image	Custom image	
* Image			Select
Code Directory			Select
* Boot Command ⑦	1		

For details about how to use custom images supported by the new-version training, see Using a Custom Image to Create a CPU- or GPU-based Training Job.

If all used images are customized, do as follows to use a specified Conda environment to start training:

Training jobs do not run in a shell. Therefore, you are not allowed to run the **conda activate** command to activate a specified Conda environment. In this case, use other methods to start training.

For example, Conda in your custom image is installed in the **/home/ma-user/ anaconda3** directory, the Conda environment is **python-3.7.10**, and the training script is stored in **/home/ma-user/modelarts/user-job-dir/code/train.py**. Use a specified Conda environment to start training in one of the following ways:

 Method 1: Configure the correct DEFAULT\_CONDA\_ENV\_NAME and ANACONDA\_DIR environment variables for the image.

Run the **python** command to start the training script. The following shows an example:

python /home/ma-user/modelarts/user-job-dir/code/train.py

• Method 2: Use the absolute path of Conda environment Python.

Run the **/home/ma-user/anaconda3/envs/python-3.7.10/bin/python** command to start the training script. The following shows an example: /home/ma-user/anaconda3/envs/python-3.7.10/bin/python /home/ma-user/modelarts/user-job-dir/ code/train.py

• Method 3: Configure the path environment variable.

Configure the bin directory of the specified Conda environment into the path environment variable. Run the **python** command to start the training script. The following shows an example:

export PATH=/home/ma-user/anaconda3/envs/python-3.7.10/bin:\$PATH; python /home/ma-user/ modelarts/user-job-dir/code/train.py

• Method 4: Run the **conda run -n** command.

Run the **/home/ma-user/anaconda3/bin/conda run -n python-3.7.10** command to execute the training. The following shows an example: /home/ma-user/anaconda3/bin/conda run -n python-3.7.10 python /home/ma-user/modelarts/userjob-dir/code/train.py

#### D NOTE

If there is an error indicating that the .so file is unavailable in the **\$ANACONDA\_DIR/envs/ \$DEFAULT\_CONDA\_ENV\_NAME/lib** directory, add the directory to **LD\_LIBRARY\_PATH** and place the following command before the preceding boot command:

export LD\_LIBRARY\_PATH=\$ANACONDA\_DIR/envs/\$DEFAULT\_CONDA\_ENV\_NAME/ lib:\$LD\_LIBRARY\_PATH;

For example, the example boot command used in method 1 is as follows:

export LD\_LIBRARY\_PATH=\$ANACONDA\_DIR/envs/\$DEFAULT\_CONDA\_ENV\_NAME/ lib:\$LD\_LIBRARY\_PATH; python /home/ma-user/modelarts/user-job-dir/code/train.py

### 11.4.2 Example: Creating a Custom Image for Training

11.4.2.1 Example: Creating a Custom Image for Development and Training (MindSpore + Ascend)

### 11.4.2.1.1 Scenarios

This section describes how to create an Ascend container image from scratch and use the image for training on ModelArts. The AI engine used in the image is MindSpore, and the resources used for training are powered by Ascend in a dedicated resource pool.

### Constraints

- This example requires the CANN commercial edition. If you do not have permission to download the CANN commercial edition, see other examples for creating a custom image.
- Pay attention to the version mapping between MindSpore and CANN, and between CANN and Ascend driver or firmware. Unmatched versions will lead to a training failure.

### Objective

Create a container image with the following configurations and use the image to create an Ascend-powered training job on ModelArts:

- Ubuntu 18.04
- CANN 6.3.RC2 (commercial edition)
- Python 3.7.13
- MindSpore 2.1.1

**NOTE** 

- CANN 6.3.RC2 and MindSpore 2.1.1 are used in the following examples.
- These examples show how to create an Ascend container image and run the image in a dedicated resource pool with the required Ascend driver or firmware installed.

### Procedure

Before using a custom image to create a training job, you need to be familiar with Docker and have development experience.

- 1. Step 1 Creating an OBS Bucket and Folder
- 2. Step 2 Preparing Script Files and Uploading Them to OBS
- 3. Step 3 Creating a Custom Image
- 4. Step 4 Uploading the Image to SWR
- 5. Step 5 Creating and Debugging a Notebook Instance on ModelArts
- 6. Step 6 Creating a Training Job on ModelArts

### 11.4.2.1.2 Step 1 Creating an OBS Bucket and Folder

### Procedure

Create a bucket and folders in OBS for storing the sample dataset and training code. In this example, create a bucket named **test-modelarts** and folders listed in **Table 11-13**.

Folder	Description
obs://test-modelarts/ascend/demo- code/	Store the Ascend training script.
obs://test-modelarts/ascend/demo- code/run_ascend/	Store the startup scripts of the Ascend training script.
obs://test-modelarts/ascend/log/	Store training log files.

### 11.4.2.1.3 Step 2 Preparing Script Files and Uploading Them to OBS

- 1. Prepare the training script **mindspore-verification.py** and Ascend startup scripts (five in total) required in this example.
  - For details about the training script, see Training Script.
  - For details about the following Ascend startup scripts, see Ascend Startup Scripts.
    - i. run\_ascend.py
    - ii. common.py
    - iii. rank\_table.py
    - iv. manager.py
    - v. fmk.py

#### **NOTE**

The **mindspore-verification.py** and **run\_ascend.py** scripts are invoked by the **Boot Command** parameter during training job creation. For details, see **Boot Command**.

The **common.py**, **rank\_table.py**, **manager.py**, and **fmk.py** scripts are invoked when the **run\_ascend.py** script is running.

- 2. Upload the training script **mindspore-verification.py** to **obs://test-modelarts/ascend/demo-code/** in the OBS bucket.
- 3. Upload the five Ascend startup scripts to the **obs://test-modelarts/ascend/ demo-code/run\_ascend/** folder in the OBS bucket.

### **Training Script**

#### mindspore-verification.py

import os import numpy as np from mindspore import Tensor import mindspore.ops as ops import mindspore.context as context

```
print('Ascend Envs')

print('-----')

print('JOB_ID: ', os.environ['JOB_ID'])

print('RANK_TABLE_FILE: ', os.environ['RANK_TABLE_FILE'])

print('RANK_SIZE: ', os.environ['RANK_SIZE'])

print('ASCEND_DEVICE_ID: ', os.environ['ASCEND_DEVICE_ID'])

print('DEVICE_ID: ', os.environ['DEVICE_ID'])

print('RANK_ID: ', os.environ['RANK_ID'])
```

print('-----')

```
context.set_context(device_target="Ascend")
x = Tensor(np.ones([1,3,3,4]).astype(np.float32))
y = Tensor(np.ones([1,3,3,4]).astype(np.float32))
```

print(ops.add(x, y))

### **Ascend Startup Scripts**

```
run_ascend.py
import sys
import os
from common import RunAscendLog
from common import RankTableEnv
from rank_table import RankTable, RankTableTemplate1, RankTableTemplate2
from manager import FMKManager
if __name__ == '__main__':
  log = RunAscendLog.setup_run_ascend_logger()
  if len(sys.argv) <= 1:
     log.error('there are not enough args')
     sys.exit(1)
  train_command = sys.argv[1:]
  log.info('training command')
  log.info(train_command)
  if os.environ.get(RankTableEnv.RANK_TABLE_FILE_V1) is not None:
     # new format rank table file
     rank_table_path = os.environ.get(RankTableEnv.RANK_TABLE_FILE_V1)
     RankTable.wait_for_available(rank_table_path)
     rank_table = RankTableTemplate1(rank_table_path)
  else:
     # old format rank table file
     rank_table_path_origin = RankTableEnv.get_rank_table_template2_file_path()
     RankTable.wait_for_available(rank_table_path_origin)
     rank_table = RankTableTemplate2(rank_table_path_origin)
  if rank_table.get_device_num() >= 1:
     log.info('set rank table %s env to %s' % (RankTableEnv.RANK_TABLE_FILE,
rank table.get rank table path()))
     RankTableEnv.set_rank_table_env(rank_table.get_rank_table_path())
  else:
     log.info('device num < 1, unset rank table %s env' % RankTableEnv.RANK_TABLE_FILE)
     RankTableEnv.unset_rank_table_env()
  instance = rank_table.get_current_instance()
  server = rank_table.get_server(instance.server_id)
  current_instance = RankTable.convert_server_to_instance(server)
  fmk_manager = FMKManager(current_instance)
  fmk_manager.run(rank_table.get_device_num(), train_command)
  return code = fmk manager.monitor()
  fmk_manager.destroy()
  sys.exit(return_code)
common.py
import logging
import os
```

logo = 'Training'
```
# Rank Table Constants
class RankTableEnv:
  RANK_TABLE_FILE = 'RANK_TABLE_FILE'
  RANK_TABLE_FILE_V1 = 'RANK_TABLE_FILE_V_1_0'
  HCCL_CONNECT_TIMEOUT = 'HCCL_CONNECT_TIMEOUT'
  # jobstart_hccl.json is provided by the volcano controller of Cloud-Container-Engine(CCE)
  HCCL_JSON_FILE_NAME = 'jobstart_hccl.json'
  RANK_TABLE_FILE_DEFAULT_VALUE = '/user/config/%s' % HCCL_JSON_FILE_NAME
  @staticmethod
  def get_rank_table_template1_file_dir():
    parent_dir = os.environ[ModelArts.MA_MOUNT_PATH_ENV]
    return os.path.join(parent_dir, 'rank_table')
  @staticmethod
  def get_rank_table_template2_file_path():
    rank_table_file_path = os.environ.get(RankTableEnv.RANK_TABLE_FILE)
    if rank_table_file_path is None:
       return RankTableEnv.RANK_TABLE_FILE_DEFAULT_VALUE
    return os.path.join(os.path.normpath(rank_table_file_path),
RankTableEnv.HCCL JSON_FILE_NAME)
  @staticmethod
  def set_rank_table_env(path):
    os.environ[RankTableEnv.RANK_TABLE_FILE] = path
  @staticmethod
  def unset_rank_table_env():
    del os.environ[RankTableEnv.RANK_TABLE_FILE]
class ModelArts:
  MA_MOUNT_PATH_ENV = 'MA_MOUNT_PATH'
  MA_CURRENT_INSTANCE_NAME_ENV = 'MA_CURRENT_INSTANCE_NAME'
  MA_VJ_NAME = 'MA_VJ_NAME'
  MA_CURRENT_HOST_IP = 'MA_CURRENT_HOST_IP'
  CACHE_DIR = '/cache'
  TMP_LOG_DIR = '/tmp/log/'
  FMK_WORKSPACE = 'workspace'
  @staticmethod
  def get_current_instance_name():
    return os.environ[ModelArts.MA_CURRENT_INSTANCE_NAME_ENV]
  @staticmethod
  def get_current_host_ip():
    return os.environ.get(ModelArts.MA_CURRENT_HOST_IP)
  @staticmethod
  def get_job_id():
    ma_vj_name = os.environ[ModelArts.MA_VJ_NAME]
    return ma_vj_name.replace('ma-job', 'modelarts-job', 1)
  @staticmethod
  def get_parent_working_dir():
    if ModelArts.MA_MOUNT_PATH_ENV in os.environ:
       return os.path.join(os.environ.get(ModelArts.MA_MOUNT_PATH_ENV),
ModelArts.FMK_WORKSPACE)
```

```
return ModelArts.CACHE_DIR
```

class RunAscendLog:

```
@staticmethod
```

def setup\_run\_ascend\_logger():
 name = logo

formatter = logging.Formatter(fmt='[run ascend] %(asctime)s - %(levelname)s - %(message)s')

handler = logging.StreamHandler()
handler.setFormatter(formatter)

logger = logging.getLogger(name) logger.setLevel(logging.INFO) logger.addHandler(handler) logger.propagate = False return logger

```
@staticmethod
def get_run_ascend_logger():
    return logging.getLogger(logo)
```

### • rank\_table.py

import json import time import os

from common import ModelArts from common import RunAscendLog from common import RankTableEnv

log = RunAscendLog.get\_run\_ascend\_logger()

class Device:

```
def __init__(self, device_id, device_ip, rank_id):
    self.device_id = device_id
    self.device_ip = device_ip
    self.rank_id = rank_id
```

```
class Instance:
    def __init__(self, pod_name, server_id, devices):
        self.pod_name = pod_name
        self.server_id = server_id
        self.devices = self.parse_devices(devices)
```

@staticmethod

```
def parse_devices(devices):
    if devices is None:
        return []
    device_object_list = []
    for device in devices:
        device_object_list.append(Device(device['device_id'], device['device_ip'], "))
```

```
return device_object_list
```

def set\_devices(self, devices):
 self.devices = devices

class Group:

```
def __init__(self, group_name, device_count, instance_count, instance_list):
    self.group_name = group_name
    self.device_count = int(device_count)
    self.instance_count = int(instance_count)
    self.instance_list = self.parse_instance_list(instance_list)
```

@staticmethod

```
def parse_instance_list(instance_list):
     instance_object_list = []
     for instance in instance_list:
        instance_object_list.append(
           Instance(instance['pod_name'], instance['server_id'], instance['devices']))
     return instance_object_list
class RankTable:
  STATUS FIELD = 'status'
  COMPLETED_STATUS = 'completed'
  def __init__(self):
     self.rank_table_path = ""
     self.rank_table = {}
  @staticmethod
  def read_from_file(file_path):
     with open(file_path) as json_file:
        return json.load(json_file)
  @staticmethod
  def wait_for_available(rank_table_file, period=1):
     log.info('Wait for Rank table file at %s ready' % rank_table_file)
     complete_flag = False
     while not complete flag:
        with open(rank_table_file) as json_file:
           data = json.load(json_file)
           if data[RankTable.STATUS_FIELD] == RankTable.COMPLETED_STATUS:
             log.info('Rank table file is ready for read')
             log.info('\n' + json.dumps(data, indent=4))
             return True
        time.sleep(period)
     return False
  @staticmethod
  def convert_server_to_instance(server):
     device_list = []
     for device in server['device']:
        device_list.append(
           Device(device_id=device['device_id'], device_ip=device['device_ip'],
rank_id=device['rank_id']))
     ins = Instance(pod_name=", server_id=server['server_id'], devices=[])
     ins.set_devices(device_list)
     return ins
  def get_rank_table_path(self):
     return self.rank_table_path
  def get_server(self, server_id):
     for server in self.rank_table['server_list']:
        if server['server_id'] == server_id:
           log.info('Current server')
           log.info('\n' + json.dumps(server, indent=4))
           return server
     log.error('server [%s] is not found' % server_id)
     return None
class RankTableTemplate2(RankTable):
  def __init__(self, rank_table_template2_path):
     super().__init__()
```

```
json data = self.read from file(file path=rank table template2 path)
     self.status = json_data[RankTableTemplate2.STATUS_FIELD]
     if self.status != RankTableTemplate2.COMPLETED_STATUS:
        return
     # sorted instance list by the index of instance
     # assert there is only one group
     json_data["group_list"][0]["instance_list"] = sorted(json_data["group_list"][0]["instance_list"],
                                          key=RankTableTemplate2.get_index)
     self.group_count = int(json_data['group_count'])
     self.group_list = self.parse_group_list(json_data['group_list'])
     self.rank_table_path, self.rank_table = self.convert_template2_to_template1_format_file()
  @staticmethod
  def parse_group_list(group_list):
     group_object_list = []
     for group in group_list:
        group_object_list.append(
           Group(group['group_name'], group['device_count'], group['instance_count'],
group['instance_list']))
     return group_object_list
  @staticmethod
  def get_index(instance):
     # pod_name example: job94dc1dbf-job-bj4-yolov4-15
     pod_name = instance["pod_name"]
     return int(pod_name[pod_name.rfind("-") + 1:])
  def get_current_instance(self):
     get instance by pod name
     specially, return the first instance when the pod name is None
     :return:
     pod_name = ModelArts.get_current_instance_name()
     if pod_name is None:
        if len(self.group_list) > 0:
          if len(self.group_list[0].instance_list) > 0:
             return self.group_list[0].instance_list[0]
        return None
     for group in self.group_list:
        for instance in group.instance_list:
           if instance.pod_name == pod_name:
             return instance
     return None
  def convert_template2_to_template1_format_file(self):
     rank_table_template1_file = {
        'status': 'completed',
        'version': '1.0',
        'server_count': '0',
        'server_list': []
     }
     logic_index = 0
     server_map = {}
     # collect all devices in all groups
     for group in self.group_list:
        if group.device_count == 0:
           continue
        for instance in group.instance_list:
           if instance.server_id not in server_map:
             server_map[instance.server_id] = []
```

```
for device in instance.devices:
             template1_device = {
                'device_id': device.device_id,
                'device_ip': device.device_ip,
                'rank_id': str(logic_index)
             logic_index += 1
             server_map[instance.server_id].append(template1_device)
     server_count = 0
     for server_id in server_map:
        rank_table_template1_file['server_list'].append({
           'server_id': server_id,
           'device': server_map[server_id]
        })
        server_count += 1
     rank_table_template1_file['server_count'] = str(server_count)
     log.info('Rank table file (Template1)')
     log.info('\n' + json.dumps(rank_table_template1_file, indent=4))
     if not os.path.exists(RankTableEnv.get_rank_table_template1_file_dir()):
        os.makedirs(RankTableEnv.get_rank_table_template1_file_dir())
     path = os.path.join(RankTableEnv.get_rank_table_template1_file_dir(),
RankTableEnv.HCCL_JSON_FILE_NAME)
     with open(path, 'w') as f:
        f.write(json.dumps(rank_table_template1_file))
        log.info('Rank table file (Template1) is generated at %s', path)
     return path, rank_table_template1_file
  def get_device_num(self):
     total_device_num = 0
     for group in self.group_list:
        total_device_num += group.device_count
     return total_device_num
class RankTableTemplate1(RankTable):
  def __init__(self, rank_table_template1_path):
     super().__init__()
     self.rank_table_path = rank_table_template1_path
     self.rank table = self.read from file(file path=rank table template1 path)
  def get_current_instance(self):
     current_server = None
     server_list = self.rank_table['server_list']
     if len(server_list) == 1:
        current_server = server_list[0]
     elif len(server_list) > 1:
        host_ip = ModelArts.get_current_host_ip()
        if host_ip is not None:
           for server in server_list:
             if server['server_id'] == host_ip:
                current_server = server
                break
        else:
           current_server = server_list[0]
     if current server is None:
        log.error('server is not found')
        return None
     return self.convert_server_to_instance(current_server)
  def get_device_num(self):
     server_list = self.rank_table['server_list']
```

```
device_num = 0
     for server in server_list:
       device_num += len(server['device'])
     return device_num
manager.py
import time
import os
import os.path
import signal
from common import RunAscendLog
from fmk import FMK
log = RunAscendLog.get_run_ascend_logger()
class FMKManager:
  # max destroy time: ~20 (15 + 5)
  # ~ 15 (1 + 2 + 4 + 8)
  MAX_TEST_PROC_CNT = 4
  def __init__(self, instance):
     self.instance = instance
     self.fmk = []
     self.fmk_processes = []
     self.get_sigterm = False
     self.max_test_proc_cnt = FMKManager.MAX_TEST_PROC_CNT
  # break the monitor and destroy processes when get terminate signal
  def term handle(func):
     def receive_term(signum, stack):
        log.info('Received terminate signal %d, try to destroyed all processes' % signum)
        stack.f_locals['self'].get_sigterm = True
     def handle_func(self, *args, **kwargs):
        origin_handle = signal.getsignal(signal.SIGTERM)
        signal.signal(signal.SIGTERM, receive term)
        res = func(self, *args, **kwargs)
        signal.signal(signal.SIGTERM, origin_handle)
        return res
     return handle_func
  def run(self, rank_size, command):
     for index, device in enumerate(self.instance.devices):
        fmk_instance = FMK(index, device)
        self.fmk.append(fmk_instance)
        self.fmk_processes.append(fmk_instance.run(rank_size, command))
  @term handle
  def monitor(self, period=1):
     # busy waiting for all fmk processes exit by zero
     # or there is one process exit by non-zero
     fmk_cnt = len(self.fmk_processes)
     zero_ret_cnt = 0
     while zero_ret_cnt != fmk_cnt:
        zero_ret_cnt = 0
        for index in range(fmk_cnt):
          fmk = self.fmk[index]
          fmk_process = self.fmk_processes[index]
          if fmk_process.poll() is not None:
             if fmk_process.returncode != 0:
                log.error('proc-rank-%s-device-%s (pid: %d) has exited with non-zero code: %d'
                      % (fmk.rank_id, fmk.device_id, fmk_process.pid, fmk_process.returncode))
                return fmk_process.returncode
```

```
zero_ret_cnt += 1
        if self.get_sigterm:
          break
        time.sleep(period)
     return 0
  def destroy(self, base_period=1):
     log.info('Begin destroy training processes')
     self.send_sigterm_to_fmk_process()
     self.wait_fmk_process_end(base_period)
     log.info('End destroy training processes')
  def send_sigterm_to_fmk_process(self):
     # send SIGTERM to fmk processes (and process group)
     for r_index in range(len(self.fmk_processes) - 1, -1, -1):
        fmk = self.fmk[r_index]
        fmk_process = self.fmk_processes[r_index]
        if fmk_process.poll() is not None:
          log.info('proc-rank-%s-device-%s (pid: %d) has exited before receiving the term signal',
                 fmk.rank_id, fmk.device_id, fmk_process.pid)
          del self.fmk_processes[r_index]
          del self.fmk[r_index]
        try:
          os.killpg(fmk_process.pid, signal.SIGTERM)
        except ProcessLookupError:
          pass
  def wait_fmk_process_end(self, base_period):
     test_cnt = 0
     period = base_period
     while len(self.fmk_processes) > 0 and test_cnt < self.max_test_proc_cnt:
        for r_index in range(len(self.fmk_processes) - 1, -1, -1):
          fmk = self.fmk[r_index]
          fmk_process = self.fmk_processes[r_index]
          if fmk_process.poll() is not None:
             log.info('proc-rank-%s-device-%s (pid: %d) has exited',
                    fmk.rank_id, fmk.device_id, fmk_process.pid)
             del self.fmk_processes[r_index]
             del self.fmk[r_index]
        if not self.fmk_processes:
          break
        time.sleep(period)
        period *= 2
        test_cnt += 1
     if len(self.fmk_processes) > 0:
        for r_index in range(len(self.fmk_processes) - 1, -1, -1):
          fmk = self.fmk[r_index]
          fmk_process = self.fmk_processes[r_index]
          if fmk_process.poll() is None:
             log.warn('proc-rank-%s-device-%s (pid: %d) has not exited within the max waiting time,
                    'send kill signal',
                   fmk.rank_id, fmk.device_id, fmk_process.pid)
             os.killpg(fmk_process.pid, signal.SIGKILL)
fmk.py
import os
import subprocess
import pathlib
from contextlib import contextmanager
from common import RunAscendLog
from common import RankTableEnv
from common import ModelArts
log = RunAscendLog.get_run_ascend_logger()
```

class FMK:

```
def __init__(self, index, device):
  self.job_id = ModelArts.get_job_id()
  self.rank_id = device.rank_id
  self.device_id = str(index)
def gen_env_for_fmk(self, rank_size):
  current_envs = os.environ.copy()
  current_envs['JOB_ID'] = self.job_id
  current_envs['ASCEND_DEVICE_ID'] = self.device_id
  current_envs['DEVICE_ID'] = self.device_id
  current_envs['RANK_ID'] = self.rank_id
  current_envs['RANK_SIZE'] = str(rank_size)
  FMK.set_env_if_not_exist(current_envs, RankTableEnv.HCCL_CONNECT_TIMEOUT, str(1800))
  log_dir = FMK.get_log_dir()
  process_log_path = os.path.join(log_dir, self.job_id, 'ascend', 'process_log', 'rank_' + self.rank_id)
  FMK.set_env_if_not_exist(current_envs, 'ASCEND_PROCESS_LOG_PATH', process_log_path)
  pathlib.Path(current_envs['ASCEND_PROCESS_LOG_PATH']).mkdir(parents=True, exist_ok=True)
  return current_envs
@contextmanager
def switch_directory(self, directory):
  owd = os.getcwd()
  try:
     os.chdir(directory)
     yield directory
  finally:
     os.chdir(owd)
def get_working_dir(self):
  fmk_workspace_prefix = ModelArts.get_parent_working_dir()
  return os.path.join(os.path.normpath(fmk_workspace_prefix), 'device%s' % self.device_id)
@staticmethod
def get_log_dir():
  parent_path = os.getenv(ModelArts.MA_MOUNT_PATH_ENV)
  if parent_path:
     log_path = os.path.join(parent_path, 'log')
     if os.path.exists(log_path):
       return log_path
  return ModelArts.TMP_LOG_DIR
@staticmethod
def set_env_if_not_exist(envs, env_name, env_value):
  if env_name in os.environ:
     log.info('env already exists. env_name: %s, env_value: %s ' % (env_name, env_value))
     return
  envs[env_name] = env_value
def run(self, rank_size, command):
  envs = self.gen_env_for_fmk(rank_size)
  log.info('bootstrap proc-rank-%s-device-%s' % (self.rank_id, self.device_id))
  log_dir = FMK.get_log_dir()
  if not os.path.exists(log_dir):
     os.makedirs(log_dir)
  log_file = '%s-proc-rank-%s-device-%s.txt' % (self.job_id, self.rank_id, self.device_id)
```

```
log_file_path = os.path.join(log_dir, log_file)
working_dir = self.get_working_dir()
if not os.path.exists(working_dir):
    os.makedirs(working_dir)
```

log.info('proc-rank-%s-device-%s (pid: %d)', self.rank\_id, self.device\_id, training\_proc.pid)

```
# https://docs.python.org/3/library/subprocess.html#subprocess.Popen.wait
subprocess.Popen(['tee', log_file_path], stdin=training_proc.stdout)
```

return training\_proc

### 11.4.2.1.4 Step 3 Creating a Custom Image

This section describes how to write a Dockerfile to create a custom image.

Create a container image with the following configurations and use the image to create a training job on ModelArts:

- Ubuntu 18.04
- CANN 6.3.RC2 (commercial edition)
- Python 3.7.13
- MindSpore 2.1.1

#### **NOTE**

Pay attention to the version mapping between MindSpore and CANN, and between CANN and Ascend driver or firmware. Unmatched versions will lead to a training failure.

The following example shows how to create an Ascend container image and run the image in a dedicated resource pool with the required Ascend driver or firmware installed.

- 1. Obtain a Linux AArch64 server running Ubuntu 18.04. Either an ECS or your local PC will do.
- 2. Install Docker.

The following uses Linux AArch64 as an example to describe how to obtain a Docker installation package. For more details about how to install Docker, see official Docker documents.

```
curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh
```

If the **docker images** command can be executed, Docker has been installed. In this case, skip this step.

Start Docker. systemctl start docker

3. Obtain the Docker engine version. docker version | grep -A 1 Engine

The following information is displayed: Engine: Version: 18.09.0

#### **NOTE**

Use the Docker engine of the preceding version or later to create a custom image.

- 4. Create a folder named **context**. mkdir -p context
- 5. Obtain the **pip.conf** file. [global] index-url = https://repo.xxx.com/repository/pypi/simple trusted-host = repo.xxx.com timeout = 120
- 6. Obtain the APT source file **Ubuntu-Ports-bionic.list**. wget -O Ubuntu-Ports-bionic.list https://repo.xxx.com/repository/conf/Ubuntu-Ports-bionic.list
- Download the CANN 6.3.RC2-linux aarch64 and mindspore-2.1.1-cp37cp37m-linux\_aarch64.whl installation files.
  - Download the Ascend-cann-nnae\_6.3.RC2\_linux-aarch64.run file by referring to CANN 6.3.RC2.
  - Download the mindspore-2.1.1-cp37-cp37m-linux\_aarch64.whl file.

### **NOTE**

ModelArts supports only the commercial CANN edition, but not the community edition.

8. Download the Miniconda3 installation file.

Download Miniconda3-py37-4.10.3 (Python 3.7.10) at https:// repo.anaconda.com/miniconda/Miniconda3-py37\_4.10.3-Linux-aarch64.sh.

9. Store the pip source file, .run file, .whl file, and Miniconda3 installation file in the **context** folder, which is as follows:

```
context

Ascend-cann-nnae_6.3.RC2_linux-aarch64.run

mindspore-2.1.1-cp37-cp37m-linux_aarch64.whl

Miniconda3-py37_4.10.3-Linux-aarch64.sh

pip.conf

Ubuntu-Ports-bionic.list
```

10. Write the container image Dockerfile.

```
Create an empty file named Dockerfile in the context folder and copy the following content to the file:
```

# The server on which the container image is created must access the Internet. FROM arm64v8/ubuntu:18.04 AS builder

```
# The default user of the base container image is root.
# USER root
# Install OS dependencies obtained from Mirrors.
COPY Ubuntu-Ports-bionic.list /tmp
RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak && \
  mv /tmp/Ubuntu-Ports-bionic.list /etc/apt/sources.list && \
  echo > /etc/apt/apt.conf.d/00skip-verify-peer.conf "Acquire { https::Verify-Peer false }" && \
  apt-get update && \
  apt-get install -y \
  # utils
  ca-certificates vim curl \
  # CANN 6.3.RC2
  gcc-7 g++ make cmake zlib1g zlib1g-dev openssl libsglite3-dev libssl-dev libfi-dev unzip pciutils
net-tools libblas-dev gfortran libblas3 && \
  apt-get clean && \
  mv /etc/apt/sources.list.bak /etc/apt/sources.list && \
  # Grant the write permission of the parent directory of the CANN 6.3.RC2 installation directory to
ma-user.
  chmod o+w /usr/local
RUN useradd -m -d /home/ma-user -s /bin/bash -g 100 -u 1000 ma-user
```

# Configure the default user and working directory of the container image.

USER ma-user WORKDIR /home/ma-user # Use the PyPI configuration provided by Mirrors. RUN mkdir -p /home/ma-user/.pip/ COPY --chown=ma-user:100 pip.conf /home/ma-user/.pip/pip.conf # Copy the installation files to the /tmp directory in the base container image. COPY --chown=ma-user:100 Miniconda3-py37\_4.10.3-Linux-aarch64.sh /tmp # https://conda.io/projects/conda/en/latest/user-guide/install/linux.html#installing-on-linux # Install Miniconda3 in the /home/ma-user/miniconda3 directory of the base container image. RUN bash /tmp/Miniconda3-py37 4.10.3-Linux-aarch64.sh -b -p /home/ma-user/miniconda3 ENV PATH=\$PATH:/home/ma-user/miniconda3/bin # Install the CANN 6.3.RC2 Python dependency package. RUN pip install numpy~=1.14.3 decorator~=4.4.0 sympy~=1.4 cffi~=1.12.3 protobuf~=3.11.3 \ attrs pyyaml pathlib2 scipy requests psutil absl-py # Install CANN 6.3.RC2 in /usr/local/Ascend. COPY --chown=ma-user:100 Ascend-cann-nnae\_6.3.RC2\_linux-aarch64.run /tmp RUN chmod +x /tmp/Ascend-cann-nnae\_6.3.RC2\_linux-aarch64.run && \ /tmp/Ascend-cann-nnae\_6.3.RC2\_linux-aarch64.run --install --install-path=/usr/local/Ascend # Install MindSpore 2.1.1. COPY --chown=ma-user:100 mindspore-2.1.1-cp37-cp37m-linux\_aarch64.whl /tmp RUN chmod +x /tmp/mindspore-2.1.1-cp37-cp37m-linux\_aarch64.whl && \ pip install /tmp/mindspore-2.1.1-cp37-cp37m-linux\_aarch64.whl # Create the container image. FROM arm64v8/ubuntu:18.04 # Install OS dependencies obtained from Mirrors. COPY Ubuntu-Ports-bionic.list /tmp RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak && \ mv /tmp/Ubuntu-Ports-bionic.list /etc/apt/sources.list && \ echo > /etc/apt/apt.conf.d/00skip-verify-peer.conf "Acquire { https::/Verify-Peer false }" && \ apt-get update && \ apt-get install -y \ # utils ca-certificates vim curl \ # CANN 6.3.RC2 gcc-7 g++ make cmake zlib1g zlib1g-dev openssl libsglite3-dev libssl-dev libfi-dev unzip pciutils net-tools libblas-dev gfortran libblas3 && \ apt-get clean && \ mv /etc/apt/sources.list.bak /etc/apt/sources.list RUN useradd -m -d /home/ma-user -s /bin/bash -g 100 -u 1000 ma-user # Copy the directories from the builder stage to the directories with the same name in the current container image. COPY --chown=ma-user:100 --from=builder /home/ma-user/miniconda3 /home/ma-user/miniconda3 COPY --chown=ma-user.100 --from=builder /home/ma-user/Ascend /home/ma-user/Ascend COPY --chown=ma-user:100 --from=builder /home/ma-user/var /home/ma-user/var COPY --chown=ma-user:100 --from=builder /usr/local/Ascend /usr/local/Ascend # Configure the preset environment variables of the container image. # Configure CANN environment variables. # Configure Ascend driver environment variables. # Set PYTHONUNBUFFERED to 1 to prevent log loss. ENV PATH=\$PATH:/usr/local/Ascend/nnae/latest/bin:/usr/local/Ascend/nnae/latest/compiler/ ccec compiler/bin:/home/ma-user/miniconda3/bin \ LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:/usr/local/Ascend/driver/lib64:/usr/local/Ascend/driver/ lib64/common:/usr/local/Ascend/driver/lib64/driver:/usr/local/Ascend/nnae/latest/lib64:/usr/local/ Ascend/nnae/latest/lib64/plugin/opskernel:/usr/local/Ascend/nnae/latest/lib64/plugin/nnengine \ PYTHONPATH=\$PYTHONPATH:/usr/local/Ascend/nnae/latest/python/site-packages:/usr/local/ Ascend/nnae/latest/opp/built-in/op\_impl/ai\_core/tbe \

ASCEND\_AICPU\_PATH=/usr/local/Ascend/nnae/latest \

ASCEND\_OPP\_PATH=/usr/local/Ascend/nnae/latest/opp \ ASCEND\_HOME\_PATH=/usr/local/Ascend/nnae/latest \ PYTHONUNBUFFERED=1

# Configure the default user and working directory of the container image. USER ma-user WORKDIR /home/ma-user

For details about how to write a Dockerfile, see official Docker documents.

- 11. Verify that the Dockerfile has been created. The following shows the **context** folder:
  - context Ascend-cann-nnae\_6.3.RC2\_linux-aarch64.run Dockerfile
  - mindspore-2.1.1-cp37-cp37m-linux\_aarch64.whl Miniconda3-py37\_4.10.3-Linux-aarch64.sh
  - pip.conf
  - Ubuntu-Ports-bionic.list
- 12. Run the following command in the directory where the Dockerfile is stored to create a container image:

docker build . -t mindspore:2.1.1-cann6.3.RC2

The following log shows that the image has been created. Successfully tagged mindspore:2.1.1-cann6.3.RC2

13. Upload the created image to SWR. For details, see **Step 4 Uploading the Image to SWR**.

## 11.4.2.1.5 Step 4 Uploading the Image to SWR

Upload the created image to SWR so that it can be used to create training jobs on ModelArts.

1. Log in to the SWR console and select the target region.

### Figure 11-15 SWR console



2. Click **Create Organization** in the upper right corner and enter an organization name to create an organization. Customize the organization name. Replace the organization name **deep-learning** in subsequent commands with the actual organization name.

### Figure 11-16 Creating an organization

### Create Organization

Each organization     You can create 5     For centralized m	n name must be globally unique. organizations. anagement of images, limit each organization to one company,		
department, or indi	department, or individual.		
Examples Company or department: cloud-hangzhou or cloud-develop			
Person: john			
Organization Name	deep-learning		

3. Click **Generate Login Command** in the upper right corner to obtain a login command.

### Figure 11-17 Login Command

Cogin Comman	nd	×
learn how to obtain a login co	mmand that has long-term validity.	
docker login -u p :	8@⊁ <sup></sup>	
	ı ت	

Valid Until: Sep 08, 2022 10:43:52 GMT+08:00

- 4. Log in to the local environment as the **root** user and enter the login command.
- 5. Upload the image to SWR.
  - a. Tag the uploaded image. # Replace the region, domain, as well as organization name deep-learning with the actual values. sudo docker tag mindspore:2.1.1-cann6.3.RC2 swr.{region}.{domain}/deep-learning/ mindspore:2.1.1-cann6.3.RC2
  - b. Upload the image.
     # Replace the region, domain, as well as organization name deep-learning with the actual values.
     sudo docker push swr.{region}.{domain}/deep-learning/mindspore:2.1.1-cann6.3.RC2
- 6. After the image is uploaded, choose **My Images** in navigation pane on the left of the SWR console to view the uploaded custom image.

### **NOTE**

Obtain the *region* and *domain* on the management console or contact the system administrator of the target region.

For example, if *region* is **cn-southwest-228** and *domain* is **cdzs.cn**, the URL of the image is as follows:

**swr.cn-southwest-228.cdzs.cn/deep-learning/mindspore:2.1.1-cann6.3.RC2**. Use the actual site information.

Mo Mo	delArts	×	+	
← →	с	Console cdzs.cn	/modelarts/?agencyld=	3&region=cn-southwest-228

## 11.4.2.1.6 Step 5 Creating and Debugging a Notebook Instance on ModelArts

- 1. Register **the image uploaded to SWR** with ModelArts Image Management.
  - Log in to the ModelArts management console. In the navigation pane on the left, choose **Image Management**. Click **Register** and register the image. The registered image can be used to create notebook instances.
- 2. Use the custom image to create a notebook instance and debug it. After the debugging is successful, save the image.
  - a. For details about how to create a notebook instance using a custom image, see **Using a Custom Image to Create a Notebook Instance**.
  - b. For details about how to save a notebook image, see **Saving a Notebook Environment Image**.
- 3. Create a training job on ModelArts.

### 11.4.2.1.7 Step 6 Creating a Training Job on ModelArts

- Log in to the ModelArts management console. In the navigation pane on the left, choose Training Management > Training Jobs. The training job list is displayed by default.
- 2. On the **Create Training Job** page, configure parameters and click **Submit**.
  - Created By: Custom algorithms
  - Boot Mode: Custom images
  - Image: swr.xxx.xxx.com/deep-learning/mindspore:2.1.1-cann6.3.RC2
  - Code Directory: OBS path to startup scripts, for example, obs://testmodelarts/ascend/demo-code/
  - Boot Command: python \${MA\_JOB\_DIR}/demo-code/run\_ascend/ run\_ascend.py python \${MA\_JOB\_DIR}/demo-code/mindsporeverification.py
  - Resource Pool: Dedicated resource pools
  - **Resource Type**: Ascend with the required driver and firmware version
  - Job Log Path: OBS path to stored training logs, for example, obs://testmodelarts/ascend/log/
- 3. Confirm the configurations of the training job and click **Submit**.
- 4. Wait until the training job is created.

After you submit the job creation request, the system will automatically perform operations on the backend, such as downloading the container image and code directory and running the boot command. A training job requires a certain period of time for running. The duration ranges from dozens of minutes to several hours, varying depending on the service logic and selected resources. After the training job is executed, logs are displayed.

**Figure 11-18** Runtime logs of a training job powered by Ascend resources in a dedicated resource pool

```
75 Ascend Envs
76 -----
77 JOB_ID: modelarts-job-2436291a-8543-4ab8-84ad-2dda8f1e4f5c
78 RANK_TABLE_FILE: /home/ma-user/modelarts/rank_table/jobstart_hccl.json
79 RANK_SIZE: 1
80 ASCEND_DEVICE_ID: 0
81 DEVICE_ID: 0
82 RANK_ID: 0
83 -----
84
   [2. 2. 2. 2.]
85
    [2. 2. 2. 2.]]
86
   [[2. 2. 2. 2.]
87
88
   [2. 2. 2. 2.]
89
    [2. 2. 2. 2.]]
90
    [[2. 2. 2. 2.]
91
92
     [2. 2. 2. 2.]
93
     [2. 2. 2. 2.]]]]
```

## 11.4.3 Preparing a Training Image

## 11.4.3.1 Specifications for Custom Images for Training Jobs

When you use a locally developed model and training script to create a custom image, ensure that the custom image complies with the specifications defined by ModelArts.

## Specifications

- Use Ubuntu 18.04 for custom images to in case versions are not compatible.
- Do not use a custom image larger than 15 GB. The size should not exceed half of the container engine space of the resource pool. Otherwise, the start time of the training job is affected.

The container engine space of ModelArts public resource pool is 50 GB. By default, the container engine space of the dedicated resource pool is also 50 GB. You can customize the container engine space when creating a dedicated resource pool.

- The **uid** of the default user of a custom image must be **1000**.
- The GPU or Ascend driver cannot be installed in a custom image. When you select GPU resources to run training jobs, ModelArts automatically places the GPU driver in the **/usr/local/nvidia** directory in the training environment. When you select Ascend resources to run training jobs, ModelArts automatically places the Ascend driver in the **/usr/local/Ascend/driver** directory.
- x86- or Arm-based custom images can run only with specifications corresponding to their architecture.
  - Run the following command to check the CPU architecture of a custom image:

docker inspect {*Custom image path*} | grep Architecture

The following is the command output for an Arm-based custom image: "Architecture": "arm64"

 If the name of a specification contains Arm, this specification is an Armbased CPU architecture.



 If the name of a specification does not contain Arm, this specification is an x86-based CPU architecture.

```
Instance Flavor
```

GPU: 1\*NVIDIA-V100(16GB) | CPU: 8 vCPUs 64GB 780GB

• ModelArts does not support the download of open source installation packages. Install the dependency packages required by the training job in the custom image.

## 11.4.3.2 Migrating an Image to ModelArts Training

To migrate an image to the training management, perform the following operations:

1. Add the default user group **ma-group** (**gid = 100**) of the training management for the image.

**NOTE** 

If the user group whose **gid** is **100** already exists, the error message "groupadd: GID '100' already exists" may be displayed. You can use the command **cat /etc/group** | **grep 100** to check whether the user group whose GID is 100 exists.

If the user group whose **gid** is **100** already exists, skip this step and delete the command **RUN groupadd ma-group -g 100** from the Dockerfile.

2. Add the default user **ma-user** (**uid = 1000**) of the training management for the image.

If the user whose **uid** is **1000** already exists, the error message "useradd: UID 1000 is not unique" may be displayed. You can use the command **cat /etc/passwd | grep 1000** to check whether the user whose UID is 1000 exists.

If the user whose **uid** is **1000** already exists, skip this step and delete the command **RUN useradd -d /home/ma-user -m -u 1000 -g 100 -s /bin/bash ma-user** from the Dockerfile.

3. Modify the permissions on files in the image to allow **ma-user** whose **uid** is **1000** to read and write the files.

You can modify an image by referring to the following Dockerfile so that the image complies with specifications for custom images of the new-version training management.

FROM {An existing image}

USER root

```
# If the user group whose GID is 100 already exists, delete the groupadd command.
RUN groupadd ma-group -g 100
# If the user whose UID is 1000 already exists, delete the useradd command.
```

RUN useradd -m -d /home/ma-user -s /bin/bash -g 100 -u 1000 ma-user

# Modify the permissions on image files so that user **ma-user** whose UID is 1000 can read and write the files.

RUN chown -R ma-user:100 {Path to the Python software package}

# Configure the preset environment variables of the container image.
# Set PYTHONUNBUFFERED to 1 to prevent log loss.
ENV PYTHONUNBUFFERED=1

# Configure the default user and working directory of the container image. USER ma-user WORKDIR /home/ma-user

After editing the Dockerfile, run the following command to build a new image:

docker build -f Dockerfile . -t {New image}

Upload the new image to SWR. For details, see How Can I Log In to SWR and Upload Images to It?

### 11.4.3.3 Using a Base Image to Create a Training Image

ModelArts provides deep learning-powered base images such as TensorFlow, PyTorch, and MindSpore images. In these images, the software mandatory for running training jobs has been installed. If the software in the base images cannot meet your service requirements, create new images based on the base images and use the new images to create training jobs.

### Procedure

Perform the following operations to create an image using a training base image:

1. Install Docker. If the **docker images** command is executed, Docker has been installed. In this case, skip this step.

The following uses Linux x86\_64 as an example to describe how to obtain the Docker installation package. Run the following command to install Docker: curl -fsSL get.docker.com -o get-docker.sh

sh get-docker.sh

- 2. Create a folder named **context**. mkdir -p context
- 3. Obtain the **pip.conf** file. [global] index-url = https://repo.xxx.com/repository/pypi/simple trusted-host = repo.xxx.com timeout = 120
- Create a new image based on a training base image provided by ModelArts. Save the edited Dockerfile in the context folder. FROM {Path to the training base image provided by ModelArts}

# Configure pip. RUN mkdir -p /home/ma-user/.pip/ COPY --chown=ma-user:ma-group pip.conf /home/ma-user/.pip/pip.conf

# Configure the preset environment variables of the container image.
# Add the Python interpreter path to the PATH environment variable.
# Set PYTHONUNBUFFERED to 1 to prevent log loss.
ENV PATH=\${ANACONDA\_DIR}/envs/\${ENV\_NAME}/bin:\$PATH \
PYTHONUNBUFFERED=1

RUN /home/ma-user/anaconda/bin/pip install --no-cache-dir numpy

- 5. Run the following command in the directory where the Dockerfile is stored to create a container image, for example, **training:v1**: docker build . -t training:v1
- 6. Upload the new image to SWR. For details, see How Can I Log In to SWR and Upload Images to It?.
- 7. Use the custom image to create a training job on ModelArts. For details, see Using a Custom Image to Create a CPU- or GPU-based Training Job.

## 11.4.4 Creating an Algorithm Using a Custom Image

Your locally developed algorithms or algorithms developed using other tools can be uploaded to ModelArts for unified management.

## Entries for Creating an Algorithm

You can create an algorithm using a custom image on ModelArts in either of the following ways:

- Entry 1: On the ModelArts console, choose **Algorithm Management** > **My algorithms**. Then, create an algorithm and use it in training jobs or publish it to Al Hub.
- Entry 2: On the ModelArts console, choose Training Management > Training Jobs, and click Create Training Job to create a custom algorithm and submit a training job. For details, see Using a Custom Image to Create a CPU- or GPU-based Training Job.

## Parameters for creating an algorithm

## Figure 11-19 Creating an algorithm using a custom image

* Boot Mode	Preset image	Custom image	
★ Image			Select
Code Directory			Select
* Boot Command (?)	1		

Table 11-14 Parameters for creating an algorithm

Parameter	Description
Boot Mode	Select <b>Custom images</b> . This parameter is mandatory.

Parameter	Description	
Image	<ul> <li>URL of an SWR image. This parameter is mandatory.</li> <li>Private images or shared images: Click Select on the right to select an SWR image. Ensure that the image has been uploaded to SWR. For details, see How Can I Log In to SWR and Upload Images to It?.</li> </ul>	
	• Public images: You can also manually enter the image path in the format of " <organization image<br="" to="" which="" your="">belongs&gt;/<image name=""/>" on SWR. Do not contain the domain name (swr.<region>.xxx.com) in the path because the system will automatically add the domain name to the path. For example: modelarts-job-dev-image/pytorch_1_8:train-pytorch_1.8.0-cuda_10.2-py_3.7-</region></organization>	
Code Directory	OBS path for storing the training code. This parameter is	
	optional.	
	Take OBS path <b>obs://obs-bucket/training-test/demo-code</b> as an example. The content in the OBS path will be automatically downloaded to <b>\${MA_JOB_DIR}/demo-code</b> in the training container, and <b>demo-code</b> (customizable) is the last-level directory of the OBS path.	
Boot Command	Command for booting an image. This parameter is mandatory. The boot command will be automatically executed after the code directory is downloaded.	
	<ul> <li>If the training boot script is a .py file, train.py for example, the boot command can be python \${MA_JOB_DIR}/demo- code/train.py.</li> </ul>	
	<ul> <li>If the training boot script is an .sh file, main.sh for example, the boot command can be bash \${MA_JOB_DIR}/demo- code/main.sh.</li> </ul>	
	Semicolons (;) and ampersands (&&) can be used to combine multiple boot commands, but line breaks are not supported. <b>demo-code</b> (customizable) in the boot command is the last-level directory of the OBS path.	

## **Configuring Pipelines**

A preset image-based algorithm obtains data from an OBS bucket or dataset for model training. The training output is stored in an OBS bucket. The input and output parameters in your algorithm code must be parsed to enable data exchange between ModelArts and OBS. For details about how to develop code for training on ModelArts, see **Developing a Custom Script**.

When you use a preset image to create an algorithm, configure the input and output pipelines.

• Input configurations

Table 11-15	Input	configurations
-------------	-------	----------------

Paramete r	Description
Parameter Name	Set the name based on the data input parameter in your algorithm code. The code path parameter must be the same as the training input parameter parsed in your algorithm code. Otherwise, the algorithm code cannot obtain the input data. For example, If you use <b>argparse</b> in the algorithm code to parse <b>data_url</b> into the data input, set the data input parameter to <b>data_url</b> when creating the algorithm.
Descriptio n	Customizable description of the input parameter,
Obtained from	Source of the input parameter. You can select Hyperparameters (default) or Environment variables.
Constraint s	Whether data is obtained from a storage path or ModelArts dataset.
	following constraints are added:
	• Labeling Type: For details, see Creating a Labeling Job.
	<ul> <li>Data Format, which can be Default, CarbonData, or both.</li> <li>Default indicates the manifest format.</li> </ul>
	<ul> <li>Data Segmentation: available only for image classification, object detection, text classification, and sound classification datasets.</li> <li>Possible values are Segmented dataset, Dataset not segmented, and Unlimited. For details, see Publishing a Data Version.</li> </ul>
Add	Multiple data input sources are allowed.

## • Output configurations

 Table 11-16 Output configurations

Parameter	Description
Parameter Name	Set the name based on the data output parameter in your algorithm code. The code path parameter must be the same as the training output parameter parsed in your algorithm code. Otherwise, the algorithm code cannot obtain the output path.
	For example, If you use <b>argparse</b> in the algorithm code to parse <b>train_url</b> into the data output, set the data output parameter to <b>train_url</b> when creating the algorithm.
Descriptio n	Customizable description of the output parameter,

Parameter	Description
Obtained from	Source of the output parameter. You can select Hyperparameters (default) or Environment variables.
Add	Multiple data output paths are allowed.

## **Defining Hyperparameters**

When you use a preset image to create an algorithm, ModelArts allows you to customize hyperparameters so you can view or modify them anytime. After the hyperparameters are defined, they are displayed in the startup command and transferred to your boot file as CLI parameters.

1. Import hyperparameters.

You can click Add hyperparameter to manually add hyperparameters.

2. Edit hyperparameters.

For details, see **Table 11-17**.

### Table 11-17 Hyperparameters

Parame ter	Description
Name	Hyperparameter name
	Enter 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Туре	Type of the hyperparameter, which can be <b>String</b> , <b>Integer</b> , <b>Float</b> , or <b>Boolean</b>
Default	Default value of the hyperparameter, which is used for training jobs by default
Constrai nts	Click <b>Restrain</b> . Then, set the range of the default value or enumerated value in the dialog box displayed.
Require	Select <b>Yes</b> or <b>No</b> .
d	• If you select <b>No</b> , you can delete the hyperparameter on the training job creation page when using this algorithm to create a training job.
	• If you select <b>Yes</b> , you cannot delete the hyperparameter on the training job creation page when using this algorithm to create a training job.
Descript	Description of the hyperparameter
ion	Only letters, digits, spaces, hyphens (-), underscores (_), commas (,), and periods (.) are allowed.

## **Adding Training Constraints**

You can add training constraints of the algorithm based on your needs.

- **Resource Type**: Select the required resource types.
- Multicard Training: Choose whether to support multi-card training. •
- Distributed Training: Choose whether to support distributed training.

## **Runtime Environment Preview**

in the lower

When creating an algorithm, click the arrow on right corner of the page to know the path of the code directory, boot file, and input and output data in the training container.

## Follow-Up Procedure

After an algorithm is created, use it to create a training job. For details, see Using a Custom Image to Create a CPU- or GPU-based Training Job.

## 11.4.5 Using a Custom Image to Create a CPU- or GPU-based **Training Job**

Model training is an iterative optimization process. Through unified training management, you can flexibly select algorithms, data, and hyperparameters to obtain the optimal input configuration and model. After comparing metrics between job versions, you can determine the most satisfactory training job.

## Prerequisites

- The data to be trained has been uploaded to an OBS directory.
- At least one empty folder for storing the training output has been created in • OBS.
- A custom image has been created based on ModelArts specifications. For details about the custom image specifications, see **Specifications for Custom Images for Training Jobs.**
- The custom image has been uploaded to SWR. For details, see How Can I Log In to SWR and Upload Images to It?.

## **Creating a Training Job**

- 1. Log in to the ModelArts management console. In the left navigation pane, choose Training Management > Training Jobs.
- Click **Create Training Job** and set parameters. 2.

Parameter	Description
Algorithm Type	Select <b>Custom algorithm</b> . This parameter is mandatory.

**Table 11-18** Creating a training job using a custom image

Parameter	Description	
Boot Mode	Select <b>Custom image</b> . This parameter is mandatory.	
Image	<ul> <li>Container image path. This parameter is mandatory. You can set the container image path in either of the following ways:</li> <li>To select your image or an image shared by others, click Select on the right and select a container image for training. The required image must be uploaded to SWR beforehand.</li> <li>To select a public image, enter the address of the public image in SWR. Enter the image path in the format of "Organization name/Image name:Version name". Do not contain the domain name (swr.<region>.xxx.com) in the path because the system will automatically add the domain name to the path. For example, if the SWR address of a public image is swr.<region>.xxx.com/test-image/ tensorflow2_1_1:1.1.1.</region></region></li> </ul>	
Code Directory	<ul> <li>Select the OBS directory where the training code file is stored. If the custom image does not contain training code, you need to set this parameter. If the custom image contains training code, you do not need to set this parameter.</li> <li>Upload code to the OBS bucket beforehand. The total size of files in the directory cannot exceed 5 GB, the number of files cannot exceed 1000, and the folder depth cannot exceed 32.</li> <li>The training code file is automatically downloaded to the \${MA_JOB_DIR}/demo-code directory of the training container when the training job is started. demo-code is the last-level OBS directory for storing the code. For example, if Code Directory is set to /test/code, the training code file is downloaded to the \${MA_JOB_DIR}/code directory of the training code file is downloaded to the \${MA_JOB_DIR}/code directory of the training code file is downloaded to the \${MA_JOB_DIR}/code directory of the training container.</li> </ul>	
User ID	User ID for running the container. The default value 1000 is recommended. If the UID needs to be specified, its value must be within the specified range. The UID ranges of different resource pools are as follows: • Public resource pool: 1000 to 65535	
	Dedicated resource pool: 0 to 65535	

Parameter	Description	
Boot Command	Command for booting an image. This parameter is mandatory.	
	When a training job is running, the boot command is automatically executed after the code directory is downloaded.	
	• If the training boot script is a .py file, <b>train.py</b> for example, the boot command is as follows. python \${MA_JOB_DIR}/demo-code/train.py	
	<ul> <li>If the training boot script is a .sh file, main.sh example, the boot command is as follows. bash \${MA_JOB_DIR}/demo-code/main.sh</li> </ul>	
	You can use semicolons (;) and ampersands (&&) to combine multiple commands. <b>demo-code</b> in the command is the last-level OBS directory where the code is stored. Replace it with the actual one.	
Local Code Directory	Specify the local directory of a training container. When a training starts, the system automatically downloads the code directory to this directory.	
	The default local code directory is <b>/home/ma-user/</b> <b>modelarts/user-job-dir</b> . This parameter is optional.	
Work Directory	During training, the system automatically runs the <b>cd</b> command to execute the boot file in this directory.	

Table 11	-19	Parameters	for	creating	а	training	iob	
14010 1		i arameters		ereating	~	ananng	,00	

Paramet er	Sub- Paramet er	Description
Input	Paramete r	The algorithm code reads the training input data based on the input parameter name.
		Set this parameter to <b>data_url</b> , which is the same as the parameter for parsing the input data in the training code. You can set multiple training input parameters. The name of each training input parameter must be unique.
		For example, if you use <b>argparse</b> in the training code to parse <b>data_url</b> into the data input, set the parameter name of the training input to <b>data_url</b> . import argparse # Create a parsing task.
		formatter_class=argparse.ArgumentDefaultsHelpFormatter) # Add parameters. parser.add argument('train_url', type=str, help='the path model
parser.add_argument saved') parser.add_argument # Parse the parametr args, unknown = par		saved') parser.add_argument('data_url', type=str, help='the training data') # Parse the parameters. args, unknown = parser.parse_known_args()
	Dataset	Click <b>Dataset</b> and select the target dataset and its version in the ModelArts dataset list.
When the tr automatical to the traini		When the training job is started, ModelArts automatically downloads the data in the input path to the training container.
		<b>NOTE</b> ModelArts data management is being upgraded and is invisible to users who have not used data management. It is recommended that new users store their training data in OBS buckets.
	Data path	Click <b>Data path</b> and select the storage path to the training input data from an OBS bucket.
		When the training job is started, ModelArts automatically downloads the data in the input path to the training container.
	How to Obtain	The following uses training input <b>data_path</b> as an example.
		<ul> <li>If you select Hyperparameters, use this code to obtain the data: import argparse parser = argparse.ArgumentParser() parser.add_argument('data_path') args, unknown = parser.parse_known_args() data_path = args.data_path</li> </ul>
		<ul> <li>If you select Environment variables, use this code to obtain the data: import os data_path = os.getenv("data_path", "")</li> </ul>

Paramet er	Sub- Paramet er	Description
Output	Paramete r	The algorithm code reads the training output data based on the output parameter name. Set this parameter to <b>train_url</b> , which is the same as the parameter for parsing the output data in the training code. You can set multiple training output parameters. The name of each training output parameter must be unique.
	Data path	Click <b>Data path</b> and select the storage path to the training output data from an OBS bucket. During training, the system automatically synchronizes files from the local code directory of the training container to the data path. <b>NOTE</b> The data path can only be an OBS path. To prevent any issues with data storage, choose an empty directory as the data path.
	How to Obtain	<ul> <li>The following uses the training output train_url as an example.</li> <li>If you select Hyperparameters, use this code to obtain the data: <ul> <li>import argparse</li> <li>parser = argparse.ArgumentParser()</li> <li>parser.add_argument('train_url')</li> <li>args, unknown = parser.parse_known_args()</li> <li>train_url = args.train_url</li> </ul> </li> <li>If you select Environment variables, use this code to obtain the data: <ul> <li>import os</li> <li>train_url = os.getenv("train_url", "")</li> </ul> </li> </ul>
	Predownl oad	<ul> <li>Indicates whether to pre-download the files in the output directory to a local directory.</li> <li>If you set Predownload to No, the system does not download the files in the training output data path to a local directory of the training container when the training job is started.</li> <li>If you set Predownload to Yes, the system automatically downloads the files in the training output data path to a local directory of the training output data path to a local directory of the training container when the training job is started.</li> <li>If you set Predownload to Yes, the system automatically downloads the files in the training output data path to a local directory of the training container when the training job is started. The larger the file size, the longer the download time. To avoid excessive training time, remove any unneeded files from the local code directory of the training container as soon as possible. To use resumable training and incremental training, Download must be selected.</li> </ul>

Paramet er	Sub- Paramet er	Description
Hyperpar ameters	-	Used for training tuning. This parameter is optional.
Environm ent Variable	-	Add environment variables based on service requirements. For details about preset environment variables in the training container, see <b>Viewing</b> <b>Environment Variables of a Training Container</b> .
Auto Restart	-	Number of retries for a failed training job. If this parameter is enabled, a failed training job will be automatically re-delivered and run. On the training job details page, you can view the number of retries for a failed training job.
		• This function is disabled by default.
		<ul> <li>If you enable this function, set the number of retries. The value ranges from 1 to 3 and cannot be changed.</li> </ul>

- 3. Select an instance flavor. The value range of the training parameters is consistent with the constraints of existing custom images. Select a public resource pool or dedicated resource pool based on your needs. For details about the parameters, see **Creating a Training Job**.
- 4. Click **Submit** to create the training job.

It takes a period of time to create a training job.

To view the real-time status of a training job, go to the training job list and click the name of the training job. On the training job details page that is displayed, view the basic information of the training job. For details, see **Viewing Training Job Details**.

# 11.4.6 Using a Custom Image to Create an Ascend-based Training Job

If preset Ascend images cannot meet your requirements, you can build a custom image and use it to create a training job. The main process of using a custom image to create a CPU- or GPU-based training job is the same as that of using a custom image to create an Ascend-based training job. Pay attention to the following differences:

• Ascend HCCL RANK\_TABLE\_FILE

Ascend HCCL RANK\_TABLE\_FILE provides the cluster used by Ascend distributed training jobs. It is used for distributed communication between Ascend chips and can be parsed by the NVIDIA Collective Communication Library (NCCL). There are two templates for the file format, template 1 and template 2. ModelArts provides the template 2 format. For details about the complete Ascend HCCL RANK\_TABLE\_FILE, see Resource Configuration Files for the Ascend AI Processor.

The **Ascend HCCL RANK\_TABLE\_FILE** file used in the ModelArts training environment is **jobstart\_hccl.json**. **Table 11-20** lists the file parameters.

 Example of the jobstart\_hccl.json file content in the ModelArts training environment (template 2)

```
{
    "group_count": "1",
    "group_list": [{
        "device_count": "1",
        "group_name": "job-trainjob",
        "instance_count": "1",
        "instance_list": [{
            "devices": [{
               "device_id": "4",
               "device_ip": "192.1.10.254"
            }],
            "pod_name": "jobxxxxxxx-job-trainjob-0",
            "server_id": "192.168.0.25"
        }]
    }],
    "status": "completed"
}
```

In **jobstart\_hccl.json**, the **status** value may not be **completed** when the training script is started. In this case, wait until the **status** value changes to **completed** and read the remaining content of the file.

If you want to use the **jobstart\_hccl.json** file in template 1 format, use the training script to convert the **jobstart\_hccl.json** file in template 2 format to the **jobstart\_hccl.json** file in template 1 format after the **status** value changes to **completed**.

Format of the jobstart\_hccl.json file after format conversion (template 1)

```
{
    "server_count": "1",
    "server_list": [{
        "device": [{
            "device_id": "4",
            "device_ip": "192.1.10.254",
            "rank_id": "0"
        }],
        "server_id": "192.168.0.25"
    }],
    "status": "completed",
    "version": "1.0"
}
```

RANK\_TABLE\_FILE

 Table 11-20 Environment variables

Environment Variable	Description
RANK_TABLE_FI LE	Directory of <b>Ascend HCCL RANK_TABLE_FILE</b> , which is <b>/</b> user/config.
	Obtain the file using <b>\${RANK_TABLE_FILE}/</b> jobstart_hccl.json.

# **11.4.7 Troubleshooting Process**

## Symptom

A training job using a custom image failed.

## **Locating Method**

- 1. Determine the image source.
  - Check whether the base image of the custom image is from ModelArts.
     Use a base image provided by ModelArts to create a custom image. For details, see .
  - If the image is from a third party, check with the creator of the custom image for how to use this image.
- 2. Determine the size of the custom image.

Do not use a custom image larger than 15 GB. The size should not exceed half of the container engine space of the resource pool. Otherwise, the start time of the training job is affected.

The container engine space of ModelArts public resource pool is 50 GB. By default, the container engine space of the dedicated resource pool is also 50 GB. You can customize the container engine space when creating a dedicated resource pool.

- 3. Determine the error type.
  - If an error message is displayed indicating that a file could not be found, see .
  - If an error message is displayed indicating that a package could not be found, see .
  - An error occurred in the Ascend startup script or initialization script.

Check whether the script is obtained from the official website and whether the script is used strictly following the instructions provided in official documents. For example, check whether the script name and path are correct.

- The driver version is incompatible with the underlying driver.

Before upgrading the driver of a custom image, check whether the upgraded version is supported by the underlying driver.

- You are not allowed to access a file.

The possible cause is that the user of the custom image is different from that of the job container. In this case, modify the Dockerfile.

```
RUN if id -u ma-user > /dev/null 2>&1 ; \
then echo 'The ModelArts user already exists.' ; \
else echo 'The ModelArts user does not exist.' && \
groupadd ma-group -g 1000 && \
useradd -d /home/ma-user -m -u 1000 -g 1000 -s /bin/bash ma-user ; fi && \
chmod 770 /home/ma-user && \
chmod 770 /root && \
usermod -a -G root ma-user
```

- For other issues, search for solutions in .

## **Summary and Suggestions**

Before using a custom image for training jobs, create the image by following the . which also provides end-to-end examples for your reference.

# 11.5 Using a Custom Image to Create AI applications for Inference Deployment

# 11.5.1 Custom Image Specifications for Creating AI Applications

When building a custom image using a locally developed model, ensure that the image complies with ModelArts specifications.

- No malicious code is allowed.
- The size of a custom image cannot exceed 30 GB.
- External APIs

Set the external service API for a custom image. The inference API must be the same as the URL defined by **apis** in **config.json**. Then, the external service API can be directly accessed when the image is started. The following is an example of accessing an MNIST image. The image contains a model trained using an MNIST dataset and can identify handwritten digits. **listen\_ip** indicates the container IP address. You can start a custom image to obtain the container IP address from the container.

- Sample request curl -X POST \ http://{*Listening IP address*}:8080/ \ -F images=@seven.jpg

Figure	11-20	Example	e of o	obtaini	ing	listen_	ip

```
root@leal140116_0:/# cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
169.254.30.2 d6211431d0e3
```

- Sample response
   {"mnist\_result": 7}
- (Optional) Health check API

If services must not be interrupted during a rolling upgrade, the health check API must be configured in **config.json** for ModelArts. The health check API returns the healthy state for a service when the service is running properly or an error when the service becomes faulty.

### NOTICE

The health check API must be configured for a hitless rolling upgrade.

The following shows a sample health check API:

- URI
- GET /health - Sample request: curl -X GET \ http://*{Listening IP address}*:8080/health
- Sample response {"health": "true"}
- Status code

### Table 11-21 Status code

Status Code	Message	Description
200	ОК	Request sent

### • Log file output

Configure standard output so that logs can be properly displayed.

### • Image boot file

To deploy a batch service, set the boot file of an image to **/home/run.sh** and use CMD to set the default boot path. The following is a sample Dockerfile:

CMD ["sh", "/home/run.sh"]

### • Image dependencies

To deploy a batch service, install dependency packages such as Python, JRE/ JDK, and ZIP in the image.

### • (Optional) Hitless rolling upgrade

To ensure that services are not interrupted during a rolling upgrade, set HTTP **keep-alive** to **200**. For example, Gunicorn does not support keep-alive by default. To ensure a hitless rolling upgrade, install Gevent and configure -- **keep-alive 200 -k gevent** in the image. The parameter settings vary depending on the service framework. Set the parameters as required.

### (Optional) Gracefully exiting a container

To ensure that services are not interrupted during a rolling upgrade, the system must capture SIGTERM signals in the container and wait for 60s before gracefully exiting the container. If the duration is less than 60s before the graceful exiting, services may be interrupted during the rolling upgrade. To ensure uninterrupted service running, the system exits the container after the system receives SIGTERM signals and processes all received requests. The whole duration is not longer than 90s. The following shows example **run.sh**:

```
#!/bin/bash
gunicorn_pid=""
handle_sigterm() {
    echo "Received SIGTERM, send SIGTERM to $gunicorn_pid"
    if [ $gunicorn_pid != "" ]; then
        sleep 60
        kill -15 $gunicorn_pid # Transfer SIGTERM signals to the Gunicorn process.
        wait $gunicorn_pid # Wait until the Gunicorn process stops.
    fi
}
```

trap handle\_sigterm TERM

# 11.5.2 Creating a Custom Image and Using It to Create an AI Application

If you want to use an AI engine that is not supported by ModelArts, create a custom image for the engine, import the image to ModelArts, and use the image to create AI applications. This section describes how to use a custom image to create an AI application and deploy the application as a real-time service.

The process is as follows:

- 1. **Building an Image Locally**: Create a custom image package locally. For details, see **Custom Image Specifications for Creating AI Applications**.
- 2. Verifying the Image Locally and Uploading It to SWR: Verify the APIs of the custom image and upload the custom image to SWR.
- 3. Using the Custom Image to Create an AI Application: Import the image to ModelArts AI application management.
- 4. **Deploying the AI Application as a Real-Time Service**: Deploy the model as a real-time service.

## **Building an Image Locally**

This section uses a Linux x86\_x64 host as an example. You can use an existing local host to create a custom image.

- 1. After logging in to the host, install Docker. For details, see **Docker official documents**. Alternatively, run the following commands to install Docker: curl -fsSL get.docker.com -o get-docker.sh sh get-docker.sh
- 2. Obtain the base image. Ubuntu 18.04 is used in this example. docker pull ubuntu:18.04
- 3. Create the **self-define-images** folder, and edit **Dockerfile** and **test\_app.py** in the folder for the custom image. In the sample code, the application code runs on the Flask framework.

```
The file structure is as follows:
self-define-images/
  --Dockerfile
  --test_app.py
     Dockerfile
     From ubuntu:18.04
     # Configure the source and install Python, Python3-PIP, and Flask.
     RUN cp -a /etc/apt/sources.list /etc/apt/sources.list.bak && \
      sed -i "s@http://.*security.ubuntu.com@http://repo.xxx.com@g" /etc/apt/sources.list && \
      sed -i "s@http://.*archive.ubuntu.com@http://repo.xxx.com@g" /etc/apt/sources.list && \
      apt-get update && \
      apt-get install -y python3 python3-pip && \
      pip3 install --trusted-host https://repo.xxx.com -i https://repo.xxx.com/repository/pypi/simple
     Flask
     # Copy the application code to the image.
     COPY test_app.py /opt/test_app.py
     # Specify the boot command of the image.
     CMD python3 /opt/test_app.py
     test_app.py
     from flask import Flask, request
     import json
     app = Flask(__name__)
```

```
@app.route('/greet', methods=['POST'])
def say_hello_func():
  print("------ in hello func ------")
  data = json.loads(request.get_data(as_text=True))
  print(data)
  username = data['name']
  rsp_msg = 'Hello, {}!'.format(username)
  return json.dumps({"response":rsp_msg}, indent=4)
@app.route('/goodbye', methods=['GET'])
def say_goodbye_func():
  print("------ in goodbye func ------")
  return '\nGoodbye!\n'
@app.route('/', methods=['POST'])
def default_func():
  print("------ in default func ------")
  data = json.loads(request.get_data(as_text=True))
  return '\n called default func !\n {} \n'.format(str(data))
# host must be "0.0.0.0", port must be 8080
if __name__ == '__main__':
  app.run(host="0.0.0.0", port=8080)
```

- Switch to the self-define-images folder and run the following command to create custom image test:v1: docker build -t test:v1.
- 5. Run **docker images** to view the custom image you have created.

## Verifying the Image Locally and Uploading It to SWR

 Run the following command in the local environment to start the custom image: docker run -it -p 8080:8080 test:v1

Figure 11-21 Starting a custom image



2. Open another terminal and run the following commands to test the functions of the three APIs of the custom image:

curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/ curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet curl -X GET 127.0.0.1:8080/goodbye

If information similar to the following is displayed, the function verification is successful.

Figure 11-22 Testing API functions



- 3. Upload the custom image to SWR. For details, see How Can I Upload Images to SWR?
- 4. View the uploaded image on the **My Images** > **Private Images** page of the SWR console.

### Figure 11-23 Uploaded images

My Ir	nages 💮		
	Private Images Shared Images		
	1 Delete		Display only s
	Image J≡	Organization	Tags J
	canary-provider-beta	hwstaff_pub_cbuinfo	1
	test	deep-learning	1

## Using the Custom Image to Create an AI Application

Import a meta model. For details, see Creating and Importing a Model Image. Key parameters are as follows:

- Meta Model Source: Select Container image.
  - **Container Image Path**: Select the created private image.

### Figure 11-24 Created private image

Select Image		,
My Images		С
My Images		Enter a name. Q
Name	Updated ↓Ξ	Organization Versions
log ggho		3

- **Container API**: Protocol and port number for starting a model. Ensure \_ that the protocol and port number are the same as those provided in the custom image.
- **Image Replication**: indicates whether to copy the model image in the container image to ModelArts. This parameter is optional.
- Health Check: checks health status of a model. This parameter is optional. This parameter is configurable only when the health check API is configured in the custom image. Otherwise, creating the AI application will fail
- **APIs**: APIs of a custom image. This parameter is optional. The model APIs must comply with ModelArts specifications. For details, see Specifications for Editing a Model Configuration File.

The configuration file is as follows:

```
[{
     "url": "/".
     "method": "post",
     "request": {
        "Content-type": "application/json"
     },
     "response": {
        "Content-type": "application/json"
     }
  },
```

{

```
"url": "/greet",
"method": "post",
"request": {
"Content-type": "application/json"
},
"response": {
"Content-type": "application/json"
}
},
{
"url": "/goodbye",
"method": "get",
"request": {
"Content-type": "application/json"
},
"response": {
"Content-type": "application/json"
}
}
```

## Deploying the AI Application as a Real-Time Service

- 1. Deploy the AI application as a real-time service. For details, see **Deploying as** a **Real-Time Service**.
- 2. View the details about the real-time service.

### Figure 11-25 Usage Guides

<   Back to Real-Time Service List	Modify Start Stop Delete C
Basic Information	
Name service-e7ff Serv	srvice ID
Status O Running(59 minutes until stop) Ø Sou	surce My Deployment
Falled Cally/Real Des Cells Des	scatption 🖉
Custom Settings Trad	affic Limit 100
Adamod Log  The Adamod Log  Th	
Usage Guides Prediction Configuration Updates Monitoring Events Logs Tags	
API UKL https://eef9fbx2943844bd998010dbx we_ 0	Note: AK/SK or token authentication is supported. APP Reference
Al Application model.	
v Post /	
V POST /greet	G
✓ GET /goodbye	

3. Access the real-time service on the **Prediction** tab page.

### Figure 11-26 Accessing a real-time service

Usage Guides	Prediction	Configuration Updates	Monitoring	Events	Logs	Tags			
Request Path	/	•	d 9 MP. Otherwise	the request will	ba intercent	ad Partrictions on t	he Request Rody Size fo	r Condeo	Dradition
The size of the rec	uest body to be upo	baded for prediction carinot excee	u o Mb. Outerwise, i	the request with	be intercept	eu. Restrictions on t	The Request Body Size to	Service	Prediction.
Inference Code								Respo	nse
1 ("nune": ")	on"}							1 2 3 4	called default func   {'name': 'Tom'}
Predict									

 $\times$ 

# 11.6 FAQs

## 11.6.1 How Can I Log In to SWR and Upload Images to It?

This section describes how to log in to SWR and upload images to it.

## Step 1 Log In to SWR

1. Log in to the SWR console and select the target region.

### Figure 11-27 SWR console

SWR	Dashboard ⑦		+ Create Organi.	+ Create Organization		
Dashboard My Images Image Resources Organization Management	Getting Started	2 Upload Image	Create Application	Add Trigger	Quick Start Guide Creating an Organization Uptoading an Image Creating an Image Package Adding Permissions	
	Create an organization to which you can upload images and manage them. Create Organization	Upload a local image to the organization or use a public image. Upload Local Image   Use Public Image	Use CCE to create an application from the image. Use CCE to Create Application	Add triggers in SWR to automatically update the application when the image is updated. Add Trigger		

2. Click **Create Organization** in the upper right corner and enter an organization name to create an organization. **deep-learning** is used as an example. Replace it in subsequent commands with the actual organization name.

Figure 11-28 Creating an organization

### Create Organization

1 You can create 5 more organizations.					
Each organization     You can create 5 o     For centralized ma     department, or indiv     Examples     Company or de     Person: john	name must be globally unique. organizations. inagement of images, limit each organization to one company idual. ipartment: cloud-hangzhou or cloud-develop				
Organization Name	deep-learning OK Cancel				

3. Click **Generate Login Command** in the upper right corner to obtain a login command.
#### Figure 11-29 Login Command

Command	×
Learn how to obtain a login command that has long-term validity.	
docker login -u @	
a80dedd swr	
Valid Until: Aug 31, 2022 11:24:16 GMT+08:00	

Log in to the ECS as user **root** and enter the login command. 4.

Figure 11-30 Login command executed on the ECS

r	root@djy-ubuntu1804-cpu:~# docker login -u ( 4@YX2DYVXMJTSMBGJSIGUE -p 400000000000000000000000000000000000	
W	WARNING! Usingpassword via the CLI is insecure. Usepassword-stdin.	
W	WARNING! Your password will be stored unencrypted in /root/.docker/config.json.	
C	Configure a credential helper to remove this warning. See	
h	https://docs.docker.com/engine/reference/commandline/login/#credentials-store	

#### Step 2 Upload Images to SWR

This section describes how to upload an image to SWR.

1. Log in to SWR and tag the image to be uploaded. Replace the organization name **deep-learning** in the following command with the actual organization name obtained in step 1.

sudo docker tag tf-1.13.2:latest swr.xxx.com/deep-learning/tf-1.13.2:latest

2. Run the following command to upload the image: sudo docker push swr.xxx.com/deep-learning/tf-1.13.2:latest

Figure 11-31 Uploading an image

<pre>rootlecs-7918:~# sudo docker tag tf-1.13.2:latest swr.</pre> .com/deep-learning/tf-1.13.2:latest
root@ecs-7918:~# sudo docker push swr
The push refers to repository [swr/deep-learning/tf-1.13.2]
c554b77ee0db: Pushed
902871f33e88: Pushed
83ade5a612e2: Pushed
20cfaf8c1ab8: Pushed
bf7955efefcb: Pushed
af6a5fe577ce: Pushed
f4c051ffa5f2: Pushed
184f790bb501: Pushed
dfbea9e01449: Pushed
f0193b2fb026: Pushed
f98177ec269a: Pushed
81e535525773: Pushed
582ab80c9f26: Pushed
4e3516398cef: Pushed
52ad947270f1: Pushed
dd841c774a30: Pushed
37b9a4b22186: Pushed
e0b3afb09dc3: Pushed
6c01b5a53aac: Pushed
2c6ac8e5063e: Pushed
cc967c529ced: Pushed

3. After the image is uploaded, choose My Images in navigation pane on the left of the SWR console to view the uploaded custom images.

#### Figure 11-32 Uploaded custom image

SWR	My Images 📀		
Dashboard My Images	Private Images Shared Images		
Image Resources -	Delete		
Organizations	Name J≡		
Experience Center	pytorci		

swr.xxx.com/deep-learning/tf-1.13.2:latest is the SWR URL of the custom image.

# 11.6.2 How Do I Configure Environment Variables for an Image?

In a Dockerfile, use the ENV instruction to configure environment variables. For details, see **Dockerfile reference**.

# 11.6.3 How Do I Use Docker to Start an Image Saved Using a Notebook Instance?

An image saved using a notebook instance contains the **Entrypoint** parameter, as shown in **Entrypoint**. The executable file or command specified in the **Entrypoint** parameter overwrites the default boot command of the image. The command input in the **Entrypoint** parameter is not preset in the image. When you run **docker run** in the local environment to start the image, an error message is displayed, indicating that the container creation task fails because the boot file or directory is not found, as shown in **Figure 11-34**.

To avoid this error, configure the --entrypoint parameter to overwrite the program specified in Entrypoint. Use the boot file or command specified by the -- entrypoint parameter to start the image. Example:

docker run -it -d --entrypoint /bin/bash image:tag

Figure 11-33 Entrypoint





# 11.6.4 How Do I Configure a Conda Source in a Notebook Development Environment?

You can install the development dependencies in Notebook as you need. Package management tools pip and Conda can be used to install regular dependencies. The pip source has been configured and can be used for installation, while the Conda source requires further configuration.

This section describes how to configure the Conda source on a notebook instance.

#### **Configuring the Conda Source**

The Conda software has been preset in images. For details, see https://mirror.tuna.tsinghua.edu.cn/help/anaconda/.

#### **Common Conda Commands**

For details about all Conda commands, see **Conda official documents**. The following table lists only common commands.

Descripti on	Command
Obtain online help.	condahelp conda updatehelp # Obtain help for a command, for example, <b>update</b> .
View the Conda version.	conda -V
Update Conda.	conda update conda # Update Conda. conda update anaconda # Update Anaconda.
Manage environm ents.	conda env list # Show all virtual environments. conda info -e # Show all virtual environments. conda create -n myenv python=3.7 # Create an environment named <b>myenv</b> with Python version <b>3.7</b> . conda activate myenv # Activate the <b>myenv</b> environment. conda deactivate # Disable the current environment. conda remove -n myenvall # Delete the <b>myenv</b> environment. conda create -n newnameclone oldname # Clone the old environment to the new environment.

Table 11-22 Common Conda commands

Descripti on	Command
Manage packages.	<ul> <li>conda list # Check the packages that have been installed in the current environment.</li> <li>conda list -n myenv # Specify the packages installed in the myenv environment.</li> <li>conda search numpy # Obtain all information of the numpy package.</li> <li>conda search numpy=1.12.0info # View the information of NumPy 1.12.0.</li> <li>conda install numpy pandas # Concurrently install the NumPy and Pandas packages.</li> <li>conda install numpy=1.12.0 # Install NumPy of a specified version.</li> <li># The install, update, and remove commands use -n to specify an environment, and the install and update commands use -c to specify a source address.</li> <li>conda install -n myenv numpy # Install the numpy package in the myenv environment.</li> <li>conda install -c https://conda.anaconda.org/anaconda numpy # Install NumPy using https://conda.anaconda.org/anaconda.</li> <li>conda update numpy pandas # Concurrently update the NumPy and Pandas packages.</li> <li>conda updateall # Update all packages in the current environment.</li> </ul>
Clear Conda.	conda clean -p  # Delete useless packages. conda clean -t  # Delete compressed packages. conda clean -yall # Delete all installation packages and clear caches.

#### Saving as an Image

After installing the external libraries, save the environment using the image saving function provided by ModelArts notebook of the new version. You can save a running notebook instance as a custom image with one click for future use. After the dependency packages are installed on a notebook instance, it is a good practice to save the instance as an image to prevent the dependency packages from being lost. For details, see **Saving a Notebook Environment Image**.

# 11.6.5 What Are Supported Software Versions for a Custom Image?

If your custom image uses software libraries such as NCCL, CUDA, and OFED, ensure that the software libraries meet the following version requirements:

- NCCL 2.7.8 or later
- OFED MLNX\_OFED\_LINUX-5.4-3.1.0.0 or later
- The CUDA version needs to be adapted to the GPU driver version of the dedicated resource pool. To obtain the GPU driver version, go to the dedicated resource pool details page.

# **12** Permissions Management

## **12.1 Basic Concepts**

ModelArts allows you to configure fine-grained permissions for refined management of resources and permissions. This is commonly used by large enterprises, but it is complex for individual users. It is recommended that individual users configure permissions for using ModelArts by referring to Assigning Permissions to Individual Users for Using ModelArts.

#### 

If you meet any of the following conditions, read this document.

- You are an enterprise user, and
  - There are multiple departments in your enterprise, and you need to control users' permissions so that users in different departments can access only their dedicated resources and functions.
  - There are multiple roles (such as administrators, algorithm developers, and application O&M personnel) in your enterprise. You need them to use only specific functions.
  - There are logically multiple environments (such as the development environment, pre-production environment, and production environment) and are isolated from each other. You need to control users' permissions on different environments.
  - You need to control permissions of specific IAM user or user group.
- You are an individual user, and you have created multiple IAM users. You need to assign different ModelArts permissions to different IAM users.
- You need to understand the concepts and operations of ModelArts permissions management.

ModelArts uses Identity and Access Management (IAM) for most permissions management functions. Before reading below, learn about *Basic Concepts*. This helps you better understand this document.

To implement fine-grained permissions management, ModelArts provides permission control, agency authorization, and workspace. The following describes the details.

#### **ModelArts Permissions and Agencies**



#### Figure 12-1 Permissions management

Exposed ModelArts functions are controlled through IAM permissions. For example, if you as an IAM user need to create a training job on ModelArts, you must have the **modelarts:trainJob:create** permission. For details about how to assign permissions to a user (you need to add the user to a user group and then assign permissions to the user group), see *Permissions Management*.

ModelArts must access other services for AI computing. For example, ModelArts must access OBS to read your data for training. For security purposes, ModelArts must be authorized to access other cloud services. This is agency authorization.

The following summarizes permissions management:

- Your access to any cloud service is controlled through IAM. You must have the permissions of the cloud service. (The required service permissions vary depending on the functions you use.)
- To use ModelArts functions, you need to grant permissions through IAM.
- ModelArts must be authorized by you to access other cloud services for Al computing.

#### **ModelArts Permissions Management**

By default, new IAM users do not have any permissions assigned. You need to add the user to a user group and grant the user group with policies, so that the users in the group can inherit the permissions. After authorization, users can perform operations on ModelArts based on permissions.

#### 

ModelArts is a project-level service deployed and accessed in specific physical regions. When you authorize an agency, you can set the scope for the permissions you select to all resources, enterprises projects, or region-specific projects. If you specify region-specific projects, the selected permissions will be applied to resources in these projects.



When assigning permissions to a user group, IAM does not directly assign specific permissions to the user group. Instead, IAM needs to add the permissions to a policy and then assign the policy to the user group. To facilitate user permissions management, each cloud service provides some preset policies for you to directly use. If the preset policies cannot meet your requirements of fine-grained permissions management, you can customize policies.

 Table 12-1 lists all the preset system-defined policies supported by ModelArts.

Policy	Description	Туре
ModelArts FullAccess	Administrator permissions for ModelArts. Users granted these permissions can operate and use ModelArts.	System-defined policy
ModelArts CommonOperations	Common user permissions for ModelArts. Users granted these permissions can operate and use ModelArts, but cannot manage dedicated resource pools.	System-defined policy
ModelArts Dependency Access	Permissions on dependent services for ModelArts	System-defined policy

 Table 12-1
 System-defined policies supported by ModelArts

Generally, ModelArts FullAccess is assigned only to administrators. If fine-grained management is not required, assigning ModelArts CommonOperations to all users will meet the development requirements of most small teams. If you want to customize policies for fine-grained permissions management, see IAM.

#### D NOTE

When you assign ModelArts permissions to a user, the system does not automatically assign the permissions of other services to the user. This ensures security and prevents unexpected unauthorized operations. In this case, however, you must separately assign permissions of different services to users so that they can perform some ModelArts operations.

For example, if an IAM user needs to use OBS data for training and the ModelArts training permission has been configured for the IAM user, the IAM user still needs to be assigned with the OBS read, write, and list permissions. The OBS list permission allows you to select the training data path on ModelArts. The read permission is used to preview data and read data for training. The write permission is used to save training results and logs.

- For individual users or small organizations, it is a good practice to configure the **Tenant** Administrator policy that applies to global services for IAM users. In this way, IAM users can obtain all user permissions except IAM. However, this may cause security issues. (For an individual user, its default IAM user belongs to the **admin** user group and has the **Tenant Administrator** permission.)
- If you want to restrict user operations, configure the minimum permissions of OBS for ModelArts users. For details about fine-grained permissions management of other cloud services, see the corresponding cloud service documents.

#### **ModelArts Agency Authorization**

ModelArts must be authorized by users to access other cloud services for AI computing. In the IAM permission system, such authorization is performed through agencies.

To simplify agency authorization, ModelArts supports automatic agency authorization configuration. You only need to configure an agency for yourself or specified users on the **Global Configuration** page of the ModelArts console.

#### **NOTE**

- Only users with the IAM agency management permission can perform this operation. Generally, members in the IAM admin user group have this permission.
- ModelArts agency authorization is region-specific, which means that you must perform agency authorization in each region you use.

Dashboard	
Workflow	
ExeML	
Data Management	•
DevEnviron	•
Algorithm Management	
Training Management	•
AI Application Managem	ent▼
Service Deployment	•
Image Management	
Al Gallery 🗁	
Dedicated Resource Poo	ols 🔻

Figure 12-2 Settings

On the **Global Configuration** page of the ModelArts console, after you click **Add Authorization**, you can configure an agency for a specific user or all users. Generally, an agency named **modelarts\_agency\_**<*Username>\_Random ID* is created by default. In the **Permissions** area, you can select the preset permission configuration or select the required policies. If both options cannot meet your requirements, you can create an agency on the IAM management page (you need to delegate ModelArts to access your resources), and then use an existing agency instead of adding an agency on the **Add Authorization** page.

ModelArts associates multiple users with one agency. This means that if two users need to configure the same agency, you do not need to create an agency for each user. Instead, you only need to configure the same agency for the two users.



#### Figure 12-3 Mapping between users and agencies

#### **NOTE**

Each user can use ModelArts only after being associated with an agency. However, even if the permissions assigned to the agency are insufficient, no error is reported when the API is called. An error occurs only when the system uses unauthorized functions. For example, you enable message notification when creating a training job. Message notification requires SMN authorization. However, an error occurs only when messages need to be sent for the training job. The system ignores some errors, and other errors may cause job failures. When you implement permission minimization, ensure that you will still have sufficient permissions for the required operations on ModelArts.

#### **Strict Authorization**

In strict authorization mode, explicit authorization by the account administrator is required for IAM users to access ModelArts. The administrator can add the required ModelArts permissions to common users through authorization policies.

In non-strict authorization mode, IAM users can use ModelArts without explicit authorization. The administrator needs to configure the deny policy for IAM users to prevent them from using some ModelArts functions.

The administrator can change the authorization mode on the **Global Configuration** page.

#### NOTICE

The strict authorization mode is recommended. In this mode, IAM users must be authorized to use ModelArts functions. In this way, the permission scope of IAM users can be accurately controlled, minimizing permissions granted to IAM users.

#### Managing Resource Access Using Workspaces

Workspace enables enterprise customers to split their resources into multiple spaces that are logically isolated and to manage access to different spaces. As an enterprise user, you can submit the request for enabling the workspace function to your technical support manager. After workspace is enabled, a default workspace is created. All resources you have created are in this workspace. A workspace is like a ModelArts twin. You can switch between workspaces in the upper left corner of the ModelArts console. Jobs in different workspaces do not affect each other.

When creating a workspace, you must bind it to an enterprise project. Multiple workspaces can be bound to the same enterprise project, but one workspace cannot be bound to multiple enterprise projects. You can use workspaces for refined restrictions on resource access and permissions of different users. The restrictions are as follows:

- Users must be authorized to access specific workspaces (this must be configured on the pages for creating and managing workspaces). This means that access to AI assets such as datasets and algorithms can be managed using workspaces.
- In the preceding permission authorization operations, if you set the scope to enterprise projects, the authorization takes effect only for workspaces bound to the selected projects.

#### D NOTE

- Restrictions on workspaces and permission authorization take effect at the same time. That is, a user must have both the permission to access the workspace and the permission to create training jobs (the permission applies to this workspace) so that the user can submit training jobs in this workspace.
- If you have enabled an enterprise project but have not enabled a workspace, all operations are performed in the default enterprise project. Ensure that the permissions on the required operations apply to the default enterprise project.
- The preceding restrictions do not apply to users who have not enabled any enterprise project.

#### Summary

Key features of ModelArts permissions management:

- If you are an individual user, you do not need to consider fine-grained permissions management. Your account has all permissions to use ModelArts by default.
- All functions of ModelArts are controlled by IAM. You can use IAM authorization to implement fine-grained permissions management for specific users.
- All users (including individual users) can use specific functions only after agency authorization on ModelArts (Settings > Add Authorization). Otherwise, unexpected errors may occur.
- If you have enabled the enterprise project function, you can also enable ModelArts workspace and use both basic authorization and workspace for refined permissions management.

## **12.2 Permission Management Mechanisms**

## 12.2.1 IAM

This section describes the IAM permission configurations for all ModelArts functions.

#### **IAM Permissions**

If you need to assign different permissions to employees in your enterprise to access your ModelArts resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access cloud resources. If your account can meet your requirements and you do not need an IAM account to manage user permissions, skip this chapter.

IAM is a free service. You only pay for the resources in your account.

With IAM, you can control access to specific cloud resources. For example, if the software developers in your enterprise need to own permissions to use ModelArts, yet you do not want them to own high-risk operation permissions such as deleting ModelArts, you can grant permissions using IAM to limit their permission on ModelArts.

For details about IAM, see What is IAM?.

#### **Role/Policy-based Authorization**

ModelArts supports role/policy-based authorization. By default, new IAM users do not have any permissions. You need to add a user to one or more groups, and assign permissions policies or roles to these groups. Users inherit permissions of the groups to which they are added. This process is called authorization. The users then inherit permissions from the groups and can perform specified operations on cloud services.

ModelArts is a project-level service deployed for specific regions. When you set **Scope** to **Region-specific projects** and select the specified projects in the specified regions , the users only have permissions for APIG resources in the selected projects. If you set **Scope** to **All resources**, the users have permissions for APIG resources in all region-specific projects. When accessing ModelArts, the users need to switch to a region where they have been authorized to use cloud services.

**Table 12-2** lists all system-defined policies supported by ModelArts. If preset ModelArts permissions cannot meet your requirements, create a custom policy by referring to **Policy Fields in JSON Format**.

Policy	Description	Туре
ModelArts FullAccess	All permissions for ModelArts administrators	System-defined policy
ModelArts CommonOperations	All operation permissions for ModelArts common users, which does not include managing dedicated resource pools.	System-defined policy

Table 12-2 System-defined	l policies	supported	by	ModelArts
---------------------------	------------	-----------	----	-----------

Policy	Description	Туре
ModelArts Dependency Access	Permissions on dependent services for ModelArts	System-defined policy

ModelArts depends on other cloud services. To check or view the cloud services, configure the corresponding permissions on the ModelArts console, as shown in the following table.

**Table 12-3** Roles or policies that are required for performing operations on the ModelArts console

Console Function	Dependency	Role/Policy Required	
Data management	Object Storage Service (OBS)	OBS Administrator	
	Data Lake Insight (DLI)	DLI FullAccess	
	MapReduce Service (MRS)	MRS Administrator	
	GaussDB(DWS)	DWS Administrator	
	Cloud Trace Service (CTS)	CTS Administrator	
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access	
Development	OBS	OBS Administrator	
environment	Cloud Secret Management Service (CSMS)	CSMS ReadOnlyAccess	
	СТЅ	CTS Administrator	
	Elastic Cloud Server (ECS)	ECS FullAccess	
	Software Repository for Container (SWR)	SWR Administrator	
	Scalable File Service (SFS)	SFS Turbo FullAccess	
	Application Operations Management (AOM)	AOM FullAccess	

Console Function	Dependency	Role/Policy Required
	Key Management Service (KMS)	KMS CMKFullAccess
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access
Training	OBS	OBS Administrator
management	Simple Message Notification (SMN)	SMN Administrator
	СТЅ	CTS Administrator
	SFS Turbo	SFS Turbo ReadOnlyAccess
	SWR	SWR Administrator
	AOM	AOM FullAccess
	KMS	KMS CMKFullAccess
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access
Workflow	OBS	OBS Administrator
	СТЅ	CTS Administrator
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access
ExeML	OBS	OBS Administrator
	СТЅ	CTS Administrator
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access
AI application	OBS	OBS Administrator
management	Enterprise Project Management Service (EPS)	EPS FullAccess
	СТЅ	CTS Administrator
	SWR	SWR Administrator
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access
Service	OBS	OBS Administrator
aeployment	Cloud Eye Service (CES)	CES ReadOnlyAccess

Console Function	Dependency	Role/Policy Required	
	SMN	SMN Administrator	
	EPS	EPS FullAccess	
	СТЅ	CTS Administrator	
	Log Tank Service (LTS)	LTS FullAccess	
	Virtual Private Cloud (VPC)	VPC FullAccess	
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access	
AI Gallery	OBS	OBS Administrator	
	СТЅ	CTS Administrator	
	SWR	SWR Administrator	
	ModelArts	ModelArts CommonOperations ModelArts Dependency Access	
Dedicated	СТЅ	CTS Administrator	
resource pool	Cloud Container Engine (CCE)	CCE Administrator	
	Bare Metal Server (BMS)	BMS FullAccess	
	Image Management Service (IMS)	IMS FullAccess	
	Data Encryption Workshop (DEW)	DEW KeypairReadOnlyAccess	
	VPC	VPC FullAccess	
	ECS	ECS FullAccess	
	SFS	SFS Turbo FullAccess	
	OBS	OBS Administrator	
	AOM	AOM FullAccess	
	ModelArts	ModelArts FullAccess	
	Billing Center	BSS Administrator	

If system-defined policies cannot meet your requirements, you can create a custom policy. For details about the actions supported by custom policies, see **ModelArts Resource Permissions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Create a JSON policy or edit an existing one.

For details, see . The following lists examples of common ModelArts custom policies.

• Example 1: Grant permission to manage images.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
            "modelarts:image:register",
            "modelarts:image:listGroup"
        ]
        }
]
```

• Example 2: Grant permission to deny creating, updating, and deleting a dedicated resource pool.

A policy with only "Deny" permissions must be used together with other policies. If the permissions granted to an IAM user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

```
{
   "Version": "1.1",
  "Statement": [
     {
        "Action": [
           "modelarts:*:*"
        L
        "Effect": "Allow"
     },
     {
        "Action": [
           "swr:*:*"
        "Effect": "Allow"
     },
     {
        "Action": [
           "smn:*:*"
        1.
        "Effect": "Allow"
     },
     {
        "Action": [
           "modelarts:pool:create",
           "modelarts:pool:update",
           "modelarts:pool:delete"
        "Effect": "Deny"
     }
  ]
}
```

• Example 3: Create a custom policy containing multiple actions.

A custom policy can contain actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

```
"Version": "1.1",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
           "modelarts:service:*"
     ]
    },
    {
        "Effect": "Allow",
        "Action": [
           "lts:logs:list"
     ]
    }
]
```

## **Policy Fields in JSON Format**

}

{

#### **Policy Structure**

A policy consists of a version and one or more statements (indicating different actions).



#### Figure 12-4 Policy structure

#### **Policy Parameters**

The following describes policy parameters. You can create custom policies by specifying the parameters.

#### Table 12-4 Policy parameters

Parameter		Description	Value	
Version		Policy version	<b>1.1</b> : indicates policy-based access control.	
Statement: authorizatio n statement of a policy	Effect	Whether to allow or deny the operations defined in the action	<ul> <li>Allow: indicates the operation is allowed.</li> <li>Deny: indicates the operation is not allowed.</li> <li>NOTE         <ul> <li>If the policy used to grant user permissions contains both Allow and Deny for the same action, Deny takes precedence.</li> </ul> </li> </ul>	
	Action	Operation to be performed on the service	Format: " <i>Service name</i> : <i>Resource type</i> : <i>Action</i> ". Wildcard characters (*) are supported, indicating all options. Example:	
			<b>modelarts:notebook:list</b> : indicates the permission to view a notebook instance list. <b>modelarts</b> indicates the service name, <b>notebook</b> indicates the resource type, and <b>list</b> indicates the operation. View all actions of a service in its <i>API</i>	
	Conditio C	Condition for	Reference.	
	n a policy to take effect including condition keys and operators	a policy to take effect, including condition keys and operators	<i>key</i> .[ <i>Value 1, Value 2</i> ]}" If you set multiple conditions, the policy takes effect only when all the conditions are met. Example:	
			StringEndWithIfExists": {"g:UserName":["specialCharacter"]}: The statement is valid for users whose names end with specialCharacter.	
	Resourc e	Resources on which a policy takes effect	Format: <i>Service</i> <i>name</i> :< <i>Region</i> >:< <i>Account ID</i> >: <i>Resource</i> <i>type</i> : <i>Resource path</i> . Asterisks (*) are supported for resource type, indicating all resources. <b>NOTE</b> ModelArts authorization does not allow you to specify a resource path.	

#### ModelArts Resource Types

Administrators can specify the scope based on ModelArts resource types. The following table lists the resource types supported by ModelArts:

**Table 12-5** Resource types supported by ModelArts role/policy-based authorization

Resource Type	Description
notebook	Notebook instances in DevEnviron
exemlProject	ExeML projects
exemlProjectInf	ExeML-powered real-time inference service
exemlProjectTrain	ExeML-powered training jobs
exemlProjectVersion	ExeML project version
workflow	Workflow
pool	Dedicated resource pool
network	Networking of a dedicated resource pool
trainJob	Training job
trainJobLog	Runtime logs of a training job
trainJobInnerModel	Preset model
trainJobVersion	Version of a training job (supported by old-version training jobs that will be discontinued soon)
trainConfig	Configuration of a training job (supported by old-version training jobs that will be discontinued soon)
tensorboard	Visualization job of training results (supported by old-version training jobs that will be discontinued soon)
model	Models
service	Real-time service
nodeservice	Edge service
workspace	Workspace
dataset	Dataset
dataAnnotation	Dataset labels
aiAlgorithm	Algorithm for training jobs

Resource Type	Description
image	Image
devserver	Elastic BMS

#### ModelArts Resource Permissions

For details, see "Permissions Policies and Supported Actions" in *ModelArts API Reference*.

## **12.2.2 Agencies and Dependencies**

#### **Function Dependency**

#### **Function Dependency Policies**

When using ModelArts to develop algorithms or manage training jobs, you are required to use other Cloud services. For example, before submitting a training job, select an OBS path for storing the dataset and logs, respectively. Therefore, when configuring fine-grained authorization policies for a user, the administrator must configure dependent permissions so that the user can use required functions.

#### **NOTE**

If you use ModelArts as the root user (default IAM user with the same name as the account), the root user has all permissions by default.

Applicati on Scenario	Dependent Service	Dependent Policy	Supported Function
Global configura tion	IAM	iam:users:listUs ers	Obtain a user list. This action is required by the administrator only.
Basic function	IAM	iam:tokens:ass ume	(Mandatory) Use an agency to obtain temporary authentication credentials.

#### Table 12-6 Basic configuration

 Table 12-7 Managing workspaces

Applicati on Scenario	Dependent Service	Dependent Policy	Supported Function
Workspac e	IAM	iam:users:listUs ers	Authorize an IAM user to use a workspace.

Applicati on Scenario	Dependent Service	Dependent Policy	Supported Function
	ModelArts	modelarts:*:*de lete*	Clear resources in a workspace when deleting it.

Table 12-8 Managing notebook instances

Application Scenario	Depend ent Service	Dependent Policy	Supported Function
Lifecycle managemen t of developmen t environment instances	ModelA rts	modelarts:notebook:cr eate modelarts:notebook:li st modelarts:notebook:u pdate modelarts:notebook:u pdate modelarts:notebook:st art modelarts:notebook:st op modelarts:notebook:u pdateStopPolicy modelarts:image:delet e modelarts:image:list modelarts:image:list modelarts:image:creat e modelarts:image:get modelarts:notebook:u pdateStopPolicy modelarts:image:delet e	Start, stop, create, delete, and update an instance.

Application Scenario	Depend ent Service	Dependent Policy	Supported Function
Dynamically mounting storage	ModelA rts	modelarts:notebook:li stMountedStorages modelarts:notebook: mountStorage modelarts:notebook:g etMountedStorage modelarts:notebook:u mountStorage	Dynamically mount storage.
	OBS	obs:bucket:ListAllMyB uckets obs:bucket:ListBucket	
lmage managemen t	ModelA rts	modelarts:image:regis ter modelarts:image:listG roup	Register and view an image on the <b>Image Management</b> page.
Saving an image	SWR	SWR Admin	The <b>SWR Admin</b> policy contains the maximum scope of SWR permissions, which can be used to:
			<ul> <li>Save a running development environment instance as an image.</li> </ul>
			<ul> <li>Create a notebook instance using a custom image.</li> </ul>
Using the SSH function	ECS	ecs:serverKeypairs:list ecs:serverKeypairs:get ecs:serverKeypairs:del ete ecs:serverKeypairs:cre ate	Configure a login key for a notebook instance.
Mounting an SFS Turbo file system	SFS Turbo	SFS Turbo FullAccess	Read and write an SFS directory as an IAM user. Mount an SFS file system that is not created by you to a notebook instance using a dedicated resource pool.

Application Scenario	Depend ent Service	Dependent Policy	Supported Function
Viewing all Instances	ModelA rts	modelarts:notebook:li stAllNotebooks	View development environment instances of all
	IAM	iam:users:listUsers	users on the ModelArts management console. This action is required by the development environment instance administrator.
Local VS Code plug- in or PyCharm Toolkit	ModelA rts	modelarts:notebook:li stAllNotebooks modelarts:trainJob:cre ate modelarts:trainJob:list modelarts:trainJob:up date modelarts:trainJobVer sion:delete modelarts:trainJob:get modelarts:trainJob:log Export modelarts:workspace: getQuotas (This policy is required if the <b>workspace</b> function is anabled.)	Access a notebook instance from local VS Code and submit training jobs.

Application Scenario	Depend ent Service	Dependent Policy	Supported Function
	OBS	obs:bucket:ListAllMyb uckets	
		obs:bucket:HeadBucke t	
		obs:bucket:ListBucket	
		obs:bucket:GetBucket Location	
		obs:object:GetObject	
		obs:object:GetObjectV ersion	
		obs:object:PutObject	
		obs:object:DeleteObje ct	
		obs:object:DeleteObje ctVersion	
		obs:object:ListMultipa rtUploadParts	
		obs:object:AbortMulti partUpload	
		obs:object:GetObjectA cl	
		obs:object:GetObjectV ersionAcl	
		obs:bucket:PutBucket Acl	
		obs:object:PutObjectA cl	
		obs:object:ModifyObje ctMetaData	
	IAM	iam:projects:listProject s	Obtain an IAM project list through local PyCharm for access configurations.

Application Scenario	Dependent Service	Dependent Policy	Supported Function
Training manageme nt	ModelArts	modelarts:trainJob:* modelarts:trainJobLog:* modelarts:aiAlgorithm:* modelarts:image:list	Create a training job and view training logs.
		modelarts:workspace:getQuot as	Obtain a workspace quota. This policy is required if the <b>workspace</b> function is enabled.
		modelarts:tag:list	Use Tag Management Service (TMS) in a training job.
	IAM	iam:credentials:listCredentials iam:agencies:listAgencies	Use the configured agency authorization.
	SFS Turbo	sfsturbo:shares:getShare sfsturbo:shares:getAllShares	Use SFS Turbo in a training job.
	SWR	swr:repository:listTags swr:repository:getRepository swr:repository:listRepositories	Use a custom image to create a training job.
	SMN	smn:topic:publish smn:topic:list	Notify training job status changes through SMN.

Table 12-9 Managing training jobs

Application Scenario	Dependent Service	Dependent Policy	Supported Function
	OBS	obs:bucket:ListAllMybuckets obs:bucket:HeadBucket obs:bucket:ListBucket obs:bucket:GetBucketLocation obs:object:GetObject obs:object:GetObjectVersion	Run a training job using a dataset in an OBS bucket.
		obs:object:PutObject obs:object:DeleteObject obs:object:DeleteObjectVer- sion	
		obs:object:ListMultipartUpload Parts obs:object:AbortMultipartUp-	
		obs:object:GetObjectAcl obs:object:GetObjectVersio- nAcl	
		obs:bucket:PutBucketAcl obs:object:PutObjectAcl obs:object:ModifyObjectMeta- Data	

Table 12-10 Using workflows

Applicatio	Depende	Dependent Policy	Supported
n Scenario	nt Service		Function
Using a dataset	ModelArts	modelarts:dataset:getDataset modelarts:dataset:createDataset modelarts:dataset:createDatasetV ersion modelarts:dataset:createImportTa sk modelarts:dataset:updateDataset modelarts:processTask:createProc essTask modelarts:processTask:getProcess Task modelarts:dataset:listDatasets	Use ModelArts datasets in a workflow.

Applicatio n Scenario	Depende nt Service	Dependent Policy	Supported Function
Managing AI application s	ModelArts	modelarts:model:list modelarts:model:get modelarts:model:create modelarts:model:delete modelarts:model:update	Manage ModelArts AI applications in a workflow.
Deploying a service	ModelArts	modelarts:service:get modelarts:service:create modelarts:service:update modelarts:service:delete modelarts:service:getLogs	Manage ModelArts real- time services in a workflow.
Training jobs	ModelArts	modelarts:trainJob:get modelarts:trainJob:create modelarts:trainJob:list modelarts:trainJobVersion:list modelarts:trainJobVersion:create modelarts:trainJob:delete modelarts:trainJobVersion:delete modelarts:trainJobVersion:stop	Manage ModelArts training jobs in a workflow.
Workspace	ModelArts	modelarts:workspace:get modelarts:workspace:getQuotas	Use ModelArts workspaces in a workflow.

Applicatio n Scenario	Depende nt Service	Dependent Policy	Supported Function
Managing data	OBS	obs:bucket:ListAllMybuckets (Obtaining a bucket list)	Use OBS data in a workflow.
		obs:bucket:HeadBucket (Obtaining bucket metadata)	
		obs:bucket:ListBucket (Listing objects in a bucket)	
		obs:bucket:GetBucketLocation (Obtaining the bucket location)	
		obs:object:GetObject (Obtaining object content and metadata)	
		obs:object:GetObjectVersion (Obtaining object content and metadata)	
		obs:object:PutObject (Uploading objects using PUT method, uploading objects using POST method, copying objects, appending an object, initializing a multipart task, uploading parts, and merging parts)	
		obs:object:DeleteObject (Deleting an object or batch deleting objects)	
		obs:object:DeleteObjectVersion (Deleting an object or batch deleting objects)	
		obs:object:ListMultipartUpload- Parts (Listing uploaded parts)	
		obs:object:AbortMultipartUpload (Aborting multipart uploads)	
		obs:object:GetObjectAcl (Obtaining an object ACL)	
		obs:object:GetObjectVersionAcl (Obtaining an object ACL)	
		obs:bucket:PutBucketAcl (Configuring a bucket ACL)	
		obs:object:PutObjectAcl (Configuring an object ACL)	

Applicatio n Scenario	Depende nt Service	Dependent Policy	Supported Function
Executing a workflow	IAM	iam:users:listUsers (Obtaining users)	Call other ModelArts
		iam:agencies:getAgency (Obtaining details about a specified agency)	services when the workflow is running.
		iam:tokens:assume (Obtaining an agency token)	
Integrating DLI	DLI	dli:jobs:get (Obtaining job details)	Integrate DLI into a workflow.
		dli:jobs:list_all (Viewing a job list)	
		dli:jobs:create (Creating a job)	
Integrating MRS	MRS	mrs:job:get (Obtaining job details)	Integrate MRS into a workflow.
		mrs:job:submit (Creating and executing a job)	
		mrs:job:list (Viewing a job list)	
		mrs:job:stop (Stopping a job)	
		mrs:job:batchDelete (Batch deleting jobs)	
		mrs:file:list (Viewing a file list)	

Table 12-11 Managing AI applications

Applicatio	Depende	Dependent Policy	Supported
n Scenario	nt Service		Function
Managing AI application s	SWR	swr:repository:deleteRepository swr:repository:deleteTag swr:repository:getRepository swr:repository:listTags	Import a model from a custom image. Use a custom engine when importing a model from OBS.

Applicatio n Scenario	Depende nt Service	Dependent Policy	Supported Function
	OBS	obs:bucket:ListAllMybuckets (Obtaining a bucket list)	Import a model from a template.
		obs:bucket:HeadBucket (Obtaining bucket metadata)	Specify an OBS path for model
		obs:bucket:ListBucket (Listing objects in a bucket)	conversion.
		obs:bucket:GetBucketLocation (Obtaining the bucket location)	
		obs:object:GetObject (Obtaining object content and metadata)	
		obs:object:GetObjectVersion (Obtaining object content and metadata)	
		obs:object:PutObject (Uploading objects using PUT method, uploading objects using POST method, copying objects, appending an object, initializing a multipart task, uploading parts, and merging parts)	
		obs:object:DeleteObject (Deleting an object or batch deleting objects)	
		obs:object:DeleteObjectVersion (Deleting an object or batch deleting objects)	
		obs:object:ListMultipartUpload- Parts (Listing uploaded parts)	
		obs:object:AbortMultipartUpload (Aborting multipart uploads)	
		obs:object:GetObjectAcl (Obtaining an object ACL)	
		obs:object:GetObjectVersionAcl (Obtaining an object ACL)	
		obs:bucket:PutBucketAcl (Configuring a bucket ACL)	
		obs:object:PutObjectAcl (Configuring an object ACL)	

Applicatio n Scenario	Depende nt Service	Dependent Policy	Supported Function
Real-time services	LTS	lts:logs:list (Obtaining the log list)	Show LTS logs.
	OBS	obs:bucket:GetBucketPolicy (Obtaining a bucket policy) obs:bucket:HeadBucket (Obtaining bucket metadata) obs:bucket:ListAllMyBuckets (Obtaining a bucket list) obs:bucket:PutBucketPolicy (Configuring a bucket policy) obs:bucket:DeleteBucketPolicy (Deleting a bucket policy)	Mount external volumes to a container when services are running.
Batch services	OBS	obs:object:GetObject (Obtaining object content and metadata) obs:object:PutObject (Uploading objects using PUT method, uploading objects using POST method, copying objects, appending an object, initializing a multipart task, uploading parts, and merging parts) obs:bucket:CreateBucket (Creating a bucket) obs:bucket:ListBucket (Listing objects in a bucket) obs:bucket:ListAllMyBuckets (Obtaining a bucket list)	Create batch services and perform batch inference.
Edge services	CES	ces:metricData:list: (Obtaining metric data)	View monitoring metrics.
	IEF	ief:deployment:delete (Deleting a deployment)	Manage edge services.

Table 12-12 Managing service deployment

Applicati on Scenario	Depende nt Service	Dependent Policy	Supported Function
Managing datasets and labels	OBS	obs:bucket:ListBucket (Listing objects in a bucket) obs:object:GetObject (Obtaining object content and metadata) obs:object:PutObject (Uploading objects using PUT method, uploading objects using POST method, copying objects, appending an object, initializing a multipart task, uploading parts, and merging parts) obs:object:DeleteObject (Deleting an object or batch deleting objects) obs:bucket:HeadBucket (Obtaining bucket metadata) obs:bucket:GetBucketAcl (Obtaining a bucket ACL) obs:bucket:PutBucketAcl (Configuring a bucket Policy (Obtaining a bucket policy) obs:bucket:DeleteBucketPolicy (Configuring a bucket policy) obs:bucket:DeleteBucketPolicy (Configuring a bucket policy) obs:bucket:PutBucketCORS (Configuring or deleting CORS rules of a bucket) obs:bucket:GetBucketCORS (Obtaining the CORS rules of a bucket) obs:object:PutObjectAcl (Configuring an object ACL)	Manage datasets in OBS. Label OBS data. Create a data management job.
Managing table datasets	DLI	dli:database:displayAllDatabases dli:database:displayAllTables dli:table:describe_table	Manage DLI data in a dataset.
Managing table datasets	DWS	dws:openAPICluster:list dws:openAPICluster:getDetail	Manage DWS data in a dataset.

Applicati on Scenario	Depende nt Service	Dependent Policy	Supported Function
Managing table datasets	MRS	mrs:job:submit mrs:job:list mrs:cluster:list mrs:cluster:get	Manage MRS data in a dataset.
Auto labeling	ModelArts	modelarts:service:list modelarts:model:list modelarts:model:get modelarts:model:create modelarts:trainJobInnerModel:list modelarts:workspace:get modelarts:workspace:list	Enable auto labeling.
Team labeling	IAM	iam:projects:listProjects (Obtaining tenant projects) iam:users:listUsers (Obtaining users) iam:agencies:createAgency (Creating an agency) iam:quotas:listQuotasForProject (Obtaining the quotas of a project)	Manage labeling teams.

#### Table 12-14 Managing resources

Applicatio n Scenario	Dependen t Service	Dependent Policy	Supported Function
Managing resource pools	BSS	bss:coupon:view bss:order:view bss:balance:view bss:discount:view bss:renewal:view bss:bill:view bss:contract:update bss:order:pay bss:unsubscribe:update bss:renewal:update bss:order:update	Create, renew, and unsubscribe from a resource pool. Dependent permissions must be configured in the IAM project view.

Applicatio n Scenario	Dependen t Service	Dependent Policy	Supported Function
	ECS	ecs:availabilityZones:list	Show AZs. Dependent permissions must be configured in the IAM project view.
Network managem ent	VPC	<pre>vpc:routes:create vpc:routes:list vpc:routes:get vpc:peerings:create vpc:peerings:accept vpc:peerings:get vpc:peerings:delete vpc:routeTables:update vpc:routeTables:get vpc:routeTables:list vpc:vpcs:create vpc:vpcs:list vpc:vpcs:get vpc:vpcs:delete vpc:subnets:create vpc:subnets:delete vpc:subnets:delete vpcep:endpoints:list vpcep:endpoints:delete vpcep:endpoints:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:ports:get vpc:networks:create vpc:networks:get vpc:networks:get vpc:networks:update vpc:networks:update vpc:networks:update vpc:networks:delete</pre>	Create and delete ModelArts networks, and interconnect VPCs. Dependent permissions must be configured in the IAM project view.

Applicatio n Scenario	Dependen t Service	Dependent Policy	Supported Function
	SFS Turbo	sfsturbo:shares:addShareNic sfsturbo:shares:deleteShareNic sfsturbo:shares:showShareNic sfsturbo:shares:listShareNics	Interconnect your network with SFS Turbo. Dependent permissions must be configured in the IAM project view.
Edge resource pool	IEF	ief:node:list ief:group:get ief:application:list ief:application:get ief:node:listNodeCert ief:node:get ief:lEFInstance:get ief:deployment:list ief:group:listGroupInstanceState ief:lEFInstance:list ief:deployment:get ief:deployment:get ief:group:list	Add, delete, modify, and search for edge pools

#### Agency authorization

To simplify operations when you use ModelArts to run jobs, certain operations are automatically performed on the ModelArts backend, for example, downloading the datasets in an OBS bucket to a workspace before a training job is started and dumping training job logs to the OBS bucket.

ModelArts does not save your token authentication credentials. Before performing operations on your resources (such as OBS buckets) in a backend asynchronous job, you are required to explicitly authorize ModelArts through an IAM agency. ModelArts will use the agency to obtain a temporary authentication credential for performing operations on your resources. For details, see Adding Authorization.





As shown in **Figure 12-5**, after authorization is configured on ModelArts, ModelArts uses the temporary credential to access and operate your resources, relieving you from some complex and time-consuming operations. The agency credential will also be synchronized to your jobs (including notebook instances and training jobs). You can use the agency credential to access your resources in the jobs.

# You can use either of the following methods to authorize ModelArts using an agency:

#### One-click authorization

ModelArts provides one-click automatic authorization. You can quickly configure agency authorization on the **Global Configuration** page of ModelArts. Then, ModelArts will automatically create an agency for you and configure it in ModelArts.

In this mode, the authorization scope is specified based on the preset system policies of dependent services to ensure sufficient permissions for using services. The created agency has almost all permissions of dependent services. If you want to precisely control the scope of permissions granted to an agency, use the second method.

#### Custom authorization

The administrator creates different agency authorization policies for different users in IAM, and configures the created agency for ModelArts users. When creating an agency for an IAM user, the administrator specifies the minimum permissions for the agency based on the user's permissions to control the resources that the user can access when they use ModelArts.

#### **Risks in Unauthorized Operations**

The agency authorization of a user is independent. Theoretically, the agency authorization scope of a user can be beyond the authorization scope of the authorization policy configured for the user group. Any improper configuration will result in unauthorized operations.
To prevent unauthorized operations, only a tenant administrator is allowed to configure agencies for users in the ModelArts global configuration to ensure the security of agency authorization.

#### **Minimal Agency Authorization**

When configuring agency authorization, an administrator must strictly control the authorization scope.

ModelArts asynchronously and automatically performs operations such as job preparation and clearing. The required agency authorization is within the basic authorization scope. If you use only some functions of ModelArts, the administrator can filter out the basic permissions that are not used according to the agency authorization configuration. Conversely, if you need to obtain resource permissions beyond the basic authorization scope in a job, the administrator can add new permissions to the agency authorization configuration. In a word, the agency authorization scope must be minimized and customized based on service requirements.

#### **Basic Agency Authorization Scope**

To customize the permissions for an agency, select permissions based on your service requirements.

Applica tion Scenari o	Depende nt Service	Agency Authorization	Description	Conf igur atio n Sug gest ion
JupyterL ab	OBS	obs:object:DeleteObject obs:object:GetObject obs:object:GetObjectVersion obs:bucket:CreateBucket obs:bucket:ListBucket obs:bucket:ListAllMyBuckets obs:object:PutObject obs:bucket:GetBucketAcl obs:bucket:PutBucketAcl obs:bucket:PutBucketCORS	Use OBS to upload and download data in JupyterLab through ModelArts notebook.	Reco mm end ed

Table 12-15 Basic agency authorization for a development environment

Applica tion Scenari o	Depende nt Service	Agency Authorization Description		Conf igur atio n Sug gest ion
Develop ment environ ment monitori ng	AOM	aom:alarm:put	Call the AOM API to obtain monitoring data and events of notebook instances and display them in ModelArts notebook.	Reco mm end ed

#### Table 12-16 Basic agency authorization for training jobs

Applicati on Scenario	Dependent Service	Agency Authorization	Description
Training jobs	OBS	obs:bucket:ListBucket obs:object:GetObject obs:object:PutObject	Download data, models, and code before starting a training job. Upload logs and models when a training job is running.

 Table 12-17 Basic agency authorization for deploying services

Applicat ion Scenari o	Dependen t Service	Agency Authorization	Description
Real- time services	LTS	lts:groups:create lts:groups:list lts:topics:create lts:topics:delete lts:topics:list	Configure LTS for reporting logs of real-time services.

Applicat ion Scenari o	Dependen t Service	Agency Authorization	Description
Batch services	OBS	obs:bucket:ListBucket obs:object:GetObject obs:object:PutObject	Run a batch service.
Edge services	IEF	ief:deployment:list ief:deployment:create ief:deployment:update ief:deployment:delete ief:node:createNodeCert ief:iefInstance:list ief:node:list	Deploy an edge service using IEF.

#### Table 12-18 Basic agency authorization for managing data

Applica tion Scenari o	Dependen t Service	Agency Authorization	Description
Dataset and data labeling	OBS	obs:object:GetObject obs:object:PutObject obs:object:DeleteObject obs:object:PutObjectAcl obs:bucket:ListBucket obs:bucket:HeadBucket obs:bucket:GetBucketAcl obs:bucket:PutBucketAcl obs:bucket:PutBucketPolicy obs:bucket:DeleteBucketPolicy obs:bucket:DeleteBucketPolicy obs:bucket:PutBucketCORS	Manage datasets in an OBS bucket.
Labelin g data	ModelArts inference	modelarts:service:get modelarts:service:create modelarts:service:update	Perform auto labeling based on ModelArts inference.

Applicati on Scenario	Depende nt Service	Agency Authorization	Description
Network managem ent (New version)	VPC	vpc:routes:create           vpc:routes:get           vpc:routes:delete           vpc:peerings:create           vpc:peerings:accept           vpc:peerings:get           vpc:routeTables:update           vpc:routeTables:get           vpc:routeTables:list           vpc:vpcs:create           vpc:vpcs:list           vpc:subnets:create           vpc:subnets:create           vpc:subnets:delete           vpc:subnets:create           vpc:subnets:create           vpc:subnets:delete           vpc:subnets:get           vpc:subnets:get           vpc:subnets:create           vpc:subnets:delete           vpc:perindpoints:list           vpcep:endpoints:create           vpc:ports:create           vpc:ports:create           vpc:ports:create           vpc:ports:create           vpc:ports:create           vpc:ports:create           vpc:ports:create           vpc:ports:create           vpc:ports:update           vpc:ports:delete           vpc:ports:delete           vpc:ports:delete           vpc:ports:update           vpc:networks:create           vpc:networks:create	Create and delete ModelArts networks, and interconnect VPCs. Dependent permissions must be configured in the IAM project view.
	SFS Turbo	sfsturbo:shares:addShareNic sfsturbo:shares:deleteShareNic sfsturbo:shares:showShareNic sfsturbo:shares:listShareNics	Interconnect your network with SFS Turbo. Dependent permissions must be configured in the IAM project view.

Table 12-19 Basic agency authorization for managing dedicated resource pools

Applicati on Scenario	Depende nt Service	Agency Authorization	Description
Managing resource pools	ECS	ecs:availabilityZones:list	Show AZs. Dependent permissions must be configured in the IAM project view.

# 12.2.3 Workspace

ModelArts allows you to create multiple workspaces to develop algorithms and manage and deploy models for different service objectives. In this way, the development outputs of different applications are allocated to different workspaces for simplified management.

Workspace supports the following types of access control:

- **PUBLIC**: publicly accessible to tenants (including both tenant accounts and all their user accounts)
- **PRIVATE**: accessible only to the creator and tenant accounts
- **INTERNAL**: accessible to the creator, tenant accounts, and specified IAM user accounts. When **Authorization Type** is set to **INTERNAL**, specify one or more accessible IAM user accounts.

A default workspace is allocated to each IAM project of each account. The access control of the default workspace is **PUBLIC**.

Workspace access control allows the access of only certain users. This function can be used in the following scenarios:

- Education: A teacher allocates an INTERNAL workspace to each student and allows the workspaces to be accessed only by specified students. In this way, students can separately perform experiments on ModelArts.
- **Enterprises**: An administrator creates a workspace for production tasks and allows only O&M personnel to use the workspace, and creates a workspace for routine debugging and allows only developers to use the workspace. In this way, different enterprise roles can use resources only in a specified workspace.

As an enterprise user, you can submit the request for enabling the workspace function to your technical support.

# **12.3 Configuration Practices in Typical Scenarios**

# 12.3.1 Assigning Permissions to Individual Users for Using ModelArts

Certain ModelArts functions require access to Object Storage Service (OBS), Software Repository for Container (SWR), and Intelligent EdgeFabric (IEF). Before using ModelArts, your account must be authorized to access these services. Otherwise, these functions will be unavailable.

#### Constraints

- Only a tenant account can perform agency authorization to authorize the current account or all IAM users under the current account.
- Multiple IAM users or accounts can use the same agency.
- A maximum of 50 agencies can be created under an account.
- If you use ModelArts for the first time, add an agency. Generally, common user permissions are sufficient for your requirements. You can configure permissions for refined permissions management.
- If you have not been authorized, ModelArts will display a message indicating that you have not been authorized when you access the **Add Authorization** page. In this case, contact your administrator to add authorization.

#### Adding Authorization

- 1. Log in to the ModelArts management console. In the navigation pane on the left, choose **Settings**. The **Global Configuration** page is displayed.
- 2. Click **Add Authorization**. On the **Add Authorization** page that is displayed, configure the parameters.

Parameter	Description
Authorized User	Options: IAM user, Federated user, Agency, and All users
	• <b>IAM user</b> : You can use a tenant account to create IAM users and assign permissions for specific resources. Each IAM user has their own identity credentials (password and access keys) and uses cloud resources based on assigned permissions.
	• Federated user: A federated user is also called a virtual enterprise user.
	Agency: You can create agencies in IAM.
	• All users: If you select this option, the agency permissions will be granted to all IAM users under the current account, including those created in the future. For individual users, choose All users.

Table 12-20 Parameters

Parameter	Description					
Authorized To	<ul> <li>This parameter is not displayed when Authorized User is set to All users.</li> <li>IAM user: Select an IAM user and configure an agency for the IAM user.</li> </ul>					
	Figure 12-6 Selecting an IAM user					
	Authorized User IAM user Federated user Agency All users					
Authorized To						
• Federated user: Enter the username or user ID of the target to user.						
	Figure 12-7 Selecting a federated user					
	Authorized User IAM user Federated user Agency All users					
	Authorized To					
	<ul> <li>Agency: Select an agency name. You can create an agency under account A and grant the agency permissions to account B. When using account B, you can switch the role in the upper right corner of the console to account A and use the agency permissions of account A.</li> <li>Figure 12-8 Switch Role</li> <li>English</li> <li>Security Settings</li> <li>My Credentials</li> <li>Enterprise Management</li> <li>Switch Role</li> </ul>					
	Tag Management Log Out					
Agency	• Use existing: If there are agencies in the list, select an available one to authorize the selected user. Click the drop-down arrow next to an agency name to view its permission details.					
	• Add agency: If there is no available agency, create one. If you use ModelArts for the first time, select Add agency.					

Parameter	Description
Add agency > Agency Name	The system automatically creates a changeable agency name.
Add agency > Authorization Method	<ul> <li>Role-based: A coarse-grained IAM authorization strategy to assign permissions based on user responsibilities. Only a limited number of service-level roles are available. When using roles to grant permissions, assign other roles on which the permissions depend to take effect. Roles are not ideal for fine-grained authorization and secure access control.</li> <li>Policy-based: A fine-grained authorization tool that defines permissions for operations on specific cloud resources under certain conditions. This type of authorization is more flexible and ideal for secure access control.</li> </ul>
Add agency > Permissions > Common User	<b>Common User</b> provides the permissions to use all basic ModelArts functions. For example, you can access data, and create and manage training jobs. Select this option generally. Click <b>View permissions</b> to view common user permissions.
Add agency > Permissions > Custom	If you need refined permissions management, select <b>Custom</b> to flexibly assign permissions to the created agency. You can select permissions from the permission list as required.

3. Click Create.

## **Viewing Authorized Permissions**

You can view the configured authorizations on the **Global Configuration** page. Click **View Permissions** in the **Authorization Content** column to view the permission details.

#### Figure 12-9 View Permissions

Authorized To $\mbox{+}$	Authorized User $\mbox{$\stackrel{\circ}{=}$}$	Authorization Type 👙	Authorization Content 💠	Creation Time 💠	Operation
	All users	Agency	modelarts_{	Jan 19, 2023 16:53:29 GMT+08:00	View Permissions Delete

#### Figure 12-10 Common user permissions

View Permissio	ns		
	Name	Туре	Description
	DLI FullAccess	System-defined policy	Full permissions for Data Lake Insight.
	VPC Administrator	System-defined role	VPC Administrator
	EPS FullAccess	System-defined policy	All operations on the Enterprise Project Management service.
	CTS Administrator	System-defined role	CTS Administrator
	ModelArts CommonOperations	System-defined policy	Common permissions of ModelArts service, except create, update, del
	SFS ReadOnlyAccess	System-defined policy	The read-only permissions to all SFS resources.
	OBS Administrator	System-defined policy	Object Storage Service Administrator
	DWS Administrator	System-defined role	Data Warehouse Service Administrator
	LTS FullAccess	System-defined policy	All permissions of Log Tank service.
	CES ReadOnlyAccess	System-defined policy	Read-only permissions for Cloud Eye.
	10 🔻 Total Records: 12 < 1	2 >	

# 12.3.2 Separately Assigning Permissions to Administrators and Developers

In small- and medium-sized teams, administrators need to globally control ModelArts resources, and developers only need to focus on their own instances. Generally, the **te\_admin** permission of a developer account must be configured by the tenant account. This section uses notebook as an example to describe how to assign different permissions to administrators and developers through custom policies.

#### Scenarios

To develop a project using notebook, administrators need full control permissions for using ModelArts dedicated resource pools, and access and operation permissions on all notebook instances.

To use development environments, developers only need operation permissions for using their own instances and dependent services. They do not need to perform operations on ModelArts dedicated resource pools or view notebook instances of other users.





## **Configuring Permissions for an Administrator**

Assign full control permissions to administrators for using ModelArts dedicated resource pools and all notebook instances. The procedure is as follows:

- **Step 1** Use a tenant account to create an administrator user group ModelArts\_admin\_group and add administrator accounts to ModelArts\_admin\_group.
- **Step 2** Create a custom policy.
  - 1. Log in to the management console using an administrator account, hover over your username in the upper right corner, and click **Identity and Access**

**Management** from the drop-down list to switch to the IAM management console.

 Create custom policy 1 and assign IAM and OBS permissions to the user. In the navigation pane of the IAM console, choose Permissions > Policies/Roles. Click Create Custom Policy in the upper right corner. On the displayed page, enter Policy1\_IAM\_OBS for Policy Name, select JSON for Policy View, configure the policy content, and click OK.

\* Policy Name Policy1 IAM OBS JSON Policy View Visual editor 1 - { \* Policy Content 2 "Version": "1.1", "Statement": [ 3 -4 -{ "Effect": "Allow", 5 6 -"Action": [ 7 "iam:users:listUsers", "iam:projects:listProjects", 8 "obs:object:PutObject", 9 10 "obs:object:GetObject", "obs:object:GetObjectVersion", 11 "obs:bucket:HeadBucket", 12 13 "obs:object:DeleteObject" 14 "obs:bucket:CreateBucket", 15 "obs:bucket:ListBucket" 16 ] 17 } 18 ] 19 }

Figure 12-12 Custom policy 1

The custom policy **Policy1\_IAM\_OBS** is as follows, which grants IAM and OBS operation permissions to the user. You can directly copy and paste the content.

```
"Version": "1.1",

"Statement": [

{

    "Effect": "Allow",

    "Action": [

    "iam:users:listUsers",

    "iam:projects:listProjects",

    "obs:object:PutObject",

    "obs:object:GetObjectVersion",

    "obs:object:GetObjectVersion",

    "obs:bucket:HeadBucket",

    "obs:bucket:CreateBucket",

    "obs:bucket:ListBucket"
```

{

}

] }

{

ļ

3. Repeat **2.2** to create custom policy 2 and grant the user the permissions to perform operations on dependent services ECS, SWR, MRS, and SMN as well as ModelArts. Set **Policy Name** to **Policy2\_AllowOperation** and **Policy View** to **JSON**, configure the policy content, and click **OK**.

The custom policy **Policy2\_AllowOperation** is as follows, which grants the user the permissions to perform operations on dependent services ECS, SWR, MRS, and SMN as well as ModelArts. You can directly copy and paste the content.

"Version": "1.1", "Statement": [ { "Effect": "Allow", "Action": [ "ecs:serverKeypairs:list", "ecs:serverKeypairs:get", "ecs:serverKeypairs:delete", "ecs:serverKeypairs:create", "swr:repository:getNamespace", "swr:repository:listNamespaces", "swr:repository:deleteTag", "swr:repository:getRepository", "swr:repository:listTags", "swr:instance:createTempCredential", "mrs:cluster:get", "modelarts:\*:\*" 1 } ]

**Step 3** Grant the policy created in **2** to the administrator group **ModelArts\_admin\_group**.

 In the navigation pane of the IAM console, choose User Groups. On the User Groups page, locate the row that contains ModelArts\_admin\_group, click Authorize in the Operation column, and select Policy1\_IAM\_OBS and Policy2\_AllowOperation. Click Next.

Figure 12-13 Select Policy/Role

1 Select Poli	cy/Role (2) Select Scope (3) Finish
Assign sele	ected permissions to ModelArts_admin_group.
View Se	elected (2) Copy Permissions from Another Project
	Policy/Role Name
	Policy2_AllowOperation
	Policy1_IAM_OBS

2. Specify the scope as **All resources** and click **OK**.

#### Figure 12-14 Select Scope

1 Select Policy/Role — 2 Select Scope — 3 Finish
The following are recommended scopes for the permissions you selected. See
Scope
<ul> <li>All resources</li> </ul>
IAM users will be able to use all resources, including those in enterprise projects,
○ Region-specific projects ⑦
◯ Global services ⑦
Show Less

- **Step 4** Configure agent-based ModelArts access authorization for an administrator to allow ModelArts to access dependent services such as OBS.
  - 1. Log in to the ModelArts console using a tenant account. In the navigation pane, choose **Settings**. The **Global Configuration** page is displayed.
  - 2. Click Add Authorization. On the Add Authorization page, set Authorized User to IAM user, select an administrator account for Authorized To, select Add agency, and select Common User for Permissions. Permissions control is not required for administrators, so use default setting Common User.

Authorized User	IAM user	Federated user	Agency	All users		
Authorized To		٣				
Agency	Use existing	Add agency				
* Agency Name		A	maximum of 50 ager	ncies can be created	. You can create 34 more.	Naming rules
Permissions		Common User			Custom	
	You can use basi manage resource	c ModelArts functions but /s.	not to	You can flexibly a agency. Select th management.	issign permissions to the c is mode for refined permis	reated sions
	View permissions	;		Select required p	ermissions	

Figure 12-15 Configuring authorization for an administrator

- 3. Click Create.
- **Step 5** Test administrator permissions.
  - 1. Log in to the ModelArts management console as the administrator. On the login page, ensure that **IAM User Login** is selected.

Change the password as prompted upon the first login.

2. In the navigation pane of the ModelArts management console, choose **Dedicated Resource Pools** and click **Create**. If the console does not display a message indicating insufficient permissions, the permissions have been assigned to the administrator.

----End

#### **Configuring Permissions for a Developer**

Use IAM for fine-grained control of developer permissions. The procedure is as follows:

- **Step 1** Use a tenant account to create a developer user group **user\_group** and add developer accounts to **user\_group**.
- **Step 2** Create a custom policy.
  - Log in to the management console using a tenant account, hover over your username in the upper right corner, and click **Identity and Access Management** from the drop-down list to switch to the IAM management console.
  - 2. Create custom policy 3 to prevent users from performing operations on ModelArts dedicated resource pools and viewing notebook instances of other users.

In the navigation pane of the IAM console, choose **Permissions** > **Policies/ Roles**. Click **Create Custom Policy** in the upper right corner. On the displayed page, enter **Policy3\_DenyOperation** for **Policy Name**, select **JSON** for **Policy View**, configure the policy content, and click **OK**.

The custom policy **Policy3\_DenyOperation** is as follows. You can copy and paste the content.



**Step 3** Grant the custom policy to the developer user group **user\_group**.

 In the navigation pane of the IAM console, choose User Groups. On the User Groups page, locate the row that contains user\_group, click Authorize in the Operation column, and select Policy1\_IAM\_OBS, Policy2\_AllowOperation, and Policy3\_DenyOperation. Click Next.

#### Figure 12-16 Select Policy/Role

1 Select I	Policy/I	Role (2) Select Scope (3) Finish
Assign s	electe	ed permissions to user_group.
Viev	v Seleo	cted (3) Copy Permissions from Another Project
		Policy/Role Name
	~	Policy3_DenyOperation
	~	Policy2_AllowOperation
~	~	Policy1_IAM_OBS

2. Specify the scope as **All resources** and click **OK**.

#### Figure 12-17 Select Scope

1 Select Policy/Role 2 Select Scope 3 Finish
The following are recommended scopes for the permissions you selected. Sel
Scope
All resources
IAM users will be able to use all resources, including those in enterprise projects,
○ Region-specific projects ⑦
◯ Global services ⑦
Show Less

- **Step 4** Configure agent-based ModelArts access authorization for a developer to allow ModelArts to access dependent services such as OBS.
  - 1. Log in to the ModelArts console using a tenant account. In the navigation pane, choose **Settings**. The **Global Configuration** page is displayed.
  - Click Add Authorization. On the Add Authorization page, set Authorized User to IAM user, select a developer account for Authorized To, add an agency ma\_agency\_develop\_user, set Permissions to Custom, and select OBS Administrator. Developers only need OBS authorization to allow developers to access OBS when using notebook.

Authorized User	IAM user Federated user Agency	All users	
Authorized To	•		
Agency	Use existing Add agency		
* Agency Name	A maximum of 50	agencies can be created. You can create 34 more. Naming rule	5
Permissions	Common User	Custom	
	You can use basic ModelArts functions but not to manage resources.	You can flexibly assign permissions to the created agency. Select this mode for refined permissions management.	
	View permissions	Select required permissions	
	Policy/Role	Module	Description
	OBS Administrator	Data Management   DevEnviron   Training Management	Object Storage Service Administrator

#### Figure 12-18 Configuring authorization for a developer

- 3. Click **Create**.
- 4. On the **Global Configuration** page, click **Add Authorization** again. On the **Add Authorization** page that is displayed, configure an agency for other developer users.

On the **Add Authorization** page, set **Authorized User** to **IAM user**, select a developer account for **Authorized To**, and select the existing agency **ma\_agency\_develop\_user** created before.

- **Step 5** Test developer permissions.
  - 1. Log in to the ModelArts management console as an IAM user in **user\_group**. On the login page, ensure that **IAM User Login** is selected.

Change the password as prompted upon the first login.

2. In the navigation pane of the ModelArts management console, choose **Dedicated Resource Pools** and click **Create**. If the console does not display a message indicating insufficient permissions, the permissions have been assigned to the developer.

Figure 12-19 Insufficient permissions



 $\times$ 

You are not allowed to perform the operation modelarts:pool:create because your permission is insufficient.



----End

# 12.3.3 Viewing the Notebook Instances of All IAM Users Under One Tenant Account

Any IAM user granted with the **listAllNotebooks** and **listUsers** permissions can click **View all** on the notebook page to view the instances of all IAM users in the current IAM project.

#### **NOTE**

Users granted with these permissions can also access OBS and SWR of all users in the current IAM project.

#### **Assigning the Required Permissions**

- Log in to the ModelArts management console as a tenant user, hover the cursor over your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
- On the IAM console, choose Permissions > Policies/Roles from the navigation pane, click Create Custom Policy in the upper right corner, and create two policies.

Policy 1: Create a policy that allows users to view all notebook instances of an IAM project, as shown in **Figure 12-20**.

- Policy Name: Enter a custom policy name, for example, Viewing all notebook instances.
- Policy View: Select Visual editor.
- Policy Content: Select Allow, ModelArts Service, modelarts:notebook:listAllNotebooks, and default resources.

Figure 12-20 Creating a custom policy

Policies/Roles / Create C	ustom Policy				
1 You can use custor	m policies to supplement system-defined policies for fine-grained permissions management. Learn	nore			
* Policy Name 1	policyM3rw				
Policy View 2	Visual editor JSDN				
* Policy Content	∧ 3 O Allow	6 Actions: 1	0 B AI	(Optional) Add request condition	Ū Đ
	Select all modelarts:notebook:listAlINotebooks			X   Q	
	A 🗹 ListOnly				
	6 v modelarts:notebook: listAllNotebooks Query the list of all development environment instances				
	Select Existing Policy/Role      Add Permissions				
Description	Enter a brief description.				
		0256			
Scope	Project-level services				
8	OK Cancel				

Policy 2: Create a policy that allows users to view all users of an IAM project.

- Policy Name: Enter a custom policy name, for example, Viewing all users of the current IAM project.
- **Policy View**: Select **Visual editor**.

- Policy Content: Select Allow, Identity and Access Management, iam:users:listUsers, and default resources.
- 3. In the navigation pane, choose **User Groups**. Then, click **Authorize** in the **Operation** column of the target user group. On the **Authorize User Group** page, select the custom policies created in **2**, and click **Next**. Then, select the scope and click **OK**.

After the configuration, all users in the user group have the permission to view all notebook instances created by users in the user group.

If no user group is available, create a user group, add users using the user group management function, and configure authorization. If the target user is not in a user group, you can add the user to a user group through the user group management function.

#### **Starting Notebook Instances of Other IAM Users**

If an IAM user wants to access another IAM user's notebook instance through remote SSH, they need to update the SSH key pair to their own. Otherwise, error **ModelArts.6789** will be reported. For details about how to update a key pair, see **Modifying the SSH Configuration for a Notebook Instance**.

Erro message: ModelArts.6789: Failed to use SSH key pair KeyPair-xxx. Update the key pair and try again later.

# 12.3.4 Logging In to a Training Container Using Cloud Shell

#### **Application Scenario**

You can use Cloud Shell provided by the ModelArts console to log in to a running training container.

#### Constraints

Only dedicated resource pools support Cloud Shell. The training job must be in the **Running** state.

#### Preparation: Assigning the Cloud Shell Permission to an IAM User

- 1. Log in to the management console as a tenant user, hover the cursor over your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
- 2. On the IAM console, choose **Permissions** > **Policies/Roles** from the navigation pane, click **Create Custom Policy** in the upper right corner, and configure the following parameters.
  - Policy Name: Enter a custom policy name, for example, Using Cloud Shell to access a running job.
  - Policy View: Select Visual editor.
  - Policy Content: Select Allow, ModelArts Service, modelarts:trainJob:exec, and default resources.

Figure 12-21 Creating a custom policy

(	0					
* Policy Name	Usin	g Cloud Shell to access	a running job			
Policy View	2	/isual editor	JSON			_
+ Policy Content	^	Allow		ModelArts Service	Actions: 1	
		Select all mode	larts:trainJob:exec			
		^ ☑ ReadWrite				
		Modelarts:tra Access runnin	inJob:exec g training jobs by CloudShell			
	⊕ Se	lect Existing Policy/Role	Add Permissions			
	Ente	r a brief description.				
Description						

3. In the navigation pane, choose **User Groups**. Then, click **Authorize** in the **Operation** column of the target user group. On the **Authorize User Group** page, select the custom policies created in **2**, and click **Next**. Then, select the scope and click **OK**.

After the configuration, all users in the user group have the permission to use Cloud Shell to log in to a running training container.

If no user group is available, create a user group, add users using the user group management function, and configure authorization. If the target user is not in a user group, you can add the user to a user group through the user group management function.

#### **Using Cloud Shell**

- 1. Configure parameters based on **Preparation: Assigning the Cloud Shell Permission to an IAM User**.
- 2. On the ModelArts console, choose **Training Management** > **Training Jobs** from the navigation pane.
- 3. In the training job list, click the name of the target job to go to the training job details page.
- 4. On the training job details page, click the **Cloud Shell** tab and log in to the training container.

Verify that the login is successful, as shown in the following figure.

#### Figure 12-22 Cloud Shell page

Events	Logs	Cloud Shell	Res	ource l	Jsages	Evalu	ation Results	Tags										
worker-0		•	Reconnect	t	Cor	nnection <												
(pytorch NGC-DL-0 <b>bin</b>	ı) ma-use CONTAINEF	er@modelar R-LICENSE	ts-job- boot cache	2a70d dev etc	ale-ea home lib	87-4ee4 lib64 media	-aele-55df84 mnt modelarts-j	46e7f41 job-2a	1-worke 70dale	er-0:/\$ -ea87-4	<b>ls</b> lee4-ae1	le-55d	f846e7f4	opt proc	root run	sbin srv	sys tmp	usr var
(pytorch	ı) ma-use	er@modelar	ts-job-	2a70d	ale-ea	87-4ee4	-ae1e-55df84	46e7f41	1-worke	er-0:/\$								

If the job is not running or the permission is insufficient, Cloud Shell cannot be used. In this case, locate the fault as prompted.

#### D NOTE

An exception may occur when some users log in to the Cloud Shell page. Click **Enter** to rectify the fault.

Figure 12-23 Abnormal path

ind/model/1\$ @97c6-b87f-4410-9f74-18a8b1d0ff9d-59x451kz-6548f94565-lrjgs:/home/mi

## 12.3.5 Prohibiting a User from Using a Public Resource Pool

This section describes how to control the ModelArts permissions of a user so that the user is not allowed to use a public resource pool to create training jobs, create notebook instances, or deploy inference services.

#### Context

Through permission control, ModelArts dedicated resource pool users can be prohibited from using a public resource pool to create training jobs, create notebook instances, or deploy inference services.

To control the permissions, configure the following permission policy items:

- modelarts:notebook:create: allows you to create a notebook instance.
- modelarts:trainJob:create: allows you to create a training job.
- modelarts:service:create: allows you to create an inference service.

#### Procedure

- 1. Log in to the management console as a tenant user, hover the cursor over your username in the upper right corner, and choose **Identity and Access Management** from the drop-down list to switch to the IAM management console.
- In the navigation pane, choose Permissions > Policies/Roles. On the Policies/ Roles page, click Create Custom Policy in the upper right corner, configure parameters, and click OK.
  - **Policy Name**: Configure the policy name.
  - **Policy View**: Select **Visual editor** or **JSON**.
  - Policy Content: Select Deny. In Select service, search for ModelArts and select it. In ReadWrite under Actions, search for modelarts:trainJob:create, modelarts:notebook:create, and modelarts:service:create and select them. All: Retain the default setting. In Add request condition, click Add Request Condition. In the displayed dialog box, set Condition Key to modelarts:poolType, Operator to StringEquals, and Value to public.

#### Figure 12-24 Create Custom Policy (visual editor)

* 策略名称 🛛 1	不允许用户使用公共资源	池			
策略配置方式 2	可视化视图	JSON视图	•		
* 策略内容	へ 3 🕓 拒絶	4 🕒 ModelArts	5 O 3项操作	C 所有资源	● 请求条件 (可选)
	■ 选择所有操作	请输入关键字			Q
	▶ □ 只读	共23项操作			
	^ 図 写 共	3项操作 已选择3项			
	✓ modelarts 创建训练化	s:trainJob:create ⊨业	✓ modelarts:notebook:create 创建notebook开发环境	■ modelarts:service:create 部署模型服务	
	> □ 列表	共21项操作			

Figure 12-25 Add Request Condition (visual editor)

* 策略名称	不允许用户使用公共资	20 No. 10 No.
策略配置方式	可视化视图	JSON规图
* 策略内容	へ () 拒絶	C ModelArts C 3项操作 C 所有资源 C 请求条件 (可选)
	(→ 添加条件	
	④ 从已有策略复制	x
策略描述	请输入策略描述(	态加请求条件 
		集件離 modelarts:poolType *
		章游符 StringEquals •
作用范围	项目级服务	í public
		<ul> <li>⑦ 滞如</li> </ul>
		allian 現2)尚

The policy content in JSON view is as follows:



3. In the navigation pane, choose **User Groups**. On the **User Groups** page, locate the row containing the target user group and click **Authorize** in the **Operation** column. On the **Authorize User Group** page, select the custom policy created in **2** and click **Next**. Then, select the scope and click **OK**.

After the configuration, all users in the user group have the permission to view all notebook instances created by users in the user group.

If no user group is available, create one, add users to it through user group management, and configure authorization for the user group. If the target user is not in a user group, add the user to a user group through user group management.

4. Add the policy to the user's agency authorization. This prevents the user from breaking the permission scope through a token on the tenant plane.

In the navigation pane, choose **Agencies**. Locate the agency used by the user group on ModelArts and click **Modify** in the **Operation** column. On the **Permissions** tab page, click **Authorize**, select the created custom policy, and click **Next**. Select the scope for authorization and click **OK**.

Figure 12-26 Modifying authorization

全局	配置 ⑦			
	添加授权	清空授权		
	用户名		授权类型	授权内容
	所有用户		委托	modelarts 查看权限
	权限详	ŧ.		
	用户名	所有用户		
	圈托名称	modelarts		
	雲托权限	14项权限 去IAM修改委托权限		

#### Verification

Log in to the ModelArts console as an IAM user, choose **Training Management** > **Training Jobs**, and click **Create Training Job**. On the page for creating a training job, only a dedicated resource pool can be selected for **Resource Pool**.

Log in to the ModelArts console as an IAM user, choose **DevEnviron** > **Notebook**, and click **Create**. On the page for creating a notebook instance, only a dedicated resource pool can be selected for **Resource Pool**.

Log in to the ModelArts console as an IAM user, choose **Service Deployment** > **Real-Time Services**, and click **Deploy**. On the page for service deployment, only a dedicated resource pool can be selected for **Resource Pool**.

# 12.4 FAQ

# 12.4.1 What Do I Do If a Message Indicating Insufficient Permissions Is Displayed When I Use ModelArts?

If a message indicating insufficient permissions is displayed when you use ModelArts, perform the operations described in this section to grant permissions for related services as needed.

The permissions to use ModelArts depend on OBS authorization. Therefore, ModelArts users require OBS system permissions as well.

 For details about how to grant a user full permissions for OBS and common operations permissions for ModelArts, see Configuring Common Operations Permissions.  For details about how to manage user permissions on OBS and ModelArts in a refined manner and configure custom policies, see Creating a Custom Policy for ModelArts.

#### **Configuring Common Operations Permissions**

To use ModelArts basic functions, assign the **ModelArts CommonOperations** permission on project-level services to users. Since ModelArts depends on OBS permissions, assign the **OBS Administrator** permission on global services to users.

The procedure is as follows:

**Step 1** Create a user group.

Log in to the IAM console and choose **User Groups** > **Create User Group**. Enter a user group name, and click **OK**.

**Step 2** Configure permissions for the user group.

In the user group list, locate the user group created in step 1, click **Authorize**, and perform the following operations.

1. Assign the **ModelArts CommonOperations** permission on project-level services to the user group and click **OK**.

**NOTE** 

The permission takes effect only in assigned regions. Assign permissions in all regions if the permission is required in all regions.

- 2. Assign the **OBS Administrator** permission on global services to the user group and click **OK**.
- **Step 3** Create a user on the IAM console and add the user to the user group created in step 1.
- **Step 4** In the authorized region, perform the following operations:
  - Choose Service List > ModelArts. Choose Dedicated Resource Pools. On the page that is displayed, select a resource pool type and click Create. You should not be able to create a new resource pool.
  - Choose any other service in **Service List**. (Assume that the current policy contains only **ModelArts CommonOperations**.) If a message appears indicating that you have insufficient permissions to access the service, the **ModelArts CommonOperations** policy has already taken effect.
  - Choose Service List > ModelArts. On the ModelArts console, choose Data Management > Datasets > Create Dataset. You should be able to access the corresponding OBS path.

----End

#### **Creating a Custom Policy for ModelArts**

In addition to the default system policies of ModelArts, you can create custom policies, which can address OBS permissions as well.

You can create custom policies using either the visual editor or JSON views. This section describes how to use a JSON view to create a custom policy to grant

permissions required to use development environments and the minimum permissions required by ModelArts to access OBS.

#### **NOTE**

A custom policy can contain actions for multiple services that are accessible globally or only for region-specific projects.

ModelArts is a project-level service, but OBS is a global service, so you need to create separate policies for the two services and then apply these policies to the users.

1. Create a custom policy for minimizing permissions for OBS that ModelArts depends on.

Log in to the IAM console, choose **Permissions** > **Policies/Roles**, and click **Create Custom Policy**. Configure the parameters as follows:

- **Policy Name**: Choose a custom policy name.
- Policy View: JSON
- Policy Content: Follow the instructions in Example Custom Policies of OBS.
- 2. Create a custom policy for the permissions to use ModelArts development environments. Configure the parameters as follows:
  - **Policy Name**: Choose a custom policy name.
  - Policy View: JSON
  - Policy Content: Follow the instructions in Example Custom Policies for Using the ModelArts Development Environment. For the actions that can be added for custom policies, see *ModelArts API Reference* > "Permissions Policies and Supported Actions" > "Introduction".
- 3. After creating a user group on the IAM console, grant the custom policy created in 1 to the user group.
- 4. Create a user on the IAM console and add the user to the user group created in **3**.
- 5. In the authorized region, perform the following operations:
  - Choose Service List > ModelArts. On the ModelArts console, choose
     Data Management > Datasets. If you cannot create a dataset, the permissions (for using the development environment) granted only to ModelArts users have taken effect.
  - Choose Service List > ModelArts. On the ModelArts console, choose DevEnviron > Notebook and click Create. If you can access the OBS path specified in Storage, the OBS permissions have taken effect.

#### **Example Custom Policies of OBS**

The permissions to use ModelArts require OBS authorization. The following example shows the minimum OBS required, including the permissions for OBS buckets and objects. After being granted the minimum permissions for OBS, users can access OBS from ModelArts without restrictions.

```
"Version": "1.1",
"Statement": [
{
"Action": [
"obs:bucket:ListAllMybuckets",
```



## **Example Custom Policies for Using the ModelArts Development Environment**



# **13** Best Practices

# 13.1 Migrating a Locally Developed MindSpore Model to the Cloud for Training

This section describes how to use PyCharm Toolkit to debug and train your local MindSpore model code on ModelArts.

Before you start, complete the requirements described in **Prerequisites**. The procedure in this case is as follows:

Step 1: Installing PyCharm Toolkit and Logging In to It

Step 2: Using PyCharm for Local Development and Debugging

Step 3: Using ModelArts Notebook for Development and Debugging

Step 4: Using PyCharm to Submit a Training Job to ModelArts

**Step 5: Releasing Resources** 

#### Prerequisites

- PyCharm 2019.2 or later has been installed on a local server. Either the community or professional edition will do. **Download PyCharm Toolkit** and install it on the local server.
  - PyCharm of only the professional edition can be used to access notebook instances.
  - PyCharm of both the community and professional editions can be used to submit training jobs.
- Access keys (AK and SK) of the current account are available. To create access keys, see **Creating Access Keys (AK and SK)**.
- Access authorization has been configured for the current account. For details, see **Configuring Access Authorization**.

#### Step 1: Installing PyCharm Toolkit and Logging In to It

1. Install PyCharm Toolkit.

In PyCharm, choose **File** > **Settings** > **Plugins**, search for **ModelArts** in Marketplace, and click **Install**.

Figure	13-1	Installation	using	Marketplace
--------	------	--------------	-------	-------------

Settings		
Q•	Plugins	Marketplace Installed
> Appearance & Behavior	Q+ ModelArts	×
Кеутар	Search Results (1)	Sort By: Relevance 🗸 🛛
> Editor	8 8 Hugurai El MadalArta	
Plugins 🔳		
> Version Control		
> Project: models		Plugin hon
> Build, Execution, Deployment		Al develop
> Languages & Frameworks		ModelArts
> Tools		With the P

- 2. Log in to PyCharm Toolkit.
  - a. Open Edit Credential.

After the plug-in is installed, **ModelArts** is displayed on the IDE menu bar. Click **ModelArts** and choose **Edit Credential**.

b. Contact the region operations company to obtain the YAML configuration file and host information.

Add the host information to the hosts file on your local PC. The hosts file is usually in C:\Windows\System32\drivers\etc.

In the **Edit Credential** dialog box, click **Config** to import the YAML configuration file. After the file is imported, the message **Import successful** is displayed, indicating that the region information is configured.

c. Verify the login information.

Enter the created access keys (AK and SK) in the toolkit area and click **OK**. If the following information is displayed, the login is successful.

To create access keys, see Creating Access Keys (AK and SK).

Figure 13-2 Login succeeded

O Validate Credential Success

The credential is valid.

#### Step 2: Using PyCharm for Local Development and Debugging

1. Download code to a local directory.

In this case, the image classification model ResNet50 is used as an example, which is stored in **./models/official/cv/resnet/**. # Use the Terminal on your local PC to download the code. git clone https://gitee.com/mindspore/models.git -b v1.5.0



Figure 13-3 Downloading code to a local directory

2. Configure the development environment on a local PC. *# Install MindSpore on the Terminal of PyCharm.* pip install mindspore



#### Figure 13-4 Installing the ResNet dependency



3. Prepare a dataset.

In this case, a dataset with five categories of flowers for recognition is used. **Download the dataset** and decompress it to the project directory. Create a dataset folder and save the decompressed dataset to the dataset folder.

#### Figure 13-5 Preparing a dataset



4. Configure a PyCharm interpreter and input parameters.

Click **Current File** in the upper right corner and choose **Edit Configuration**. The **Run/Debug Configurations** dialog box is displayed. In the dialog box, click **+** and select **Python**.



Figure 13-6 Entrance to a PyCharm interpreter

Select **train.py** for **Script path**, configure **Parameters** (shown in the following command), select a Python interpreter, and click **OK**.

--net\_name=resnet50 --dataset=imagenet2012 --data\_path=../../../dataset/flower\_photos/ -class\_num=5 --config\_path=./config/resnet50\_imagenet2012\_config.yaml --epoch\_size=1 -device\_target="CPU"

Run/Debug Configurations				
+ - 🖻 📭 🐙				
🗠 🍦 Python	<u>N</u> ame: train		Allow parallel r <u>u</u> n <u>S</u> tore as project f	ile 🔍
😴 train				
	Configuration Logs			
	Python interpreter:	O Python 3.8 C:\ProgramData\Anaconda3\p		
	✓ Add source roots to			
	Run with Python Con			
	Redirect input from:			
	▼ <u>B</u> efore launch			
Edit configuration templates				
?				

#### Figure 13-7 Configuring a PyCharm interpreter

#### **NOTE**

Configure **Parameters** according to the README file. **device\_target="CPU"** indicates that the system runs on CPUs, and **device\_target="Ascend"** indicates that the system runs on Ascend.

5. Develop and debug code locally.

Local CPUs typically feature average compute power and small memory, which may lead to memory overflow. Therefore, change the value of **batch\_size** in **config/resnet50\_imagenet2012\_config.yaml** from **256** to **32** for fast training job execution.

#### Figure 13-8 Changing the batch\_size value

✓ ■ resnet	13	device_target: "Ascend"
> 🖿 ascend310_infer		<pre>checkpoint_path: "./checkpoint/"</pre>
🗸 🖿 config		checkpoint_file_path: ""
<del>州</del> resnet18_cifar10_config.yaml		
📶 resnet18_cifar10_config_gpu.yaml		# =====================================
🚛 resnet 18_imagenet 2012_config. yaml		# Training options
🛻 resnet 18_imagenet 2012_config_gpu. yaml		optimizer: "Momentum"
攝 resnet34_imagenet2012_config.yaml		infer label: ""
resnet50_cifar10_config.yaml		class num: 1001
机 resnet50_imagenet2012_Ascend_Thor_config.yaml		
提 resnet50_imagenet2012_Boost_config.yaml	22	Datch_size: 32
🛻 resnet50_imagenet2012_config.yaml	23	loss_scale: 1024

During AI development, the development of datasets and models are irrelevant to hardware specifications and takes the longest time when being compared with other phases. Therefore, develop and debug datasets and models on local CPUs.

**NOTE** 

In this case, the sample code supports training on CPUs. Therefore, the entire training can be executed on CPUs. If your code supports training only on GPUs or Ascend, an error may be reported. In this case, use a notebook instance for debugging code.

After setting a breakpoint, click the debugging icon to debug code step by step and view variable values.

Figure 13-9 Debugging icon







Click the execution icon and check whether the training job is running properly based on logs.

#### Figure 13-11 Execution icon



#### Figure 13-12 Training logs



#### Step 3: Using ModelArts Notebook for Development and Debugging

The features of using ModelArts notebook for development and debugging are as follows:

- Consistent in-cloud and on-premises environments
- One-click configuration
- Remote code debugging
- On-demand resource usage

#### **NOTE**

The functions described in this section are supported by PyCharm of only the professional edition. If a PyCharm community edition is used, go to **Step 4: Using PyCharm to Submit a Training Job to ModelArts** to create a training job.

- 1. Access a notebook instance.
  - a. Create or open an existing Ascend-powered notebook instance. To create a notebook instance, see **Creating a Notebook Instance**. Notebook specifications are as follows:

Image: mindspore1.7.0-cann5.1.0-py3.7-euler2.8.3

Resource Type: Public resource pool

#### Type: Ascend

Flavor: Ascend: 1\*Ascend910|CPU: 24vCPUs 96 GB

#### Storage: Default

#### Remote SSH: enabled

Key Pair: Select an existing key pair, or click Create on the right to create one.

- 2. Access the notebook instance using the toolkit.
  - On the IDE menu bar, choose ModelArts > Notebook > Remote Config. On the page that is displayed, select the notebook instance to be accessed.

#### 

If **Connect To Remote** is unavailable, create a notebook instance with remote SSH enabled. For details, see **Creating a Notebook Instance**.

If the fault persists, check whether the PyCharm Toolkit version is the latest one. If not, download the latest version.

Before downloading PyCharm Toolkit, clear the browser cache. If PyCharm Toolkit of an earlier version has been downloaded, the browser cache may lead to the failure in downloading a new version.

b. In KeyPair, select the key of the target notebook instance. Then, click Apply to perform one-click remote notebook configuration. After a period of time, a confirmation dialog box will be displayed, asking you to restart the IDE. Click OK to restart the IDE. The configuration takes effect after the restart.

🖺 DevContainer List	×
DevConatiner:	best_prac 🔻
RunningStatus:	STOPPED
Flavor:	modelarts.bm.d910.xlarge.1
ImageName:	best_prac_image
SshUrl:	ssh://ma-use
KeyPairName:	KeyPair-shp
KeyPair:	D:\Develop\KeyPair-s pem
PathMappings:	/home/ma-user/work/models
	<b>Apply</b> Cancel

Figure 13-13 Configuration for accessing notebook using PyCharm Toolkit

#### D NOTE

- **KeyPair**: Select the locally stored key pair of the notebook instance for authentication. The key pair created during the notebook instance creation is saved in your browser's default downloads folder.
- **PathMappings**: Synchronization directory for the local IDE project and notebook, which defaults to **/home/ma-user/work**/*Project name* and is adjustable.
- 3. Synchronize code and data to the notebook instance.
  - a. Synchronize code to the notebook instance.

Right-click the **resnet** folder and choose **Deployment** > **Upload to** from the shortcut menu to upload code to the notebook instance.

90		inguining code by	incini of inzuci		
>	psenet			364	
>	🖿 resnet	New		330	
>	resnet50_quant				
>	resnet_thor	X Cu <u>t</u>	Ctrl+X		config
>	resnext	值 <u>С</u> ору	Ctrl+C		🛛 🏳 🗌 logger
>	resnext101	Copy Path/Reference	ce		
>	🖿 retinaface_resne	🛱 <u>P</u> aste	Ctrl+V		# define c
>	🖿 retinanet	Ead Users	ALL 17		time_cb =
>	🖿 se_resnext50	Find <u>o</u> sages			loss_cb =
>	shufflenetv1		Ctri+Snitt+F		cb = [time
>	shufflenetv2	Repl <u>a</u> ce in Files	Ctri+Sniπ+K	365	cknt save
>	simclr	Inspect Code		366	if config
>	simple_pose	<u>R</u> efactor			aknt o
>	squeezenet	Clean Python Comp	oiled Files		
>	srcnn				CONTIC
>	ssd	BOOKMARKS			
>	tinydarknet	<u>R</u> eformat Code	Ctrl+Alt+L		
2	unet	Optimi <u>z</u> e Imports	Ctrl+Alt+O		ckpt_c
2	unet3d	<u>D</u> elete	Delete		⊂ cb +=
~	vgg16	Open In			run_eval(t
~	vit	Open in			# train mo
(	warpctc	Local <u>H</u> istory			if config.
	xception	<u>G</u> it			confic
	yolov3_darknet5	Repair IDE on File			dataset_si
Ś	volov2_respect19	🕄 Reload from Disk			config.pre
Ś		Compare With	Ctel L D		model trai
Ś			Curto		induct et al
ŕ	respet zin	Mark Directory as		300	
> •	ann	<b>↓↑</b> Deployment	>	→ Upload to bes	t_prac
	lite			Upload to	Ctrl+Alt+Shift+X
>	nlp			<u> <b>↓</b>   D</u> ownload fror	n best_prac חדנ
>	recommend			Download fror	n
>	utils			Sunc with Dool	oved to best pres
🖿 re	esearch			Sync with Depl	eved to best_prac ==
L ut	ils			Sync with Depi	et(

Figure 13-14 Configuring code synchronization

- b. Synchronize data to the notebook instance.
  - (Recommended) Method 1: Upload the dataset package to notebook and decompress the package.

Right-click the dataset package and choose **Deployment** > **Upload to** from the shortcut menu to upload the package to the notebook instance. Run the following command to decompress the dataset package in the notebook instance:

tar -zxvf work/models/dataset/flower\_photos.tgz

_	5					5		•		5					
PC	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	R <u>u</u> n			<u>W</u> indow	<u>M</u> odelAr	ts <u>H</u> el	p m		
m	odels $\rangle$	datas	et 👌 🕌					<u>T</u> as	ks 81	Contexts					
н	🔲 Pr	oiect						🐣 Co	de Wi	th Me	Ctrl+Sł	hift+Y	n.pv ×		
ē ē	× 🖿	models	: D-\D					IDE		oting Cons					
	>		e					XM	L Acti						
	>	.qith	ub					Ma	rkdov	wn Conver	rter				
umit	>	🖿 .jenk						<u>∔† De</u>						if c	onfig.opti
5	>	com	munity												from src.l
<del>۰</del>	×	🖿 data						Vay	jrani						damping =
		> 🖿 fl	ower_	photos				HT	rp Cli	ent					split_indi
		f î	ower_	photos.tgz				🍺 Spa							opt = thor
	>	how	_to_co	ntribute				💰 Pyt		r Debug (	Console				
	>	ottic	ial					Syn	c Pytl	10n Requi	rements				model = Co
		rese	arch												
		uuis 4. aitia	nora												
		no .giug L CON	ITRIRI	ITING md											config.run
			ITRIBL	JTING CN.r	nd			Go -	ogle <i>i</i>	App Engin					logger.war
			NSE					🎨 Ор	en CP	rofile sna	pshot				
		offic													fine callb
		🖞 owi	NERS												
		<del>4</del> Rea	DME.n	nd								train_	net() →	it config	.optimizer ==
	Termin	ial: _d	ev-mo	delhuawe	ei.com⇒	<u>+                                    </u>									
	(Mind	Spore	) [ma	i-user ∼]	tar	-zxvf wo	rk/m	odels/	data	iset/flo	wer_pho	tos.t	gz		
	flowe	r_pho	tos/												
	flowe	r_pho	tos/r	oses/											
	flowe	r_pho	tos/r	oses/148	310868	100_87eb	739f	26_m.j	pg						
	flowe	r_pho	tos/r	oses/144	460904:	16_f0cad	15fde	4.jpg							

Figure 13-15 Uploading a dataset package to a notebook instance

Method 2: Upload the data folder to the notebook instance.

Similar to uploading code to a notebook instance, directly upload the data folder. (In this case, there are a large number of images in the dataset. Uploading images using the IDE is time-consuming. Method 1 is recommended.)

5		1 5					
>	dataset	New			342	Ĭ	
Ĺ		94.0.					
		R Cut	Ctrl+X				els
Ś	research	∎ <u>С</u> ору	Ctrl+C				
<i></i>	aitianore	Copy Path/Referen	ce				
		⊔ <u>P</u> aste	Ctrl+V				
		Find <u>U</u> sages	Alt+F7				
		Find in Files	Ctrl+Shift+F				if
	official.zip	Repl <u>a</u> ce in Files	Ctrl+Shift+R				~~~~~
	d OWNERS	Inspect Code					
	뤮 README.md	Refactor					
	撮 README_CN.m	Clean Python Com	oiled Files				
> III	I External Libraries						
- <b>*</b>	Scratches and Cor	Bookmarks					
		<u>R</u> eformat Code	Ctrl+Alt+L				
		Optimi <u>z</u> e Imports	Ctrl+Alt+O				
		<u>D</u> elete	Delete				
		Open In					
		Local <u>H</u> istory					# r
		<u>G</u> it					
		Repair IDE on File					
		S Reload from Disk					LU2 06
		📌 Compare With	Ctrl+D				CD okr
					305		скр ч е
		Mark Directory as					11
		<b>↓</b> T Deployment		↑ Upload to best_	prac	01.10	
				Upload to	Ctrl+Alt	+Shift+X	5
				<u> <u> </u></u>	best_prac		
				Download from.			
				S Sync with Deploy	yed to bes	t_prac	
				Sync with Deploy	yed to		
					575		rur

Figure 13-16 Uploading a data folder to a notebook instance

#### **▲** CAUTION

- If a dataset is in GB, it is a good practice to upload the dataset to OBS and then to the target notebook instance through OBS. PyCharm is applicable only to upload small files.
- Use a small dataset subset for debugging. This facilitates rapid data synchronization and loading.
- 4. Configure the Python interpreter.

```
Modify Parameters and select a Python interpreter.
--net_name=resnet50 --dataset=imagenet2012 --data_path=../../../dataset/flower_photos/ --
class_num=5 --config_path=./config/resnet50_imagenet2012_config.yaml --epoch_size=1 --
device_target="Ascend"
```

5	
Run/Debug Configurations	×
+ — № № ↓8 ~ ∲Python ∳train	
	Configuration Logs
	Interpreter options: O Project Default (Python 3.8 (best_prac_pycharm_ascend)) C\Uters\00519530, condatenvs\best_prac_pycharm_ascend/python.exe
	🖉 Add content roots to
	🗳 best prac Python 3.7.6 (sttp://ma-user
	✓ <u>B</u> efore launch

Figure 13-17 Configuring the Python interpreter

5. Install the notebook dependency.

Choose **Tool** > **Start SSH Section** to install the dependency software.

# Access MindSpore. source /home/ma-user/anaconda3/bin/activate MindSpore # Install the ResNet dependency. pip install -r work/models/official/cv/resnet/requirements.txt



Figure 13-18 Installing the notebook dependency

6. Debug and run data.

After the interpreter is configured, PyCharm can directly use the Python interpreter and hardware of the remote notebook instance. This allows you to experience the real hardware environment locally and perform debugging and verification throughout the entire process.
#### 

In Ascend-powered cases, an error may occur.

ModuleNotFoundError: No module named 'te'

The cause is that PyCharm **PYTHONPATH** overwrites **PYTHONPATH** specified in notebook environment variables. To resolve this issue, add the path to the TE package to PyCharm **PYTHONPATH**.

Run the **pip show te** command to obtain the path to the TE package. For example, if the path to the TE package is **/usr/local/Ascend/nnae/5.0.3/ compiler/python/site-package**, the **PYTHONPATH** value is **\$PYTHONPATH:/usr/local/Ascend/nnae/5.0.3/compiler/python/site-package**.

Figure 13-19 Adding the path to the TE package to PyCharm PYTHONPATH

	<u>P</u> arameters:	_config.yaml —epoch_size=1 —device_target="Ascend" + "				
	Environment					
	Environment variables:	/local/Ascend/nnae/5.0.3/compiler/python/site-packages				
P	🖺 Environment Variables X					
ι	J <u>s</u> er environment variables:					
	+ - 🖻 🛱					
	Name	Value				
	PYTHONUNBUFFERED	1				
	PYTHONPATH	<pre>\$PYTHONPATH:/usr/local/Ascend/nnae/5.0.3/</pre>				
			]			
	Include system environm	nent variables:				

7. Save the development environment image.

After the notebook instance is debugged, the notebook instance contains all dependent environments for training models. Save the debugged development environment as an image. To do so, select the target notebook instance and choose **More** > **Save Image** in the **Operation** column. The notebook instance will be frozen during the image saving. It requires several minutes to unfreeze the instance. (You only need to save a notebook instance once.)





Saved images are available in **Image Management** on the ModelArts management console. Obtain a complete image name in **SWR Address** on the image details page.

#### Figure 13-21 Viewing a saved image

 
 Image Version/ID /E
 Status
 Resource Type
 Size
 SWR Address

 0.10 e.st.et
 © A valiable
 ASCEND
 4.51 GB
 Image Version/ID /E
 Image Version/ID /E
 Image Version/ID /E
 SWR Address
 Image Version/ID /E
 Image Version/ID /E
 SWR Address
 Image Version/ID /E
 Image Version/ID /E
 Image Version/ID /E
 SWR Address
 Image Version/ID /E
 Image Version/ID /E

#### D NOTE

After debugging code and saving an image, release the notebook instance if it not required.

- 8. Access, stop, start, or disconnect a notebook instance.
  - Access a notebook instance.

If a notebook instance is in the green triangle state, the instance is running (but not connected to PyCharm). Click the instance name. The instance state changes to a green tick, indicating that it has been connected to PyCharm.

Figure 13-22 A running instance

<u>M</u> odelArts	<u>H</u> elp		models [D:\Develop\mo
🗸 Edit Cree	dential		
Noteboo	ok	>	bestprac
Training	Job	>	Remote Config נטאא_אטאלי F:

- Stop a notebook instance.

If a notebook instance is in the green tick state, the instance has been connected to PyCharm. Click the instance name. The instance state changes to a yellow exclamation mark, indicating that the notebook instance is stopped.

<u>M</u> odelArts	<u>H</u> elp		models [D:\Develop\n	nc
🗸 Edit Cre	dential			
Noteboo	ok -	>	🗸 bestprac	
Training	Job	>	Remote Config	Fi

- Start a notebook instance.

If a notebook instance is in the yellow exclamation mark state, the instance is stopped. Click the instance name. The instance state changes to a green tick, indicating that it has been started and connected to PyCharm. (The default connection holding duration is 4 hours.)

Figure 13-23 An instance that is running and connected to PyCharm





- Disconnect a notebook SSH connection from PyCharm Toolkit.

Choose **File** > **Settings** > **Tool** > **SSH Configurations**, click the instance to be disconnected, select -, and click **OK**. Then, the notebook instance in **ModelArts** > **Notebook** on the IDE menu bar is disconnected.

After this step is performed, the notebook instance will not be available in PyCharm Toolkit, but it is still available on the management console. To release notebook resources, log in to the ModelArts management console and delete the notebook instance on the **Notebook** page.

PC Settings							×
		Tools → SSH Configura	ations 🔳				$\leftarrow \rightarrow$
Project: models		2) + <mark>-</mark> 恒 ∠	Visible only for the second	is project			
> Build, Execution, Deployment		bestprac		dev-modelart			31700
> Languages & Frameworks							
✓ Tools						Local port:	<dynami< td=""></dynami<>
			Authentication type:	Key pair OpenSSH			
Web Browsers and Preview			Private key file:	D:\Develop\KeyPai			
File Watchers			Passphrase:				ssphrase
				Z Parca config fila	,/sch/config		·
> Database				Test Connection			
SSH Configurations	0						
			Z Sand kaon-ali			second	
Code With Me				te messages every			- -
			📃 Strict host key	checking			
> Diff & Merge				known hosts file			
External Documentation							
			> HTTP/SOCKS Prox	y			
Python Integrated Tools							
Python Scientific							
Server Certificates							
							Apply

**Figure 13-25** Disconnecting a notebook SSH connection from PyCharm Toolkit

#### Step 4: Using PyCharm to Submit a Training Job to ModelArts

The ModelArts training platform provides massive compute power specifications and training optimization. You can submit a training job in PyCharm based on the locally debugged code and the saved development environment image.

1. Create an OBS bucket and upload data to the bucket.

Training jobs run on ModelArts. Therefore, upload the training data and code to the in-cloud notebook instance. Upload downloaded training data to OBS through OBS Browser+.

Create the **data-flower** bucket, upload the **flower\_photos** folder with training data to OBS through OBS Browser+, and create the **train** folder to store training job data.

#### Figure 13-26 Uploading data to OBS

OBS Browse	er+	☆ Favorite
Object Storage	← → ↑ Bucket List / data-flower	er
Parallel File System	• cn-north-7   Total objects: 3681	Used storage space: 222.21 MB
External Bucket	⊥ Upload ☐ Create Folder	L Download ☐ Copy More
Auto Upload	Object Name ↓Ξ	Storage Class ↓Ξ Siz
Task	🗌 🗮 flower_photos	
Management	🗌 📄 train	

2. Create a training job.

On the IDE menu bar, choose **ModelArts** > **Training Job** > **New** to create a training job.

#### Figure 13-27 Creating a training job



The following table describes the parameters on the **Create Training Job** page.

#### Table 13-1 Parameters

Parameter	Description
JobName	Name of a training job, which defaults to the job creation time
Al Engine	Training engine, including the engine type and version
Boot File Path	Path to the code for booting local training
Code Directory	Local code directory
Image Path(optional)	(Optional) SWR address of a custom image (The engine of the custom image is the same as that of the preset training image.)
Data Obs Path	OBS path to a dataset (The data must be uploaded to OBS beforehand.)

Parameter	Description
Training Obs Path	OBS path (must be reachable), which is used to store the code, trained models, and logs
Running Parameters	Parameters received by a training script
Specifications	Compute specifications. Select NPU:1*Ascend 910CPU:24*vCPUs 96GB for this case.
Compute Node	Number of nodes ( <b>1</b> for single-node training by default)

PyCharm allows you to create a training job using either a preset image or a custom image.

- Use a preset image to create a training job.

Configure the following training parameters in **RunningParameters** and configure other parameters based on site requirements: --net\_name=resnet50 --dataset=imagenet2012 --enable\_modelarts=True --class\_num=5 -config\_path=/home/ma-user/modelarts/user-job-dir/resnet/config/ resnet50\_imagenet2012\_config.yaml --epoch\_size=10 --device\_target=Ascend

After configuring the parameters, click **Apply and Run** to create the training job.

🖺 Edit Training Job (	Configurations			×
* JobName:	MA-new-models-09-2	6-15-11-774		
Job Description:				
	Frequently-used C			
	AI Engine:	Ascend-Powered-Engine 🔹	mindspore_1.7.0-cler_2.8.3-aarch64 🔻	
Algorithm Source:	Boot File Path:	D:\Develop\models\official\cv\resne	et\train.py	
	Code Directory:	D:\Develop\models\official\cv\resne	et 📂	
	Image Path(optional):			
* Data OBS Path:	obs://data-flower/flow	ver_photos/		
Training OBS Path:	obs://data-flower/trai			
Running Parameters:	net/config/resnet5	9_imagenet2012_config.yamlep	och_size=10device_target=Ascend $_{\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!$	
Specifications:	NPU:1*Ascend-910 CP	U:24*vCPUs 96GB		
* Compute Nodes:		Debug	gger:	
			Apply and Run Cancel App	

Figure 13-28 Using a preset image to create a training job

- Use a custom image to create a training job.

The difference between using a custom image and using a preset image lies in **Image Path**. Set **Image Path** to the path to the custom image.

After configuring the parameters, click **Apply and Run** to create the training job.

#### **NOTE**

Ensure that the AI engine of the selected preset image is the same as that of the custom image. In this way, the boot command of the preset image can be used to start the custom image.

For example, if a custom image is based on MindSpore, select a preset image with MindSpore.

Figure 13-29	Using a	custom	image to	create	a training	job
-						

🔋 Edit Training Job (	Edit Training Job Configurations					
* JobName:	MA-new-models-09-2	MA-new-models-09-28-11-38-121				
Job Description:						
	Frequently-used (	Custom				
	AI Engine:	Ascend-Powered-Engine 🔹	mindspore_1.7.0-cler_2.8.3-aarch64 💌			
* Algorithm Source:	Boot File Path:	D:\Develop\models\official\cv\resnet\	train.py			
	Code Directory:	D:\Develop\models\official\cv\resnet				
	Image Path(optional):	aide " gif an dag "gene gant	best_prac_image:0.1.0			
* Data OBS Path:	obs://data-flower/flo	wer_photos/				
Training OBS Path:	obs://data-flower/tra					
Running Parameters:	net/config/resnet50_imagenet2012_config.yamlepoch_size=10device_target=Ascend $\Bbbk^{^{(n)}}$					
Specifications:	NPU:1*Ascend-910 CPU:24*vCPUs 96GB					
* Compute Nodes:	□					
			Apply and Run Cancel App	oly		

3. View training logs.

After you click **Apply and Run**, training logs are displayed in the PyCharm window in real time. You can also click the console link in **Event Log** to view training logs on a web page.

#### Figure 13-30 Viewing training logs in PyCharm

ModelArts Event Event Lo	g 🌣 🗕	ModelArts Training Training Log
2022/09/17 15:54:07	Job log path is /modelerts-test/plugin/job/NW-	===save flag===
2022/09/17 15:54:20	Begin to upload training code.	Finish sync data from /home/ma-user/madmlmrtm/matputm/train_wrl.
plugin/job/MA-new-m	odels-09-17-15-54-79/code/models/official/cv/re	Workspace downloaded: []
2022/09/17 15:54:20	Begin to create obs dir.	epoch: 1 step: 114, loss is 1.6526103
2022/09/17 15:54:21	Begin to create training job.	epoch time: 80659.367 ms, per step time: 707.538 ms
2022/09/17 15:54:21	Training job is created successfully.	epoch: 2 step: 114, loss is 1.6303644
2022/09/17 15:54:21	Job name is MA-new-models-09-17-15-54-79.	epoch time: 2150.736 ms, per step time: 18.866 ms
2022/09/17 15:54:21	Job id is bf218f18-49bf-40d6-a559-bd30712e7f8!	epoch: 3 step: 114, loss is 1.7353387
2022/09/17 15:54:21	View your job detail in ModelArts console.	epoch time: 2149.199 ms, per step time: 18.853 ms
https://console.cdz		epoch: 4 step: 114, loss is 1.638027
2022/09/17 15:57:34		epoch time: 2150.259 ms, per step time: 18.862 ms
2022/09/17 15:57:36	Job output path is /modelents-test/plugin/job.	epoch: 5 step: 114, loss is 1.6032746
2022/09/17 15:57:36	Job log path is /modelerts-test/plugin/job/MA	epoch time: 3747.228 ms, per step time: 32.870 ms
		epoch: 6 step: 114, loss is 1,6764398

4. Stop a training job.

To stop training, choose **ModelArts** > **Training Job** > **Stop** in PyCharm or click **Stop** on the web page.

### ModelArts <u>H</u>elp Edit Credential Notebook New... Stop

#### Step 5: Releasing Resources

Release resources, such as the real-time service, training job, and OBS directories after trial use.

- To stop a notebook instance, go to the **Notebook** page, and click **Stop** in the **Operation** column of the instance.
- On the PyCharm menu bar, choose **ModelArts** > **Stop Training Job** to stop the training job.
- Log in to OBS management console and delete the created OBS bucket. Delete folders and files in the bucket one by one and then delete the bucket.

# 13.2 Creating an AI Application Using a Custom Engine

When you use a custom engine to create an AI application, you can select your image stored in SWR as the engine and specify a file directory in OBS as the model package. In this way, bring-your-own images can be used to meet your dedicated requirements.

Before deploying such an AI application as a service, ModelArts downloads the SWR image to the cluster and starts the image as a container as the user whose UID is 1000 and GID is 100. Then, ModelArts downloads the OBS file to the / **home/mind/model** directory in the container and runs the boot command preset in the SWR image. The service available to port 8080 in the container is automatically registered with APIG. You can access the service through the APIG URL.

#### Specifications for Using a Custom Engine to Create an AI Application

To use a custom engine to create an AI application, ensure the SWR image, OBS model package, and file size comply with the following requirements:

- SWR image specifications
  - A common user named ma-user in group ma-group must be built in the SWR image. Additionally, the UID and GID of the user must be 1000 and 100, respectively. The following is the dockerfile command for the built-in user:

groupadd -g 100 ma-group && useradd -d /home/ma-user -m -u 1000 -g 100 -s /bin/bash mauser

Specify a command for starting the image. In the dockerfile, specify **cmd**. The following shows an example: CMD sh /home/mind/run.sh

Customize the startup entry file **run.sh**. The following is an example.

#### #!/bin/bash

# User-defined script content

# run.sh calls app.py to start the server. For details about app.py, see "HTTPS Example". python app.py

- The service must be HTTPS enabled, and it is available on port 8080. For details, see the HTTPS example.
- (Optional) On port 8080, enable health check with URL /health. (The health check URL must be /health.)
- OBS model package specifications

The name of the model package must be **model**. For details about model package specifications, see **Introduction to Model Package Specifications**.

• File size specifications

When a public resource pool is used, the total size of the downloaded SWR image (not the compressed image displayed on the SWR page) and the OBS model package cannot exceed 30 GB.

#### **HTTPS Example**

Use Flask to start HTTPS. The following is an example of the web server code:

```
from flask import Flask, request
import json
app = Flask(__name__)
@app.route('/greet', methods=['POST'])
def say_hello_func():
  print("-----" in hello func -----")
  data = json.loads(request.get_data(as_text=True))
  print(data)
  username = data['name']
  rsp_msg = 'Hello, {}!'.format(username)
  return json.dumps({"response":rsp_msg}, indent=4)
@app.route('/goodbye', methods=['GET'])
def say_goodbye_func():
  print("------ in goodbye func ------")
  return '\nGoodbye!\n'
@app.route('/', methods=['POST'])
def default_func():
  print("------ in default func ------")
  data = json.loads(request.get_data(as_text=True))
  return '\n called default func !\n {} \n'.format(str(data))
@app.route('/health', methods=['GET'])
def healthy():
  return "{\"status\": \"OK\"}"
# host must be "0.0.0.0", port must be 8080
if __name__ == '__main_
app.run(host="0.0.0.0", port=8080, ssl_context='adhoc')
```

#### Debugging on a Local Computer

Perform the following operations on a local computer with Docker installed to check whether a custom engine complies with specifications:

- 1. Download the custom image, for example, **custom\_engine:v1** to the local computer.
- 2. Copy the model package folder **model** to the local computer.
- 3. Run the following command in the same directory as the model package folder to start the service: docker run --user 1000:100 -p 8080:8080 -v model:/home/mind/model custom\_engine:v1

#### **NOTE**

This command is used for simulation only because the directory mounted to **-v** is assigned the root permission. In the cloud environment, after the model file is downloaded from OBS to **/home/mind/model**, the file owner will be changed to **mauser**.

4. Start another terminal on the local computer and run the following command to obtain the expected inference result: curl https://127.0.0.1:8080/\${Request path to the inference service}

#### **Deployment Example**

The following section describes how to use a custom engine to create an AI application.

1. Create an AI application and viewing its details.

Log in to the ModelArts console, choose **AI Application Management** > **AI Applications**, and click **Create**. On the page that is displayed, configure the following parameters:

- Meta Model Source: OBS
- **Meta Model**: a model package selected from OBS
- Al Engine: Custom
- Engine Package: an SWR image

Retain the default settings for other parameters.

Click **Create Now**. In the AI application list that is displayed, check the AI application status. When its status changes to **Normal**, the AI application has been created.

#### Figure 13-32 Creating an AI application

<	Create	
	* Name	model-1959-custom
	* Version	00.1
	Description	
		0010
	* Meta Model Source	Training job         OBS         Container Image         Template           • Import one of the following models from OBS: TensorFlow, PyTorch, MindSpore, Custom. To import a model image, you are advised to select Container image. Ensure that ^         the model file is stored in the model editectory and specify the parent directory of the model directory as the path. If the model requires inference code, ensure that the code is stored in the model editectory. The finame must be "containe_service.pr". Finame that the model meets the Model Reactage Specifications Parameters.           • If the meta model is from a custom image, ensure the size of the meta model complies with Restrictions on the Image Size for Importing an Al Application .
		* Meta Mo         * Al Engine         Custom         * Engine Package         Swr.c         *           * Container         HTTPS         '// (host) : 8080         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *         *
		Health Check (2) Come
	AI Application Description	O Add AI Application Description

Click the AI application name. On the page that is displayed, view details about the AI application.

Figure 13-33 Viewing details about an AI application

Back to AI Applicat	tions 0.0.1 *			Deploy - Delete C	1
Basic Information				Model Precision	
Name	model-d961-custom	Status	S Normal	Recall	
Version	0.0.1	ID	8a512a64-9484-659-200d-a8079c80cd85	Precision	
Size	212.09 MB	Runtime Environment	switt	Accuracy	
Meta Model Source	/model	Al Engine	Custom	F1 Score	
Deployment Type	Real-Time Services	Model Source	Custom algorithm		
Dynamic loading	Enabled	Description	- 🖉		
AI Application					
Description					
Inference Environme	nt				
Instruction Set Architect	ure X86				
Inference Accelerator					
Parameter Configura	tion Runtime Dependency Events Constraint Associated	l Services			
View apis Definition					-
V POST /					3

2. Deploy the AI application as a service and view service details.

On the AI application details page, choose **Deploy** > **Real-Time Services** in the upper right corner. On the **Deploy** page, select a proper compute node specification, retain the default settings for other parameters, and click **Next**. When the service status changes to **Running**, the service has been deployed.

Figure 13-34 Deploying a service

de Manue	remies fifty surprise	
* Name	Service-Orike-Custoni	
Auto Stop 🕐		
	1 Enable this option to automatically stop the real-time service at the time you specify. You will not be billed for the service after the s	ervice is stopped.
	1 hour later      2 hours later      4 hours later      6 hours later      Custom	
Description	0/100	
* Resource Pool	Public Resource Pool Dedicated Resource Pool Dedicated Resource Pool New	
* AI Application and Configuration	Al Application Source My Al Applications My Subscriptions	
	Al Application and Version model-d961-custom(synchronous re • 0.0.1(Normal) • C Traffic Ratio (%) (	) - 100 +
	Specifications CPU: 2 vCPUs SGB Compute Nodes C Application scenario: Standard Ascend specifications, meeting the running and prediction requirements of NPU-accelerated AI applications	D <u> </u>
	Environment Variable ⑦ ③ Add Environment Variable Do not enter sensitive information, such as plaintext passwords, to ensure data security.	
	Timeout ① - 20 + minutes	

Click the service name. On the page that is displayed, view the service details. Click the **Logs** tab to view the service logs.

#### Figure 13-35 Logs

Usage Guides	Prediction	Configuration Upda	ates Mor	nitoring	Events	Logs	Tags	
model-1959-custom_	.0 <b>•</b> A	Il nodes	•	Latest 5 mi	nutes Lates	at 30 minutes	Latest 1 hour	Custom
<pre>* Serving Flask app 'xlz_app' (lazy loading) * Environment: production WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead. * Debug mode: off * Running on all addresses. WARNING: This is a development server. Do not use it in a production deployment. * Running on https://172.16.0.60:8080/ (Press CTRL+C to quit) * Control of the form the server. * Running of the server instead to appear to app</pre>								

#### 3. Use the service for prediction.

On the service details page, click the **Prediction** tab to use the service for prediction.

#### Figure 13-36 Prediction



# 13.3 Using a Large Model to Create an AI Application and Deploying a Real-Time Service

#### Context

Currently, a large model can have hundreds of billions or even trillions of parameters, and its size becomes larger and larger. A large model with hundreds of billions of parameters exceeds 200 GB, and poses new requirements for version management and production deployment of the platform. For example, importing AI applications requires dynamic adjustment of the tenant storage quota. Slow model loading and startup requires a flexible timeout configuration in the deployment. The service recovery time needs to be shortened in the event that the model needs to be reloaded upon a restart caused by a load exception.

To address the preceding requirements, the ModelArts inference platform provides a solution to AI application management and service deployment in large model application scenarios.

#### Constraints

- You need to apply for the size quota of an AI application and add the whitelist cached using the local storage of the node.
- You need to use the custom engine **Custom** to configure dynamic loading.
- A dedicated resource pool is required to deploy the service.
- The disk space of the dedicated resource pool must be greater than 1 TB.

#### Procedure

- 1. Applying for Increasing the Size Quota of an AI Application and Using the Local Storage of the Node to Cache the Whitelist
- 2. Uploading Model Data and Verifying the Consistency of Uploaded Objects
- 3. Creating a Dedicated Resource Pool
- 4. Creating an AI Application
- 5. Deploying a Real-Time Service

# Applying for Increasing the Size Quota of an AI Application and Using the Local Storage of the Node to Cache the Whitelist

During service deployment, the dynamically loaded model package is stored in the temporary disk space by default. When the service is stopped, the loaded files are deleted, and they need to be reloaded when the service is restarted. To avoid repeated loading, the platform allows the model package to be loaded from the local storage space of the node in the resource pool and keeps the loaded files valid even when the service is stopped or restarted (using the hash value to ensure data consistency).

To use a large model, you need to use a custom engine and enable dynamic loading when importing the model. In this regard, you need to perform the following operations:

- If the model size exceeds the default quota, submit a service ticket to increase the size quota of a single AI application. The default size quota of an AI application is 20 GB.
- Submit a service ticket to add the whitelist cached using the local storage of the node.

#### Uploading Model Data and Verifying the Consistency of Uploaded Objects

To ensure data integrity during dynamic loading, you need to verify the consistency of uploaded objects when uploading model data to OBS. obsutil, OBS Browser+, and OBS SDKs support verification of data consistency during upload. You can select a method that meets your requirements. For details, see "Verifying Data Consistency During Upload" in Object Storage Service documentation.

For example, if you upload data via OBS Browser+, enable MD5 verification, as shown in **Figure 13-37**. When dynamic loading is enabled and the local persistent storage of the node is used, OBS Browser+ checks data consistency during data upload.

o		
System Settings		×
Basic Configurations	Advanced Settings	
MD5 Verification		
QoS Rate Limit	Disable 🔻	
Timeout Interval	30 seconds	
Maximum Connections	25	
	OK Cancel	

Figure 13-37 Configuring MD5 verification for OBS Browser+

#### **Creating a Dedicated Resource Pool**

To use the local persistent storage, you need to create a dedicated resource pool whose disk space is greater than 1 TB. You can view the disk information on the **Specifications** tab of the **Basic Information** page of the dedicated resource pool. If a service fails to be deployed and the system displays a message indicating that the disk space is insufficient, see **What Do I Do If Resources Are Insufficient When a Real-Time Service Is Deployed, Started, Upgraded, or Modified**.

< a30 1 +								More + C
Basic Information								
Name a30	1			Resource Pool ID	a) .		7	
Resource Pool Type Physical				Status	Running			
DevEnviron				Training Job				
Inference Service 🔮 Enabled				Billing Mode	Pay-per-use			
Description				Network	ne	) 5 resource pools as	ssociated	
Interconnect VPC vpc-da8a	Interconnect VPC vpc-dalla GPU Driver 470.5702 © Running							
Obtained At Apr 28, 2023 15:15:1	2 GMT+08:00							
Events Nodes Specification	Monitoring Subpools							
							1	C 🐵
Specifications	Metering ID	CPU Cores	CPU Architecture	Memory	AI Accelerator	Disk Capacity		Quantity
GPU: 1*rwidia-a30   CPU: 24 vCPUs 96G8 (	. maos.modelarts.vm.gpu.a30.1682666112.g	24 :	486	96GB	1mvidia-a30	1300GB		1

Figure 13-38 Viewing the disk information of the dedicated resource pool

#### **Creating an AI Application**

If you use a large model to create an AI application and import the model from OBS, complete the following configurations:

1. Use a custom engine and enable dynamic loading.

To use a large model, you need to use a custom engine and enable dynamic loading when importing the model. You can create a custom engine to meet special requirements for image dependency packages and inference frameworks in large model scenarios. For details about how to create a custom engine, see **Creating an AI Application Using a Custom Engine**.

When you use a custom engine, dynamic loading is enabled by default. The model package is separated from the image, and the model is dynamically loaded to the service load during service deployment.

2. Configure health check.

Health check is mandatory for the AI applications imported using a large model to identify unavailable services that are displayed as started.

**Figure 13-39** Using a custom engine, enabling dynamic loading, and configuring health check

Meta Model Source	Training job	OBS C	ontainer image	Template			
	• Import one of the follo select Container image. Inference code, ensure the Specifications. Paramete	wing models from OE Ensure that the mode nat the code is stored rs .	IS: TensorFlow, PyTor I file is stored in the in the model directo	ch, MindSpore, Spark_ML model directory and spec ry. The file name must be	lib, Scikit_Learn, XGBoost, Custom. 1 ify the parent directory of the mode "customize_service.py". Ensure that	To import a model image, you are advised to I directory as the path. If the model requires the model meets the Model Package	^
	When uploading a mod	del larger than 5 GB, s	elect Dynamic loadii	ng. It is a good practice to	set AI Engine to Custom and enabl	e health check for model uploading.	
	* Meta Mo		l5-100g/	ð			
	* Al Engine	Iustom	•	* Engine Package	swr.cn-north-7.myhuawek 🗎		
	* Container A	HTTPS	:// {host}: 8080				
	Health Check 🤅						
		* Check M	HTTP request	Command			
		* Health C	/health				
		* Health C	10				
		* Delay( se	1,800				
		* Maximu	120				
	V Dynamic loading	?					

#### **Deploying a Real-Time Service**

When deploying the service, complete the following configurations:

1. Customize the deployment timeout interval.

Generally, the time for loading and starting a large model is longer than that for a common model. Set **Timeout** to a proper value. Otherwise, the timeout may elapse prior to the completion of the model startup, and the deployment may fail.

2. Add an environment variable.

During service deployment, add the following environment variable to set the service traffic load balancing policy to cluster affinity, preventing unready service instances from affecting the prediction success rate: MODELARTS\_SERVICE\_TRAFFIC\_POLICY: cluster

**Figure 13-40** Customizing the deployment timeout interval and adding an environment variable

AI Application and Configuration	AI Application Source	My AI Applications My Subscriptions		
	AI Application and Version	model-aa5c(synchronous request) • 0.0.1 • C	Traffic Ratio (%) 🕜	- 100 +
	Specifications	GPU: 1 * P4 (8GB)   CPU: 8 vCPUs 32 •	Compute Nodes	- 1 +
	Environment Variable	MODELARTS_SERVICE_TI         =         cluster         ☑           ③ Add Environment Variable		
		Do not enter sensitive information, such as plaintext passwords, to ensure data security.		
	Timeout 🕐	- 20 + minutes		

You are advised to deploy multiple instances to improve service reliability.

# 13.4 Importing a Model from OBS to Create an AI Application and Deploying a Real-Time Service

This section describes how to upload a model package to Object Storage Service (OBS), import the model from OBS to create an AI application, deploy the AI application as a real-time service, and perform prediction.

Before you start, complete the requirements described in **Prerequisites**. The procedure in this case is as follows:

Step 1: Upload a Model Package to OBS

Step 2: Import the Model from OBS to Create an AI Application

Step 3: Deploy the AI Application as a Real-Time Service

**Step 4: Perform Prediction** 

#### Prerequisites

Prepare the required files based on the **model package specifications**. The model package must contain the **model** folder. The **model** folder stores the model file, model configuration file, and model inference code file.

#### Step 1: Upload a Model Package to OBS

- 1. Log in to the OBS console.
- Create a bucket and a directory for storing the model package. Click Upload Object to upload the required model package to the OBS bucket directory. For example, the structure of the MindSpore model package is as follows:
   OBS bucket or directory name
   mest
   mest

model (Mandatory) Name of a fixed subdirectory, which is used to store model-related files

| | config.json (Mandatory) Model configuration file. The file name is fixed to **config.json**. Only one model configuration file is supported.

| — customize\_service.py (Mandatory) Model inference code. The file name is fixed to **customize\_service.py**. Only one model inference code file exists. The files on which **customize\_service.py** depends can be directly stored in the model directory.

tmp.om (Mandatory) An empty .om file that enables the model package to be imported

#### Figure 13-41 Uploading a model package

bject Storage /	lect Storage / / cB5 / mindspore_2.0.0-cann_6.3.0 / model 🗇						
Objects	Objects Deleted Objects Fragments						
Objects are b	Objects are basic units of data storage. In OBS, files and folders are treated as objects. Any file type can be uploaded and managed in a bucket. Learn more						
Upload C	Upload Object Create Folder Delete						
	Object Name ↓Ξ Size ↓Ξ Last Modified ↓ <del>.</del> Operation						
	← Back						
	customize_service.py	3.61 KB	May 05, 2023 09:59:28 GMT+08:00	Download   Copy Path   More 👻			
	tmp.om	0 Byte	May 05, 2023 09:59:28 GMT+08:00	Download   Copy Path   More 👻			
	config.json	1.27 KB	May 05, 2023 09:59:28 GMT+08:00	Download   Copy Path   More 🔻			
	checkpoint_lenet_1-1_1875.ckpt	482.57 KB	May 05, 2023 09:59:28 GMT+08:00	Download   Copy Path   More 💌			

#### Step 2: Import the Model from OBS to Create an AI Application

- Log in to the ModelArts management console, choose AI Application Management > AI Applications. On the My AI Applications tab page, click Create.
- 2. Set **Meta Model Source** to **OBS** and select the model package uploaded in **Step 1: Upload a Model Package to OBS**. AI engine, runtime environment, and runtime dependency will be automatically configured. The following is an example.

#### Figure 13-42 Importing a model from OBS

Mata Model Source	Training Job OPS	Container Image Tr	mplate		
weta wodet source	Import one of the following models for	m OBS: TensorFlow, DriTersh, Min	Ispace Custom To import a model	Image you are advised to select Cent	niner image Foru
	model file is stored in the model direct	prv and specify the parent director	v of the model directory as the pat	th. If the model requires inference cod	e, ensure that the
	in the model directory. The file name must be "customize, service.py".				
	* Meta Mo	/c85/mindsnore 1 🗃			
		, costimuspore_ E			
	di Al Casina MindCasa				
	* Al Engine Mindspore	<ul> <li>minuspore_2.0.0-</li> </ul>	.d *		
	🔽 Dynamic loading 🕐				
Inference Code	/c8	5/mindspore_2.0.0-cann_6.3.0-py_	8.7-euler_2.8.3-aarch64/model/cust	comize_service.py	
Runtime Dependency	Installation Method	Name	Version	Constraint	Operation
annual corporation (					
	Add				

- 3. Click **Create now** after you finish the configuration to create the AI application.
- 4. On the AI application list page, check the AI application status. After the status changes to **Normal**, the AI application is created.

#### Step 3: Deploy the AI Application as a Real-Time Service

 On the AI application list page, locate the AI application created in Step 2: Import the Model from OBS to Create an AI Application and click the down arrow on the left to show its AI application version list. Then, choose Deploy > Real-Time Services in the Operation column.

Figure 13-43 Deploying a real-time service

Operatio	n
Deploy 🔺	Publish   Delete
Real-Time Services	
Batch Services	

2. Verify the AI application and version number on the real-time service deployment page. They are configured automatically. Turn on auto stop to make the service stop by itself at the time you choose. Configure other parameters as required and then click **Create now** to deploy the real-time service. The following is an example.

* AI Application and Configuration	AI Application Source	My AI Applications My Subscriptions	
	AI Application and Version	model-dBed-Mindspore(synchronous • 0.0.1 • C	Traffic Ratio (%) ⑦ - 100 +
	Specifications	Ascend: 1* D910 (32GB)   ARM: 24 v 💌	Compute Nodes ⑦ - 1 +
	Environment Variable	O Add Environment Variable Do not enter sensitive information, such as plaintext passwords, to ensure data security.	
	Timeout 🕐	20 + minutes	

Figure 13-44 AI Application and Configuration

3. Check the service status on the real-time service list page. If the status changes to **Running**, the real-time service is deployed.

### **Step 4: Perform Prediction**

- 1. Click the name of the real-time service deployed in **Step 3**: **Deploy the AI Application as a Real-Time Service**.
- 2. Click the **Prediction** tab to perform prediction. The following is an example.

#### Figure 13-45 Prediction

Usage Guides Prediction Configuration Updates Monitoring Events Logs Tags Request Publ /   Image File Upload Predict	
Test Image Preview	Test Result

#### **Step 5: Clear Resources**

Once the prediction finishes, delete any unused resources.

- To stop or delete a real-time service, go to the Real-Time Services page, locate the row that contains the target service, and choose More > Stop or Delete in the Operation column.
- Log in to OBS management console and delete the created OBS bucket. Delete folders and files in the bucket one by one and then delete the bucket.

# **14** Full-Process Development of WebSocket Real-Time Services

#### Context

WebSocket is a network transmission protocol that supports full-duplex communication over a single TCP connection. It is located at the application layer in an OSI model. The WebSocket communication protocol was established by IETF in 2011 as standard RFC 6455 and supplemented by RFC 7936. The WebSocket API in the Web IDL is standardized by W3C.

WebSocket simplifies data exchange between the client and the server and allows the server to proactively push data to the client. In the WebSocket API, if the initial handshake between the client and the server is successful, a persistent connection will be established between them and data can be transferred bidirectionally.

#### Prerequisites

- You are experienced in developing Java and familiar with JAR packaging.
- You have basic knowledge and calling methods of WebSocket.
- You are familiar with the method of creating an image using Docker.

#### Constraints

- WebSocket supports only the deployment of real-time services.
- WebSocket supports only real-time services deployed using AI applications imported from custom images.

#### Preparations

Before using WebSocket in ModelArts for inference, bring your own custom image. The custom image must be able to provide complete WebSocket services in a standalone environment, for example, completing WebSocket handshakes and exchanging data between the client to the server. The model inference is implemented in the custom image, including downloading the model, loading the model, performing preprocessing, completing inference, and assembling the response body.

#### Procedure

To develop a WebSocket real-time service, perform the following operations:

- Uploading the Image to SWR
- Creating an AI Application Using the Image
- Deploying the AI Application as a Real-Time Service
- Calling the WebSocket Real-Time Service

#### Uploading the Image to SWR

Upload the local image to SWR. For details, see **How Can I Log In to SWR and Upload Images to It?** 

#### Creating an AI Application Using the Image

- Log in to the ModelArts management console, choose AI Application Management > AI Applications, and click Create under My AI Applications. The page for creating an AI application is displayed.
- 2. Configure the AI application.
  - Meta Model Source: Select Container image.
  - Container Image Path: Select the path specified in Uploading the Image to SWR.
  - **Container API**: Configure this parameter based on site requirements.
  - Health Check: Retain default settings. If health check has been configured in the image, configure the health check parameters based on those configured in the image.

#### Figure 14-1 AI application parameters

Meta Model Source	Tra	aining job	OBS	Container image	Template	
	A mode	l imported from	n a containe	r image is of the image t	ype. Ensure the imag	e can be properly started and provides inference APIs. During
	service	deployment, M	odelArts use	s the image to deploy inf	erence services. Lear	n more about image specifications . Parameters
						and the second sec
	*	Container Im	age Path	swr.cn-north-7.myhuav	veicloud.com/modela	arts-modelhub-image2/simj
	*	Container AF	9	HTTP		8887
		In the second se				
		Image Replic	ation	Million Alain from ations in alia	ablad AL and instinut	he second eviably but medificing as deleting impress in
				when this function is dis	abled, Al application	s can be created quickty, but mounying or detering images in
				be created quickly, but w	ou can modify or del	ete images in the source directory as that would not affect
				service deployment.	ou can mouny or dea	ete images in the source directory as that would not anect
			~			
		Health Check	(?)			
AL Analization Description	0	⊕ Add	AI Applicatio	n Description		
AI Application Description	$\odot$			1 State 1 Stat		

3. Click **Create now**. In the AI application list that is displayed, check the AI application status. When it changes to **Normal**, the AI application has been created.

#### Deploying the AI Application as a Real-Time Service

- Log in to the ModelArts management console, choose Service Deployment > Real-Time Services, and click Deploy.
- 2. Configure the service.

- AI Application and Version: Select the AI application and version created in Creating an AI Application Using the Image.
- WebSocket: Enable this function.

#### Figure 14-2 WebSocket

* AI Application and Configuration	AI Application Source	My Al Applications My Subscriptions
	AI Application and Version	model-2f34(synchronous request)         ▼         0.0.1         ▼         C         Traffic Ratio (%)         ⑦         −         100         +
	Specifications	CPU: 2 vCPUs 8GE           Compute Nodes
	Environment Variable 🕜	Add Environment Variable     Do not enter sensitive information, such as plaintext passwords, to ensure data     security.
	Timeout ⑦	- 20 + minutes
WebSocket		

3. Click **Next**, confirm the configuration, and click **Submit**. In the real-time service list you will be redirected to, check the service status. When it changes to **Running**, the real-time service has been deployed.

#### Calling a WebSocket Real-Time Service

WebSocket itself does not require additional authentication. ModelArts WebSocket is WebSocket Secure-compliant, regardless of whether WebSocket or WebSocket Secure is enabled in the custom image. WebSocket Secure supports only one-way authentication, from the client to the server.

You can use one of the following authentication methods provided by ModelArts:

• Access Authenticated Using a Token

The following section uses GUI software Postman for prediction and token authentication as an example to describe how to call WebSocket.

- 1. Establish a WebSocket connection.
- 2. Exchange data between the WebSocket client and the server.

**Step 1** Establish a WebSocket connection.

 Open Postman of a version later than 8.5, for example, 10.12.0. Click in the upper left corner and choose File > New. In the displayed dialog box, select WebSocket Request (beta version currently).

#### Figure 14-3 WebSocket Request



2. Configure parameters for the WebSocket connection.

Select **Raw** in the upper left corner. Do not select **Socket.IO** (a type of WebSocket implementation, which requires that both the client and the server run on **Socket.IO**). In the address box, enter the **API Address** obtained on the **Usage Guides** tab on the service details page. If there is a finer-grained URL in the custom image, add the URL to the end of the address. If **queryString** is available, add this parameter in the **params** column. Add authentication information into the header. The header varies depending on the authentication mode, which is the same as that in the HTTPS-compliant inference service. Click **Connect** in the upper right corner to establish a WebSocket connection.

#### Figure 14-4 Obtaining the API address

Usage Guides	Prediction	Configuration Updates	Monitoring	Events	Logs	Tags
API Address	wss://inferen	/infers/90734aa(	D-3ad4-41f9 🗇			

#### 

- If the information is correct, **CONNECTED** will be displayed in the lower right corner.
- If establishing the connection failed and the status code is 401, check the authentication.
- If a keyword such as WRONG\_VERSION\_NUMBER is displayed, check whether the port configured in the custom image is the same as that configured in WebSocket or WebSocket Secure.

The following shows an established WebSocket connection.

#### Figure 14-5 Connection established

Ram v wss://mference.ulangab.huawei.com/v1/mfers/90734aaO-3ad4-1119-0116-2725440548a Docs > Feedback > 🔯 Sere v					
wss:/	inferen 548a				Disconnect
Messag	Params Headers Settings				
Headers	6 hidden				
	Key	Value		Description	Bulk Edit
	X-Auth-Token				
	Key	Value		Description	
Messag	ns Al Massages √ ∰ Clear Messages			[	• Connected ~
□ ↓	welcome: /			68	) 19:10:00 ~
Ø	Connected to wss://inferenc	8a			19:10:00 🗸

#### NOTICE

Preferentially check the WebSocket service provided by the custom image. The type of implementing WebSocket varies depending on the tool you used. Possible issues are as follows: A WebSocket connection can be established but cannot be maintained, or the connection is interrupted after one request and needs to be reconnected. ModelArts only ensures that it will not affect the WebSocket status in a custom image (the API address and authentication mode may be changed on ModelArts).

**Step 2** Exchange data between the WebSocket client and the server.

After the connection is established, WebSocket uses TCP for full-duplex communication. The WebSocket client sends data to the server. The implementation types vary depending on the client, and the lib package may also be different for the same language. Different implementation types are not considered here.

The format of the data sent by the client is not limited by the protocol. Postman supports text, JSON, XML, HTML, and Binary data. Take text as an example. Enter the text data in the text box and click **Send** on the right to send the request to the server. If the text is oversized, Postman may be suspended.

#### Figure 14-6 Sending data

Rarr v wss/inference. Docs <sup>3</sup> Feedback	> 🖺 Save 🗸
wss//inference 3a	Disconnect
Message Params Headers  Settings	
1 8555	
Tet v	Send
Messages	Connected
Search All Messages 🗸 🌐 Clear Messages	
we have received your message:sss, and this is our reply:OK.	19:11:38 👻
↑ 5555	19:11:38 🗸
😺 welcome: /	19:10:00 🗸
Connected to wss://inference la	19:10:00 ~

----End

# **15** FAQs

# **15.1 General Issues**

### 15.1.1 What Is ModelArts?

ModelArts is a one-stop AI development platform geared toward developers and data scientists of all skill levels. It enables you to rapidly build, train, and deploy models anywhere (from the cloud to the edge), and manage full-lifecycle AI workflows. ModelArts accelerates AI development and fosters AI innovation with key capabilities, including data preprocessing and auto labeling, distributed training, automated model building, and one-click workflow executing.

The one-stop ModelArts platform covers all stages of AI development, including data processing, AI application creation, and model training and deployment. The underlying layer of ModelArts supports various heterogeneous computing resources. You can flexibly select and use the resources without having to consider the underlying technologies. In addition, ModelArts supports popular open-source AI development frameworks such as TensorFlow. Developers can also use self-developed algorithm frameworks to match their usage habits.

ModelArts aims to achieve simple, convenient AI development.

# 15.1.2 What Are the Relationships Between ModelArts and Other Services?

OBS

ModelArts uses Object Storage Service (OBS) to securely and reliably store data and models at low costs. For more details, see *Object Storage Service Console Operation Guide*.

### CCE ModelArts uses Cloud Container Engine (CCE) to deploy models as real-time services. CCE enables high concurrency and provides elastic scaling. For more information about CCE, see Cloud Container Engine User Guide. **SWR** To use an AI framework that is not supported by ModelArts, use Software Repository for Container (SWR) to customize an image and import the image to ModelArts for training or inference. For details about SWR, see . **Cloud Eve** ModelArts uses Cloud Eye to monitor online services and model loads in real time and send alarms and notifications automatically. For details about Cloud Eye, see Cloud Eye User Guide. **CTS** ModelArts uses Cloud Trace Service (CTS) to record operations for later query, audit, and backtrack operations. For details about CTS, see *Cloud Trace Service* User Guide.

### 15.1.3 What Are the Differences Between ModelArts and DLS?

Deep Learning Service (DLS) is a one-stop deep learning platform based on the high-performance computing capabilities. With various optimized neural network models, DLS allows you to easily implement model training and evaluation with the flexibility of on-demand scheduling.

However, DLS supports only the deep learning technologies, while ModelArts integrates both the deep learning and machine learning technologies. In addition, ModelArts is a one-stop AI development platform, which manages the AI development lifecycle from data labeling, algorithm development, to model training and deployment. To be specific, ModelArts contains and supports the functions and features of DLS. Currently, DLS is terminated. The functions related to deep learning can be directly used in ModelArts. If you are a DLS user, you can also migrate the data in DLS to ModelArts.

## 15.1.4 Which Ascend Chips Are Supported?

Currently, Ascend 310 and Ascend 910 are supported.

- **Model training**: Ascend 910 can be used to train models. ModelArts provides algorithms designed for model training with Ascend 910.
- **Model inference**: When a model is deployed as a real-time service on ModelArts, you can use Ascend 310 resources for model inference.

### 15.1.5 How Do I Obtain an Access Key?

#### **Obtaining an Access Key**

- 1. Log in to the console, enter the **My Credentials** page, and choose **Access Keys** > **Create Access Key**.
- 2. In the **Create Access Key** dialog box that is displayed, use the login password for verification.
- 3. Click **OK**, open the **credentials.csv** file, and save the key file as prompted. The access key file is saved in the default downloads folder of the browser. Then, the access key (**Access Key Id** and **Secret Access Key**) is obtained.

### 15.1.6 How Do I Upload Data to OBS?

Before using ModelArts to develop AI models, data needs to be uploaded to an OBS bucket. You can log in to the OBS console to create an OBS bucket, create a folder in it, and upload data. For details about how to upload data, see *Object Storage Service User Guide*.

# 15.1.7 What Do I Do If the System Displays a Message Indicating that the AK/SK Pair Is Unavailable?

#### **Issue Analysis**

An AK and SK form a key pair required for accessing OBS. Each SK corresponds to a specific AK, and each AK corresponds to a specific user. If the system displays a message indicating that the AK/SK pair is unavailable, it is possible that the account is in arrears or the AK/SK pair is incorrect.

#### Solution

- 1. Use the current account to log in to the OBS console and check whether the current account can access OBS.
  - If the account can access OBS, rectify the fault by referring to 2.
- 2. If yes, .
  - If not, replace the AK/SK with those created using the current account. For details, see **Access Keys**.

# 15.1.8 What Do I Do If a Message Indicating Insufficient Permissions Is Displayed When I Use ModelArts?

If a message indicating insufficient permissions is displayed when you use ModelArts, perform the operations described in this section to grant permissions for related services as needed.

The permissions to use ModelArts depend on OBS authorization. Therefore, ModelArts users require OBS system permissions as well.

 For details about how to grant a user full permissions for OBS and common operations permissions for ModelArts, see Configuring Common Operations Permissions.  For details about how to manage user permissions on OBS and ModelArts in a refined manner and configure custom policies, see Creating a Custom Policy for ModelArts.

#### **Configuring Common Operations Permissions**

To use ModelArts basic functions, assign the **ModelArts CommonOperations** permission on project-level services to users. Since ModelArts depends on OBS permissions, assign the **OBS Administrator** permission on global services to users.

The procedure is as follows:

**Step 1** Create a user group.

Log in to the IAM console and choose **User Groups** > **Create User Group**. Enter a user group name, and click **OK**.

**Step 2** Configure permissions for the user group.

In the user group list, locate the user group created in step 1, click **Authorize**, and perform the following operations.

1. Assign the **ModelArts CommonOperations** permission on project-level services to the user group and click **OK**.

**NOTE** 

The permission takes effect only in assigned regions. Assign permissions in all regions if the permission is required in all regions.

- 2. Assign the **OBS Administrator** permission on global services to the user group and click **OK**.
- **Step 3** Create a user on the IAM console and add the user to the user group created in step 1.
- **Step 4** In the authorized region, perform the following operations:
  - Choose Service List > ModelArts. Choose Dedicated Resource Pools. On the page that is displayed, select a resource pool type and click Create. You should not be able to create a new resource pool.
  - Choose any other service in **Service List**. (Assume that the current policy contains only **ModelArts CommonOperations**.) If a message appears indicating that you have insufficient permissions to access the service, the **ModelArts CommonOperations** policy has already taken effect.
  - Choose Service List > ModelArts. On the ModelArts console, choose Data Management > Datasets > Create Dataset. You should be able to access the corresponding OBS path.

----End

#### **Creating a Custom Policy for ModelArts**

In addition to the default system policies of ModelArts, you can create custom policies, which can address OBS permissions as well.

You can create custom policies using either the visual editor or JSON views. This section describes how to use a JSON view to create a custom policy to grant

permissions required to use development environments and the minimum permissions required by ModelArts to access OBS.

#### **NOTE**

A custom policy can contain actions for multiple services that are accessible globally or only for region-specific projects.

ModelArts is a project-level service, but OBS is a global service, so you need to create separate policies for the two services and then apply these policies to the users.

1. Create a custom policy for minimizing permissions for OBS that ModelArts depends on.

Log in to the IAM console, choose **Permissions** > **Policies/Roles**, and click **Create Custom Policy**. Configure the parameters as follows:

- **Policy Name**: Choose a custom policy name.
- Policy View: JSON
- Policy Content: Follow the instructions in Example Custom Policies of OBS.
- 2. Create a custom policy for the permissions to use ModelArts development environments. Configure the parameters as follows:
  - **Policy Name**: Choose a custom policy name.
  - Policy View: JSON
  - Policy Content: Follow the instructions in Example Custom Policies for Using the ModelArts Development Environment. For the actions that can be added for custom policies, see *ModelArts API Reference* > "Permissions Policies and Supported Actions" > "Introduction".
- 3. After creating a user group on the IAM console, grant the custom policy created in 1 to the user group.
- 4. Create a user on the IAM console and add the user to the user group created in **3**.
- 5. In the authorized region, perform the following operations:
  - Choose Service List > ModelArts. On the ModelArts console, choose
     Data Management > Datasets. If you cannot create a dataset, the permissions (for using the development environment) granted only to ModelArts users have taken effect.
  - Choose Service List > ModelArts. On the ModelArts console, choose
     DevEnviron > Notebook and click Create. If you can access the OBS path specified in Storage, the OBS permissions have taken effect.

#### **Example Custom Policies of OBS**

The permissions to use ModelArts require OBS authorization. The following example shows the minimum OBS required, including the permissions for OBS buckets and objects. After being granted the minimum permissions for OBS, users can access OBS from ModelArts without restrictions.

```
"Version": "1.1",
"Statement": [
{
"Action": [
"obs:bucket:ListAllMybuckets",
```



**Example Custom Policies for Using the ModelArts Development Environment** 



# 15.1.9 How Do I Use ModelArts to Train Models Based on Structured Data?

For more advanced users, ModelArts provides the notebook creation function of DevEnviron for code development. It allows the users to create training tasks with large volumes of data in training jobs and use the engines such as Scikit\_Learn, XGBoost, or Spark\_MLlib in the development and training processes.

### 15.1.10 How Do I View All Files Stored in OBS on ModelArts?

To view all files stored in OBS when using notebook instances or training jobs, use either of the following methods:

• OBS console

Log in to OBS console using the current account, and search for the OBS buckets, folders, and files.

 You can use an API to check whether a given directory exists. In an existing notebook instance or after creating a new notebook instance, run the following command to check whether the directory exists: import moxing as mox mox.file.list\_directory('obs://bucket\_name', recursive=True)

2024-04-30

If there are a large number of files, wait until the final file path is displayed.

# 15.1.11 Where Are Datasets of ModelArts Stored in a Container?

Datasets of ModelArts and data in specific data storage locations are stored in OBS.

### 15.1.12 Which AI Frameworks Does ModelArts Support?

The AI frameworks and versions supported by ModelArts vary slightly based on the development environment notebook, training jobs, and model inference (AI application management and deployment). The following describes the AI frameworks supported by each module.

#### **Unified Image List**

ModelArts provides unified images of Arm+Ascend specifications, including MindSpore and PyTorch. You can use the images to develop environment, train models, and deploy services. For details, see **Unified Image List**.

Table	15-1	MindSpore
iubic		ivinius porc

Preset Image	Supported Processor	Applicable Scope
mindspore_2.2.0-cann_7.0.1-py_3.9- euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment
mindspore_2.1.0-cann_6.3.2-py_3.7- euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment
mindspore_2.2.10-cann_8.0.rc1- py_3.9-hce_2.0.2312-aarch64-snt9c	Ascend snt9c	Notebook, training, and inference deployment

#### Table 15-2 PyTorch

Preset Image	Supported Processor	Applicable Scope
pytorch_1.11.0-cann_6.3.2-py_3.7- euler_2.10.7-aarch64-snt9b	Ascend snt9b	Notebook, training, and inference deployment
pytorch_2.1.0-cann_8.0.rc1-py_3.9- hce_2.0.2312-aarch64-snt9c	Ascend snt9c	Notebook, training, and inference deployment
pytorch_1.11.0-cann_8.0.rc1-py_3.9- hce_2.0.2312-aarch64-snt9c	Ascend snt9c	Notebook, training, and inference deployment

### **Development Environment Notebook**

The image and versions supported by development environment notebook instances vary based on runtime environments.

Image	Description	Suppor ted Chip	Remot e SSH	Online Jupyter Lab
pytorch_1.11.0-cann_7.0.1- py_3.9-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes
pytorch_2.1.0-cann_7.0.1- py_3.9-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes
mindspore_2.2.0-cann_7.0.1- py_3.9-euler_2.10.7-aarch64- snt9b	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
mindspore_2.1.0-cann_6.3.2- py_3.7-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine MindSpore	Ascend	Yes	Yes
pytorch_1.11.0-cann_6.3.2- py_3.7-euler_2.10.7-aarch64- snt9b	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes

Table 15-3 Images supported by notebook of the new version

Image	Description	Suppor ted Chip	Remot e SSH	Online Jupyter Lab
mindspore1.7.0-cann5.1.0- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
mindstudio5.0.rc1-ascend- cann5.1.rc1-euler2.8.3- aarch64	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	No
mindspore1.8.0-cann5.1.2- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
tensorflow1.15-cann5.1.0- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. TensorFlow is preset in the Al engine.	Ascend	Yes	Yes
mindspore_2.0.0-cann_6.3.0- py_3.7-euler_2.8.3	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine MindSpore	Ascend	Yes	Yes
pytorch_1.11.0-cann_6.3.0- py_3.7-euler_2.8.3	Ascend- and Arm- powered public image for algorithm development and training, with built- in AI engine PyTorch	Ascend	Yes	Yes

Image	Description	Suppor ted Chip	Remot e SSH	Online Jupyter Lab
tensorflow1.15- mindspore1.7.0-cann5.1.0- euler2.8-aarch64	Ascend+Arm algorithm development and training. TensorFlow and MindSpore are preset in the Al engine.	Ascend	Yes	Yes
tensorflow_1.15.0- cann_6.3.0-py_3.7- euler_2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes
tensorflow1.15.0-cann5.1.2- py3.7-euler2.8.3	Ascend+Arm algorithm development and training. MindSpore is preset in the Al engine.	Ascend	Yes	Yes

#### Table 15-4 Images supported by notebook of the old version

Runtime Environment	Built-in AI Engine and Version	Supported Chip
Ascend-Powered-Engine	MindSpore 1.2.0	Ascend
1.0 (Python3)	MindSpore 1.1.1	Ascend
	TensorFlow 1.15.0	Ascend

### **Training Jobs**

The following table lists the AI engines.

The built-in training engines are named in the following format: <Training engine name\_version>-[cpu | <cuda\_version | cann\_version >]-<py\_version>-<OS name\_version>-< x86\_64 | aarch64>

	Table 15-5 /	Al engines	supported	by t	raining	jobs
--	--------------	------------	-----------	------	---------	------

Runtime Environmen t	System Archite cture	System Version	AI Engine and Version	Supported CUDA or Ascend Version
Ascend- Powered- Engine	aarch6 4	Euler2.8	mindspore_2.0.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0
PyTorch	aarch6 4	Euler2.8	pytorch_1.11.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0
TensorFlow	aarch6 4	Euler2.8	tensorflow_1.15.0- cann_6.3.0-py_3.7- euler_2.8.3-aarch64	cann_6.3.0

#### **NOTE**

Supported AI engines vary depending on regions.

#### Supported AI Engines for ModelArts Inference

If you import a model from a template or OBS to create an AI application, the following AI engines and versions are supported.

#### **NOTE**

- Runtime environments marked with **recommended** are unified runtime images, which will be used as mainstream base inference images.
- Images of the old version will be discontinued. Use unified images.
- The base images to be removed are no longer maintained.
- Naming a unified runtime image: <*AI engine name and version*> <*Hardware and version*: CPU, CUDA, or CANN> <*Python version*> <*OS version*> <*CPU architecture*>

Engine	Runtime
TensorFlow	tensorflow_1.15.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64
MindSpore	mindspore_2.0.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64
PyTorch	pytorch_1.11.0-cann_6.3.0-py_3.7-euler_2.8.3-aarch64

#### Table 15-6 Supported AI engines and their runtime

# 15.1.13 What Are the Functions of ModelArts Training and Inference?

ModelArts training includes ExeML, training management, and dedicated resource pools (for development/training).

ModelArts inference includes AI application management and deployment.

# 15.1.14 Can AI-assisted Identification of ModelArts Identify a Specific Label?

After a model with multiple labels is trained and deployed as a real-time service, all the labels are identified. If only one type of label needs to be identified, train a model dedicated for identifying the label. To speed up the label identification, select a high flavor for deploying the model.

# 15.1.15 Why Is the Job Still Queued When Resources Are Sufficient?

- If a public resource pool is used, the resources may be used by other users. Please wait or find solutions in Why Is a Training Job Always Queuing?.
- If a dedicated resource pool is used, perform the following operations:
  - a. Check whether other jobs (including inference jobs, training jobs, and development environment jobs) are running in the dedicated resource pool.

On the **Dashboard** page, you can go to the details page of the running jobs or instances to check whether the dedicated resource pool is used. You can stop them based on your needs to release resources.

b. Click the dedicated resource pool to go to the details page and view the job list.

If other jobs are waiting in the queue, the new job must also join the queue.

- c. Check whether resources are fragmented.
  - For example, the cluster has two nodes, and there are four idle cards on each node. However, your job requires eight cards on one node. In this case, the idle resources cannot be allocated to your job.

# 15.2 Data Management (Old Version)

### 15.2.1 Are There Size Limits for Images to be Uploaded?

For data management, there are limits on the image size when you upload images to the datasets whose labeling type is object detection or image classification. The size of an image cannot exceed 8 MB, and only JPG, JPEG, PNG, and BMP formats are supported.

#### Solutions:
- Import the images from OBS. Upload images to any OBS directory and import the images from the OBS directory to an existing dataset.
- Use data source synchronization. Upload images to the input directory or its subdirectory of a dataset, and click **Synchronize Data Source** on the dataset details page to add new images. Note that synchronizing a data source will delete the files deleted from OBS from the dataset. Exercise caution when performing this operation.
- Create a dataset. Upload images to any OBS directory. You can directly use the image directory as the input directory to create the dataset.

# 15.2.2 What Do I Do If Images in a Dataset Cannot Be Displayed?

# Symptom

Images in a created dataset cannot be displayed during labeling, and they cannot be viewed by clicking them. Alternatively, the system displays a message indicating that an error occurred in image loading.

# **Possible Cause**

- The local network may be faulty. As a result, OBS cannot be accessed and images cannot be loaded.
- You are not allowed to access the target OBS bucket.
- The OBS bucket or file may be encrypted.
- The OBS storage class does not allow the parallel file system to process images. Therefore, the thumbnails cannot be displayed.

# Solution

1. The following uses Google Chrome as an example. Press **F12** to open the browser console, locate the image, and copy the image URL.

### Figure 15-1 Obtaining the image URL



- Enter the URL in a new browser. The "Your Connection Is Not Private" message is displayed. Click Advanced on the page and choose Proceed to <link> (unsafe) to go to the target website.
- 3. After the image is successfully accessed, return to the ModelArts console to access the dataset. The image is displayed.

# 15.2.3 How Do I Integrate Multiple Object Detection Datasets into One Dataset?

Create a parent directory in an OBS bucket, in the directory add the same number of folders as that of datasets, export one dataset to one folder, and use the parent directory to create a dataset.

Log in to the ModelArts management console and choose **Data Management** > **Datasets**. Click the target dataset to switch to its **Dashboard** page. Then, click **Export** in the upper right corner of the page to export the dataset to a folder in the OBS parent directory.

# 15.2.4 What Do I Do If Importing a Dataset Failed?

The possible cause is that the storage class of the target OBS bucket is incorrect. In this case, select a bucket of the standard storage class to import data.

Region	CN North-Beijing4
	Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region. Once a bucket is created, the region cannot be changed.
Data Redundancy Policy  ?	Multi-AZ storage Single-AZ storage
	Multi-AZ storage improves data availability. Multi-AZ storage has higher price than single-AZ sotrage. Pricing details
	Once the multi- AZ storage is enabled, it cannot be disabled.
Bucket Name	Enter a bucket name.
	Naming conventions - The name must be globally unique in OBS. - The name must contain 3 to 63 characters. Only lowercase letters, digits, hyphens (-), and periods (.) are allowed. - The name cannot start or end with a period (.) or hyphen (-), and cannot contain two consecutive periods (.) or contain a period (.) and a hyphen (-) adjacent to each other. - The name cannot be an IP address. - If the name contains any periods, a security certificate verification message may appear when you access the bucket or its objects by entering a domain name. - The name of a bucket or parallel file system can be reused 30 minutes after the bucket or parallel file system is deleted.
Storage Class	Standard Infrequent Access Archive
	Optimized for frequently accessed (multiple times per month) data such as small and essential files that require low latency.
	The storage class of a bucket is inherited by objects uploaded to the bucket by default. You can also change the storage class of an object when uploading it to the bucket. Learn more

# 15.2.5 Can a Table Dataset Be Labeled?

Table datasets cannot be labeled. They are suitable for processing structured data such as tables. Table files are in CSV format. You can preview up to 100 data records in a table.

# 15.2.6 What Do I Do to Import Locally Labeled Data to ModelArts?

ModelArts allows you to import data by importing datasets. Locally labeled data can be imported from an OBS directory or the manifest file. After the import, you can label the data again or modify the labels in ModelArts Data Management.

For details about how to import data from an OBS directory or manifest file, see .

# 15.2.7 Why Does Data Fail to Be Imported Using the Manifest File?

### Symptom

Failed to use the manifest file of the published dataset to import data again.

#### Possible Cause

Data has been changed in the OBS directory of the published dataset, for example, images have been deleted. Therefore, the manifest file is inconsistent with data in the OBS directory. As a result, an error occurs when the manifest file is used to import data again.

### Solution

- Method 1 (recommended): Publish a new version of the dataset again and use the new manifest file to import data.
- Method 2: Modify the manifest file on your local PC, search for data changes in the OBS directory, and modify the manifest file accordingly. Ensure that the manifest file is consistent with data in the OBS directory, and then import data using the new manifest file.

# 15.2.8 Where Are Labeling Results Stored?

The ModelArts console provides data visualization capabilities, which allows you to view detailed data and labeling information on the console. To learn more about the path for storing labeling results, see the following description.

# Background

When creating a dataset in ModelArts, set both **Input Dataset Path** and **Output Dataset Path** to OBS.

- Input Dataset Path: OBS path where the raw data is stored.
- **Output Dataset Path**: Under this path, directories are generated based on the dataset version after data is labeled in ModelArts and datasets are published. The manifest files (containing data and labeling information) used in ModelArts are also stored in this path. For details about the files, see .

# Procedure

- 1. Log in to the ModelArts console and choose **Data Management** > **Datasets**.
- 2. Select your desired dataset and click the triangle icon on the left of the dataset name to expand the dataset details. You can obtain the OBS path set for **Output Dataset Path**.

#### **NOTE**

Before obtaining labeling results, ensure that at least one dataset version is available.

#### Figure 15-2 Dataset details

∧ <u>dataset-car-pers</u>	Object detection	Apr 23, 2021 11:34:5 🖉	Auto Label Process Data Deploy Model 👻
ID	cTPhr0G0Gp1xWsdwjfq	Name	dataset-car-person
Labeling Type	Object detection	Created	Apr 23, 2021 11:34:59 GMT+08:00
Input Dataset Path	/modelarts-test08/dataset-car-person/data_1619148897910/	Output Dataset Path	/modelarts-test08/dataset-car-person/work_1619148897910/
Version	V002	Label Set	car person
Description		Import Status	Completed View Task History

3. Log in to the OBS console and locate the directory of the corresponding dataset version from the OBS path obtained in 2 to obtain the labeling result of the dataset.

iguie is soluting the tabeling result	Figure	15-3	Obtaining	the	labeling	result
---------------------------------------	--------	------	-----------	-----	----------	--------

Dbject Storage / modelarts-test08 / dataset-car-person / work_1619148897910 / dataset-car-person-cTPhr0 / annotation 🗓								
Objects Deleted Objects Fragments								
Objects are basic units of data storage. In OBS, files and folders are treated as objects. Any file type can be uploaded and managed in a bucket.Learn more								
Upload Object	Upload Object Create Folder Restore More 👻							
Name J≡	Storage Class ↓≡	Size ↓≡	Encrypted J≡					
Sack								
🗌 🖻 V001								
🗋 🔁 V002								

# 15.2.9 How Do I Download Labeling Results to a Local PC?

After being published, the labeling information and data in ModelArts datasets are stored as manifest files in the OBS path set for **Output Dataset Path**.

To obtain the OBS path, do as follows:

- Log in to the ModelArts management console and choose Data Management > Datasets.
- 2. Locate the target dataset and click the triangle icon on the left of the dataset name to expand the dataset details. You can obtain the OBS path set for **Output Dataset Path**.
- 3. Log in to the OBS management console and locate the version directory from the obtained OBS path to obtain the labeling result of the dataset.

To download the labeling results to a local PC, go to the OBS path where the manifest files are stored and click **Download**.

Figure 15-4 Downloading labeling results

Name ↓≡	Storage Class ↓≡	Size ↓≡	Encrypted ↓≡	Restoration Status ↓Ξ	Last Modified 1	Operation
← Back						
V001.manifest	Standard	27.98 KB	No	-	Apr 23, 2021 11:35:45	Download Share More -

# 15.2.10 Why Cannot Team Members Receive Emails for a Team Labeling Task?

The possible causes are as follows:

- All dataset data has been labeled. An email can be sent to team members only if there is unlabeled data in the dataset when the team labeling task is created.
- Team members receive emails for team labeling tasks. No email will be sent when you create a labeling team or add members to a labeling team.
- Your email address has not been configured or has been incorrectly configured. For details about how to configure an email address, see .
- Team members' email addresses are blocked.

# 15.2.11 Can Two Accounts Concurrently Label One Dataset?

Multiple accounts (annotators) are allowed to concurrently label one dataset. However, if multiple annotators concurrently label one image, only the labeling of the last annotator will be used as the labeling result. It is a good practice to label one image by multiple annotators in turn and save the labeling result of each annotator promptly.

# 15.2.12 Can I Delete an Annotator from a Labeling Team with a Labeling Task Assigned? What Is the Impact on the Labeling Result After Deletion? If the Annotator Cannot Be Deleted, Can I Separate the Annotator's Labeling Result?

No annotator cannot be deleted from a labeling team with labeling tasks assigned.

The labeling result of an annotator can be synchronized to the overall labeling result only after the annotator's labeling is approved, and the labeling result cannot be filtered.

# 15.2.13 How Do I Define a Hard Example in Data Labeling? Which Samples Are Identified as Hard Examples?

Hard examples are samples that are difficult to identify. Only image classification and object detection support hard examples.

# 15.2.14 Can I Add Multiple Labeling Boxes to an Object Detection Dataset Image?

Yes.

For an object detection dataset, you can add multiple labeling boxes and labels to an image during labeling. Note that the labeling boxes cannot extend beyond the image boundary.

# 15.2.15 How Do I Merge Two Datasets?

Datasets cannot be merged.

However, you can perform the following operations to merge the data of two datasets into one dataset.

For example, to merge datasets A and B, do the following:

- 1. Publish datasets A and B.
- 2. Obtain the manifest files of the two datasets from the OBS path set for **Output Dataset Path**.
- 3. Create empty dataset C and select an empty OBS folder for **Input Dataset Path**.
- 4. Import the manifest files of datasets A and B to dataset C.

After the import is complete, data in datasets A and B is merged into dataset C. To use the merged dataset, publish dataset C.

# 15.2.16 Does Auto Labeling Support Polygons?

No. Polygons cannot be used in auto labeling. Only rectangles can be used in auto labeling. If a sample is labeled using other bounding boxes, the sample will not be trained.

# 15.2.17 What Do the Options for Accepting a Team Labeling Task Mean?

Modifying Labeled Data	Not overwritten	Overlays		
Acceptance scop	All passed	All rejects	All remaining items pass	All remaining items rejects

- All passed: All items, including the rejected ones will pass the review.
- All rejects: All items, including the ones that have passed the review will be rejected. In this case, the passed items must be labeled and reviewed again in the next acceptance.
- All remaining items pass: The rejected items are still rejected, and the remaining items will automatically pass the review.
- All remaining items rejects: The selected items that have passed the review do not need to be labeled. All the selected items that have been rejected and the items that have not been selected must be labeled again for acceptance.

# 15.2.18 Why Are Images Displayed in Different Angles Under the Same Account?

There are rotation angles of certain images, and the rules of processing such images vary depending on browsers. The following figures show compatibility with browsers.

• L indicates the latest version. L3 indicates the latest three stable browser versions when the product is released.

- If your browser is of an earlier version, the page display will be adversely affected, and the system will prompt you to upgrade your browser.
- If your browser is not compatible with the management console, the system will advise you to upgrade your browser or install a desired browser.

# 15.2.19 Do I Need to Train Data Again If New Data Is Added After Auto Labeling Is Complete?

After auto labeling is complete, confirm the labeled data. If you add new data before confirming the labeled data, all unlabeled data will be automatically labeled again. If you add new data after confirming the labeled data, the data must be trained again.

# 15.2.20 Why Does the System Display a Message Indicating My Label Fails to Save on ModelArts?

# Symptom

Take the Google Chrome browser as an example. When an image is labeled for the first time, the system displays a message in the upper right corner, indicating that the label fails to save. But when the same image is labeled the second time, a message is displayed, indicating that the label is saved. This issue occurs occasionally. When this issue occurs, the request status is

(failed)net::ERR\_ADDRESS\_IN\_USE, which is obtained by pressing F12 on the Google Chrome browser and clicking Network.

Vame	Status	Туре	Initiator	Size	Time	Waterfall
samples	200	xhr	jguerujs8629	762 8	599 ms	4
0040009abe08fbe8af1fc9d18c125b20?worker_id=2a43868ea24e65b6de7443e87c966194	200	xhr	jguery.js:8629	5.1 k8	405 ms	4
2021-12-20.11-05-12-676.jpg?AccessKeyId=S78PNLDJW434Da4%3D8/Signature=TW0J45rpuvm9Le0	200	jpeg	datalabelAnnotationCtrl.jp:1	76.6 kB	312 ms	1
001a56ad54100ecbb66e5e4850a083cb?worker_id=2a43868ea24e65b6de7443e87c966194	200	xhr	jguenuis8629	5.3 k8	158 ms	1
2021-12-20.11-54-52-9130.jpg?AccessKeyId=IMM8S6Q833D8:Signature=%28c36F%2FrQHpnGnDfw	200	jpeg	olis:1	444 k8	232 ms	1
samples	(failed) net::ERR_ADDRESS_IN	, xhr	jguery.js:8629	0.8	105 ms	
0040009abe08fbe8af1fc9d18c125b20?worker_jd=2a43868ea24e65b6de7443e87c966194	200	xhr	jguencis:8629	5.1 kB	357 ms	4
2021-12-20.11-05-12-676.jpg?AccessKeyId=B5HZEF3R27pmxR0%3D8/Signature=Fxk3hYAve4qD5g6o	200	jpeg	dataLabelAnnotationCtrl.jp:1	76.6 k8	97 ms	)
me	200	xhr	jguen, is:8629	914 B	270 ms	

# **Possible Cause**

The local network is faulty, for example, the network is unstable, or the network configuration is incorrect.

# Solution

- 1. Use a stable network and try again.
- 2. Initialize the network configuration. To do so, open Command Prompt as an administrator and execute the **netsh winsock reset** command. Once the initialization is complete, restart your computer and log in to the data labeling platform again.

# 15.2.21 Can One Label By Identified Among Multiple Labels?

After a model is trained with multiple labels and deployed as a real-time service, all the labels are identified. If only one type of label needs to be identified, train a model dedicated for identifying the label. To speed up the label identification, select a high flavor for deploying the model.

# 15.2.22 Why Are Newly Added Images Not Automatically Labeled After Data Amplification Is Enabled?

After data amplification is enabled, images newly added in image classification datasets cannot be automatically labeled, but those added in object detection datasets can be.

# 15.2.23 Why Cannot Videos in a Video Dataset Be Displayed or Played?

If the issue occurs, check the video format. Only MP4 videos can be displayed and played.

# 15.2.24 Why All the Labeled Samples Stored in an OBS Bucket Are Displayed as Unlabeled in ModelArts After the Data Source Is Synchronized?

This issue occurs if automatic encryption is enabled in the OBS bucket. To resolve this issue, create an OBS bucket and upload data to it, or disable bucket encryption and upload data to it again.

# 15.2.25 How Do I Use Soft-NMS to Reduce Bounding Box Overlapping?

YOLOv3 algorithms subscribed to in AI Gallery can use Soft-NMS to reduce overlapped bounding boxes. No official information has been released to show that YOLOv5 algorithms support this function. Use this function in custom algorithms.

# 15.2.26 Why ModelArts Image Labels Are Lost?

The default labeling job is deleted. As a result, the labels are deleted.

# 15.2.27 How Do I Add Images to a Validation or Training Dataset?

You are not allowed to manually add images to a training or validation dataset, but can only set a training and validation ratio. Then, the system randomly allocates the images to the training and validation datasets based on the ratio.

# Setting a Training and Validation Ratio

When you publish a dataset, only the dataset of the image classification, object detection, text classification, or sound classification type supports data splitting.

By default, data splitting is disabled. After this function is enabled, set a training and validation ratio.

Enter a value ranging from 0 to 1 for the training set ratio. After the training set ratio is set, the validation set ratio is determined. The sum of the training set ratio and the validation set ratio is 1.

The training set ratio is the ratio of sample data used for model training. The validation set ratio is the ratio of the sample data used for model validation. The training and validation ratios affect the performance of training templates.

# 15.2.28 Can I Customize Labels for an Object Detection Dataset?

Yes. You can add custom labels to the label set of an object detection dataset by modifying the dataset.

#### Figure 15-5 Modify Dataset

Modify Datas	set	×
Name	auto	
Description	0/256	
Label Set	blue + ঊ none + ঊ ↔ Add Label	
	<b>OK</b> Cancel	

# 15.2.29 What ModelArts Data Management Can Be Used for?

The functions provided ModelArts data management vary depending on the type of the dataset.

Data set Type	Label ing Type	Creat ing a Datas et	lmpo rting Data	Expo rting Data	Publi shing a Datas et	Modi fying a Data set	Mana ging Datas et Versi ons	Auto Grou ping	Data Featu re Engin eerin g
Files	lmag e classif icatio n	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	Supp orted
	Objec t detec tion	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	Supp orted

Data set Type	Label ing Type	Creat ing a Datas et	Impo rting Data	Expo rting Data	Publi shing a Datas et	Modi fying a Data set	Mana ging Datas et Versi ons	Auto Grou ping	Data Featu re Engin eerin g
	lmag e segm entati on	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	N/A
	Soun d classif icatio n	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Speec h labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Speec h parag raph labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Text classif icatio n	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Name d entity recog nition	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Text triplet	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Video s	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Free forma t	Supp orted	N/A	Supp orted	Suppo rted	Supp orted	Supp orted	N/A	N/A
Table s	Table s	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A

# 15.2.30 Will My Old-Version Datasets Be Cleared After the Old Version Is Discontinued? The existing datasets and the ones newly created in the old version will be retained after the old version is discontinued.

The new version is compatible with the old version. However, the datasets created in the new version cannot be displayed in the dataset list of the old version.

# 15.2.31 Why Is My New Dataset Version Unavailable in Versions?

The version list can be zoomed in or out. Zoom out the page before searching.

Click the name of the target dataset to go to the dataset overview page. Then, zoom out the **Versions** page.





# 15.2.32 How Do I View the Size of a Dataset?

Only the number of samples in a dataset is collected in data management. There is no entrance to view the dataset size.

# 15.2.33 How Do I View Labeling Details of a New Dataset?

- Log in to the ModelArts management console and choose Data Management > Datasets from the navigation pane on the left.
- 2. Locate the target dataset by name and click its name. The **Dashboard** tab page is displayed.
- 3. On the **Dashboard** tab page, click **View Details** in the **Labeling Information** area.

Labeling Information	
Object detection	
Name	Labels 🍦
no_mask	306
yes_mask	354

# 15.2.34 How Do I Export Labeled Data?

Only datasets of image classification, object detection, and image segmentation types can be exported.

- For image classification datasets, only the label files in TXT format can be exported.
- For object detection datasets, only XML label files in Pascal VOC format can be exported.
- For image segmentation datasets, only XML label files in Pascal VOC format and mask images can be exported.

For other types of datasets, use to publish the datasets.

# 15.2.35 Why Cannot I Find My Newly Created Dataset?

The datasets of the new version are not displayed on the dataset page of the old version. To view the datasets of the new version, switch to the dataset page of the new version.

Datasets	어 New Version	ew						
Datasets of the old version will be discontinued soon. Use Dataset New . Learn more Documentation.								
Create	Create Maximum datasets: 1000, Available for creation: 419							
	Name	Labeling Type	Labeling Progress (Labe	led/Total)				
~	xianao-text-3	Text classification		38% (15/40)				
~	dataset-9c2c	Text classification		99% (21120/21176)				

# 15.2.36 What Do I Do If the Database Quota Is Incorrect?

The quota for the datasets of both the old and new versions is 100. On the dataset page of the new version, all created datasets are displayed. However, the dataset page of the old version does not display the new-version datasets. Go to the dataset page of the new version to view the datasets.

# 15.2.37 How Do I Split a Dataset?

When you publish a dataset, only the dataset of the image classification, object detection, text classification, or sound classification type supports data splitting.

By default, data splitting is disabled. After this function is enabled, set the training and validation ratios.

Enter a value ranging from 0 to 1 for the training set ratio. After the training set ratio is set, the validation set ratio is determined. The sum of the training set ratio and the validation set ratio is 1.

The training set ratio is the ratio of sample data used for model training. The validation set ratio is the ratio of the sample data used for model validation. The training and validation ratios affect the performance of training templates.

# 15.2.38 How Do I Delete a Dataset Image?

- Log in to the ModelArts management console. In the navigation pane, choose Data Management > Label Data. The data labeling list is displayed. Click the dataset from which you want to delete images. The labeling details page is displayed.
- 2. On the All statuses, Unlabeled, or Labeled tab page, select the images to be

deleted or click **Select Images on Current Page**, and click  $\begin{tabular}{ll} \hline U \end{tabular}$  to delete them. In the displayed dialog box, select or deselect **Delete the source files from OBS** as required. After confirmation, click **Yes** to delete the images.

If a tick is displayed in the upper left corner of an image, the image is

selected. If no image is selected on the page,  $\Box$  is unavailable.

#### Figure 15-6 Deleting a dataset image



# 15.2.39 Why Is There No Sample in the ModelArts Dataset Downloaded from AI Gallery and Then an OBS Bucket?

Check the format of the data downloaded from AI Gallery. For example, compressed packages and Excel files will be ignored. The following table lists the supported formats.

Data set Type	Label ing Type	Creat ing a Datas et	lmpo rting Data	Expo rting Data	Publi shing a Datas et	Modi fying a Data set	Mana ging Datas et Versi ons	Auto Grou ping	Data Featu re Engin eerin g
-iles	lmag e classif icatio n	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	Supp orted
	Objec t detec tion	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	Supp orted
	lmag e segm entati on	Supp orted	Supp orted	Supp orted	Suppo rted	Supp orted	Supp orted	Supp orted	N/A
	Soun d classif icatio n	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Speec h labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Speec h parag raph labeli ng	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Text classif icatio n	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Name d entity recog nition	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Text triplet	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A

Data set Type	Label ing Type	Creat ing a Datas et	lmpo rting Data	Expo rting Data	Publi shing a Datas et	Modi fying a Data set	Mana ging Datas et Versi ons	Auto Grou ping	Data Featu re Engin eerin g
	Video s	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A
	Free forma t	Supp orted	N/A	Supp orted	Suppo rted	Supp orted	Supp orted	N/A	N/A
Table s	Table s	Supp orted	Supp orted	N/A	Suppo rted	Supp orted	Supp orted	N/A	N/A

# 15.3 Notebook (New Version)

# 15.3.1 Constraints

# 15.3.1.1 Is sudo Privilege Escalation Supported?

For security purposes, notebook instances do not support sudo privilege escalation.

# 15.3.1.2 Does ModelArts Support apt-get?

**Terminal** in ModelArts DevEnviron does not support **apt-get**. You can use a **custom image** to support it.

# 15.3.1.3 Is the Keras Engine Supported?

Notebook instances in **DevEnviron** support the Keras engine. The Keras engine is not supported in job training and model deployment (inference).

Keras is an advanced neural network API written in Python. It is capable of running on top of TensorFlow, CNTK, or Theano. Notebook instances in **DevEnviron** support **tf.keras**.

# How Do I View Keras Versions?

- 1. On the ModelArts management console, create a notebook instance with image **TensorFlow-1.13** or **TensorFlow-1.15**.
- 2. Access the notebook instance. In JupyterLab, run **!pip list** to view Keras versions.

🔳 Untit	led.ipynb	
8 +	% ⊡ ₿ ■	C ↦ Code ∨ () git
	idna	3.3
	importlib-resources	5.7.1
	inicontig ipykernel	1.1.1 6.7.0
	ipython	7.31.1
	ipython-genutils jedi	0.2.0 0.18.1
	Jinja2	3.1.2
	jinja2-time jmespath	0.2.0 0.10.0
	joblib	1.1.0
	jsonschema jupyter-client	7.1.2
	jupyter-core Kapas	4.9.1
	Keras-Applications	1.0.8
	Keras-Preprocessing	1.1.2
	keyboard	0.13.5
	TxmT	4.8.0.post20220315201753

Figure 15-7 Viewing Keras versions

# 15.3.1.4 Does ModelArts Support the Caffe Engine?

The Python 2 environment of ModelArts supports Caffe, but the Python 3 environment does not support it.

# 15.3.1.5 Can I Install MoXing in a Local Environment?

No. MoXing can be used only on ModelArts.

# 15.3.1.6 Can Notebook Instances Be Remotely Logged In?

The notebook instances of the new version can be remotely logged in. To do so, enable remote SSH when you create the notebook instances. Remotely log in to a notebook instance from a local IDE through or .

# 15.3.2 Data Upload or Download

# 15.3.2.1 How Do I Upload a File from a Notebook Instance to OBS or Download a File from OBS to a Notebook Instance?

In a notebook instance, you can call the ModelArts MoXing API or SDK to exchange data with OBS for uploading a file to OBS or downloading a file from OBS to the notebook instance.





### Method 1: Using MoXing to Upload and Download a File

Developed by the ModelArts team, MoXing is a distributed training acceleration framework built on open-source deep learning engines such as TensorFlow and PyTorch. MoXing makes model coding easier and more efficient.

MoXing provides a set of file object APIs for reading and writing OBS files.

Sample code:

import moxing as mox

# Download the OBS folder sub\_dir\_0 from OBS to a notebook instance. mox.file.copy\_parallel('obs://bucket\_name/sub\_dir\_0', '/home/ma-user/work/sub\_dir\_0') # Download the OBS file obs\_file.txt from OBS to a notebook instance. mox.file.copy('obs://bucket\_name/obs\_file.txt', '/home/ma-user/work/obs\_file.txt')

# Upload the OBS folder sub\_dir\_0 from a notebook instance to OBS. mox.file.copy\_parallel('/home/ma-user/work/sub\_dir\_0', 'obs://bucket\_name/sub\_dir\_0') # Upload the OBS file obs\_file.txt from a notebook instance to OBS. mox.file.copy('/home/ma-user/work/obs\_file.txt', 'obs://bucket\_name/obs\_file.txt')

### Method 2: Using SDK to Upload and Download a File

Call the ModelArts SDK for downloading a file from OBS.

Sample code: Download **file1.txt** from OBS to **/home/ma-user/work/** in the notebook instance. All the bucket name, folder name, and file name are customizable.

from modelarts.session import Session session = Session() session.obs.download\_file(src\_obs\_file="obs://bucket-name/dir1/file1.txt", dst\_local\_dir="/home/ma-user/ work/")

Call the ModelArts SDK for downloading a folder from OBS.

Sample code: Download **dir1** from OBS to **/home/ma-user/work/** in the notebook instance. The bucket name and folder name are customizable.

from modelarts.session import Session session = Session() session.obs.download\_dir(src\_obs\_dir="obs://bucket-name/dir1/", dst\_local\_dir="/home/ma-user/work/")

Call the ModelArts SDK for uploading a file to OBS.

Sample code: Upload **file1.txt** in the notebook instance to OBS bucket **obs://bucket-name/dir1/**. All the bucket name, folder name, and file name are customizable.

from modelarts.session import Session
session = Session()
session.obs.upload\_file(src\_local\_file='/home/ma-user/work/file1.txt', dst\_obs\_dir='obs://bucket-name/dir1/')

Call the ModelArts SDK for uploading a folder to OBS.

Sample code: Upload /work/ in the notebook instance to **obs://bucket-name/ dir1/work/** of **bucket-name**. The bucket name and folder name are customizable.

from modelarts.session import Session
session = Session()
session.obs.upload\_dir(src\_local\_dir='/home/ma-user/work/', dst\_obs\_dir='obs://bucket-name/dir1/')

# 15.3.2.2 How Do I Upload Local Files to a Notebook Instance?

For details about how to upload files to JupyterLab in notebook of the new version, see **Uploading Files to JupyterLab**.

### 15.3.2.3 How Do I Import Large Files to a Notebook Instance?

• Large files (files larger than 100 MB)

Use OBS to upload large files. To do so, use OBS Browser to upload a local file to an OBS bucket and use ModelArts SDK to download the file from OBS to a notebook instance.

For details about how to use ModelArts SDK or MoXing to download files from OBS, see **How Do I Upload a File from a Notebook Instance to OBS** or **Download a File from OBS to a Notebook Instance?** 

• Folders

Compress a folder into a package and upload the package in the same way as uploading a large file. After the package is uploaded to a notebook instance, decompress it on the **Terminal** page.

unzip xxx.zip # Directly decompress the package in the path where the package is stored.

For more details, search for the decompression command in mainstream search engines.

### 15.3.2.4 Where Will the Data Be Uploaded to?

If you use OBS to store the notebook instance, after you click **upload**, the data is directly uploaded to the target OBS path, that is, the OBS path specified when the notebook instance is created.

# 15.3.2.5 How Do I Download Files from a Notebook Instance to a Local Computer?

For details about how to download files from JupyterLab in notebook of the new version, see **Downloading a File from JupyterLab to a Local Path**.

# **15.3.2.6 How Do I Copy Data from Development Environment Notebook A to Notebook B?**

Data cannot be directly copied from notebook A to notebook B. To copy data, do as follows:

- 1. Upload the data of notebook A to OBS.
- 2. Download data from OBS to notebook B.

For details about how to upload and download files, see **How Do I Upload a File** from a Notebook Instance to OBS or Download a File from OBS to a Notebook Instance?

# 15.3.2.7 What Can I Do If a File Fails to Be Uploaded to a Notebook Instance?

### Symptom

- The file upload process is fast but unsuccessful.
- When a file is uploaded to a notebook instance, the uploading is consistently in progress on the GUI. When a file is uploaded through MoXing, an error occurred. When an OBS file is uploaded, no bucket is displayed, and the message "Obtaining data" is displayed.
- When you click <sup>1</sup> to upload files on the JupyterLab page, "Failed to obtain data" is displayed.

#### Figure 15-9 OBS file upload

Add files to Notebook

OBS file upload				
Enter the OBS file path, or selec	t a path from OBS File Browser		UPLO	DAD
CLOSE OBS FILE BROWSE	ER ^			
obs				
Back	E	inter a name for query		C
Name	Last Modified	Туре	Size	
	لے Failed to obtain dat	a		
	Rows per	page: 10 ▼ 0-0 of 0 K	< >	>

When you check the notebook log (usually **notebook-<date>.log** in / **home/ma-user/log/**), the error message **List objects failed, obs\_client resp: {'status' : 403, 'reason' : 'Forbidden' , 'errorCode' : 'AccessDenied'** is displayed.

### **Possible Cause**

The first symptom is because that the size of a file is limited when it is uploaded through the intranet.

The possible causes of other symptoms are as follows:

X

- OBS access is not authorized.
- You do not have the permission to access the OBS bucket or file.
- The OBS bucket has been deleted.

# Solution

- Check agency authorization. Go to the Global Configuration page and check whether you have the OBS access permission. If you do not, see "Configuring Access Authorization (Global Configuration)".
- Check whether you have the permission to access the OBS bucket. Go to the OBS console, access the target bucket, and check if there is an error showing that you do not have the access permission.
- Go to the OBS console and check whether the OBS bucket exists.

# 15.3.2.8 Failed to View the Local Mount Point of a Dynamically Mounted OBS Parallel File System in JupyterLab of a Notebook Instance

### Symptom

When an OBS parallel file system is dynamically mounted to a notebook instance, the local mount directory is **/data/demo-yf/**, which, however, is not displayed in the navigation pane on the left of JupyterLab.

F <b>igure 15-10</b> Loca	l mount directory
---------------------------	-------------------

Туре	Status	Storage Path	Cloud Mount Path
Parallel File System	Mounted	obs:// /	/data/demo/

Figure 15-11 JupyterLab of notebook

$\bigwedge$	File	Edit	View	Run	Kernel	Git	Tabs	Settings	Help	
		+			1	È	C	¢		
0	Fil	ter file: /	s by na	me					Q	Γ
	Nam	те					•	Last M	odified	L
		.model	arts					a c	lay ago	L
≣		lost+fo	ound					a c	lay ago	L

### **Possible Causes**

The local mount directory is the **demo-yf** folder created in the **~/data** directory of the notebook container. However, the default path of the navigation pane on the left of JupyterLab is the **~/work** directory, which means that **/data** and **/work** are of the same level. As a result, the directory cannot be displayed in JupyterLab.

After Terminal is opened, the default directory is **~work**. Run the following commands to go to the **~data** directory and view the local mount directory:

(PyTorch-1.8) [ma-user work]\$cd (PyTorch-1.8) [ma-user ~]\$cd /data (PyTorch-1.8) [ma-user data]\$ls

(PyTorch-1.8)	[ma-user	work]\$cd
(PyTorch-1.8)	[ma-user	~]\$cd /data
(PyTorch-1.8)	[ma-user	data]\$ls
demo-yf		

# 15.3.3 Data Storage

# 15.3.3.1 How Do I Rename an OBS File?

OBS files cannot be renamed on the OBS console. To rename an OBS file, call a MoXing API in an existing or newly created notebook instance.

The following shows an example:

Rename **obs\_file.txt** to **obs\_file\_2.txt**. import moxing as mox mox.file.rename('obs://bucket\_name/obs\_file.txt', 'obs://bucket\_name/obs\_file\_2.txt')

# 15.3.3.2 Do Files in /cache Still Exist After a Notebook Instance is Stopped or Restarted? How Do I Avoid a Restart?

Temporary files are stored in the **/cache** directory and will not be saved after the notebook instance is stopped or restarted. Data stored in the **/home/ma-user/work** directory will be retained after the notebook instance is stopped or restarted.

To avoid a restart, do not execute heavy-load training jobs that consume large amounts of resources in the development environment.

# 15.3.3.3 How Do I Use the pandas Library to Process Data in OBS Buckets?

- Step 1 Download data from OBS to a notebook instance. For details, see Downloading a File from JupyterLab to a Local Path.
- Step 2 Process pandas data by following the instructions provided in *pandas User Guide*.

----End

# **15.3.4 Environment Configurations**

# 15.3.4.1 How Do I Check the CUDA Version Used by a Notebook Instance?

Run the following command to view the CUDA version of the target notebook instance:

ll /usr/local | grep cuda

The following shows an example.

#### Figure 15-12 Checking the CUDA version in the current environment



In the preceding example, the CUDA version is 10.2.

### 15.3.4.2 How Do I Enable the Terminal Function in DevEnviron of ModelArts?

- 1. Log in to the ModelArts management console, and choose **DevEnviron** > **Notebooks**.
- 2. Create a notebook instance. When the instance is running, click **Open** in the **Operation** column. The **JupyterLab** page is displayed.
- 3. Choose File > New > Terminal. The Terminal page is displayed.

#### Figure 15-13 Going to the Terminal page

M	File	Edit	View	Run	Kernel	Git	Tabs	Set	ting	gs	Help
	N	lew							Þ	۶.,	Console
_	N	lew Lau	incher			(	Ctrl+Shi	ft+L			Notebook
Ø	0	pen fro	om Path							\$_	Terminal

### 15.3.4.3 How Do I Install External Libraries in a Notebook Instance?

Multiple environments such as Jupyter and Python have been integrated into ModelArts notebook to support many frameworks, including TensorFlow, MindSpore, PyTorch, and Spark. You can use **pip install** to install external libraries in Jupyter Notebook or on the **Terminal** page.

### Installing External Libraries in Jupyter Notebook

You can use JupyterLab to install Shapely in the **TensorFlow-1.8** environment.

- 1. Open a notebook instance and access the **Launcher** page.
- 2. In the **Notebook** area, click **TensorFlow-1.8** and create an IPYNB file.
- 3. In the new notebook instance, enter the following command in the code input bar:

#### !pip install Shapely

# Installing External Libraries on the Terminal Page

You can use **pip** to install external libraries in the **TensorFlow-1.8** environment on the **Terminal** page. For example, to install Shapely:

- 1. Open a notebook instance and access the **Launcher** page.
- 2. In the **Other** area, click **Terminal** and create a terminal file.
- 3. Enter the following commands in the code input box to obtain the kernel of the current environment and activate the Python environment on which the installation depends:

#### cat /home/ma-user/README

#### source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8

#### **NOTE**

To install TensorFlow in another Python environment, replace **TensorFlow-1.8** in the command with the target engine.

#### Figure 15-14 Activating the environment

h-4.3\$cat /home/ma-user/README
lease use one of following command to start the specifed framework environment.
or Conda-python3 source /home/ma-user/anaconda3/bin/activate base
or MXNet-1.2.1 source /home/ma-user/anaconda3/bin/activate MXNet-1.2.1
or PySpark-2.3.2PySpark-2.3.2
or Pytorch-1.0.0Pytorch-1.0.0
or TensorFlow-1.13.1 source /home/ma-user/anaconda3/bin/activate TensorFlow-1.13.1
or TensorFlow-1.8 source /home/ma-user/anaconda3/bin/activate TensorFlow-1.8
or XGBoost-SklearnXGBoost-Sklearn source /home/ma-user/anaconda3/bin/activate XGBoost-Sklearn

4. Run the following command in the code input box to install Shapely: **pip install Shapely** 

### 15.3.4.4 How Do I Obtain the External IP Address of My Local PC?

Search for "IP address lookup" in a mainstream search engine.

#### Figure 15-15 IP address lookup

WhatIsMyIP.com<sup>®</sup> » Tools » IP Address Lookup



# 15.3.4.5 How Can I Resolve Abnormal Font Display on a ModelArts Notebook Accessed from iOS?

#### Symptom

When a ModelArts notebook is accessed from iOS, the font is displayed abnormally.

### Solution

Set fontFamily of Terminal to Menlo.

### Procedure

**Step 1** Log in to the ModelArts management console and choose **DevEnviron** > **Notebook**.

- **Step 2** Locate the row containing the target notebook instance and click **Open** in the **Operation** column. The **JupyterLab** page is displayed.
- **Step 3** On the **JupyterLab** page, choose **Settings > Advanced Settings Editor**. The **Settings** tab page is displayed.



**Step 4** Choose **Terminal** in the navigation pane on the left and set **fontFamily** to **Menlo**.



----End

### 15.3.4.6 Is There a Proxy for Notebook? How Do I Disable It?

There is a proxy for Notebook.

Run the **env|grep proxy** command to obtain the notebook proxy.

Run the **unset https\_proxy unset http\_proxy** command to disable the proxy.

# 15.3.5 Notebook Instances

# 15.3.5.1 What Do I Do If I Cannot Access My Notebook Instance?

Troubleshoot the issue based on error code.

### A Black Screen Is Displayed When a Notebook Instance Is Opened

A black screen is displayed after a notebook instance is opened, which is caused by a proxy issue. Change the proxy to rectify the fault.

### A Blank Page Is Displayed When a Notebook Instance Is Opened

- If a blank page is displayed after a notebook instance is opened, clear the browser cache and open the notebook instance again.
- Check whether the ad filtering component is installed for the browser. If yes, disable the component.

# Error 404

If this error is reported when an IAM user creates an instance, the IAM user does not have the permissions to access the corresponding storage location (OBS bucket).

Solution

- 1. Log in to the OBS console using the primary account and grant access permissions for the OBS bucket to the IAM user.
- 2. After the IAM user obtains the permissions, log in to the ModelArts console, delete the instance, and use the OBS path to create a notebook instance.

### Error 503

If this error is reported, it is possible that the instance is consuming too many resources. If this is the case, stop the instance and restart it.

### Error 500

Notebook JupyterLab cannot be opened, and error 500 is reported. The possible cause is that the disk space in the **work** directory is used up. In this case, identify the fault cause and clear the disk by referring to .

### Error "This site can't be reached"

After a notebook instance is created, click **Open** in the **Operation** column. The error message shown in the following figure is displayed.

This site can't be reached	
The webpace at https://authoring-modelarts. /lab might be tempo permanently to a new web address.	.huaweicloud.com/dfc45125- orarily down or it may have moved
ERR_TUNNEL_CONNECTION_FAILED	

To solve the problem, copy the domain name of the page, add it to the **Do not use proxy server for addresses beginning with** text box, and save the settings.

手动设置代理
将代理服务器用于以太网或 Wi-Fi 连接。这些设置不适用于 VPN 连接。
使用代理服务器
💭 л
地址
http:// i.com 8080
请勿对以下列条目开头的地址使用代理服务器。若有多个条目,请使用英 文分号 (:) 来分隔。
n;
✓ 请勿将代理服务器用于本地(Intranet)地址
保存

# 15.3.5.2 What Should I Do When the System Displays an Error Message Indicating that No Space Left After I Run the pip install Command?

### Symptom

In the notebook instance, error message "No Space left..." is displayed after the **pip install** command is run.

### Solution

You are advised to run the **pip install --no-cache \*\*** command instead of the **pip install \*\*** command. Adding the **--no-cache** parameter can solve such problem.

# 15.3.5.3 What Do I Do If "Read timed out" Is Displayed After I Run pip install?

### Symptom

After I run **pip install** in a notebook instance, the system displays error message "ReadTimeoutError..." or "Read timed out...".



# Solution

Run **pip install --upgrade pip** and then **pip install**.

# 15.3.5.4 What Do I Do If the Code Can Be Run But Cannot Be Saved, and the Error Message "save error" Is Displayed?

If the notebook instance can run the code but cannot save it, the error message "save error" is displayed when you save the file. In most cases, this error is caused by a security policy of Web Application Firewall (WAF).

On the current page, some characters in your input or output of the code are intercepted because they are considered to be a security risk. Submit a service ticket and contact customer service to check and handle the problem.

# 15.3.5.5 When the SSH Tool Is Used to Connect to a Notebook Instance, Server Processes Are Cleared, but the GPU Usage Is Still 100%

This fault occurs because code execution is suspended and the GPU memory is not released. Alternatively, the program is cleared due to memory overflow during code execution. In this case, you need to release the GPU memory and restart the instance. To avoid unsaved code caused by the end of processes, you are advised to periodically save the code to an OBS bucket or the **./work** directory of the container.

# 15.3.6 Code Execution

# 15.3.6.1 What Do I Do If a Notebook Instance Won't Run My Code?

If a notebook instance fails to execute code, you can locate and rectify the fault as follows:

 If the execution of a cell is suspended or lasts for a long time (for example, the execution of the second and third cells in Figure 15-16 is suspended or lasts for a long time, causing execution failure of the fourth cell) but the notebook page still responds and other cells can be selected, click interrupt the kernel highlighted in a red box in the following figure to stop the execution of all cells. The notebook instance retains all variable spaces.

#### Figure 15-16 Stopping all cells

8	+	ж	ē	Ĩ	►		C	**	Code
75 ms	[2]	a =	1						
	[]	whil	le(1) pass	): 5					
	[]	impo time	ort t e.sle	time eep(1	000	9)			
	[]	a							

- 2. If the notebook page does not respond, close the notebook page and the ModelArts console. Then, open the ModelArts console and access the notebook instance again. The notebook instance retains all the variable spaces that exist when the notebook instance is unavailable.
- 3. If the notebook instance still cannot be used, access the **Notebook** page on the ModelArts console and stop the notebook instance. After the notebook instance is stopped, click **Start** to restart the notebook instance and open it. The instance will have preserved all the spaces for the variables that were unable to run.

# 15.3.6.2 Why Does the Instance Break Down When dead kernel Is Displayed During Training Code Running?

The notebook instance breaks down during training code running due to insufficient memory caused by large data volume or excessive training layers.

After this error occurs, the system automatically restarts the notebook instance to fix the instance breakdown. In this case, only the breakdown is fixed. If you run the training code again, the failure will still occur. To solve the problem of insufficient memory, you are advised to create a new notebook instance and use a resource pool of higher specifications, such as a dedicated resource pool, to run the training code. An existing notebook instance that has been successfully created cannot be scaled up using resources with higher specifications.

# 15.3.6.3 What Do I Do If cudaCheckError Occurs During Training?

#### Symptom

The following error occurs when the training code is executed in a notebook:

cudaCheckError() failed : no kernel image is available for execution on the device

### Possible Cause

Parameters **arch** and **code** in **setup.py** have not been set to match the GPU compute power.

# Solution

For Tesla V100 GPUs, the GPU compute power is **-gencode** arch=compute\_70,code=[sm\_70,compute\_70]. Set the compilation parameters in setup.py accordingly.

### 15.3.6.4 What Should I Do If DevEnviron Prompts Insufficient Space?

If space is insufficient, use notebook instances of the EVS type.

Upload code and data to an OBS bucket for the original notebook instance by referring to How Do I Upload a File from a Notebook Instance to OBS or Download a File from OBS to a Notebook Instance?. Then, create a notebook instance of the EVS type, and download files from OBS to the new notebook instance.

# 15.3.6.5 Why Does the Notebook Instance Break Down When opency.imshow Is Used?

### Symptom

When opency imshow is used in a notebook instance, the notebook instance breaks down.

### **Possible Causes**

The cv2.imshow function in OpenCV malfunctions in a client/server environment such as Jupyter. However, Matplotlib does not have this problem.

### Solution

Display images by referring to the following example. Note that OpenCV displays BGR images while Matplotlib displays RGB images.

Python:

from matplotlib import pyplot as plt
import cv2
img = cv2.imread('Image path')
plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))
plt.title('my picture')
plt.show()

# 15.3.6.6 Why Cannot the Path of a Text File Generated in Windows OS Be Found In a Notebook Instance?

# Symptom

When a text file generated in Windows is used in a notebook instance, the text content cannot be read and an error message may be displayed indicating that the path cannot be found.

### **Possible Causes**

The notebook instance runs Linux and its line feed format (CRLF) differs from that (LF) in Windows.

### Solution

Convert the file format to Linux in your notebook instance.

Shell:

dos2unix *File name* 

# 15.3.6.7 What Do I Do If Files Fail to Be Saved in JupyterLab?

### Symptom

When a file is saved in JupyterLab, an error message is displayed.

File Save Error fo	r rebar	count.ipynb
--------------------	---------	-------------

Failed to fetch

# Possible Cause

• A third-party plug-in has been installed on the browser, and the proxy intercepts the request. As a result, the file cannot be saved.

Dismiss

- The runtime file in the notebook is too large.
- You have stayed on the Jupyter page for too long.
- There is a network error. Check whether a network proxy is connected.

### Solution

- Disable the plug-in and save the file again.
- Reduce the file size.
- Open the Jupyter page again.
- Check the network.

# 15.3.7 Failures to Access the Development Environment Through VS Code

# 15.3.7.1 What Do I Do If the VS Code Window Is Not Displayed?

# Possible Cause

VS Code is not installed or the installed version is outdated.

# Solution

Download and install VS Code. (Windows users click **Windows**. Users of other operating systems click **another OS**.) After the installation, click **refresh** to complete the connection.



# 15.3.7.2 What Do I Do If a Remote Connection Failed After VS Code Is Opened?

NOTICE

If your local PC runs Linux, see possible cause 2.

### Possible Cause 1

Automatically installing the VS Code plug-in failed.

### Solution

Method 1: Verify that the VS Code network is accessible. Search for **ModelArts** in the VS Code marketplace. If the following information is displayed, a network error occurred. In this case, switch to another proxy or use another network.



Search for **ModelArts** again. If the following information is displayed, the network is normal. Then, switch back to the ModelArts management console and try to access VS Code again.



Method 2: If the error message shown in the following figure is displayed, the VS Code version is outdated. Upgrade the VS Code to 1.57.1 or the latest version.



# Possible Cause 2

The local PC runs Linux, and VS Code is installed as user **root**. When you access VS Code, the information "It is not recommended to run Code as root user" is displayed.



#### Solution

Install VS Code as a non-**root** user, return to the ModelArts management console, and click **Access VS Code**.

:~/VSCode\$ sudo dpkg -i code_1.67.2-1652812855_amd64.deb
[sudo] password for dc:
(Reading database 200705 files and directories currently installed.)
Preparing to unpack code_1.67.2-1652812855_amd64.deb
Jnpacking code (1.67.2-1652812855) over (1.67.2-1652812855)
Setting up code (1.67.2-1652812855)
Processing triggers for gnome-memus (3.13.3-11ubuntu1.1)
Processing triggers for desktop-file-utils (0.23-lubuntu3.18.04.2)
Processing triggers for mime-support (3.60ubuntul)
Processing triggers for shared-mime-info (1.9-2)
::~/VSCode\$ code
, Alecadet shred COO KeyDair Masda ser



# 15.3.7.3 Basic Problems Causing the Failures to Access the Development Environment Through VS Code

If the VS Code fails to connect to the development environment, perform the following steps:

**Step 1** Check whether the plug-in package is of the latest version. Search for the plug-in in extensions and check whether it needs to be upgraded.



- **Step 2** Check whether the instance is running. If yes, go to the next step.
- Step 3 Run the following command in VS Code's Terminal to connect to the remote development environment: ssh -tt -o StrictHostKeyChecking=no -i \${IdentityFile} \${User}@\${HostName} -p \${Port}

Parameters:

- IdentityFile: path to the local key
- User: username, for example, ma-user
- HostName: IP address
- Port: port number



If the connection is successful, go to the next step.

**Step 4** Check whether the configuration is correct. If yes, go to the next step.

#### Check the **config** file.

Select SSH configuration file to update						
C:\Users\	,.ssh\config					
C:\ProgramData\ssh\ssh_config						
Settings specify a custom configuration file						
Help about SSH configuration files						

#### HOST remote-dev

- hostname <instance connection host> port <instance connection port> user ma-user IdentityFile ~/.ssh/test.pem StrictHostKeyChecking no UserKnownHostsFile /dev/null ForwardAgent yes
- **Step 5** Check the key file. You are advised to save the key file in C:\Users\xx.ssh and ensure that the file does not contain Chinese characters.
- **Step 6** If the fault persists, rectify it by referring to the FAQs in **follow-up sections**.

#### ----End
# 15.3.7.4 What Do I Do If Error Message "Could not establish connection to xxx" Is Displayed During a Remote Connection?

# Symptom



# **Possible Cause**

Establishing a remote SSH connection to an instance through VS Code failed.

### Solution

Close the displayed dialog box, view the error information in **OUTPUT**, and rectify the fault by referring to the troubleshooting methods provided in the following sections.

# 15.3.7.5 What Do I Do If the Connection to a Remote Development Environment Remains in "Setting up SSH Host xxx: Downloading VS Code Server locally" State for More Than 10 Minutes?

# Symptom



# **Possible Cause**

The local network is faulty. As a result, it takes a long time to automatically install the VS Code server remotely.

# Solution

Manually install the VS Code server.

**Step 1** Obtain the VS Code commit ID.

Visual Studio Code

 $\times$ 

15 FAQs

1	Visual Studio Code
	Version: 1.57.1 (user setup) Commit: 507ce72a4466fbb27b715c3722558bb15afa9f48
	Date: 2021-06-17113:28:07.7552 (10 mos ago) Electron: 12.0.7 Chrome: 89.0.4389.128 Node.js: 14.16.0 V8: 8.9.255.25-electron.0 OS: Windows_NT x64 10.0.19042
	ОК Сору

**Step 2** Download the VS Code server package of the required version. Select Arm or x86 based on the CPU architecture of the development environment.

#### **NOTE**

Replace *\${commitID}* in the following link with the commit ID obtained in **1**.

- For Arm, download vscode-server-linux-arm64.tar.gz. https://update.code.visualstudio.com/commit:\${commitID}/server-linuxarm64/stable
- For x86, download vscode-server-linux-x64.tar.gz. https://update.code.visualstudio.com/commit:\${commitID}/server-linux-x64/ stable
- **Step 3** Access the remote environment.

Switch to **Terminal** in VS Code.



Run the following command in VS Code Terminal to access the remote development environment:

ssh -tt -o StrictHostKeyChecking=no -i \${IdentityFile} \${User}@\${HostName} -p \${Port}

Parameters:

- IdentityFile: Path to the local key
- User: Username, for example, ma-user
- HostName: IP address
- Port: Port number

ame	notebook	Flavor	modelarts.vm.cpu.2u 👻
atus	Stopped	Image	pytorch1.4-cuda10.1-cudnn7-ubuntu18.04
	Ũ	Created At	May,18,2022 16:19:08 GMT+08:00
orage Path	/home/ma-user/work/	Updated At	May,18,2022 18:33:53 GMT+08:00
orage Capacity	50 GB (Default)	Address	ssh://ma-user@dev-modelartsiuaweicloud.com 30004
hitelist	🖉	Authentication	KeyPair-9559

#### **Step 4** Manually install the VS Code server.

Run the following commands on the VS Code terminal to clear the residual data (replace *\${commitID}* in the commands with the commit ID obtained in 1):

rm -rf /home/ma-user/.vscode-server/bin/\${commitID}/\* mkdir -p /home/ma-user/.vscode-server/bin/\${commitID}

Upload the VS Code server package to the development environment.

exit scp -i xxx.pem -P 31205 Local path to the VS Code server package ma-user@xxx:/home/ma-user/.vscodeserver/bin ssh -tt -o StrictHostKeyChecking=no -i \${IdentityFile} \${User}@\${HostName} -p \${Port}

Parameters:

- IdentityFile: Path to the local key
- User: Username, for example, ma-user
- HostName: IP address
- Port: Port number

Take Arm as an example. Decompress the VS Code server package to **\$HOME/.vscode-server/bin**. Replace *\${commitID}* in the command with the commit ID obtained in **1**.

cd /home/ma-user/.vscode-server/bin tar -zxf vscode-server-linux-arm64.tar.gz mv vscode-server-linux-arm64/\* \${commitID}

**Step 5** Establish the remote connection again.

----End

15.3.7.6 What Do I Do If the Connection to a Remote Development Environment Remains in the State of "Setting up SSH Host xxx: Downloading VS Code Server locally" for More Than 10 Minutes?

### Symptom

×1 -	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>T</u> erminal <u>H</u> elp	Get S	tarted - \	√isual Studio C	ode		пx
(J)	≺ Get Started ×						
	Start				Walkthroughs		
2º					Learn the Fun Jump right into	ndamentals o VS Code and get an overview of the must-have features.	
<del>a</del> >							
6	Recent						
s	20a592e7-83ab-41f2-a6b0-8c00ac468512 [SSH: ModelArts-devserv PROBLEMS OUTPUT TERMINAL PORTS	er-gyx-bi				Remote - SSH	ግ ^ X
7 7	<pre></pre>	tar.gz" Local/T 9 north7. 9% 22% 44% 66% 87% 100% 9%	"vsco emp/vs ulanqa 0 12MB 23MB 35MB 46MB 53MB 6	de-scp-dor <u>code_serve</u> b.huawei.c 0.0KB/s 11.9MB/s 11.8MB/s 11.7MB/s 11.0MB/s 0.0KB/s	e.flag" "ModelAi r_1654051611455 om]:32538,[1 : ETA 00:03 ETA 00:02 ETA 00:02 ETA 00:04 ETA 00:04 ETA 00:04 ETA	rts-devserver-78f5":"/root/.vscode-server/bin/	
8	<pre>[10:46:57.569] &gt; [10:46:58.802] "Copy server to host" terminal command done [10:49:47.820] &gt;</pre>					(U) Setting Up SSH Host ModelArts-devserver-78fs: Copying VS Code Server to host with scp Source: Remote - SSH (Extension)	@ <u>~</u>
Ø Ope	ning Remote ⑧ 0 쇼 0 钟 0						R C

### **Possible Cause**

Logs show that **vscode-scp-done.flag** has been uploaded locally, but it is not received on the remote end.

# Solution

Close all VS Code windows, return to the ModelArts management console, and click **Access VS Code**.

# 15.3.7.7 What Do I Do If the Connection to a Remote Development Environment Remains in the State of "ModelArts Remote Connect: Connecting to instance xxx..." for More Than 10 Minutes?

# Symptom



# Solution

Click **Cancel**, return to the ModelArts management console, and click **Access VS Code**.

# 15.3.7.8 What Do I Do If a Remote Connection Is in the Retry State?

#### Symptom



#### 15 FAQs

### **Possible Cause**

Downloading the VS Code server failed before, leading to residual data. As a result, new download cannot be performed.

#### Solution

Method 1 (performed locally): Open the command panel (**Ctrl+Shift+P** for Windows and **Cmd+Shift+P** for macOS), search for **Kill VS Code Server on Host**, and locate the affected instance, which will be automatically cleared. Then, establish the connection again.



Figure 15-17 Clearing the affected instance

Method 2 (performed remotely): Delete the files that are being used in / home/ma-user/.vscode-server/bin/ on the VS Code terminal. Then, establish the connection again.

ssh -tt -o StrictHostKeyChecking=no -i \${IdentityFile} \${User}@\${HostName} -p \${Port} rm -rf /home/ma-user/.vscode-server/bin/

#### Parameters:

- IdentityFile: Path to the local key
- User: Username, for example, ma-user
- HostName: IP address
- **Port**: Port number

### D NOTE

The preceding methods can also be used to resolve issues related to the VS Code server.

# 15.3.7.9 What Do I Do If Error Message "The VS Code Server failed to start" Is Displayed?

# Symptom



# Solution

**Step 1** Check whether the VS Code version is 1.65.0 or later. If so, check the Remote-SSH version. If the version is earlier than 0.76.1, upgrade Remote-SSH.

	Eile <u>E</u> dit	Selection View Go	<u>R</u> un <u>T</u> erminal <u>H</u> elp	Extension: Remote - SSH - V	'isual Studio Code		o ×	<
Ω		ONS: MARKETPLACE	⊽ບ≣…	🗋 Extension: Remote - SSH 🗙			⊳ ⊞	
þ	Remot				Remote - SSI	v0.80.0 Preview		
ဠၟၜ		Remote - SSH Open any folder on a Microsoft	ی ۱3ms remote machine using SS ن	>_	Microsoft 0 11,4 Open any folder on a rer	420,715   ★★★★★( mote machine using SSH	1	
å		Remote - Containers Open any folder or rep Microsoft	Ф 12.9M ★ 4.5 pository inside a Docker Install ∨		Disable Uninstall ✓ S This extension is enabled g	witch to Pre-Release Version ई lobally.	22	
		Remote - WSL Open any folder in the Microsoft	✿ 14.8M ★ 5 Windows Subsystem for Install	<u>Details</u> Feature C	ontributions Extension	Pack Runtime Status		
₩		Remote - SSH: Editin Edit SSH configuratior Oticrosoft	<b>g Configuration Files</b> n files 發	Visual Studi	o Code	Categories		
	3	Remote Developmen An extension pack tha Oticrosoft	nt	Coher Coher Coher Coher Coher Coher Resources Marketplace Repository License		Resources		
8	2	Azure Machine Learn This extension is used Microsoft	ning - Rem $\Phi$ 316K $\star$ 3.5 by the Azure Machine Le Install					
<u>کېځ</u>		Remote VSCode			and troubleshooting in a		ጽ ር	þ

Step 2 Open the command panel (Ctrl+Shift+P for Windows and Cmd+Shift+P for macOS), search for Kill VS Code Server on Host, and locate the affected instance, which will be automatically cleared. Then, establish the connection again.



Figure 15-18 Clearing the affected instance

----End

≺1

ſĽ

Ð

Q

مړ

 $\Rightarrow$ 

L0

٦O

# 15.3.7.10 What Do I Do If Error Message "Permissions for 'x:/xxx.pem' are too open" Is Displayed?

# Symptom

[15:39:18.228] Running script with connection command: ssh -T -D 5915 "ModelArts-notebook-2fd7	" bash
<pre>[15:39:18.231] Terminal shell path: C:\windows\System32\cmd.exe</pre>	
<pre>[15:39:18.460] &gt; 2 [0;C:\windows\System32\cmd.exe</pre>	
[15:39:18.460] Got some output, clearing connection timeout	
[15:39:18.601] > Warning: Permanently added '[dev-modelarts-cnnorth7.ulanqab.huawei.com]:30648	,[1
> 00.85.124.207]:30648' (RSA) to the list of known hosts.	
[15:39:18.730] > @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@	
[15:39:18.739] > @ WARNING: UNPROTECTED PRIVATE KEY FILE! @	
> 0000000000000000000000000000000000000	
> Permissions for 'D:/ , ' ', pem' are too open.	
> It is required that your private key files are NOT accessible by others.	
> This private key will be ignored.	
> Load key "D:/provinger.pem": bad permissions	
x ma-user@dev-modelarts-cnnorth7.ulanqab.huawei.com: Permission denied (publickey)	

### **Possible Cause**

Possible cause 1: The key file is not stored in the specified path. For details, see the **security restrictions** or **VS Code document**. Resolve this issue by referring to solution 1.

Possible cause 2: For macOS or Linux, the permission on the key file or the folder where the key is stored may be incorrect. Resolve this issue by referring to solution 2.

### Solution

Solution 1:

Place the key file in a specified path or its sub-path:

Windows: C:\Users\{{user}}

macOS or Linux: Users/{{user}}

Solution 2:

#### Check the file and folder permissions.

Local SSH file and folder permissions

macOS / Linux:

On your local machine, make sure the following permissions are set:

Folder / File	Permissions
.ssh in your user folder	chmod 700 ~/.ssh
.ssh/config in your user folder	<pre>chmod 600 ~/.ssh/config</pre>
.ssh/id_rsa.pub in your user folder	chmod 600 ~/.ssh/id_rsa.pub
Any other key file	chmod 600 /path/to/key/file

#### Windows:

The specific expected permissions can vary depending on the exact SSH implementation you are using. We recommend using the out of box Windows 10 OpenSSH Client.

In this case, make sure that all of the files in the .ssh folder for your remote user on the SSH host is owned by you and no other user has permissions to access it. See the Windows OpenSSH wiki for details.

For all other clients, consult your client's documentation for what the implementation expects.

# 15.3.7.11 What Do I Do If Error Message "Bad owner or permissions on C:\Users\Administrator/.ssh/config" or "Connection permission denied (publickey)" Is Displayed?

#### Symptom

The following error message is displayed: "Bad owner or permissions on C:\Users \Administrator/.ssh/config" or "Connection permission denied (publickey). Please

make sure the key file is correctly selected and the file permission is correct. You can view the instance keypair information on ModelArts console."

### **Possible Causes**

The permission to the SSH folder has been granted to other users, not only to the current Windows user, or the current user does not have the permission. In these cases, you only need to modify the permission.

### Solution

1. Find the SSH folder, which is typically located in **C:\Users**, for example, **C:\Users\xxx**.

**NOTE** 

The file name in C:\Users must be the same as the Windows login username.

- 2. Right-click the folder and choose **Properties**. Then, click the **Security** tab.
- 3. Click **Advanced**. In the displayed window, click **Disable inheritance**. Then, in the **Block Inheritance** dialog box, click **Remove all inherited permissions from this object**. In this case, all users will be deleted.
- 4. Add an owner. In the same window, click Add. In the displayed window, click Select a principal next to Principal. In the displayed Select User, Computer, Service Account, or Group dialog box, click Advanced, enter the username, and click Find Now. Then, the search results will be displayed. Select your account and click OK to close all windows.

			Select User, Computer, Service Account,	Select User, Computer, Service Account, or Group		x
Name: Owner: Permissions	C:\Users\ywx1161482\Desktop\eddx Administrators (SZXG4YWX1161481\A Auditing Effective Access	2 Principal: Select a principal Type: Allow Applies to: This folder, subfolders	Select this object type: [User, Group, or Built'in security principal From this location: china huawei com Enter the object name to select ( <u>examples</u> ):	Select this object type: User, Group, or Built-in security principal From this location: chrina huavei.com Common Queries 4	<u>O</u> bjec	t Types ations
Permission ent	information, double-click a permission tries:	Basic permissions:	3	Name: Starts with V	Ξ	<u>C</u> olumns
No groups or	users have permission to access this obj	Full control Modify Read & execute List folder contents Read Write	Advanced	Descriptor: Stats with  Dissipled accounts Non expiring password Days since last logon:	_ 5	Stop
Add	Remove View	Special permissions	objects and/or containers within this conta	Search results:	OK	Cancel
Enable inhe	eritance	Add a condition to limit access. The	principal will be granted the specified per			
Replace all o	child object permission entries with inhe	Add a condition				

#### Figure 15-19 Adding an owner

5. Close and open VS Code again and try to remotely access the SSH host. Ensure that the target key is stored in the SSH folder.

# 15.3.7.12 What Do I Do If Error Message "ssh: connect to host xxx.pem port xxxxx: Connection refused" Is Displayed?

# Symptom

[16:42:24.876] [16:42:24.878] [16:42:25.094]	Running script with connection command: ssh -T -D 7616 "ModelArts-notebook-2fd7" Terminal shall path: C:\windows\System32\cmd.exe > cm20;C:\windows\System32\cmd.exe	bash
[16:42:25.094]	Got some output, clearing connection timeout	
[16:42:27.257]	> ssh: connect to host	
[16:42:27.278]		
[16:42:28.544]	"install" terminal command done	
[16:42:28.544]	Install terminal quit with output:	
[16:42:28.544]	Received install output	
[16:42:28.544]	Failed to parse remote port from server output	

# **Possible Cause**

The target instance is not running.

### Solution

Log in to the ModelArts management console and check the status of the instance. If the instance is stopped, start it. If the instance is in other states, such as **Error**, stop and then start it. After the instance status changes to **Running**, establish the remote connection again.

# 15.3.7.13 What Do I Do If Error Message "ssh: connect to host ModelArts-xxx port xxx: Connection timed out" Is Displayed?

Symptom

# **Possible Cause**

Possible cause 1: The whitelisted IP addresses configured for the instance are different from the ones used in the local network.

**Change the whitelist** so that the whitelisted IP addresses are the same as those used in the local network or disable the whitelist.

Possible cause 2: The local network is inaccessible.

Solution: Check the local network and network restrictions.

# 15.3.7.14 What Do I Do If Error Message "Load key "C:/Users/xx/test1/ xxx.pem": invalid format" Is Displayed?

# Symptom

[17:20:18.402] Running script with conne	ction command: ssh -T -D 8578 "ModelArts-notebook-2fd7" bas
[17:20:18.404] Terminal shell path: C:\w	vindows\System32\cmd.exe
<pre>[17:20:18.630] &gt; 20;C:\windows\System3;</pre>	2\cmd.exe
<pre>[17:20:18.630] Got some output, clearing</pre>	connection timeout
<pre>[17:20:18.630] Got some output, clearing [17:20:18.777] &gt; Warning: Permanently ad</pre>	; connection timeout ided '[dev-modelarts-cnnorth7.ulanqab.huawei.com]:30648,[1
<pre>[17:20:18.630] Got some output, clearing [17:20:18.777] &gt; Warning: Permanently ad &gt; 00.85.124.207]:30648' (RSA) to the lis</pre>	connection timeout ded '[dev-modelarts-cnnorth7.ulanqab.huawei.com]:30648,[1 it of known hosts.
<pre>[17:20:18.630] Got some output, clearing [17:20:18.777] &gt; Warning: Permanently ad &gt; 00.85.124.207]:30648' (RSA) to the lis [17:20:18.904] &gt; Load key "C:/Users/d</pre>	<pre>connection timeout ided '[dev-modelarts-cnnorth7.ulanqab.huawei.com]:30648,[1 it of known hosts. %/test1/; r.pem": invalid format</pre>

### **Possible Cause**

The content or format of the key file is incorrect.

### Solution

Use the correct key file for remote access. If there is no correct key file locally or the file is damaged, perform the following operations:

1. Log in to the console, search for **DEW**. On the DEW management console, choose **Key Pair Service** and click **Private Key Pairs**. Then, view and download the correct key file.

DEW Console	Kay Pair Service 🛞				
Key Management Service	A user's profile leg pers are only available to that user. Account leg pers are available to all users under the same account.				
Kay Pair Service	Private Key Pairs Account Key Pairs ECS List				
Cloud Secret Management Service	Upgrade Key Pair Craste Key Pair Inspirit Key Pair				
Dedicated HSM		Status	Private Keys	Opeartion	
Diect Storage Service a		Sonal		Delete Export Private Kay Clear	

2. If the key cannot be downloaded and the originally downloaded key was lost, create a new development environment instance and a new key file. Replacing a key file in a running development environment will be supported later.

# 15.3.7.15 What Do I Do If Error Message "An SSH installation couldn't be found" or "Could not establish connection to instance xxx: 'ssh' ..." Is Displayed?

# Symptom



When VS Code attempts to access a notebook instance, the system always prompts you to select a certificate, and the message, excepting the title, consists of garbled characters. After the certificate is selected, the system still does not respond and the connection failed.

# **Possible Cause**

OpenSSH is not installed in the current environment or is not installed in the default path. For details, see the **VS Code document**.

# Solution

• If OpenSSH is not installed in the current environment, **download and install** it.

#### Installing a supported SSH client

os	Instructions
Windows 10 1803+ / Server 2016/2019 1803+	Install the Windows OpenSSH Client.
Earlier Windows	Install Git for Windows.
macOS	Comes pre-installed.
Debian/Ubuntu	Run sudo apt-get install openssh-client
RHEL / Fedora / CentOS	Run sudo yum install openssh-clients

VS Code will look for the ssh command in the PATH. Failing that, on Windows it will attempt to find ssh.exe in the default Git for Windows install path. You can also specifically tell VS Code where to find the SSH client by adding the remote.SSH.path property to settings.json.

If OpenSSH fails to be installed, manually **download the OpenSSH installation package** and perform the following operations:

- Step 1 Download the .zip package and decompress it into C:\Windows\System32.
- Step 2 In C:\Windows\System32\OpenSSH-xx, open CMD as the administrator and run the following command: powershell.exe -ExecutionPolicy Bypass -File install-sshd.ps1
- **Step 3** Add **C:\Program Files\OpenSSH-xx** (in which the SSH executable .exe file is stored) to environment system variables.
- **Step 4** Open CMD again and run **ssh**. If the following information is displayed, the installation is successful. Otherwise, go to **5** and **6**.

C:\windows\system32>ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
[-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
[-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
[-i identity_file] [-J [user@]host[:port]] [-L address]
[-1 login_name] [-m mac_spec] [-0 ctl_cmd] [-o option] [-p port]
[-Q query_option] [-R address] [-S ct1_path] [-W host:port]
[-w local_tun[:remote_tun]] destination [command]

**Step 5** Enable port 22 (default OpenSSH port) on the firewall and run the following command in Command Prompt:

netsh advfirewall firewall add rule name=sshd dir=in action=allow protocol=TCP localport=22

**Step 6** Run the following command to start OpenSSH: Start-Service sshd

----End

 If OpenSSH is not installed in the default path, open the command panel (Ctrl+Shift+P for Windows and Cmd+Shift+P for macOS).

Search for **Open settings**.



Add **remote.SSH.path** to **settings.json**, for example, **"remote.SSH.path":** "*Installation path of the local OpenSSH*".



15.3.7.16 What Do I Do If Error Message "no such identity: C:/Users/xx / test.pem: No such file or directory" Is Displayed?

# Symptom

PROBLEMS OUTPUT TERMINAL PORTS	
[17:55:48.396] Running script with connection command: "C:\Windows\System32\OpenSSH\ssh.exe" -T -D 63262 "ModelArts-notebook- " b	ash
[17:55:48.397] Terminal shell path: C:\Windows\System32\cmd.exe	
[17:55:48.670] > ssc]0;C:\Windows\System32\cmd.exest	
[17:55:48.671] Got some output, clearing connection timeout	
[17:55:48.821] > Warning: Permanently added '[authoring-ssh-modelarts- uuawei.com	
> ]:31397,[ (RSA) to the list of known hosts.	
[17:55:48.956] > no such identity: c:\\Users\\ \Downloads\\test.pem: No such file or dir	
> ectory	
[17:55:48.985] > ma-user@authoring-ssh-modelarts- huawei.com: Permission denied (	
> publickey).	

### **Possible Cause**

The key file is not in the path, or the name of the key file in the path has been changed.

### Solution

Select the key path again.

# 15.3.7.17 What Do I Do If Error Message "Host key verification failed" or "Port forwarding is disabled" Is Displayed?

# Symptom

∢	File Edit Selection View Go Terminal Help Getting Started - Visual Studio C — 🛛 🗙
பு	✓ Getting Started ×
$\cap$	
7	problems output terminal ports Remote - SSH v 🗮 🔓 ^ 🗠 x
ço	<pre>[17:30:29.025] &gt; @ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @ &gt; @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@</pre>
å	> Someone could <sup>Visual Studio Code</sup>
Ē	> The fingerprine Close Remote More Actions Retry nost is > SHA256:eBkiPY1M7zfEoX/+x/JagkK4V2N1GyiesP0TvmiyOCM. > Please contact your system administrator.
₿	<pre>&gt; Add correct host key in</pre>
	<pre>&gt; Offending RSA key in /.ssh/known_hosts:1 &gt; RSA host key for [10.155.101.174]:20615 has changed and you have requested stric</pre>
8	<pre>&gt; t checking. &gt; Host key verification failed.</pre>

Or



#### **Possible Cause**

After the notebook instance is restarted, its public key changes. The alarm is generated when OpenSSH detected the key change.

#### Solution

 Add -o StrictHostKeyChecking=no for remote access through the CLI in VS Code.

ssh -tt -o StrictHostKeyChecking=no -i \${IdentityFile} \${User}@\${HostName} -p \${Port} Parameters:

- IdentityFile: Path to the local key

- User: Username, for example, ma-user
- HostName: IP address
- Port: Port number
- Add StrictHostKeyChecking no and UserKnownHostsFile=/dev/null to the local ssh config file for manual configuration of remote access in VS Code.
  - Host xxx HostName x.x.x.x # IP address Port 22522 User ma-user IdentityFile C:/Users/my.pem StrictHostKeyChecking no UserKnownHostsFile=/dev/null ForwardAgent yes

Note that SSH logins will be insecure after the preceding parameters are added because the **known\_hosts** file will be ignored during the logins.

# 15.3.7.18 What Do I Do If Error Message "Failed to install the VS Code Server" or "tar: Error is not recoverable: exiting now" Is Displayed?

# Symptom



# **Possible Cause**

The disk space of /home/ma-user/work is insufficient.

# Solution

Delete unnecessary files in /home/ma-user/work.

# 15.3.7.19 What Do I Do If Error Message "XHR failed" Is Displayed When a Remote Notebook Instance Is Accessed Through VS Code?

# **Possible Cause**

The network of the environment may be faulty.

# Solution

Rectify the fault by referring to Troubleshooting Failed XHR.

# 15.3.7.20 What Do I Do for an Automatically Disconnected VS Code Connection If No Operation Is Performed for a Long Time?

# Symptom

After an SSH connection is set up through VS Code, no operation is performed for a long time and the window retains open. When the connection is used again, it is found that the connection is disconnected and no error message is displayed. The following figure shows the reconnection information.



According to VS Code Remote-SSH logs, the connection was disconnected about two hours after the setup.



# **Possible Cause**

After SSH interaction stops for a period of time, the firewall disconnects idle connections (http://bluebiu.com/blog/linux-ssh-session-alive.html). The default SSH configuration does not lead to a proactive disconnection upon timeout. Since the instance runs stably on the backend, set up the connection again to resolve this issue.

# Solution

To retain connections if no operation is performed for a long time, configure periodic message sending through SSH. In this way, the connection will not become idle on the firewall.

• Configure the client as needed. If the client is not configured, no heartbeat packet will be sent to the server by default.









The configuration is as follows:

Host ModelArts-xx

ServerAliveInterval 3600 # Add this configuration in the unit of second, indicating that the client will actively send a heartbeat packet to the server every hour.

ServerAliveCountMax 3 # Add this configuration, indicating that if the server does not respond after the heartbeat packet is sent for three times, the connection will be disconnected.

For example, if the firewall is configured to disconnect a connection if the connection is idle for two hours, set **ServerAliveInterval** to a value less than two hours (for example, one hour) on the client to prevent the firewall from disconnecting the connection.

• Configure the server in **/home/ma-user/.ssh/etc/sshd\_config**. (Notebook has been configured, and 24 hours is longer than the time configured on the firewall for disconnecting connections. This configuration does not need to be manually modified. It is only used to help understand the SSH configuration.)

/modelarts/authoring(MindSpore) [ma-user work]\$cat /home/ma-user/.ssh/etc/sshd\_config |grep Client ClientAliveInterval 1440m ClientAliveCountMax 3

The preceding configuration shows that the server actively sends a heartbeat packet to the client every 24 hours, and the connection will be disconnected if the client does not respond after the heartbeat packet is sent for three times.

For details, see https://unix.stackexchange.com/questions/3026/what-dooptions-serveraliveinterval-and-clientaliveinterval-in-sshd-config-d.  If a connection must be consistently retained, it is a good practice to write logs in a separate log file and run the script on the backend. For example: nohup train.sh > output.log 2>&1 & tail -f output.log

# 15.3.7.21 What Do I Do If It Takes a Long Time to Set Up a Remote Connection After VS Code Is Automatically Upgraded?

# Symptom



### **Possible Cause**

VS Code is automatically upgraded. As a result, download the new VS Code server to set up a new connection.

### Solution

Disable automatic VS Code upgrade. To do so, click **Settings** in the lower left corner, search for **Update: Mode**, and set it to **none**.



#### Figure 15-22 Settings



Figure 15-23 Setting the update mode to none

# 15.3.7.22 What Do I Do If Error Message "Connection reset" Is Displayed During an SSH Connection?

### Symptom

C:\Users\.................ssh>ssh -tt -o StrictHostKeyChecking=no -i KeyPair-.........pem ma-user@dev-modelarts-cneast3.huaweicloud. om -p 30 kex\_exchange\_identification: read: Connection reset

### **Possible Causes**

The user network is restricted. For example, SSH is disabled by default on some enterprise networks.

#### Solution

Apply for the SSH permission.

15.3.7.23 What Can I Do If a Notebook Instance Is Frequently Disconnected or Stuck After I Use MobaXterm to Connect to the Notebook Instance in SSH Mode?

#### Symptom

After MobaXterm is connected to a development environment, it is disconnected after a period of time.

### **Possible Cause**

When MobaXterm is configured, **SSH keepalive** is not selected or **Stop server after** of MobaXterm Professional is set to a value that is too small.

# Solution

**Step 1** Open MobaXterm and click **Settings** on the menu bar.

Figure	15-24	Settings
--------	-------	----------



**Step 2** On the MobaXterm configuration page, click the **SSH** tab and select **SSH keepalive**.

Figure 15-25 Selecting SSH keepalive

-	General	🚺 Termina	il 🔀 X11	🔦 SSH	👤 Display	🔑 Toolbar	🔯 Misc	
	SSH-browser settings       Image: Constraint of the setting se							
	SSH settin	gs			٥	SSH engir	ne: <new></new>	~
2	SSH 🛛	keepalive	🖂 Display	SSH banne	r 🖂 Valida	te host identity	/ at first cor	nnection
	GSSAPI Kerberos Domain: GSSAPI library: <a href="https://www.saparton.org">SSAPI library:</a>							
	Defaults for commandline SSH: 🖸 Compression 🗹 X11-Forwarding 🖓 Fix connection issues							

**NOTE** 

If MobaXterm Professional is used, go to step 3.

Step 3 Change the default value 360 seconds to 3600 seconds or a larger value for Stop server after.



Figure 15-26 Setting Stop server after

----End

# 15.3.8 Others

# 15.3.8.1 How Do I Use Multiple Ascend Cards for Debugging in a Notebook Instance?

An Ascend multi-card training job runs in multi-process, multi-card mode. The number of cards is equal to the number of Python processes. The Ascend underlayer reads the environment variable **RANK\_TABLE\_FILE**, which has been configured in the development environment, without requiring manual configuration. For example, to run a job on eight cards, the code is as follows:

```
export RANK_SIZE=8
current_exec_path=$(pwd)
echo 'start training'
for((i=0;i<=$RANK_SIZE-1;i++));
do
echo 'start rank '$i
mkdir ${current_exec_path}/device$i
cd ${current_exec_path}/device$i
echo $i
export RANK_ID=$i
dev=`expr $i + 0`
echo $dev
export DEVICE_ID=$dev
python train.py > train.log 2>&1 &
done
```

Set the environment variable **DEVICE\_ID** in **train.py**.

devid = int(os.getenv('DEVICE\_ID'))
context.set\_context(mode=context.GRAPH\_MODE, device\_target="Ascend", device\_id=devid)

# **15.3.8.2 Why Is the Training Speed Similar When Different Notebook Flavors Are Used?**

If your training job is single-process in code, the training speed is basically the same no matter when the notebook flavor of 8 vCPUs and 64 GB of memory or the flavor of 72 vCPUs and 512 GB of memory is used. For example, if your training job uses 2 vCPUs and 4 GB of memory, the training speed is similar no matter when you use the notebook flavor of 4 vCPUs and 8 GB of memory or the flavor of 8 vCPUs and 64 GB of memory.

If your training job is multi-process in code, the training speed backed by the notebook flavor of 72 vCPUs and 512 GB of memory is higher than that backed by the notebook flavor of 8 vCPUs and 64 GB of memory.

# 15.3.8.3 How Do I Perform Incremental Training When Using MoXing?

If you are not satisfied with training results when using MoXing to build a model, you can perform incremental training after modifying some data and label information.

#### Adding Incremental Training Parameters to mox.run

After modifying labeling data or datasets, you can modify the **log\_dir** parameter in and add the **checkpoint\_path** parameter to **mox.run**. Set **log\_dir** to a new directory and **checkpoint\_path** to the output path of the previous training results. If the output path is an OBS directory, set the path to a value starting with **obs://**.

If labels are changed for label data, perform operations in **If Labels Are Changed** before running **mox.run**.

```
mox.run(input_fn=input_fn,
    model_fn=model_fn,
    optimizer_fn=optimizer_fn,
    run_mode=flags.run_mode,
    inter_mode=mox.ModeKeys.EVAL if use_eval_data else None,
    log_dir=log_dir,
    batch_size=batch_size_per_device,
    auto_batch=False,
    max_number_of_steps=max_number_of_steps,
    log_every_n_steps=flags.log_every_n_steps,
    save_summary_steps=save_summary_steps,
    save_nodel_secs=save_model_secs,
    checkpoint_path=flags.checkpoint_url,
    export_model=mox.ExportKeys.TF_SERVING)
```

# If Labels Are Changed

If the labels in a dataset have changed, execute the following statement. The statement must be executed before running **mox.run**.

In the statement, the **logits** variable indicates classification layer weights in different networks, and different parameters are configured. Set this parameter to the corresponding keyword.

mox.set\_flag('checkpoint\_exclude\_patterns', 'logits')

If the built-in network of MoXing is used, the corresponding keyword needs to be obtained by calling the following API. In this example, the **Resnet\_v1\_50** keyword is the value of **logits**.

import moxing.tensorflow as mox

model\_meta = mox.get\_model\_meta(mox.NetworkKeys.RESNET\_V1\_50)
logits\_pattern = model\_meta.default\_logits\_pattern
print(logits\_pattern)

You can also obtain a list of networks supported by MoXing by calling the following API:

import moxing.tensorflow as mox
print(help(mox.NetworkKeys))

The following information is displayed:

Help on class NetworkKeys in module moxing.tensorflow.nets.nets\_factory:

class NetworkKeys(builtins.object) | Data descriptors defined here:

\_\_dict\_\_ dictionary for instance variables (if defined)

\_\_weakref\_\_ list of weak references to the object (if defined)

Data and other attributes defined here:

ALEXNET\_V2 = 'alexnet\_v2'

CIFARNET = 'cifarnet'

INCEPTION\_RESNET\_V2 = 'inception\_resnet\_v2'

INCEPTION\_V1 = 'inception\_v1'

INCEPTION\_V2 = 'inception\_v2'

INCEPTION\_V3 = 'inception\_v3'

INCEPTION\_V4 = 'inception\_v4'

LENET = 'lenet'

MOBILENET\_V1 = 'mobilenet\_v1'

MOBILENET\_V1\_025 = 'mobilenet\_v1\_025'

MOBILENET\_V1\_050 = 'mobilenet\_v1\_050'

MOBILENET\_V1\_075 = 'mobilenet\_v1\_075'

MOBILENET\_V2 = 'mobilenet\_v2'

MOBILENET\_V2\_035 = 'mobilenet\_v2\_035'

MOBILENET\_V2\_140 = 'mobilenet\_v2\_140'

NASNET\_CIFAR = 'nasnet\_cifar'

NASNET\_LARGE = 'nasnet\_large'

NASNET\_MOBILE = 'nasnet\_mobile' OVERFEAT = 'overfeat' PNASNET\_LARGE = 'pnasnet\_large' PNASNET\_MOBILE = 'pnasnet\_mobile' PVANET = 'pvanet' RESNET\_V1\_101 = 'resnet\_v1\_101' RESNET\_V1\_110 = 'resnet\_v1\_110' RESNET\_V1\_152 = 'resnet\_v1\_152' RESNET\_V1\_18 = 'resnet\_v1\_18' RESNET\_V1\_20 = 'resnet\_v1\_20' RESNET\_V1\_200 = 'resnet\_v1\_200' RESNET\_V1\_50 = 'resnet\_v1\_50' RESNET\_V1\_50\_8K = 'resnet\_v1\_50\_8k' RESNET\_V1\_50\_MOX = 'resnet\_v1\_50\_mox' RESNET\_V1\_50\_OCT = 'resnet\_v1\_50\_oct' RESNET\_V2\_101 = 'resnet\_v2\_101' RESNET\_V2\_152 = 'resnet\_v2\_152' RESNET\_V2\_200 = 'resnet\_v2\_200' RESNET\_V2\_50 = 'resnet\_v2\_50' RESNEXT\_B\_101 = 'resnext\_b\_101' RESNEXT\_B\_50 = 'resnext\_b\_50' RESNEXT\_C\_101 = 'resnext\_c\_101' RESNEXT\_C\_50 = 'resnext\_c\_50' VGG\_16 = 'vgg\_16' VGG\_16\_BN = 'vgg\_16\_bn' VGG\_19 = 'vgg\_19' VGG\_19\_BN = 'vgg\_19\_bn'  $VGG_A = 'vgg_a'$ VGG\_A\_BN = 'vgg\_a\_bn' XCEPTION\_41 = 'xception\_41' XCEPTION\_65 = 'xception\_65' XCEPTION\_71 = 'xception\_71'

# 15.3.8.4 How Do I View GPU Usage on the Notebook?

If you select GPU when creating a notebook instance, perform the following operations to view GPU usage:

- 1. Log in to the ModelArts management console, and choose **DevEnviron** > **Notebooks**.
- 2. In the **Operation** column of the target notebook instance in the notebook list, click **Open** to go to the **Jupyter** page.
- 3. On the **Files** tab page of the **Jupyter** page, click **New** and select **Terminal**. The **Terminal** page is displayed.
- 4. Run the following command to view GPU usage: nvidia-smi
- Check which processes in the current notebook instance use GPUs. Method 1:

python /modelarts/tools/gpu\_processes.py

The following figure shows the case that the current process is using GPUs.

<pre>V V V V V V V V V V V V V V V V V V V</pre>			
Processes:	PID	Process name	GPU Memory Usage
+======================================	4608	python	785 MiB
0	4731	python	785 MiB
0	4860	python	785 MiB
0	5000	python	785 MiB
(D. Tarah 4 4) [	1		•+

The following figure shows the case that the current process is not using GPUs.



Method 2:

Open **/resource\_info/gpu\_usage.json** and view the processes that are using GPUs.

If no process is using GPUs, the file may be unavailable or empty.

# 15.3.8.5 How Can I Obtain GPU Usage Through Code?

Run the shell or python command to obtain the GPU usage.

# Using the shell Command

1. Run the **nvidia-smi** command.

This operation relies on CUDA NVCC.

watch -n 1 nvidia-smi

Every 1.0s: nvidia-smi

Mon Oct 25 15:20:11 2021

NVIDIA-	SMI 440.	33.01 Driver	Version: 440.33.01	CUDA Versio	n: 10.2
GPU Na	ame emp Perf	Persistence-M Pwr:Usage/Cap	Bus-Id Disp.A Memory-Usage	Volatile   GPU-Util	Uncorr. ECC Compute M.
0 Te   N/A 3	esla V100 31C P0	-SXM2 On 43W / 300W	00000000:5F:00.0 Off   0MiB / 32510MiB	0%	0 Default
1 Te   N/A 3	esla V100 34C P0	-SXM2 On 44W / 300W	00000000:B5:00.0 Off   0MiB / 32510MiB	   0%	0 Default
+					
Processes:     GPU Memory       GPU     PID       Type     Process name       Usage					
No running processes found					

#### 2. Run the **gpustat** command.

pip install gpustat gpustat -cp -i

notebook-6a654129-698e-4635-b6be-67aedbdd4c54 Mon Oct 25 15:19:11 2021 440.33.01
[0] Tesla V100-SXM2-32GB | 31'C, 0 % | 0 / 32510 MB |
[1] Tesla V100-SXM2-32GB | 34'C, 0 % | 0 / 32510 MB |

To stop the command execution, press **Ctrl+C**.

### Using the python Command

```
1. Run the nvidia-ml-py3 command (commonly used).
!pip install nvidia-ml-py3
import nvidia_smi
nvidia_smi
nvidia_smi.nvmllnit()
deviceCount = nvidia_smi.nvmlDeviceGetCount()
for i in range(deviceCount):
    handle = nvidia_smi.nvmlDeviceGetHandleByIndex(i)
    util = nvidia_smi.nvmlDeviceGetUtilizationRates(handle)
    mem = nvidia_smi.nvmlDeviceGetMemoryInfo(handle)
    print(f"|Device {i}| Mem Free: {mem.free/1024**2:5.2f}MB / {mem.total/1024**2:5.2f}MB | gpu-util:
    {util.gpu:3.1%} | gpu-mem: {util.memory:3.1%} |")
    Output:
    [Device 0| Mem Free: 32510.44MB / 32510.50MB | gpu-util: 0.0% | gpu-mem: 0.0% |
    [Device 1| Mem Free: 32510.44MB / 32510.50MB | gpu-util: 0.0% | gpu-mem: 0.0% |
```

#### 2. Run the nvidia\_smi, wapper, and prettytable commands.

Use the decorator to obtain the GPU usage in real time during model training.

```
def gputil_decorator(func):
  def wrapper(*args, **kwargs):
     import nvidia_smi
     import prettytable as pt
     try:
        table = pt.PrettyTable(['Devices','Mem Free','GPU-util','GPU-mem'])
        nvidia_smi.nvmlInit()
        deviceCount = nvidia_smi.nvmlDeviceGetCount()
        for i in range(deviceCount):
          handle = nvidia_smi.nvmlDeviceGetHandleByIndex(i)
          res = nvidia_smi.nvmlDeviceGetUtilizationRates(handle)
          mem = nvidia_smi.nvmlDeviceGetMemoryInfo(handle)
          table.add_row([i, f"{mem.free/1024**2:5.2f}MB/{mem.total/1024**2:5.2f}MB",
f"{res.gpu:3.1%}", f"{res.memory:3.1%}"])
     except nvidia_smi.NVMLError as error:
        print(error)
     print(table)
     return func(*args, **kwargs)
  return wrapper
Output:
```

Devices	Mem Free	   GPU-util	 GPU-mem	+
0   1	32510.44MB/32510.50MB 32510.44MB/32510.50MB	0.0%	0.0%	I I

3. Run the **pynvml** command.

Run **nvidia-ml-py3** to directly obtain the nvml c-lib library, without using **nvidia-smi**. Therefore, this command is recommended.

```
from pynvml import *
nvmllnit()
handle = nvmlDeviceGetHandleByIndex(0)
info = nvmlDeviceGetMemoryInfo(handle)
print("Total memory:", info.total)
print("Free memory:", info.free)
print("Used memory:", info.used)
```

```
Output:
Total memory: 34089730048
Free memory: 34089664512
Used memory: 65536
```

### 4. Run the **gputil** command.

!pip install gputil import GPUtil as GPU GPU.showUtilization()

Output:

import GPUtil as GPU
GPUs = GPU.getGPUs()
for gpu in GPUs:
 print("GPU RAM Free: {0:.0f}MB | Used: {1:.0f}MB | Util {2:3.0f}% | Total
{3:.0f}MB".format(gpu.memoryFree, gpu.memoryUsed, gpu.memoryUtil\*100, gpu.memoryTotal))
Output:

GPU RAM Free: 32510MB | Used: 0MB | Util 0% | Total 32510MB GPU RAM Free: 32510MB | Used: 0MB | Util 0% | Total 32510MB

When using a deep learning framework such as PyTorch or TensorFlow, you can also use the APIs provided by the framework for query.

# 15.3.8.6 Which Real-Time Performance Indicators of an Ascend Chip Can I View?

The real-time performance indicator that can be viewed is **npu-smi**, which is similar to **nvidia-smi** of a GPU chip.

# 15.3.8.7 What Are the Relationships Between Files Stored in JupyterLab, Terminal, and OBS?

- Files stored in JupyterLab are the same as those in the work directory on the **Terminal** page. That is, the files are created on your notebook instances or synchronized from OBS.
- Notebook instances with OBS storage mounted can synchronize files from OBS to JupyterLab using the JupyterLab upload and download functions. The files on the **Terminal** page are the same as those in JupyterLab.
- Notebook instances with EVS storage mounted can read files from OBS to JupyterLab using the MoXing API or SDKs. The files on the **Terminal** page are the same as those in JupyterLab.

# 15.3.8.8 How Do I Migrate Data from an Old-Version Notebook Instance to a New-Version One?

The old-version notebook has been discontinued. This section describes how to migrate data from a notebook instance of the old version to a notebook instance of the new version.

#### 15 FAQs

# Storage Differences Between the Old and New Versions

Storage	Old- Version Notebook	New- Version Notebook	Description
OBS	Supported	Not supported	OBS is a storage system, not a file system. In old-version notebook, remote replication and local replication of OBS data may be confused, leading to issues in controlling operations on data. Therefore, OBS mounting is removed from notebook of the new version. You can flexibly obtain and operate OBS data using code.
OBS parallel file system	Not supported	Supported	The new-version notebook allows dynamic mounting of OBS parallel file systems. You can mount storage on the details page of a running notebook instance. Data migration from the old version to the new version is not involved.
EVS	Supported	Supported	EVS disks can be attached to notebook instances of both the old and new versions. Data stored in the old version needs to be migrated to the new version.
SFS	Not supported	Supported	SFS is used in dedicated resource pools. This function has been discontinued in notebook of the old version. Therefore, data migration is not involved.
EFS	Not supported	Supported	EFS is used in notebook of the new version only.

 Table 15-7 Storage supported by notebook of the old and new versions

# **OBS Used in Notebook of the Old Version**

When notebook instances of the old version use OBS for storage, data is stored in OBS and does not need to be migrated. After a new-version notebook instance is created, directly use the data in the OBS directory. For details, see **How Do I Read and Write OBS Files in a Notebook Instance?** 

Notebooks C New Y	Version New	
Note: The notebook instances in the "	Running " status are billed. If you no	longer require a notebook instance, :
Create The maximum nur	nber of Notebooks: 0; Increase	e quota.
Name ↓Ξ	Status	Work Environment
✓ notebook-7508	<ul><li>Stopped</li></ul>	Multi-Engine 1.0 (py
<ul> <li>notebook-zhangyan-obs</li> </ul>	Stopped	Multi-Engine 2.0 (py
ID		
Work Environment	Multi-Engine 2.0 (python3)	
Storage	OBS (obs://	

Figure 15-27 OBS used in notebook of the old version

# EVS Used in Notebook of the Old Version

If EVS disks are attached to a notebook instance of the old version for storing data, back up and migrate the EVS data to a notebook instance of the new version.

- If the volume of data stored in EVS is small, download the data to a local directory, create a notebook instance of the new version, and upload the data to the new notebook instance.
- If a large amount of data is stored in EVS, upload the data to an OBS bucket. After a notebook instance of the new version is created, read data from the the OBS bucket.

For more details, see Uploading and Downloading Data in Notebook.

Notebooks ← New Y	Version
Note: The notebook instances in the "	' 🧿 Running '' status are billed. If you no longer requi
Create The maximum nur	mber of Notebooks: 0; Increase quota.
Name J≡	Status
∧ notebook-7508	Stopped
ID	
Work Environment	Multi-Engine 1.0 (python3)-cpu   CPU
Storage	EVS (Mount path: /home/ma-user/work)

Figure 15-28 EVS storage used in notebook of the old version

# 15.3.8.9 How Do I Use the Datasets Created on ModelArts in a Notebook Instance?

Datasets created on ModelArts are stored in OBS. To use these datasets in a notebook instance, download them from OBS to the notebook instance.

For details, see How Do I Upload a File from a Notebook Instance to OBS or Download a File from OBS to a Notebook Instance?

# 15.3.8.10 pip and Common Commands

pip is a common Python package management tool. It allows you to search for, download, install, and uninstall Python packages.

Common pip commands:

```
pip --help # Obtain help information.
pip install SomePackage==XXXX # Install a specified version.
pip install SomePackage # Install the latest version.
pip uninstall SomePackage # Uninstall a software version.
```

For other commands, run the **pip** --**help** command.

# 15.3.8.11 What Are Sizes of the /cache Directories for Different Notebook Specifications in DevEnviron?

When creating a notebook instance, you can select resources based on the data volume.

ModelArts mounts disks to **/cache**. You can use this directory to store temporary files. The **/cache** directory shares resources with the code directory. The directory size varies depending on resource specifications.

No disks can be mounted to **/cache** for CPUs. When only one GPU or Ascend card is used, the **/cache** directory size is limited to 500 GB. If multiple GPUs or Ascend cards are used, the **/cache** directory size is limited to 3 TB and calculated using the following formula: **/cache** directory size = Number of cards x 500 GB. For details, see Table 15-8.

Specification	/cache Directory Size
GPU, 0.25 cards	500 GB x 0.25
GPU, 0.5 cards	500 GB x 0.5
GPU, 1 card	500 GB
GPU, dual cards	500 GB x 2
GPU, four cards	500 GB x 4
GPU, eight cards	3 TB
Ascend, single card	500 GB
Ascend, dual cards	500 GB x 2
Ascend, four cards	500 GB x 4
Ascend, eight cards	3 TB
CPU	N/A

 Table 15-8 /cache directory sizes for different notebook specifications

# 15.3.8.12 What Is the Impact of Resource Overcommitment on Notebook Instances?

Notebook overcommitment refers to the sharing of GPUs and memory within a node. To fully utilize resources, they are overcommitted in dedicated pools.

Example: A dedicated pool has one CPU node with 8 vCPUs and 64 GB memory. If you create a notebook instance with 2 vCPUs and 8 GB memory, a maximum of 6.67 notebook instances (8 vCPUs/(2 vCPUs x 0.6)) can be started due to overcommitment with an overcommitment ratio of 0.6. In this case, at least 1.2 vCPUs are required for starting the notebook instance, and a maximum of 2 vCPUs are used for running the notebook instance. Similarly, at least 4.8 GB memory is required, and a maximum of 8 GB memory is used for running the notebook instance. Instances may be forcibly terminated due to overcommitment. For example, if six instances with 2 vCPUs are started on an 8 vCPUs node and the CPU usage of one instance exceeds the upper limit (8 vCPUs) of the node, Kubernetes forcibly terminates the instance that uses the most resources.

Do not overcommit resources as it may result in instance restart.

# 15.4 Training Jobs

# **15.4.1 Functional Consulting**

# 15.4.1.1 What Are the Solutions to Underfitting?

- 1. Increasing model complexity
  - For an algorithm, add more high-order items to the regression model, improve the depth of the decision tree, or increase the number of hidden layers and hidden units of the neural network to increase model complexity.
  - Discard the original algorithm and use a more complex algorithm or model. For example, use a neural network to replace the linear regression, and use the random forest to replace the decision tree.
- 2. Adding more features to make input data more expressive
  - Feature mining is critical. In particular, a small set of highly expressive features can often be more effective than a larger set of less expressive ones.
  - Feature quality is the focus.
  - To explore highly expressive features, you must have an in-depth understanding of data and application scenarios, which depends on experience.
- 3. Adjusting parameters and hyperparameters
  - Neural network: learning rate, learning attenuation rate, number of hidden layers, number of units in a hidden layer, β1 and β2 parameters in the Adam optimization algorithm, and batch\_size
  - Other algorithms: number of trees in the random forest, number of clusters in k-means, and regularization parameter λ
- 4. Adding training data, which is of little effect

Underfitting is usually caused by weak model learning capabilities. Adding data cannot significantly increase the training performance.

5. Reducing regularization constraints Regularization aims to prevent model overfitting. If a model is underfitting instead of overfitting, reduce the regularization parameter  $\lambda$  or directly remove the regularization item.

# 15.4.1.2 What Are the Precautions for Switching Training Jobs from the Old Version to the New Version?

The differences between the new version and the old version lie in:
- Differences in Training Job Creation
- Differences in Training Code Adaptation
- Differences in Built-in Training Engines

### **Differences in Training Job Creation**

- In earlier versions, you can create a training job using **Algorithm Management**, **Frequently-used**, and **Custom**.
- In the new version, you can create a training job using Custom algorithmor My algorithm.

This allows you to select algorithms by category.

- The saved algorithms in **Algorithm Management** in the old version are in **My** algorithm in the new version.
- The Frequently-used in the old version is the Custom algorithm in the new version. Select Preset image for Boot Mode when you create jobs using the new version.
- The **Custom** in the old version is the **Custom algorithm** in the new version. Select **Custom image** for **Boot Mode** when you create jobs using the new version.

### **Differences in Training Code Adaptation**

In the old version, you are required to configure data input and output as follows:

# Download data to your local container. In the code, **local\_data\_path** specifies the training input path. mox.file.copy\_parallel(args.data\_url, local\_data\_path)

# Upload the local container data to the OBS path. mox.file.copy\_parallel(local\_output\_path, args.train\_url)

In the new version, you only need to configure training input and output. In the code, **arg.data\_url** and **arg.train\_url** are used as local paths. For details, see Developing a Custom Script.

# Upload the local container data to the OBS path. #mox.file.copy\_parallel(local\_output\_path, args.train\_url)

### **Differences in Built-in Training Engines**

- In the new version, MoXing 2.0.0 or later is installed by default for built-in training engines.
- In the new version, Python 3.7 or later is used for built-in training engines.
- In the new image, the default home directory has been changed from /home/ work to /home/ma-user. Check whether the training code contains hard coding of /home/work.
- Built-in training engines are different between the old and new versions. Commonly used built-in training engines have been upgraded in the new version.

To use a training engine in the old version, switch to the old version. **Table 15-9** lists the differences between the built-in training engines in the old and new versions.

**Table 15-9** Differences between the built-in training engines in the old and new versions

Runtime Environment	Built-in Training Engine and Version	Old Versio n	New Version
Ascend-Powered-Engine	Mindspore-1.3.0	√	х
	Mindspore-1.7.0	х	$\checkmark$
	TensorFlow-1.15	√	$\checkmark$

### 15.4.1.3 How Do I Obtain a Trained ModelArts Model?

Models generated using ModelArts ExeML can be deployed only on ModelArts and cannot be downloaded to your local PC.

Models trained using a custom or subscription algorithm are stored in specified OBS paths for you to download.

### 15.4.1.4 What Is TensorBoard Used for in Model Visualization Jobs?

Visualization jobs are powered by TensorBoard. For details about TensorBoard functions, see the **TensorBoard official website**.

### 15.4.1.5 How Do I Obtain RANK\_TABLE\_FILE on ModelArts for Distributed Training?

ModelArts automatically provides the **RANK\_TABLE\_FILE** file for you. Obtain the file location through environment variables.

- Open the notebook terminal and run the following command to view RANK\_TABLE\_FILE: env | grep RANK
- In a training job, add the following code to the first line of the training startup script to print the value of **RANK\_TABLE\_FILE**:

os.system('env | grep RANK')

### 15.4.1.6 How Do I Obtain the CUDA and cuDNN Versions of a Custom Image?

Obtain a CUDA version:

cat /usr/local/cuda/version.txt

Obtain a cuDNN version:

cat /usr/local/cuda/include/cudnn.h | grep CUDNN\_MAJOR -A 2

### 15.4.1.7 How Do I Obtain a MoXing Installation File?

MoXing installation files cannot be downloaded or installed by users. The MoXing installation package is preset in ModelArts notebook and training job images, and can be directly used.

# 15.4.1.8 In a Multi-Node Training, the TensorFlow PS Node Functioning as a Server Will Be Continuously Suspended. How Does ModelArts Determine Whether the Training Is Complete? Which Node Is a Worker?

In a TensorFlow-powered distributed training, the PS task and worker task are started. The worker task is a key task. ModelArts will use a process exit code of the worker task to determine whether the training job is complete.

A task name will be used to determine which node is a worker. A Volcano job is issued for training, which contains a PS task and a worker task. The startup commands of the two tasks are different. The hyperparameter **task\_name** will be automatically generated, which is **ps** for the PS task and **worker** for the worker task.

### 15.4.1.9 How Do I Install MoXing for a Custom Image of a Training Job?

To prevent automatic installation of MoXing from affecting the package environment in the custom image, manually install MoXing for the custom image. The MoXing installation package is stored in the **/home/ma-user/modelarts/ package/** directory after the job is started. Before using MoXing, run the following code to install it:

import os os.system("pip install /home/ma-user/modelarts/package/moxing\_framework-\*.whl")

### **NOTE**

This case applies only to the training environment.

### 15.4.2 Reading Data During Training

### 15.4.2.1 How Do I Configure the Input and Output Data for Training Models on ModelArts?

ModelArts allows you to upload a custom algorithm for creating training jobs. Create the algorithm and upload it to an OBS bucket. For details about how to create an algorithm, see Creating an Algorithm. For details about how to create a training job, see Creating a Training Job.

### **Parsing Input and Output Paths**

When a ModelArts model reads data stored in OBS or outputs data to a specified OBS path, perform the following operations to configure the input and output data:

1. Parse the input and output paths in the training code. The following method is recommended:

After the parameters are parsed, use **data\_url** and **train\_url** to replace the paths to the data source and the data output, respectively.

- 2. When using a preset image to create an algorithm, set the defined input and output parameters based on the code parameters in **1**.
  - Training data is a must for algorithm development. You are advised to set the input parameter name to data\_url, indicating the input data source. You can also customize code parameters based on the algorithm code in 1.
  - After model training is complete, the trained model and the output information must be stored in an OBS path. By default, **Output** specifies the model output and the code path parameter is **train\_url**. You can also customize the output path parameters based on the algorithm code in 1.
- 3. When creating a training job, configure the input and output paths.

Select an OBS path or dataset path as the training input, and an OBS path for the output.

### **15.4.2.2 How Do I Improve Training Efficiency While Reducing Interaction** with OBS?

### **Scenario Description**

When you use ModelArts for custom deep learning training, training data is typically stored in OBS. If the volume of training data is large (for example, greater than 200 GB), a GPU resource pool is required, and the training efficiency is low.

To improve training efficiency while reducing interaction with OBS, perform the following operations for optimization.

### **Optimization Principles**

For the GPU resource pool provided by ModelArts, 500 GB NVMe SSDs are attached to each training node for free. The SSDs are attached to the **/cache** 

directory. The lifecycle of data in the **/cache** directory is the same as that of a training job. After the training job is complete, all content in the **/cache** directory is cleared to release space for the next training job. Therefore, you can copy data from OBS to the **/cache** directory during training so that data can be read from the **/cache** directory until the training is finished. After the training is complete, content in the **/cache** directory will be automatically cleared.

### **Optimization Methods**

TensorFlow code is used as an example.

The following is code before optimization:

```
tf.flags.DEFINE_string('data_url', ", 'dataset directory.')
FLAGS = tf.flags.FLAGS
mnist = input_data.read_data_sets(FLAGS.data_url, one_hot=True)
```

The following is an example of the optimized code. Data is copied to the **/cache** directory.

tf.flags.DEFINE\_string('data\_url', '', 'dataset directory.') FLAGS = tf.flags.FLAGS import moxing as mox TMP\_CACHE\_PATH = '/cache/data' mox.file.copy\_parallel('FLAGS.data\_url', TMP\_CACHE\_PATH) mnist = input\_data.read\_data\_sets(TMP\_CACHE\_PATH, one\_hot=True)

### 15.4.2.3 Why the Data Read Efficiency Is Low When a Large Number of Data Files Are Read During Training?

If a dataset contains a large number of data files (massive small files) and data is stored in OBS, files need to be repeatedly read from OBS during training. As a result, the training process is waiting for reading files, resulting in low read efficiency.

### Solution

- 1. Compress the massive small files into a package on your local PC, for example, a .zip package.
- 2. Upload the package to OBS.
- 3. During training, directly download this package from OBS to the **/cache** directory of your local PC. Perform this operation only once.

For example, you can use mox.file.copy\_parallel to download the .zip package to the **/cache** directory, decompress the package, and then read files for training.

```
tf.flags.DEFINE_string('<obs_file_path>/data.zip', '', 'dataset directory.')
FLAGS = tf.flags.FLAGS
import os
import moxing as mox
TMP_CACHE_PATH = '/cache/data'
mox.file.copy_parallel('FLAGS.data_url', TMP_CACHE_PATH)
zip_data_path = os.path.join(TMP_CACHE_PATH, '*.zip')
unzip_data_path = os.path.join(TEMP_CACHE_PATH, '*.zip')
unzip_data_path = os.path.join(TEMP_CACHE_PATH, 'unzip')
# You can also decompress .zip Python packages.
os.system('unzip '+ zip_data_path + ' -d ' + unzip_data_path)
mnist = input_data.read_data_sets(unzip_data_path, one_hot=True)
```

### 15.4.2.4 How Do I Define Path Variables When Using MoXing?

### Symptom

mox.file.copy\_parallel(src\_obs\_dir=input\_storage,'obs://dyyolov8/yolov5\_test/yolov5-7.0/datasets'),

How do I define an OBS path as a variable in the mox function?

### Solution

The following is an example of defining a variable:

input\_storage = './test.py'
import moxing as mox
mox.file.copy\_parallel(input\_storage,'obs://dyyolov8/yolov5\_test/yolov5-7.0/datasets')

### 15.4.3 Compiling the Training Code

### 15.4.3.1 How Do I Create a Training Job When a Dependency Package Is Referenced by the Model to Be Trained?

ModelArts allows you to install third-party dependency packages for model training. After the **pip-requirements.txt** file is stored in the training code directory, the system runs the command below to install the specified Python packages before the training boot file is executed.

pip install -r pip-requirements.txt

Only training jobs created using a preset image can reference dependency packages for model training.

### **NOTE**

Any one of the following file names can be used. This section uses **pip-requirements.txt** as an example.

- pip-requirement.txt
- pip-requirements.txt
- requirement.txt
- requirements.txt
- For details about the code directory, see Storing the Installation File in the Code Directory.
- For details about the specifications of pip-requirements.txt, see Installation File Specifications.

### Storing the Installation File in the Code Directory

- If you use **My algorithm** to create a training job, you can store related files in the configured **Code Directory** when creating an algorithm. The **Boot Mode** of the algorithm must be **Preset image**.
- If you use **Custom algorithm** to create a training job, you can store related files in the configured **Code Directory**. The **Boot Mode** must be **Preset image**.

Before creating a training job, upload related files to OBS. For details about the file packaging requirements, see **Installation File Specifications**.

### **Installation File Specifications**

The installation file varies depending on the dependency package type.

Open-source installation packages

**NOTE** 

Installation using the source code from GitHub is not supported.

Create a file named **pip-requirements.txt** in the code directory, and specify the name and version number of the dependency package in the file. The format is *[Package name]==[Version]*.

Take for example, an OBS path specified by **Code Dir** that contains model files and the **pip-requirements.txt** file. The code directory structure would be as follows:

```
|---OBS path to the model boot file
|---model.py #Model boot file
|---pip-requirements.txt #Defined configuration file, which specifies the name and version of the
dependency package
```

The following shows the content of the **pip-requirements.txt** file:

alembic==0.8.6 bleach==1.4.3 click==6.6

#### WHL packages

If the training background does not support the download of open source installation packages or use of user-compiled WHL packages, the system cannot automatically download and install the package. In this case, place the WHL package in the code directory, create a file named **pip-requirements.txt**, and specify the name of the WHL package in the file. The dependency package must be a **.whl** file.

Take for example, an OBS path specified by **Code Dir** that contains model files, the **.whl** file, and the **pip-requirements.txt** file. The code directory structure would be as follows:

---OBS path to the model boot file

|---model.py #Model boot file

|---XXX.whl #Dependency package. If multiple dependencies are required, place multiple dependency packages here.

|---pip-requirements.txt #Defined configuration file, which specifies the name of the dependency package

The following shows the content of the **pip-requirements.txt** file:

```
numpy-1.15.4-cp36-cp36m-manylinux1_x86_64.whl
tensorflow-1.8.0-cp36-cp36m-manylinux1_x86_64.whl
```

### 15.4.3.2 What Is the Common File Path for Training Jobs?

The path to the training environment and the code directory in the container are generally obtained using the environment variable **\${MA\_JOB\_DIR}**, which is **/ home/ma-user/modelarts/user-job-dir**.

### 15.4.3.3 How Do I Install a Library That C++ Depends on?

A third-party library may be used during job training. The following uses C++ as an example to describe how to install a third-party library.

- 1. Download source code to a local PC and upload it to OBS. .
- 2. Use MoXing to copy the source code uploaded to OBS to a notebook instance in the development environment.

The following is a code example for copying data to a notebook instance in a development environment running on an EVS: import moxing as mox mox.file.make\_dirs('/home/ma-user/work/data') mox.file.copy\_parallel('obs://bucket-name/data', '/home/ma-user/work/data')

- 3. On the **Files** tab page of the **Jupyter** page, click **New** and select **Terminal**. Run the following command to go to the target path, and check whether the source code has been downloaded, that is, whether the **data** file exists.
- 4. Compile code in **Terminal** based on service requirements.
- 5. Use MoXing to copy the compilation results to OBS. The following is a code example.

import moxing as mox mox.file.make\_dirs('/home/ma-user/work/data') mox.file.copy\_parallel('/home/ma-user/work/data', 'obs://bucket-name/file)

 During training, use MoXing to copy the compilation result from OBS to the container. The following is a code example. import moxing as mox mox.file.make\_dirs('/cache/data') mox.file.copy\_parallel('obs://bucket-name/data', '/cache/data')

### 15.4.3.4 How Do I Check Whether a Folder Copy Is Complete During Job Training?

In the script for training job boot file, run the following commands to obtain the sizes of the copied folders and the folders to be copied. Then determine whether folder copy is complete based on the command output.

import moxing as mox mox.file.get\_size('obs://bucket\_name/obs\_file',recursive=True)

**get\_size** indicates the size of the file or folder to be obtained. **recursive=True** indicates that the type is folder. **True** indicates that the type is folder, and **False** indicates that the type is file.

If the command output is consistent, the folder copy is complete. If the command output is inconsistent, the folder copy is not complete.

### 15.4.3.5 How Do I Load Some Well Trained Parameters During Job Training?

During job training, some parameters need to be loaded from a pre-trained model to initialize the current model. You can use the following methods to load the parameters:

- View all parameters by using the following code. from moxing.tensorflow.utils.hyper\_param\_flags import mox\_flags print(mox\_flags.get\_help())
- Specify the parameters to be restored during model loading. checkpoint\_include\_patterns is the parameter that needs to be restored, and checkpoint\_exclude\_patterns is the parameter that does not need to be restored.

checkpoint\_include\_patterns: Variables names patterns to include when restoring checkpoint. Such as: conv2d/weights.

checkpoint\_exclude\_patterns: Variables names patterns to include when restoring checkpoint. Such as: conv2d/weights.

Specify a list of parameters to be trained. trainable\_include\_patterns is a list of parameters that need to be trained, and trainable\_exclude\_patterns is a list of parameters that do not need to be trained.
 --trainable\_exclude\_patterns: Variables names patterns to exclude for trainable variables. Such as: conv1,conv2.
 --trainable\_include\_patterns: Variables names patterns to include for trainable variables. Such as: logits.

15.4.3.6 How Do I Obtain Training Job Parameters from the Boot File of the Training Job?

Training job parameters can be automatically generated in the background or you can enter them manually. To obtain training job parameters:

- When creating a training job, enter the name of Input (generally set to data\_url) and specify a data path to the training input, and enter the name of Output (generally set to train\_url) and specify a data path to the training output.
- 2. After the training job is executed, you can click the job name in the training job list to view its details. You can obtain the parameter input mode from logs, as shown in **Figure 15-29**.

### Figure 15-29 Viewing logs

```
[ModelArts Service Log]modelarts-pipe: will create log file /tmp/log/trainjob-4bac.log
* Restarting DNS forwarder and DHCP server dnsmasq
...done.
[Modelarts Service Log]user: uid=1101(work) gid=1101(work) groups=1101(work)
[Modelarts Service Log]pwd: /home/work
[Modelarts Service Log]ap_url: s3://donotdel-modelarts-test/AI/code/PyTorch/
[Modelarts Service Log]log_url: /tmp/log/trainjob-4bac.log
[Modelarts Service Log]log_url: /tmp/log/trainjob-4bac.log
[Modelarts Service Log]command: PyTorch/PyTorch.py
[Modelarts Service Log]command: PyTorch/PyTorch.py
[Modelarts Service Log]command: PyTorch/PyTorch.py
--data_url=s3://donotdel-modelarts-
test/AI/data/PyTorch/ --init_method=tcp://job1f00a54e-job-trainjob-4bac-0:6666
--test=test --
train_url=s3://donotdel-modelarts-test/out/
```

3. To obtain the values of **train\_url**, **data\_url**, and **test** during training, add the following code to the boot file of the training job:

import argparse
parser = argparse.ArgumentParser()
parser.add\_argument('--data\_url', type=str, default=None, help='test')
parser.add\_argument('--train\_url', type=str, default=None, help='test')
parser.add\_argument('--test', type=str, default=None, help='test')

### 15.4.3.7 Why Can't I Use os.system ('cd xxx') to Access the Corresponding Folder During Job Training?

If you cannot access the corresponding folder by using **os.system('cd xxx')** in the boot script of the training job, you are advised to use the following method:

import os os.chdir('/home/work/user-job-dir/xxx')

### 15.4.3.8 How Do I Invoke a Shell Script in a Training Job to Execute the .sh File?

ModelArts enables you to invoke a shell script, and you can use Python to invoke **.sh**. The procedure is as follows:

- 1. Upload the **.sh** script to an OBS bucket. For example, upload the **.sh** script to **/ bucket-name/code/test.sh**.
- Create the .py file on a local PC, for example, test.py. The background automatically downloads the code directory to the /home/work/user-job-dir/ directory of the container. Therefore, you can invoke the .sh file in the test.py boot file as follows: import os

os.system('bash /home/work/user-job-dir/code/test.sh')

- 3. Upload **test.py** to OBS. Then the file storage path is **/bucket-name/code/ test.py**.
- 4. When creating a training job, set the code directory to **/bucket-name/code**/, and the boot file directory to **/bucket-name/code/test.py**.

After the training job is created, you can use Python to invoke the **.sh** file.

### 15.4.3.9 How Do I Obtain the Dependency File Path to be Used in Training Code?

Since locally developed code must be uploaded to the ModelArts backend, you may set an invalid dependency file path. A recommended general solution to this problem is that you to use the OS API to obtain the absolute path of the dependency files.

The following shows an example of obtaining the path of dependency files in other folders using the OS API.

File directory structure:

project\_root #Root directory of code bootfile.py #Boot file otherfileDirectory #Directory of dependency files #Dependency files

Add the following code to the boot file to obtain the path (**otherfile\_path**) of dependency files:

```
import os
```

```
current_path = os.path.dirname(os.path.realpath(__file__)) # Obtain the path of the boot file bootfile.py.
project_root = os.path.dirname(current_path) # Obtain the root directory of the project using the path of
the boot file, which is the code directory set on ModelArts console.
```

otherfile\_path = os.path.join(project\_root, "otherfileDirectory", "otherfile.py") # Obtain the path of the dependency files using the root directory of the project.

### 15.4.3.10 What Is the File Path If a File in the model Directory Is Referenced in a Custom Python Package?

To obtain the actual path to a file in a container, use Python.

os.getcwd() # Obtain the current work directory (absolute path) of the file. os.path.realpath(\_\_ file \_\_) # Obtain the absolute path of the file.

You can also use other methods of obtaining a file path through the search engine and use the obtained path to read and write the file.

### 15.4.4 Creating a Training Job

### 15.4.4.1 What Can I Do If the Message "Object directory size/quantity exceeds the limit" Is Displayed When I Create a Training Job?

### **Issue Analysis**

The code directory for creating a training job has limits on the size and number of files.

### Solution

Delete the files except the code from the code directory or save the files in other directories. Ensure that the size of the code directory does not exceed 128 MB and the number of files does not exceed 4,096.

### 15.4.4.2 What Are Sizes of the /cache Directories for Different Resource Specifications in the Training Environment?

When creating a training job, you can select resources based on the size of the training job.

ModelArts mounts a disk to **/cache**. You can use this directory to store temporary files. The **/cache** directory shares resources with the code directory. The directory has different capacities for different resource specifications.

### **NOTE**

- The eviction policy of Kubernetes disks is 90%. Therefore, the effective size of a disk is 90% of the **cache** directory capacity.
- The local disks of BMSs are physical disks that have a fixed capacity. If you need to store a large amount of data, you can use SFS, which provides scalable storage.
- GPU resources

Table 15-10 Capacities of the cache directories for GPU resour	ces
----------------------------------------------------------------	-----

GPU Specifications	cache Directory Capacity
V100	800 GB
8*V100	3 TB
P100	800 GB

• CPU resources

#### Table 15-11 Capacities of the cache directories for CPU resources

CPU Specifications	cache Directory Capacity
2 vCPUs   8 GiB	50 GB
8 vCPUs   32 GiB	50 GB

### 15.4.4.3 Is the /cache Directory of a Training Job Secure?

The program of a ModelArts training job runs in a container. The address of a directory to which the container is mounted is unique, and can be accessed only by the running container. Therefore, the **/cache** directory of the training job is secure.

### 15.4.4.4 Why Is a Training Job Always Queuing?

If the training job is always queuing, the selected resources are limited in the resource pool, and the job needs to be queued. In this case, wait for resources. To speed up resource obtaining, do as follows:

1. If you use a public resource pool:

Resources in a public resource pool are limited. During peak hours, resources may be insufficient if service traffic is heavy. Try to take the following measures:

- If a free flavor was used, change it to a charged one. Few resources are provided for free flavors, leading to a high queuing probability.
- The less number of cards in the selected flavor leads to the lower queuing probability. For example, the probability of queuing when selecting a 1-card flavor is much less than that of queuing when selecting an 8-card flavor.
- Switch to another region.
- If resources will be used for a long term, purchase a dedicated resource pool.
- 2. If you use a dedicated resource pool:
  - If there are multiple available dedicated resource pools, switch to an idle one.
  - Release resources in the current resource pool, for example, stop notebook instances that are not used for a long time.
  - Submit a training job during off-peak hours.
  - Contact the account administrator of the resource pool to expand the resource pool based on the usage.

Helpful link: Why Is the Job Still Queued When Resources Are Sufficient?

### 15.4.4.5 What Determines the Hyperparameter Directory (/work or /mauser) When Creating a Training Job?

### Symptom

The hyperparameter directory for the input and output parameters varies between **/work** and **/ma-user** when creating a training job.

#### Figure 15-30 /ma-user directory

Input (?)	Enter a name			Dataset	Data path
	Obtained from	<ul> <li>Hyperparameters</li> </ul>	Environment variables		
	=/home <mark>/ma-user/</mark> nodel	arts/inputs/_0			

### Figure 15-31 /work directory

Input 🕐	▲ data_url	Dataset Data path
	Obtained from:Hyperparameters	-data_url=/home/work/modelarts/inputs/data_url_0

### Solution

The directory varies depending on the selected algorithm for the training job.

• If the selected algorithm is created using an old-version image, the hyperparameter directory of the input and output parameters is **/work**.

### Figure 15-32 Creating an algorithm

* Boot Mode	Preset image	Custom image		
	PyTorch	V PyTorch-1.0.0-pytho	n3.6 Old 🗸	Show Old Images
* Code Directory ⑦			Select	)

• If the selected algorithm is not created using an old-version image, the hyperparameter directory of the input and output parameters is **/ma-user**.

### **15.4.5 Managing Training Job Versions**

### 15.4.5.1 Does a Training Job Support Scheduled or Periodic Calling?

ModelArts training jobs do not support scheduled or periodic calling. When your job is in the **Running** state, you can call the job based on service requirements.

### **15.4.6 Viewing Job Details**

### 15.4.6.1 How Do I Check Resource Usage of a Training Job?

In the left navigation pane of the ModelArts management console, choose **Training Management > Training Jobs** to go to the **Training Jobs** page. In the training job list, click a job name to view job details. You can view the following metrics on the **Resource Usages** tab page.

- CPU: CPU usage (cpuUsage) percentage (Percent)
- **MEM**: Physical memory usage (memUsage) percentage (Percent)
- **GPU**: GPU usage (gpuUtil) percentage (Percent)
- **GPU\_MEM**: GPU memory usage (gpuMemUsage) percentage (Percent)

### 15.4.6.2 How Do I Access the Background of a Training Job?

ModelArts does not support access to the background of a training job.

### 15.4.6.3 Is There Any Conflict When Models of Two Training Jobs Are Saved in the Same Directory of a Container?

Storage directories of ModelArts training jobs do not affect each other. Environments are isolated from each other, and data of other jobs cannot be viewed.

### 15.4.6.4 Only Three Valid Digits Are Retained in a Training Output Log. Can the Value of loss Be Changed?

In a training job, only three valid digits are retained in a training output log. When the value of **loss** is too small, the value is displayed as **0.000**. Log content is as follows:

INFO:tensorflow:global\_step/sec: 0.382191 INFO:tensorflow:step: 81600(global step: 81600) sample/sec: 12.098 loss: 0.000 INFO:tensorflow:global\_step/sec: 0.382876 INFO:tensorflow:step: 81700(global step: 81700) sample/sec: 12.298 loss: 0.000

Currently, the value of **loss** cannot be changed. You can multiply the value of **loss** by 1000 to avoid this problem.

### 15.4.6.5 Can a Trained Model Be Downloaded or Migrated to Another Account? How Do I Obtain the Download Path?

You can download a model trained by a training job and upload the downloaded model to OBS in the region corresponding to the target account.

### **Obtaining a Model Download Path**

- Log in to the ModelArts console. In the left navigation pane, choose Training Management > Training Jobs. The Training Jobs page is displayed.
- 2. In the training job list, click a job name to view job details.
- 3. Obtain the **Output Path** on the left, which is the download path of the trained model.

### Migrating a Model to Another Account

Use either of the following methods to migrate a trained model to another account:

- Download the trained model and then upload it to the OBS bucket in the region corresponding to the target account.
- Configure a policy for the folder or bucket where the model is stored to authorize other accounts to perform read and write operations. For details, see "Creating a Custom Bucket Policy (Visual Editor)" in OBS documentation.

### **15.5 Service Deployment**

### 15.5.1 Model Management

### 15.5.1.1 Importing Models

### 15.5.1.1.1 How Do I Import the .h5 Model of Keras to ModelArts?

ModelArts does not support the import of models in .h5 format. You can convert the models in .h5 format of Keras to the TensorFlow format and then import the models to ModelArts.

For details about how to convert the Keras format to the TensorFlow format, see the **Keras official website**.

### 15.5.1.1.2 How Do I Edit the Installation Package Dependency Parameters in a Model Configuration File When Importing a Model?

### Symptom

When importing a model from OBS or a container image, edit a model configuration file. The model configuration file describes the model usage, computing framework, precision, inference code dependency package, and model API. The configuration file must be in JSON format. **dependencies** in the model configuration file specifies the dependencies required for configuring the model inference code. This parameter requires the package name, installation method, and version constraints. For details, see **Specifications for Editing a Model Configuration File** The following section describes how to edit **dependencies** in the model configuration file during model import.

### Solution

The installation packages must be installed in sequence. For example, before installing **mmcv-full**, install **Cython**, **pytest-runner**, and **pytest**. In the configuration file, **Cython**, **pytest-runner**, and **pytest** are ahead of **mmcv-full**.

#### Example:

```
"dependencies": [
  ł
  "installer": "pip",
  "packages": [
     {
        "package_name": "Cython"
     },
     {
        "package_name": "pytest-runner"
     },
     {
        "package_name": "pytest"
     },
{
        "restraint": "ATLEAST"
        "package_version": "5.0.0",
        "package_name": "Pillow"
     },
{
        "restraint": "ATLEAST",
        "package_version": "1.4.0",
        "package_name": "torch"
     },
{
        "restraint": "ATLEAST",
        "package_version": "1.19.1",
```

```
"package_name": "numpy"
   },
   {
      "package_name": "mmcv-full"
   }
 ]
}
```

If installing mmcv-full failed, the possible cause is that GCC was not installed in the base image, leading to a compilation failure. In this case, use the wheel package on premises to install mmcv-full.

Example:

]

1

```
"dependencies": [
  "installer": "pip",
  "packages": [
     {
       "package_name": "Cython"
     },
     {
       "package_name": "pytest-runner"
     },
{
        "package_name": "pytest"
     },
     {
       "restraint": "ATLEAST",
        "package_version": "5.0.0",
        "package_name": "Pillow"
     },
{
        "restraint": "ATLEAST",
        "package_version": "1.4.0",
        "package_name": "torch"
    },
{
       "restraint": "ATLEAST",
       "package_version": "1.19.1",
        "package_name": "numpy"
     },
     {
        "package_name": "mmcv_full-1.3.9-cp37-cp37m-manylinux1_x86_64.whl"
     }
   ]
  }
```

dependencies in the model configuration file supports multiple dependency structure arrays in list format.

Example: "dependencies": [ { "installer": "pip", "packages": [ { "package\_name": "Cython" }, { "package\_name": "pytest-runner" }, { "package\_name": "pytest" }, {

1

```
"package_name": "mmcv_full-1.3.9-cp37-cp37m-manylinux1_x86_64.whl"
   }
 ]
},
"installer": "pip",
"packages": [
   {
      "restraint": "ATLEAST",
      "package_version": "5.0.0",
      "package_name": "Pillow"
   },
   {
      "restraint": "ATLEAST",
      "package_version": "1.4.0",
"package_name": "torch"
   },
{
      "restraint": "ATLEAST"
      "package_version": "1.19.1",
      "package_name": "numpy"
   }.
 ]
}
```

### 15.5.1.1.3 What Do I Do If Error ModelArts.0107 Is Reported When I Use MindSpore to Create an AI Application?

In the supercomputing ecosystem, OM models are checked according to MindSpore model package specifications. Add an empty .om file to the model package so that the model package can be imported.

### 15.5.1.1.4 How Do I Change the Default Port to Create a Real-Time Service Using a Custom Image?

A port number (for example, 8443) has been specified in a model configuration file. If you do not specify a port (default port 8080 will be used then) or specify another port during AI application creation, deploying the AI application as a service will fail. In this case, set the port number to 8443 in the AI application to resolve this issue.

To change the default port, do as follows:

- 1. Log in to the ModelArts management console. In the navigation pane, choose AI Application Management > AI Applications.
- 2. Click **Create**. On the page for creating an AI application, set **Meta Model Source** to **Container image** and select a custom image.
- 3. Configure the container API and port number. Ensure that the port number is the same as that specified in the model configuration file.

### Figure 15-33 Changing the port

* Meta Model Source	Training job	OBS	Container image	Template	
	A model imported from a service deployment, Mod client or browser . Param	econtainer ir elArts uses ti eters	mage is of the image type. I he image to deploy inferenc	nsure the image e services. Learr	e can be properly started and provides inference APIs. During n more about image specifications . Upload an image through a
	* Container Imag	e Path			6
	Container API		HTTPS	▼ :// {host} :	8443
	Image Replicati	on W th be set	hen this function is disabled e source directory may affer e created quickly, but you ca rvice deployment.	d, AI applications at service deploy n modify or delo	s can be created quickly, <b>but modifying or deleting images in</b> ment. When this function is enabled, AI applications cannot te images in the source directory as that would not affect
	Health Check	0			

- 4. After the configuration, click **Create now**. Wait until the AI application runs properly.
- 5. Deploy the AI application as a real-time service again.

### 15.5.1.1.5 Does ModelArts Support Multi-Model Import?

Importing a model package from OBS to ModelArts applies to single-model scenarios. If multiple models are required, you are advised to import custom images from SWR to create AI applications and deploy services. For details about how to create a custom image, see **Creating a Custom Image and Using It to Create an AI Application**.

### 15.5.1.1.6 Restrictions on the Size of an Image for Importing an AI Application

ModelArts uses containers for deploying services. There are size limitations during container runtime. If the size of your model file, custom file, or system file exceeds the container engine space, a message will be displayed, indicating that the image space is insufficient.

The maximum container engine space in a public resource pool is 50 GB, and that for a dedicated resource pool is 50 GB by default. You can set the container engine space for a dedicated resource pool when you create it, which does not increase costs.

If the AI application is imported from OBS or a training job, the total size of the base image, model files, code, data files, and software packages cannot exceed the limit.

If the AI application is imported from a custom image, the total size of the decompressed image and image dependencies cannot exceed the limit.

### **15.5.2 Service Deployment**

### 15.5.2.1 Functional Consulting

### 15.5.2.1.1 What Types of Services Can Models Be Deployed as on ModelArts?

Models can be deployed as real-time services or batch services.

### 15.5.2.1.2 What Are the Differences Between Real-Time Services and Batch Services?

Real-Time Services

Models are deployed as web services. You can access the services through the management console or APIs.

Batch Services

A batch service performs inference on batch data and automatically stops after data processing is completed.

A batch service processes batch data at a time. A real-time service provides APIs for you to call.

### 15.5.2.1.3 What Is the Maximum Size of a Prediction Request Body?

After a service is deployed and running, you can send an inference request to the service. The requested content can be text, images, voice, or videos, depending on the model of the service.

If you use the inference request address (URL of APIG) displayed on the **Usage Guides** tab of the service details page for prediction, the maximum size of the request body is 12 MB. If the request body is oversized, the request will be intercepted.

If you perform the prediction on the **Prediction** tab of the service details page, the size of the request body cannot exceed 8 MB. The size limit varies between the two tab pages because they use different network links.

Ensure that the size of a request body does not exceed the upper limit. If there are high-concurrency and heavy-traffic inference requests, submit a service ticket to professional service support.

### 15.5.2.1.4 How Do I Select Compute Node Specifications for Deploying a Service?

Before deploying a service, specify node specifications. The node specifications displayed on the GUI are calculated by ModelArts based on the target AI application and the node specifications available in the resource pool. You can select the specifications provided by ModelArts or customize the specifications (supported only in dedicated resource pools).

Selecting compute node specifications based on the resources required by your AI application. For example, if an AI application requires 3 CPUs and 10 GB of memory, select compute node specifications higher than 3 CPUs and 10 GB of memory. This ensures that the service can be successfully deployed and run properly.

* Resource Pool	Public Resource Pool	Dedicated Resource Pool
* AI Application and Configuration	AI Application Source	My AI Applications My Subscriptions
	AI Application and Version	model-78ac(synchronous request) • 0.0.1(Normal) • C Traffic Ratio (%) ⑦ - 100 +
	Specifications	GPU: 1*NVIDIA-P4(8G8)   CPU: 8 vCP  Compute Nodes  Compute Nodes
	Environment Variable	[Free (restrictions apply)]CPU: 1 vCPUs 4 GPU: 1+NVIDIA-P4(8GB)   CPU: 8 vCPUs 3
		GPU: 1*NVIDIA-T4(16GB)   CPU: 8 vCPUs htext passwords, to ensure data
		CPU: 2 vCPUs 8GB

### Figure 15-34 Compute node specifications

When using compute node specifications, pay attention to the following:

### **Permission control**

Permissions on general-purpose compute node specifications, for example, **modelarts.vm.cpu.2u** are not controlled. You can select the specifications as long as there are idle resources in the resource pool. ModelArts provides two specifications by default, CPU-powered **modelarts.vm.cpu.2u** and GPU-powered **modelarts.vm.gpu.p4**.

For some special specifications, contact the system administrator to request for permissions.

### Unavailable public resource pool specifications

Resources in a public resource pool are limited. If a specification is in gray, resources of the current specification have been used up. In this case, select other specifications or create your own dedicated resource pool.

### **Custom specifications**

You can customize resource specifications only when a dedicated resource pool is used. Specifications cannot be customized in public resource pools.

### Figure 15-35 Custom specifications

Specifications	Custom				•					
	CPUs	_	0.25	+	core(s)	Memory	_	512	+	MB

### 15.5.2.1.5 What Is the CUDA Version for Deploying a Service on GPUs?

CUDA 10.2 is supported by default. If a later version is required, submit a service ticket to apply for technical support.

### 15.5.2.2 Real-Time Services

### 15.5.2.2.1 What Do I Do If a Conflict Occurs in the Python Dependency Package of a Custom Prediction Script When I Deploy a Real-Time Service?

Before importing a model, save the inference code and configuration file in the model folder. When coding with Python, import custom packages in relative import (Python import) mode.

If there are packages with duplicate names in the ModelArts inference framework code and they are imported not in relative import mode, a conflict will occur, leading to a service deployment or prediction failure.

### 15.5.2.2.2 What Is the Format of a Real-Time Service API?

After an AI application is deployed as a real-time service, you can use the API for inference.

#### The format of an API is as follows:

https://Domain name/Version/infer/service ID

Example:

https://6ac81cdfac4f4a30be95xxxbb682.apig.xxx.xxx.com/v1/infers/	
468d146d-278a-4ca2-8830-0b6fb37d3b72	

#### Figure 15-36 API

< Back to Real-Time	Servic				Modify
Basic Information					
Name	service-ebec			Service ID	(             -7!
Status	Running(59 minutes until stop) ໍ ່ 0			Source	My Deployment
Failed Calls/Total Calls	0/1 View Details			Description	i t-cpu 🖉
Synchronize Data	Synchronize Data			Custom Settings	
Data Collection				Filter (?)	
Traffic Limit					
Usage Guides Prec	liction Configuration Updates	Filter	Monitoring	Events Logs	
API Address https://aa5i	025c4032d4b21816a93206fb08				Note: AK/SK or token authenticati

### 15.5.2.2.3 Why Did My Service Deployment Fail with Proper Deployment Timeout Configured?

A model can properly start after a service is deployed. The startup status of a model can be detected through a health check.

Check whether a service is deployed using a health check API for custom images. When creating an AI application, configure a health check delay to ensure the initialization of containers.

It is a good practice to configure a proper health check delay for service deployment.

### 15.6 API/SDK

# 15.6.1 Can ModelArts APIs or SDKs Be Used to Download Models to a Local PC?

ModelArts APIs or SDKs cannot be used to download models to a local PC. However, the output models of training jobs are stored in OBS. You can use OBS APIs or SDKs to download the models.

### **15.6.2 What Installation Environments Do ModelArts SDKs Support?**

ModelArts SDKs can run in notebook or local environments. However, the supported environments vary depending on architectures. For details, see **Table 15-12**.

Development Environment	Architecture	Supported
Notebook	Arm	Yes
	x86	Yes
Local environment	Arm	No
	x86	Yes

 Table 15-12
 SDK installation environments

# 15.6.3 Does ModelArts Use the OBS API to Access OBS Files over an Intranet or the Internet?

In the same region, ModelArts uses the OBS API to access files stored in OBS over an intranet and does not consume public network traffic.

If you download data from OBS through the Internet, you will be charged for the OBS public network traffic.

# 15.6.4 How Do I Obtain a Job Resource Usage Curve After I Submit a Training Job by Calling an API?

After submitting a training job by calling an API, log in to the ModelArts console, choose **Training Management** > **Training Jobs**, and click the name or ID of the target training job to go to its details page. In the **Resource Usages** area, view the resource usage curve of the job.

# 15.6.5 How Do I View the Old-Version Dedicated Resource Pool List Using the SDK?

You can view the old-version dedicated resource pool list by referring to the following code:

from modelarts.session import Session
from modelarts.estimator import Estimator
algo\_info = Estimator(modelarts\_session=Session()).get\_job\_pool\_list()print(algo\_info)

### 15.7 Using PyCharm Toolkit

# 15.7.1 What Should I Do If an Error Occurs During Toolkit Installation?

### Symptom

The following error message is displayed during Toolkit installation.

### Figure 15-37 Error



### Solution

This issue occurs because the plug-in version is inconsistent with the PyCharm version. You need to obtain the plug-in of the same version as the PyCharm version, that is, version 2019.2 or later.

# 15.7.2 What Should I Do If an Error Occurs When I Edit a Credential in PyCharm Toolkit?

### Symptom

When you edit a credential in PyCharm Toolkit, the message "Validate Credential error" is displayed.



#### Or

■ TODO ● Problems ■ Terminal ● Python Console
Validate Credential Error: Authentication failed, please check your AK/SK and access authority. (5 minutes ago)

### **Possible Causes**

- Possible cause 1: Information such as the region is incorrectly configured.
- Possible cause 2: The hosts file is not configured or is incorrectly configured.
- Possible cause 3: The network proxy settings are incorrect.
- Possible cause 4: The AK/SK is incorrect.
- Possible cause 5: The computer time is incorrectly set.

### Solution

### 1. Information such as the region is incorrectly configured.

Correctly configure the region, projects, and endpoint. . For details, see Configuring a Local IDE Accessed Using PyCharm Toolkit

For example, if the endpoint is incorrect, the authentication fails.

Incorrect example: The endpoint is preceded by https.

### Figure 15-38 Configuring PyCharm Toolkit

→ ModelArts	
Region(s) :	cn-central-?  cn-central-?**
	Example: region-name1 region-id1;region-name2 region-id2
Project(s):	cn-central I cn-central-
	Example 1: keep default value. By default the project is same as region id. Example 2: custom projects in format region-name1 project1;region-name2 project2
ModelArts Endpoint :	https://modelarts.com
OBS Endpoint :	https://obs.cn-cention-contexpactom
IAM Endpoint :	https://iam-pub.cn-centr ' ^ ` i, js.com
Console Endpoint:	default
	Endpoint Example: service-name. <region-id>.domain-name</region-id>

### 2. The hosts file is not configured or is incorrectly configured.

Configure the domain names and IP addresses in the **hosts** file on the local PC. For details, see **Configuring a Local IDE Accessed Using PyCharm Toolkit**.

#### 3. Network proxy settings are incorrect.

If the network requires proxy settings, check whether the proxy settings are correct. You can also use the mobile hotspot to test.

Check whether the proxy settings are correct.

Ē	<u>File E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode	<u>Refactor Run Tools VCS Window ModelArts Help</u> yolov5-toolK
	🖆 Settings	
Project		Appearance & Behavior $ ightarrow$ System Settings $ ightarrow$ HTTP Proxy
	✓ Appearance & Behavior	• No proxy
	Appearance	Auto-detect proxy settings
	Menus and Toolbars	Automatic proxy configuration URL:
	✓ System Settings	
	Passwords	Clear passwords
	HTTP Proxy	Manual proxy configuration
	Data Sharing	HTTP O SOCKS
	Date Formats	Host name:
	Updates	
	File Colors 🛛 🔳	Port number:
	Scopes 🔳	No proxy for: *.1 **********************************
	Notifications	Example: *.domain.com, 192.168.*
	Quick Lists	✓ Proxy <u>a</u> uthentication
	Path Variables	Login:

#### Figure 15-39 PyCharm network proxy settings

### 4. The AK/SK is incorrect.

Obtained correct AK/SK and try again. For details, see **How Do I Obtain an Access Key?** 

If you use a RightCloud account, contact the technical support of the region to obtain the AK/SK.

#### 5. The computer time is incorrectly set.

Set the computer time to the correct time.

### 15.7.3 Why Cannot I Start Training?

If code that does not belong to the used project is selected in a boot script, training cannot be started. The following figure shows error information. You are advised to add the boot script to the project or open the project where the boot script is located, and then start the training job.

### Figure 15-40 Error

Error	
×	Boot File Path must in the project directory D:\EI\ ModelArts-Lab-master\official_examples\Using_MXNet_to_Train_Caltech101\codes.
	OK

# 15.7.4 What Should I Do If Error "xxx isn't existed in train\_version" Occurs When a Training Job Is Submitted?

### Symptom

Error "xxx isn't existed in train\_version" occurs when a training job is submitted. See the following figure.

Figure 15-41 Error "xxx isn't existed in train\_version"



### **Possible Causes**

The preceding error occurs because the user logs in to the ModelArts management console and deletes the training job after submitting the training job using PyCharm Toolkit.

PyCharm Toolkit records the training job IDs of ModelArts on the cloud. If you manually delete the job on the ModelArts management console, a message is displayed indicating that the job with the ID cannot be found when you submit the job locally.

### Solution

If you have deleted a job on the ModelArts management console, you also need to delete the local configuration from Toolkit. To delete the local configuration, click **Edit Training Configuration**, find the job name, click the minus sign in the upper right corner, and confirm the deletion.

Figure 15-42 Deleting the local configuration



In the displayed confirmation dialog box, confirm the information and click **Yes** to delete the configuration. After the deletion, you can create a training job configuration and submit the training job.

# 15.7.5 What Should I Do If Error "Invalid OBS path" Occurs When a Training Job Is Submitted?

When a training job is running, the "Invalid OBS path" error is reported.

Figure 15-43 "Invalid OBS path" error



To locate the fault, perform the following operations:

- If you are using ModelArts for the first time, log in to the ModelArts management console and complete access authorization configuration. The agency authorization mode is recommended. After the global configuration is complete, submit the job again.
- Check whether the configured **Data Path in OBS** exists and whether data files exist in the directory. If the directory does not exist, create a directory on OBS and upload the training data to the directory.

### 15.7.6 What Should I Do If Error "NoSuchKey" Occurs When PyCharm Toolkit Is Used to Submit a Training Job?

### Symptom

When PyCharm Toolkit is used to submit a training job, an error is reported. The log is as follows.

		Q Aa AN * ^
	trace_Ealse_twoe_'common'_verbose_Ealse)	
134	[ Mode] HTS Service Lon2023-07-03 15:16:21.914 - file jo.pv[]ine:703] - WARNING: Retrv=9. Wait=0.1. Timestamp=1688368581.9141176	
135	5 [ModelArts Service Logl2023-07-03 15:16:22.035 - file io.pv[]ine:703] - WARNING: Retry=8, Wait=0.2, Timestamp=1688368582.0354207	
136	6 [Mode]Arts Service Log[2023-07-03 15:16:22.265 - file_io.pv[]ine:703] - WARNING: Retrv=7, Wait=0.4, Timestamp=1688368582.2653368	
137	7 [ModelArts Service Log]2023-07-03 15:16:22,702 - file_io.py[]ine:703] - WARNING: Retry=6, Wait=0.8, Timestamp=1688368582.7021663	
138	8 [ModelArts Service Log]2023-07-03 15:16:23.521 - file_io.py[]ine:703] - WARNING: Retry=5, Wait=1.6, Timestamp=1688368583.5216513	
139	9 [ModelArts Service Log]2023-07-03 15:16:25,142 - file_io.py[line:703] - WARNING: Retry=4, Wait=3.2, Timestamp=1688368585.1427376	
140	0 [ModelArts Service Log]2023-07-03 15:16:28,364 - file_io.py[line:703] - WARNING: Retry=3, Wait=6.4, Timestamp=1688368588.3648236	
141	1 [ModelArts Service Log]2023-07-03 15:16:34,786 - file_io.py[line:703] - WARNING: Retry=2, Wait=12.8, Timestamp=1688368594.786392	
142	2 [ModelArts Service Log]2023-07-03 15:16:47,623 - file_io.py[]ine:703] - WARNING: Retry=1, Wait=25.6, Timestamp=1688368607.6239572	
143	3 [Mode]Arts Service Log]2023-07-03 15:17:13.250 - file_io.py[]ine:718] - ERROR: Failed to call:	
144	4 func= <bound 0x7fbbc8ebda58="" <moxing.framework.file.src.obs.client.obsclient="" at="" method="" object="" obsclient.getobject="" of="">&gt;</bound>	
145	<pre>5 args=('test-dwi'. 'lbk/tookit_test_code/MA-new-modelarts_test-07-03-15-15-159/code/modelarts_test')</pre>	
146	6 kwargs={loadStreamInMemory:False, cache:False, }	
147	7 [ModelArts Service Log]2023-07-03 15:17:13,250 - file_io.pv[]ine:725] - ERROR:	
148	8	
145	9 enrorCode:NoSuchKey	
150	0 errorMessage:The specified key does not exist.	
151	1 reason:Not Found	
157	2	
153	3 retry:0	
154	4 [ModelArts Service Log12023-07-03 15:17:13.250 - modelarts-downloader.pv[]ine:106] - ERROR: modelarts-downloader.pv: Download directory	failed: [Errno
	<pre>{'status': 404, 'reason': 'Not Found', 'errorCode': 'NoSuchKey', 'errorMessage': 'The specified key does not exist.', 'body': None, 're '00000189149C71799013208C235593F7', 'hostId': 'y2VDvug8y5dfKW63cUD0y7TZaW]yvpRbahxYPUG+dCsmICCoxX6071Ha2aX+Brl4', 'header': [('date', ' 0716547 GWT'), ('content-type', 'application/xml'), ('content-length', '369'), ('connection', 'close'), ('x-reserved', 'amazon, aws an services are trademarks or registered trademarks of Amazon Technologies, Inc'), ('request-id', '000001891A9C71799013208C235593F7'), ('i '32A40A24ABAA0A0AEABAA0A6AEABACS70H50bH=01394LUHB(YGYS-formfip')])] file or directory or bucket not found.</pre>	equestId': 'Mon, O3 Jul 2023 nd amazon web id-2',

### **Possible Causes**

The image version is too old and incompatible with the training job.

### Solution

When using PyCharm Toolkit to submit a training job, select a frequently-used engine version supported by the training job. For details about the supported versions, see **Al engines supported by training jobs**. Do not select PyTorch-1.0.0, PyTorch-1.3.0, or PyTorch-1.4.0.

Figure 15-44 Selecting an AI engine supported by the training job

🖺 Edit Training Job Configurations				
* JobName:	MA-new-models-10-	11-09-23-817		
Job Description:				
	Frequently-used	Custom		
	Al Engine:	PyTorch	pytorch_1.8.0-cuduntu_18.04-x86_64	
Algorithm Source:	Boot File Path:			
	Code Directory:			

# 15.7.7 What Should I Do If an Error Occurs During Service Deployment?

Before deploying a model as a service, edit the configuration file and inference code based on the trained model.

If the **confi.json** configuration file or the **customize\_service.py** inference code is missing in the model storage path, an error is displayed, as shown in the following figure.

Solutions:

Write the configuration file and inference code, and save them to the OBS directory where the model to be deployed resides. For details, see **Introduction to Model Package Specifications**.

### Figure 15-45 Error

odelartszvent Log teemt log teemt log teem tog t

### 15.7.8 How Do I View Error Logs of PyCharm Toolkit?

The error logs of PyCharm Toolkit are recorded in the **idea.log** file of PyCharm. For example, in the Windows operating system, the path of the **idea.log** file is C:\Users\xxx\.ldeaIC2019.2\system\log\idea.log.

Search for **modelarts** in the log file to view all logs related to PyCharm Toolkit.

# 15.7.9 How Do I Use PyCharm ToolKit to Create Multiple Jobs for Simultaneous Training?

PyCharm ToolKit supports only one job at a time. To run another job, you must manually stop the current one.

# 15.7.10 What Should I Do If "Error occurs when accessing to OBS" Is Displayed When PyCharm ToolKit Is Used?

### Symptom

The PyCharm ToolKit log showed "Error occurs when accessing to OBS".

### **Possible Causes**

You do not have OBS permissions.

### Solution

Check whether you have the OBS permissions.

- Step 1 Log in to the ModelArts console, choose Data Management > Datasets, and click Create. You have the OBS permissions if you can access the OBS path. If you do not have the OBS permissions, go to Configure the OBS permis... to configure the OBS permissions.
- **Step 2** Configure the OBS permissions.

----End

# **16** Troubleshooting

### **16.1 General Issues**

### 16.1.1 Incorrect OBS Path on ModelArts

### Symptom

- When an OBS bucket path is used in ModelArts, a message indicating that the created OBS bucket cannot be found or message "ModelArts.2791: Invalid OBS path" is reported.
- "Error: stat:403" is reported when you perform operations on an OBS bucket.
- "Permission denied" is reported when a file is downloaded from OBS to Notebook.

### **Possible Causes**

- You do not have access to OBS buckets of other users.
- Access authorization has not been configured on ModelArts.
- Encrypted files are to upload to OBS. ModelArts does not support encrypted OBS files.
- The permissions and access control lists (ACLs) of the OBS bucket are incorrectly configured.
- When a training job is created, the code directory and boot file are configured incorrectly.

### Solution

### Check whether you have the permission to access the OBS bucket.

Check whether you have the permission to access OBS buckets of other users from a notebook instance.

Check delegation authorization.

Go to the **Global Configuration** page and check whether you have the OBS access authorization. If you do not, see Configuring Access Authorization (Global Configuration).

#### Check whether the OBS bucket is encrypted.

- 1. Log in to the OBS management console and click the bucket name to go to the **Overview** page.
- 2. Ensure that default encryption is disabled for the OBS bucket. If the OBS bucket is encrypted, click **Default Encryption** and disable it.

#### **NOTE**

Basic Configurations

When you create an OBS bucket, do not select **Archive** or **Deep Archive**. Otherwise, training models will fail.

#### Figure 16-1 Bucket encryption status

C C			
Lifecycle Rules	Not configured	Static Website Hosting	Not configured
CORS Rules	Not configured	URL Validation	Not configured
Event Notification	Not configured	Tags	Not configured
Logging	Not configured	Default Encryption	Not configured
Direct Reading	Not supported		

#### Check whether the OBS file is encrypted.

- 1. Log in to the OBS management console and click the bucket name to go to the **Overview** page.
- 2. In the navigation pane on the left, choose **Objects**. The object list is displayed. Click the name of the object that stores files and find the target file. In the **Encrypted** column of the file list, check whether the file is encrypted. File encryption cannot be canceled. In this case, cancel bucket encryption and upload images or files again.

#### Check the ACLs of the OBS bucket.

- 1. Log in to the OBS management console and click the bucket name to go to the **Overview** page.
- 2. In the navigation pane, choose **Permissions** and click **Bucket ACLs**. Then, check whether the current account has the read and write permissions. If it does not, contact the bucket owner to obtain the permissions.
- 3. In the navigation pane on the left, choose **Permissions** > **Bucket Policy**, and check whether the current OBS bucket can be accessed by IAM users.

#### Check the code directory and boot file of a training job.

- Log in to the ModelArts management console, choose Training Management
   > Training Jobs, locate the failed training job, and click its name or ID to go
   to the job details page.
- 2. In the pane on the left, check whether the code directory and startup file are correct, and ensure that the OBS file name does not contain spaces.

- Select an OBS directory for code directory. If a file is selected, the system will display a message indicating an invalid OBS path.
- The boot file must be in the .py format. Otherwise, the system will display a message indicating an invalid OBS path.

Figure 16-2 Code Directory and Boot File of a training job

Algorithm Name	223
Custom images	
Code Directory	. Edit Code
Boot File	.ру
Boot Command	

If the fault persists, see for further troubleshooting.

### 16.2 ExeML

### 16.2.1 Preparing Data

### 16.2.1.1 Failed to Publish a Dataset Version

If this fault occurs, the data does not meet the requirements of the data management module. As a result, the dataset fails to be published and the following operations cannot be performed.

Check your data, exclude the data that does not meet the following requirements, and restart the ExeML training task.

### ModelArts.4710 OBS Permission Issues

This fault is caused by OBS permissions when ModelArts interacts with OBS. If the message "OBS service Error Message" is displayed, the fault is caused by OBS permissions. Perform the following steps to rectify the fault. If this information is not contained in the error message, the fault is caused by backend services. Contact technical support.

1. Check whether the current account has OBS permissions.

Perform this step if you log in to ModelArts as an IAM user.

Grant the current IAM user with the **Tenant Administrator** permission on global services so that the user has all OBS operation permissions. For details, see "Service Overview" > "User Permissions" in *OBS User Guide*..

To restrict the IAM user account' permissions, configure the minimum OBS operation permissions for it. For details, see "**Creating a Custom Policy**".

2. Check whether the user has OBS bucket permissions.

**NOTE** 

The OBS bucket described in the following steps is specified when you create an ExeML project or the one where the dataset selected during project creation is stored.

- Check whether the current account has been granted with the read and write permissions on the OBS bucket (specified in bucket ACLs).
  - Go to the OBS management console, select the OBS bucket used by the ExeML project, and click the bucket name to go to the Overview page.
  - In the navigation pane, choose Permissions > Bucket ACLs. On the Bucket ACLs page that is displayed, check whether the current account has the read and write permissions. If it does not, contact the bucket owner to grant the permissions.
- Check whether the OBS bucket is unencrypted.

Basic Configurations

- i. Go to the OBS management console, select the OBS bucket used by the ExeML project, and click the bucket name to go to the **Overview** page.
- ii. Ensure that the default encryption function is disabled for the OBS bucket. If the OBS bucket is encrypted, click **Default Encryption** and change its encryption status.

**Figure 16-3** Checking whether the default encryption function is enabled for the OBS bucket

-			
Lifecycle Rules	Not configured	Static Website Hosting	Not configured
CORS Rules	🗢 1 rule	URL Validation	Not configured
Event Notification	O Not configured	Tags	Not configured
Logging	Not configured	Default Encryption	Not configured
Direct Reading	Not configured		

- Check whether the direct reading function of archived data is disabled.
  - i. Go to the OBS management console, select the OBS bucket used by the ExeML project, and click the bucket name to go to the **Overview** page.

ii. Ensure that the direct reading function is disabled for the archived data in the OBS bucket. If this function is enabled, click **Direct Reading** and disable it.

Figure 16-4 Disabling the direct reading function



### ModelArts.4711 Number of Labeled Samples in the Dataset Does Not Meet Algorithm Requirements

Each labeling type must contain at least five images.

### ModelArts.4342 Labeling Information Does Not Meet Splitting Conditions

If this fault occurs, modify the labeling data based on the following suggestions and try again.

- At least two multi-label samples (that is, an image contains multiple labels) are required. If you enable dataset splitting when starting training and the number of images with multiple labels is less than 2, the dataset splitting fails. Check your labeling information and ensure that more than two images with multiple labels are labeled.
- After the dataset is split, the label classes contained in the training set and validation set are different. Cause: In the multi-label scenario, after random data segmentation, samples containing a certain type of labels are classified into the training set. As a result, the verification set does not contain the label samples. This issue rarely occurs. You can try to release a new version to handle the issue.

### ModelArts.4371 Dataset Version Already Exists

If this error code is displayed, the dataset version already exists. In this case, republish the dataset version.

### ModelArts.4712 Datasets Are Being Imported or Synchronized

If the dataset used in ExeML is being imported or synchronized, this error occurs during training. In this case, start the ExeML training task after other tasks are complete.

### 16.2.1.2 Invalid Dataset Version

If this issue occurs, the dataset version is successfully released but does not meet the requirements of the ExeML training jobs. As a result, an error message is displayed, indicating that the dataset version does not meet the requirements.

### Labeling Information Does Not Meet the Trainning Requirements

For different types of ExeML projects, training jobs have the following requirements on datasets:

- Image classification: There are at least two classes (that is, at least two labels) for the images to be trained, and the number of images in each class cannot be less than 5.
- Object detection: There is at least one class (that is, at least one label) for the images to be trained, and the number of images for each class cannot be less than 5.
- Predictive analytics: The dataset of the predictive analytics task is not managed in a unified manner. Even if the data does not meet the requirements, no fault information is displayed in this issue.
- Sound classification: There are at least two classes (that is, at least two labels) for the audio files to be trained, and the number of audio files in each class cannot be less than 5.
- Text classification: There are at least two classes (that is, at least two labels) for the text files to be trained, and the number of text files in each class cannot be less than 20.

### 16.2.2 Training a Model

### 16.2.2.1 Failed to Create an ExeML-powered Training Job

This fault is typically caused by a backend service failure. Recreate the training job later. If the fault persists after three retries, contact .

### 16.2.2.2 ExeML-powered Training Job Failed

A training job that is successfully created fails to be executed due to some faults.

To rectify this fault, check whether your account is in arrears first. If your account is normal, rectify the fault based on the job type.

- For details about how to rectify the job training faults related to Image Classification, Sound Classification, and Text Classification, see Checking Whether Data Exists in OBS, Checking the OBS Access Permission, and Checking Whether the Images Meet the Requirements.
- For details about how to rectify the job training faults related to **Object Detection**, see **Checking Whether Data Exists in OBS**, **Checking the OBS Access Permission**, **Checking Whether the Images Meet the Requirements**, and **Checking Whether the Marking Boxes Meet the Object Detection Requirements**.
- For details about how to rectify the job training faults related to **Predictive Analytics**, see **Checking Whether Data Exists in OBS**, **Checking the OBS**

### Access Permission, and Troubleshooting of a Predictive Analytics Job Failure.

### Checking Whether Data Exists in OBS

If the images or data stored in OBS is deleted and not synchronized to ModelArts ExeML or datasets, the task will fail.

Check whether data exists in OBS. For Image Classification, Sound Classification, Text Classification, and Object Detection, you can click **Synchronize Data Source** on the **Data Labeling** page of ExeML to synchronize data from OBS to ModelArts.

### **Checking the OBS Access Permission**

If the access permission of the OBS bucket cannot meet the training requirements, the training fails. Do the following to check the OBS permissions:

- Check whether the current account has been granted with the read and write permissions on the OBS bucket (specified in bucket ACLs).
  - a. Go to the OBS management console, select the OBS bucket used by the ExeML project, and click the bucket name to go to the **Overview** page.
  - b. In the navigation pane, choose **Permissions** and click **Bucket ACLs**. Then, check whether the current account has the read and write permissions. If it does not, contact the bucket owner to obtain the permissions.
- Check whether the OBS bucket is unencrypted.
  - a. Go to the OBS management console, select the OBS bucket used by the ExeML project, and click the bucket name to go to the **Overview** page.
  - b. Ensure that the default encryption function is disabled for the OBS bucket. If the OBS bucket is encrypted, click **Default Encryption** and change its encryption status.

#### Figure 16-5 Default encryption status

Lifecycle Rules	0 Not configured	Static Website Hosting	O Not configured
CORS Rules	O Not configured	URL Validation	0 Not configured
Event Notification	0 Not configured	Tags	O Not configured
Logging	Not configured	Default Encryption	Not configured
Direct Reading	Not configured		

- Check whether the direct reading function of archived data is disabled.
  - a. Go to the OBS management console, select the OBS bucket used by the ExeML project, and click the bucket name to go to the **Overview** page.

b. Ensure that the direct reading function is disabled for the archived data in the OBS bucket. If this function is enabled, click **Direct Reading** and disable it.

#### Figure 16-6 Disabled direct reading

Lifecycle Rules	O Not configured	Static Website Hosting	Not configured
CORS Rules	O Not configured	URL Validation	• Not configured
Event Notification	O Not configured	Tags	Not configured
Logging	O Not configured	Default Encryption	• Not configured
Direct Reading	O Not configured		

• Ensure that files in OBS are not encrypted.

Do not select KMS encryption when uploading images or files. Otherwise, the dataset fails to read data. File encryption cannot be canceled. In this case, cancel bucket encryption and upload images or files again.

#### Figure 16-7 File encryption status

Name	Storage Cla	Size (?) J⊟	Encrypted	Restoration
← Back				
3179751458_9646d839f6_n.jpg	Standard	25.54 KB	No	

### Checking Whether the Images Meet the Requirements

Currently, ExeML does not support four-channel images. Check your data and exclude or delete this format of images.

### Checking Whether the Marking Boxes Meet the Object Detection Requirements

Currently, object detection supports only rectangular labeling boxes. Ensure that the labeling boxes of all images are rectangular ones.

If a non-rectangle labeling box is used, the following error message may be displayed:

Error bandbox.

For other types of projects (such as image classification and sound classification), skip this checking item.
#### Troubleshooting of a Predictive Analytics Job Failure

1. Check whether the data used for predictive analytics meets the following requirements.

The predictive analytics task releases datasets without using the data management function. If the data does not meet the requirements of the training job, the job will fail to run.

Check whether the data used for training meets the requirements of the predictive analytics job. The following lists the requirements. If the requirements are met, go to the next step. If the requirements are not met, adjust the data based on the requirements and then perform the training again.

- The name of files in a dataset consists of letters, digits, hyphens (-), and underscores (\_), and the file name suffix is .csv. The files cannot be stored in the root directory of an OBS bucket, but in a folder in the OBS bucket, for example, /obs-xxx/data/input.csv.
- The files are saved in CSV format. Use newline characters (\n or LF) to separate lines and commas (,) to separate columns of the file content. The file content cannot contain Chinese characters. The column content cannot contain special characters such as commas (,) and newline characters. The quotation marks are not supported. It is recommended that the column content consist of letters and digits.
- The number of training columns is the same. There are at least 100 different data records (a feature with different values is considered as different data) in total. The training columns cannot contain data of the timestamp format (such as *yy-mm-dd* or *yyyy-mm-dd*). Ensure that there are at least two values in the specified label column and no data is missing. In addition to the label column, the dataset must contain at least two valid feature columns. Ensure that there are at least two values in each feature column and that the percentage of missing data must be lower than 10%. The training data CSV file cannot contain the table header. Otherwise, the training fails. Due to the limitation of the feature filtering algorithm, place the label column in the last column of the dataset. Otherwise, the training may fail.
- 2. ModelArts automatically filters data and then starts the training job. If the preprocessed data does not meet the training requirements, the training job fails to be executed.

Filter policies for columns in a dataset:

- If the vacancy rate of a column is greater than the threshold (0.9) set by the system, the data in this column will be deleted during training.
- If a column has only one value (that is, the data in each row is the same), the data in this column will be deleted during training.
- For a non-numeric column, if the number of values in this column is equal to the number of rows (that is, the values in each row are different), the data in this column will be deleted during training.

After the preceding filtering, if the data in the dataset does not meet the training requirements in Item 1, the training fails or cannot be executed. Complete the data before starting the training.

3. Restrictions for a dataset file:

a. If you use the 2U8G flavor (2 vCPUs and 8 GB of memory), it is recommended that the size of the dataset file be less than 10 MB. If the file size meets the requirements but the data volume (product of the number of rows and the number of columns) is extremely large, the training may still fail. It is recommended that the product be less than 10,000.

If you use the 8U32G flavor (8 vCPUs and 32 GB of memory), it is recommended that the size of the dataset file be less than 100 MB. If the file size meets the requirements but the data volume (product of the number of rows and the number of columns) is extremely large, the training may still fail. It is recommended that the product be less than 1,000,000.

4. If the fault persists, contact .

#### 16.2.3 Deploying a Model

#### **16.2.3.1 Failed to Submit the Real-time Service Deployment Task**

This fault is typically caused by the limited quota of the account.

In an ExeML project, after the deployment is started, the model is automatically deployed as a real-time service. If the number of real-time services exceeds the quota limit, the model cannot be deployed as a service. In this case, an error message is displayed in the ExeML project, indicating that the real-time service deployment task fails to be submitted.

#### Troubleshooting

- Method 1: Choose Service Deployment > Real-time Services. On the displayed page, delete services that are no longer used to release resources.
- Method 2: If the deployed real-time service still needs to be used, you are advised to apply for a higher quota.

#### 16.2.3.2 Failed to Deploy a Real-time Service

This fault is typically caused by a backend service failure. You are advised to redeploy the real-time service later. If the fault persists after three retries, obtain the following information and contact .

• Obtain a service ID.

Go to the **Service Deployment > Real-Time Services** page. In the service list, find the real-time service deployed in the ExeML task. All the services of ExeML start with **exeML-** Click the service name to go to the service details page. In the basic information area, obtain **Service ID**.

#### Figure 16-8 Obtaining a service ID

Name	exeML-5766_ExeML_1605	Service ID	b3d0b279-ef02-44ae-8b62-51ff7c559934
Status	<ul> <li>Stop</li> </ul>	Source	My Deployment
Failed Calls/Total Calls	0 /1 Details	Network Configuration	Unconfigured
Description	Created by Exeml project(name: exeML-5766). 🖉	Custom Settings	
Sample 🕐		Filter 🕐	
Synchronize Data 🕐	Synchronize Data to Dataset		

• Obtain events about the real-time service.

On the service details page, click the **Events** tab. Take a screenshot of the event information table, and send the screenshot to technical support personnel.

#### Figure 16-9 Obtaining events

Usage Guides Pre	diction Configuration Updates Filter Monitoring Events Logs Sharing	
		2020/11/19 10:37:44 - 202X 🗎 All statuses 🔻 C
Event Type	Event Message	Occurred 4
Normal	stop service	Nov 19, 2020 10:52:19 GMT+08:00
Normal	start model succes	Nov 19, 2020 10:44:19 GMT+08:00
Normal	pull image success	Nov 19, 2020 10:44:19 GMT+08:00
Normal	pulling model image	Nov 19, 2020 10:42:39 GMT+08:00
Normal	schedule resource success	Nov 19, 2020 10:42:39 GMT+08:00
Normal	prepare environment success	Nov 19, 2020 10:42:39 GMT+08:00
Normal	preparing environment	Nov 19, 2020 10:42:22 GMT+08:00
Normal	model (eneML-5766, ExeML_47a/51cb 00.3) build image success	Nov 19, 2020 10:42:19 GMT+08:00
Normal	building image for model [eveML-5766_EveML_47af51cb 0.0.3]	Nov 19, 2020 10:37:45 GMT+08:00
Normal	start to deploy service	Nov 19, 2020 10:37:44 GMT+08:00

#### 16.2.4 Publishing a Model

#### 16.2.4.1 Failed to Submit the Model Publishing Task

This fault is typically caused by a backend service failure. You are advised to recreate the training job later. If the fault persists after three retries, contact .

#### 16.2.4.2 Failed to Publish a Model

This fault is typically caused by a backend service failure. You are advised to recreate the training job later. If the fault persists after three retries, obtain the following information and contact .

• Obtain a model ID.

Choose **AI Application Management** > **AI Applications**. In the AI application list, find the applications automatically created in the ExeML task. All the AI

applications generated by ExeML start with **exeML**-. Click the model name to go to the model details page. In the **Basic Information** area, obtain the value of **ID**.

#### Figure 16-10 Obtaining a model ID

Basic Information			
Name	exeML-5766_ExeML_47af51cb	Label	
Status	✓ Normal	Version	0.0.3
ID	21d3be95-180d-43ec-a2d0-b6cdc7836b2a	Size	91.03 MB
Runtime Environment	tf1.13-python3.7-cpu	AI Engine	TensorFlow
Deployment Type	Real-Time Services/Batch Services	Description	🖉
Model Document			

• Obtain model events.

On the model details page, click the **Events** tab. Take a screenshot of the event information table, and send the screenshot to technical support personnel.

#### Figure 16-11 Obtaining events

Pa	rameter Con	figuration Runtime Dependency Events	
			2020/11/19 10:26:19 − 202X 🗎 🛛 All statuses 💌 C
Eve	nt Type	Event Message	Occurred ↓≡
0	Normal	Image built successfully.	Nov 19, 2020 10:42:19 GMT+08:00
0	Normal	The status of the Image building task is READY.	Nov 19, 2020 10:42:19 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:41:59 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:41:38 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:41:18 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:40:58 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:40:38 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:40:18 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:39:58 GMT+08:00
0	Normal	The status of the image building task is CREATING.	Nov 19, 2020 10:39:37 GMT+08:00

#### 16.3 DevEnviron

#### **16.3.1 Environment Configuration Faults**

#### 16.3.1.1 Disk Space Used Up

#### Symptom

- Error message "No Space left on Device" is displayed when a notebook instance is used.
- Error message "Disk quota exceeded" is displayed when code is executed in a notebook instance.

	382 det	get_result(self):	
	183	if selfex	
> 3	184	raise s	
	385	else:	
- 3	386	return selfresult	

#### Possible Causes

- After a file is deleted from the navigation pane on the left of JupyterLab, the file is moved to the recycle bin by default. This occupies memory, leading to insufficient disk space.
- The disk quota is insufficient.

#### Solution

Check the storage space used by the VM, check the memory used by files in the recycle bin, and delete unnecessary large files from the recycle bin.

1. On the notebook instance details page, view the storage capacity of the instance.

< notebook-cb61			
Name	···· <i>L</i>	Flavor	CPU: 2vCPUs 8GB
Status	<ul> <li>Running ()</li> </ul>	Image	spark2.4.5-ubuntu18.04
ID	ď	Created At	Aug 15, 2023 21:11:49 GMT+08:00
Storage Mount	/home/ma-user/work/	Updated At	Aug 16, 2023 09:59:30 GMT+08:00
Storage Capacity	20 GB (EVS) Expansion	Dedicate Pool	

 Check the storage space used by the VM. The storage space is typically close to the storage capacity. cd /home/ma-user/work

du -h --max-depth 0



 Run the following commands to check the memory used by the recycle bin (recycle bin files are stored in /home/ma-user/work/.Trash-1000/files by default): cd /home/ma-user/work/.Trash-1000/ du -ah

2024-04-30

(PyTorc	h-1.4) [ma-user work]\$cd /home/ma-user/work/.Trash-1000/
(PyTorc	:h-1.4) [ma-user .Trash-1000]\$du -ah
2.0K	./files/Untitled.ipynb
1000M	./files/bigFile-Copy1.txt
977K	./files/bigFile.txt
512	./files/bigFile1.txt
9.8G	./files/bigFile10.txt
9.8G	./files/bigFile11.txt
<b>21</b> G	./files
512	./info/Untitled.ipynb.trashinfo
512	./info/bigFile-Copy1.txt.trashinfo
512	./info/bigFile.txt.trashinfo
512	./info/bigFile1.txt.trashinfo
512	./info/bigFile10.txt.trashinfo
512	./info/bigFile11.txt.trashinfo
512	
512	
512	
512	
512	
512	
<b>10</b> K	./info
<b>21</b> G	•
(PyTorc	:h-1.4) [ma-user .Trash-1000]\$[]

4. Delete unnecessary large files from the recycle bin. Deleted files cannot be restored.

rm *{File path}* 



#### **NOTE**

If the name of the folder or file you want to delete contains spaces, add single quotation marks to the name.



5. Run the following commands to check the storage space used by the VM again:

cd /home/ma-user/work du -h --max-depth 0

6. If the notebook instance uses an EVS disk for storage, expand the storage capacity on the notebook instance details page.

< notebook		
Name	notebook- 1	
Status	💿 Running 🔿	
ID		٥
Storage Mount	/home/ma-user/work/	
Storage Capacity	20 GB (EVS) Expansion	

#### **Summary and Suggestions**

It is a good practice to delete unnecessary files when using a notebook instance to prevent a training failure caused by insufficient disk space.

### 16.3.1.2 An Error Is Reported When Conda Is Used to Install Keras 2.3.1 in Notebook

#### Symptom

An error is reported when Conda is used to install Keras 2.3.1.

) n O	conda install keras=2.3.1			
8	<pre>/home/ma-user/anaconda3/lib/python3.7/site-packages/requests/initpy:91: RequestsDependencyWarning: urllib3 (1.26.12)     RequestsDependencyWarning) Collecting package metadata (current_repodata.json): done Solving environment: failed with initial frozen solve. Retrying with flexible solve. Collecting package metadata (repodata.json): done Solving environment: failed with initial frozen solve. Retrying with flexible solve. Solving environment: failed with initial frozen solve. Retrying with flexible solve. Solving environment: Found conflicts! Looking for incompatible packages. This can take several minutes. Press CTRL-C to abort.</pre>			
	failed			
	# >>>>>>>>>> ERROR REPORT <<<<<<<			
	<pre>Traceback (most recent call last): File "/home/ma-user/anaconda3/lib/python3.7/site-packages/conda/cli/install.py", line 265, in install should_retry_solve=(_should_retry_unfrozen or repodata_fn != repodata_fns[-1]), File "/home/ma-user/anaconda3/lib/python3.7/site-packages/conda/core/solve.py", line 117, in solve_for_transaction should_retry_solve) File "/home/ma-user/anaconda3/lib/python3.7/site-packages/conda/core/solve.py", line 158, in solve_for_diff force_remove, should_retry_solve) File "/home/ma-user/anaconda3/lib/python3.7/site-packages/conda/core/solve.py", line 275, in solve_final_state ssc = selfadd_specs(ssc) File "/home/ma-user/anaconda3/lib/python3.7/site-packages/conda/core/solve.py", line 696, in _add_specs raise UnsatisfiableError({}) conda.exceptions.UnsatisfiableError: Did not find conflicting dependencies. If you would like to know which packages conflict ensure that you have enabled unsatisfiable hints.</pre>			
	conda configset unsatisfiable_hints True			

#### **Possible Cause**

There are network issues with Conda. Run the **pip install** command to install Keras 2.3.1.

#### Solution

#### Run the **!pip install keras==2.3.1** command to install Keras.

]  pip install keras2.3.1
<pre>Looking in indexes: http://repo.myhuameicloud.com/repository/pyl/simple Collecting kerns=-2.3.1 Using cached http://repo.myhuameicloud.com/repository/pyl/spackages/ad/fd/fd/fdf875acd28500a4A82b668447082bbc06fe0e585060cb/Keras-2.3.1.py2.py3-none-my.whl (377 kf Requirement already satisfied: keras-perprocessing=1.0.5 in /home/ma-user/anaconda3/envs/fensorFlow-1.13-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.0.8) Requirement already satisfied: keras-perprocessing=1.0.5 in /home/ma-user/anaconda3/envs/fensorFlow-1.13-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.1.2) Requirement already satisfied: pymal in /inne/ma-user/anaconda3/envs/fensorFlow-1.13-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.2.5) Requirement already satisfied: stpy=0.1 /inne/ma-user/anaconda3/envs/fensorFlow-1.13-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.2.5) Requirement already satisfied: stpy=0.14 in /home/ma-user/anaconda3/envs/fensorFlow-1.13-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.7.3) Requirement already satisfied: stpy=0.14 in /home/ma-user/anaconda3/envs/fensorFlow-1.15-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.7.3) Requirement already satisfied: stpy=0.14 in /home/ma-user/anaconda3/envs/fensorFlow-1.15-gpu/lb/python3.7/site-packages (from keras=-2.3.1) (1.7.3)</pre>
<pre>Nequirement aiready satisfied: six&gt;1.9.6 in /nome/ma-user/anaconaas/envs/tensorriow-1.13-gpu/lid/python5.//site-packages (trom keras=+2.5.1) (1.10.0) Installing uninstall: keras Found existing installation: keras 2.2.4 Uninstalling keras=2.2.4 Successfully uninstalled Keras=2.3.1</pre>

### 16.3.1.3 Error "HTTP error 404 while getting xxx" Is Reported During Dependency Installation in a Notebook

#### Symptom

An error is reported during dependency installation in a notebook instance. The following shows the error.



#### Possible Causes

The dependency is not in the PyPI source or the source is unavailable.

#### Solution

Run the following command to download the dependency from another source:

pip install -i Source address Dependency name

## 16.3.1.4 The numba Library Has Been Installed in a Notebook Instance and Error "import numba ModuleNotFoundError: No module named 'numba'" Is Reported

#### Symptom

After you install the **numba** library in a notebook instance by running the **!pip install numba** command, the library is running properly and is saved as a custom image. However, an error is reported indicating that the library does not exist when you run the script in DataArts Studio.

#### Possible Causes

Multiple virtual environments are created and the **numba** library is installed in python-3.7.10, as shown in **Figure 16-12**.



<pre>[ma-user work]\$con /home/ma-user/anac d version!    RequestsDependen # conda environmen #</pre>	da inf onda3/ cyWarn ts:	oenvs lib/python3.7/site-packages/requests/init ing)
base		/home/ma-user/anaconda3
PyTorch-1.8	*	<pre>/home/ma-user/anaconda3/envs/PyTorch-1.8</pre>
python-3.7.10		<pre>/home/ma-user/anaconda3/envs/python-3.7.10</pre>

#### Solution

Run the **conda deactivate** command in Termina to exit the current virtual environment and enter the default base environment. Run the **pip list** command to query the installed packages. Install and save the required dependencies, switch to the specified virtual environment, and run the script.

Using user ma-user	
Ubuntu 18.04.6 LTS, C	UDA-10.2
Tips:	
1) Navigate to the ta	rget conda environment. For details, see /home/ma-user/README.
2) Copy (Ctrl+C) and	paste (Ctrl+V) on the jupyter terminal.
3) Store your data in	/home/ma-user/work, to which a persistent volume is mounted.
(PyTorch-1.8) [ma-use	r work]\$conda deactivate
(base) [ma-user work]	\$conda deactivate
[ma-user work]\$pip li	st
Package	Version
absl-py	1.3.0
addict	2.4.0
APScheduler	3.9.1

#### 16.3.2 Instance Faults

### 16.3.2.1 Failed to Create a Notebook Instance and JupyterProcessKilled Is Displayed in Events

#### Symptom

A user failed to create a notebook instance, and **JupyterProcessKilled** was displayed in **Events**.

#### **Possible Causes**

This fault occurs because the Jupyter process is killed. Generally, the notebook instance automatically restarts. If it does not restart, its creation fails. Check whether the failure is caused by the custom image issue.

#### Solution

Check whether the custom image is correct.

When registering a custom image on the ModelArts console after it is created, ensure that its architecture and type are the same as those of the source image.

Figure	16-13	Registering	an image

* SWR Source	
	Example: <swr-domain-name>/<namespace>/<repository>:<tag></tag></repository></namespace></swr-domain-name>
Description	
	0/256 /
+ Architecture	¥86.64 ADM
* Architecture	
<b>★</b> Туре	🗹 CPU 🗌 GPU

< Register Image

#### 16.3.2.2 What Do I Do If I Cannot Access My Notebook Instance?

Troubleshoot the issue based on error code.

#### A Black Screen Is Displayed When a Notebook Instance Is Opened

A black screen is displayed after a notebook instance is opened, which is caused by a proxy issue. Change the proxy to rectify the fault.

#### A Blank Page Is Displayed When a Notebook Instance Is Opened

- If a blank page is displayed after a notebook instance is opened, clear the browser cache and open the notebook instance again.
- Check whether the ad filtering component is installed for the browser. If yes, disable the component.

#### Error 404

If this error is reported when an IAM user creates an instance, the IAM user does not have the permissions to access the corresponding storage location (OBS bucket).

#### Solution

- 1. Log in to the OBS console using the primary account and grant access permissions for the OBS bucket to the IAM user.
- 2. After the IAM user obtains the permissions, log in to the ModelArts console, delete the instance, and use the OBS path to create a notebook instance.

#### Error 503

If this error is reported, it is possible that the instance is consuming too many resources. If this is the case, stop the instance and restart it.

#### Error 500

Notebook JupyterLab cannot be opened, and error 500 is reported. The possible cause is that the disk space in the **work** directory is used up. In this case, identify the fault cause and clear the disk by referring to .

#### Error "This site can't be reached"

王动公署代理

After a notebook instance is created, click **Open** in the **Operation** column. The error message shown in the following figure is displayed.

This site can't be reached	
The webcace at https://authoring-modelarts. /lab might be temp permanently to a new web address.	huaweicloud.com/dfc45125
ERR_TUNNEL_CONNECTION_FAILED	

To solve the problem, copy the domain name of the page, add it to the **Do not use proxy server for addresses beginning with** text box, and save the settings.

于幼仪直心生
将代理服务器用于以太网或 Wi-Fi 连接。这些设置不适用于 VPN 连接。
使用代理服务器
サ 地址 端口
http:// i.com 8080
请勿对以下列条目开头的地址使用代理服务器。若有多个条目,请使用英 文分号 (:) 来分隔。 ————————————————————
n;
✓ 请勿将代理服务器用于本地(Intranet)地址
保存

2024-04-30

#### 16.3.2.3 What Should I Do When the System Displays an Error Message Indicating that No Space Left After I Run the pip install Command?

#### Symptom

In the notebook instance, error message "No Space left..." is displayed after the **pip install** command is run.

#### Solution

You are advised to run the **pip install --no-cache \*\*** command instead of the **pip install \*\*** command. Adding the **--no-cache** parameter can solve such problem.

### 16.3.2.4 What Do I Do If the Code Can Be Run But Cannot Be Saved, and the Error Message "save error" Is Displayed?

If the notebook instance can run the code but cannot save it, the error message "save error" is displayed when you save the file. In most cases, this error is caused by a security policy of Web Application Firewall (WAF).

On the current page, some characters in your input or output of the code are intercepted because they are considered to be a security risk. Submit a service ticket and contact customer service to check and handle the problem.

#### 16.3.2.5 ModelArts.6333 Error Occurs

#### Symptom

When you use a notebook instance, the ModelArts.6333 error is displayed.

#### **Possible Cause**

The fault may be caused by instance overload. The notebook instance automatically restores. Refresh the page and wait for several minutes. The common cause is that the memory is used up.

#### Solution

When this error occurs, the notebook instance automatically restores. You can refresh the page and wait for several minutes.

The common cause is that the memory is used up. You can use the following methods to rectify the fault.

- Method 1: Replace the notebook instance with a resource with higher specifications.
- Method 2: Adjust the parameters in the code to reduce memory occupation. If the memory is still insufficient after the code is modified, use method 1.
  - a. Call the sklearn method **silhouette\_score(addr\_1,siteskmeans.labels)** and specify the **sample\_size** parameter to reduce memory occupation.
  - b. When calling the **train** method, you can try to decrease the value of **batch\_size**.

### 16.3.2.6 What Can I Do If a Message Is Displayed Indicating that the Token Does Not Exist or Is Lost When I Open a Notebook Instance?

#### Symptom

You shared your notebook URL with others, but they receive an error message "... lost token or incorrect token...." when attempting to access the URL.

#### **Possible Cause**

They do not have the token of the account.

#### Solution

Add the token of the notebook owner to the end of the URL.

#### 16.3.3 Code Running Failures

### 16.3.3.1 Error Occurs When Using a Notebook Instance to Run Code, Indicating That No File Is Found in /tmp

#### Symptom

When the a notebook instance is used to run code, the following error occurs:

FileNotFoundError: [Error 2] No usable temporary directory found in ['/tmp', '/var/tmp', '/usr/tmp', 'home/ma-user/work/SR/RDN\_train\_base']

#### Figure 16-14 Code running error



#### **Possible Cause**

Check whether a large amount of data is saved in /tmp.

#### Solution

 Go to the Terminal page. In the /tmp directory, run the du -sh \* command to check the space usage of the directory. sh-4.3\$cd /tmp sh-4.3\$du -sh \* 4.0K core-js-banners 0 npm-19-41ed4c62 6.7M v8-compile-cache-1000

- 2. Delete unnecessary large files.
  - a. Delete the sample file test.txt: rm -f /home/ma-user/work/data/ test.txt
  - b. Delete the sample folder data: rm -rf /home/ma-user/work/data/

#### 16.3.3.2 What Do I Do If a Notebook Instance Won't Run My Code?

If a notebook instance fails to execute code, you can locate and rectify the fault as follows:

 If the execution of a cell is suspended or lasts for a long time (for example, the execution of the second and third cells in Figure 16-15 is suspended or lasts for a long time, causing execution failure of the fourth cell) but the notebook page still responds and other cells can be selected, click interrupt the kernel highlighted in a red box in the following figure to stop the execution of all cells. The notebook instance retains all variable spaces.



Figure 16-15 Stopping all cells

- 2. If the notebook page does not respond, close the notebook page and the ModelArts console. Then, open the ModelArts console and access the notebook instance again. The notebook instance retains all the variable spaces that exist when the notebook instance is unavailable.
- 3. If the notebook instance still cannot be used, access the **Notebook** page on the ModelArts console and stop the notebook instance. After the notebook instance is stopped, click **Start** to restart the notebook instance and open it. The instance will have preserved all the spaces for the variables that were unable to run.

### 16.3.3.3 Why Does the Instance Break Down When dead kernel Is Displayed During Training Code Running?

The notebook instance breaks down during training code running due to insufficient memory caused by large data volume or excessive training layers.

After this error occurs, the system automatically restarts the notebook instance to fix the instance breakdown. In this case, only the breakdown is fixed. If you run the training code again, the failure will still occur. To solve the problem of insufficient memory, you are advised to create a new notebook instance and use a resource pool of higher specifications, such as a dedicated resource pool, to run the training code. An existing notebook instance that has been successfully created cannot be scaled up using resources with higher specifications.

#### 16.3.3.4 What Do I Do If cudaCheckError Occurs During Training?

#### Symptom

The following error occurs when the training code is executed in a notebook:

cudaCheckError() failed : no kernel image is available for execution on the device

#### Possible Cause

Parameters **arch** and **code** in **setup.py** have not been set to match the GPU compute power.

#### Solution

For Tesla V100 GPUs, the GPU compute power is **-gencode** arch=compute\_70,code=[sm\_70,compute\_70]. Set the compilation parameters in setup.py accordingly.

#### 16.3.3.5 What Do I Do If Insufficient Space Is Displayed in DevEnviron?

If space is insufficient, use notebook instances with EVS disks.

Upload the code and data of the affected notebook instance to an OBS bucket. Then, create a notebook instance with EVS disks, and download the data from OBS to the new notebook instance. For details, see **How Do I Upload a File from a Notebook Instance to OBS or Download a File from OBS to a Notebook Instance?** 

### 16.3.3.6 Why Does the Notebook Instance Break Down When opency.imshow Is Used?

#### Symptom

When opency imshow is used in a notebook instance, the notebook instance breaks down.

#### Possible Causes

The cv2.imshow function in OpenCV malfunctions in a client/server environment such as Jupyter. However, Matplotlib does not have this problem.

#### Solution

Display images by referring to the following example. Note that OpenCV displays BGR images while Matplotlib displays RGB images.

Python:

from matplotlib import pyplot as plt
import cv2
img = cv2.imread('Image path')
plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB))
plt.title('my picture')
plt.show()

### 16.3.3.7 Why Cannot the Path of a Text File Generated in Windows OS Be Found In a Notebook Instance?

#### Symptom

When a text file generated in Windows is used in a notebook instance, the text content cannot be read and an error message may be displayed indicating that the path cannot be found.

#### **Possible Causes**

The notebook instance runs Linux and its line feed format (CRLF) differs from that (LF) in Windows.

#### Solution

Convert the file format to Linux in your notebook instance.

Shell:

dos2unix *File name* 

### 16.3.3.8 What Do I Do If No Kernel Is Displayed After a Notebook File Is Created?

#### Symptom

After a notebook file is created, "No Kernel" is displayed in the upper right corner of the page.

6 vCPU + 28 GiB + 1 x Tesla T4 👘 No Kernel 🔘 <
+ + + +

#### Possible Causes

The **code.py** file in the work directory conflicts with the name of the import code file on which the kernel depends.

#### Solution

1. View the latest log file starting with **kernelgateway** in **/home/ma-user/log/** and search for the logs near **Starting kernel**. If the stack similar to the following is displayed, the possible cause is that the name of the **code.py** file in the work directory conflicts with the name of the import code file on which the kernel depends.

[KernelGatewayApp] Starting kernel: ['/home/ma-user/anaconda3/envs/PyTorch-1.8/bin/python', '-m', 'ipykernel', '-f', '/home/ma-user/.local/share/jupyter/runtime// dff-8d3a-bd22ef8a17c3.json']	cernel-6df62665-ebde-4
[KernelGatewayApp] Connecting to: tcp://127.0.0.1:52075	
Traceback (most recent call last):	
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/runpy.py", line 193, in _run_module_as_main " main " mod spec	
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/runpy.py", line 85, in _run_code exec(code, run_globals)	
File "homes/maiser/anacondal/envs/PyTorch-1.8/11b/python3.7/site-packages/ipykernel/_mainpy", line 2, in <module> from ipykernel import kernelapp as app</module>	
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/ipykernel/kernelapp.py", line 42, in <module> from .ipkernel import IPythonKernel</module>	
File "/home/maiuser/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/lpykernel/ipkernel.py", line 38, in <module> from.debugger import Debugger</module>	
File "/home/ma_user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/Ipykernel/debugger.py", line 21, in <module> from debugpy.server import api # noga</module>	
File "/home/maiuser/anaconda3/envs/PyTorch-1.8/11b/python3.7/site-packages/debugpy/server/_initpy", line 7, in <module> import debugpy. vendored.force pythed # noqe</module>	
File "/home/maiuser/anaconda3/envs/PyTorch-1.8/11b/python3.7/site-packages/debugpy/_vendored/force_pydevd.py", line 28, in <pre>cmodule&gt;</pre> pydevd constants = import module( pydevd constants = import module( pydevd constants)	
File "/home/ma_user/anaconda3/envs/PyTorch-1.8/lib/gython3.7/importlib/_initpy", line 127, in import_module return, bootstrapgcd_import(name[level], package, level)	
File "homes/maiuser/anacondal/envs/PyTorch-1.8/11b/python3.7/site-packages/debugpy/_vendored/pydevd/_pydevd_bundle/pydevd_constants.py", line 379, in cmodule> from _pydev_bundlepydev_aved modules import thread, threading	
File "/homes/maiuser/amaconda3/envs/PyTorch-1.8/11b/python3.7/site-packages/debugy/vendored/pydevd/pydevb_bundle/_pydev_saved_modules.py", line 91, in emodule import code as code; verify shadowed.check(code, ['compile command', 'InteractiveInterpreter'])	
File "/home/mauser/anaconda3/envs/PyTorch-1.8/11b/python3.7/site-packages/debugpy/_vendored/pydevd/_pydev_bundle/_pydev_saved_modules.py", line 75, in check raise DebuggerInitiation(from(esg)	
_pydev_bundlepydev_saved_modules.DebuggerInitializationError: It was not possible to initialize the debugger due to a module name conflict.	
i.e.: the module "code" could not be imported because it is shadowed by: //eme/smillene/nork/transf.code.nv	
Please rename this file/folder so that the original module from the standard library can be imported.	

To resolve this issue, rename the code.py file in the work directory.
 code.py and select.py are typically prone to conflict.

#### 16.3.4 JupyterLab Plug-in Faults

#### 16.3.4.1 What Do I Do If the Git Plug-in Password Is Invalid?

#### Symptom

If the Git plug-in is used in JupyterLab, when a private repository is cloned or a file is pushed, an error occurs.





#### **Possible Causes**

The authorization using a password has been canceled in GitHub. When cloning a private repository or pushing a file, you are required to enter a token in the authorization text box.

#### Solution

Use a token for authorization. When cloning a private repository or pushing a file, enter the token in the authorization text box. For details about how to obtain a token, see **Using the Git Plug-in**.



#### 16.3.5 Save an Image Failures

# 16.3.5.1 What If the Error Message "there are processes in 'D' status, please check process status using'ps -aux' and kill all the 'D' status processes" or "Buildimge,False,Error response from daemon,Cannot pause container xxx" Is Displayed When I Save an Image?

#### Symptom

- When an image is saved in a notebook instance, error "there are processes in 'D' status, please check process status using 'ps -aux' and kill all the 'D' status processes" is displayed.
- When an image is saved in a notebook instance, error "Buildimge,False,Error response from daemon: Cannot pause container xxx" is displayed.

#### **Possible Causes**

If there is a process in the **D** state in the notebook instance, saving an image will fail.

#### Solution

1. Run the **ps** -**aux** on the terminal to check the process.

(PyTorch-	1.8) [ma	i-user	r work	:]\$ps -a	ux					
USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
ma-user	1	0.0	0.0	4532	392	2	Ss	10:47	0:00	/modelarts/authoring/scrip
ma-user	8	0.0	0.0	22028	2196	5	S	10:47	0:00	/bin/bash /modelarts/autho
ma-user	103	0.0	0.2	137000	76276	?	SN	10:47	0:02	/modelarts/authoring/noteb
ma-user	115	0.0	0.0	13444	808	?	S	10:47	0:00	/bin/bash /modelarts/autho
ma-user	116	0.0	0.0	7940	660	?	S	10:47	0:00	tee /home/ma-user/log/note
ma-user	119	1.5	0.3	3800480	13093	36 ?	<b>S1</b>	10:47	0:47	/modelarts/authoring/noteb
ma-user	3134	0.0	0.0	38536	18876	pts/0	SNs	10:58	0:00	/bin/bash -l
ma-user	11045	0.0	0.0	4388	392	pts/0	DN+	11:37	0:00	./d_process
ma-user	11046	0.0	0.0	4388	392	pts/0	SN+	11:37	0:00	./d_process
ma-user	11069	4.2	0.0	22148	2408	pts/1	SNs	11:37	0:00	/bin/bash -l
ma-user	11128	0.0	0.0	7936	656	?	S	11:37	0:00	sleep 3
ma-user	11131	0.0	0.0	37796	1616	pts/1	RN+	11:37	0:00	ps -aux
(PyTorch-	1.8) [ma	-user	r work	:]\$						

2. Run the **kill -9 <pid>** command to stop the process. Then, save the image again.

### 16.3.5.2 What Do I Do If Error "container size %dG is greater than threshold %dG" Is Displayed When I Save an Image?

#### Symptom

When an image is saved in a notebook instance, error "container size %dG is greater than threshold %dG" is displayed.

#### **Possible Causes**

The size of the notebook container exceeded the threshold.

#### Solution

Reduce the container size. The size of a notebook container consists of the image size and the size of the files newly installed in the container. To resolve this issue, use either of the following methods:

- Reduce the size of the files newly installed in the container.
  - a. Delete the files newly installed in a notebook instance. For example, if a large number of files have been downloaded to the notebook instance, delete them. This method applies only to directories other than the / home/ma-user/work and /cache directories. The persistent storage data in home/ma-user/work will not be stored in the created container image, and the temporary files stored in /cache do not consume the container storage space.
  - b. If no file can be deleted or it is unknown which files can be deleted, use the same image to create a notebook instance. When using the new notebook instance, minimize software package installations or file downloads to reduce the container size.
- Reduce the size of the image file.

If you are not sure which packages or files do not need to be installed, use a small image to create a notebook instance and install the required software or files in it. Among all the public images, **mindspore1.7.0-py3.7-ubuntu18.04** takes the minimum size.

### 16.3.5.3 What Do I Do If Error "too many layers in your image" Is Displayed When I Save an Image?

#### Symptom

When an image is saved, error "too many layers in your image" is displayed.

#### **Possible Causes**

The image selected for creating the target notebook instance is a bring-your-own image or a custom image that has been saved for multiple times. No image can be saved for the notebook instance that is created using such an image.

#### Solution

Use a public image or another custom image to create a notebook instance and save the image.

### 16.3.5.4 What Do I Do If Error "The container size (xG) is greater than the threshold (25G)" Is Reported When I Save an Image?

#### Symptom

The error **The container size (30G) is greater than the threshold (25G)** is reported when an image is saved, and the image fails to be created.

#### **Possible Causes**

To save an image, you need to run the **docker commit** command on the agent of a resource cluster node. Administrative data will be uploaded and updated automatically. Each time you run the command, the image becomes larger. After the image is saved for multiple times, its actual size is larger than it shows. If the image is too large, various problems may occur. You can rebuild the original image environment and save the image to solve the problem.

#### Solution

Rebuild the original image environment. You can use a base image with minimized installation and run the dependencies. Clear the installation cache and save the image.

#### 16.3.6 Other Faults

#### 16.3.6.1 Failed to Open the checkpoints Folder in Notebook

**checkpoints** is a keyword in notebook. If a created folder is named **checkpoints**, the folder will not be opened, renamed, or deleted on JupyterLab. To access **checkpoints**, you have two options: either execute the command line in the terminal to load the checkpoint files, or create a folder and transfer the checkpoint data to that folder.



Figure 16-16 Unavailable checkpoints in the JupyterLab navigation pane

#### Procedure

Open the terminal and perform operations using the CLI.

Method 1: Run the **cd checkpoints** command to open the **checkpoints** folder.

Method 2: Create a folder and move the data in the **checkpoints** folder to that folder.

- 1. Run the **mkdir** *xxx* command to create a folder, in which *xxx* is the folder name. Do not use **checkpoints** to name the folder.
- Move the data in the checkpoints folder to the new folder and delete the checkpoints folder in the root directory. mv checkpoints/\* xxx rm -r checkpoints

#### 16.3.6.2 Failed to Use a Purchased Dedicated Resource Pool to Create New-Version Notebook Instances

#### Symptom

A dedicated resource pool that has been purchased cannot be selected for creating a notebook instance, resulting in the creation failure.

A message is displayed, indicating that the development environment has not been initialized in the dedicated resource pool.

#### Possible Causes

A newly purchased dedicated resource pool can be used to create notebook instances only after its development environment is initialized.

#### Solution

Initialize the development environment on the dedicated resource pool page.

Step 1 Go to the Dedicated Resource Pools page and choose More > Set Job Type in the Operation column.

Dedicated Resource	Pools								
Resource Pools	Network								
Create A max	imum of 15 resource	pools can be created. Yo	u can create 7 more.						Enter a na
Name/ID	Status 🏹	Training Job	Inference Service	DevEnviron	Accelerator Driver	Nodes (Availa	Obtained At JF	Description	Operation
pool-os-kwx112 pool-os-kwx112	Running	<ul> <li>Enabled</li> </ul>	Enabled	Enable Failed		1/0/1	Oct 28, 2022 15:09:34 GMT+	🖉	Adjust Capacity   More 🔺
pool-ostest-d91 pool-ostest-d91	Running	Enabled	Enabled	Enabled	c81-21.0.2 💿 Running	1/0/1 (?)	Oct 18, 2022 20:17:18 GMT+		Adjus Set Job Type

**Step 2** In the **Set Job Type** dialog box, select **DevEnviron** and click **OK**. Then, the development environment is being initialized. After its status changes to **Running**, the newly purchased dedicated resource pool can be used to create notebook instances.

Figure 16-17 Setting job type to DevEnviron



Figure 16-18 Initializing the development environment



----End

### 16.3.6.3 Error Message "Permission denied" Is Displayed When the tensorboard Command Is Used to Open a Log File in a Notebook Instance

#### Symptom

When the **tensorboard** --logdir ./ command is executed on the terminal of a notebook instance, the error message "[Errno 13] Permission denied..." is displayed.

(PyTorch-1.8) [ma-user work]\$tensorboardlogdir ./
/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/requests/initpy:104: RequestsDependencyWarning: urllib3 (1.26.12) or chardet (5.1.0)/charset normalizer
ed version!
RequestsDependencyWarning)
TensorFlow installation not found - running with reduced feature set.
Serving TensorBoard on localhost; to expose to the network, use a proxy or passbind all
TensorBoard 2.1.1 at http://localhost:6006/ (Press CTRL+C to quit)
Exception in thread Reloader:
Traceback (most recent call last):
File "home/wa-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/threading.py", line 926, in _bootstrap_inner self.run()
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/threading.py", line 870, in run
self. target(*self. args, **self. kwargs)
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/tensorboard/backend/application.py", line 586, in _reload
multiplexer.AddRunsFromDirectory(path, name)
File "home/wa-user/anaconda3/emvs/hyTorch-1.8/lib/python3.7/site-packages/tensorboard/backend/event_processing/plugin_event_multiplexer.py", line 199, in AddRunsFromDirectory for subdir in io warpopre.fottoggitSubdirectories(path):
File "home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/tensorboard/backend/event_processing/io_wrapper.py", line 200, in <genexpr> subdir</genexpr>
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/tensorboard/backend/event processing/io wrapper.py", line 155, in ListRecursivelyViaWalking
for dir path, _, filenames in tf.io.gfile.walk(top, topdown=True):
File "home/ma-user/anaconda3/emvs/byTorch-1.8/lib/python3.//site-packages/tensorboard/compat/tensorflow_stub/io/gfile.py", line 687, in walk for subitem in walk(sining subdir, ropdoma, omergo-energor):
File "home/ma-user/anaconda3/emvs/byTorch-1.8/lib/python3.//site-packages/tensorboard/compat/tensorflow_stub/io/gfile.py", line 687, in walk for sublire in walk'sioning subdir. rondown, omergenoemerge):
File "/home/wa-user/anacondal/zervs/PyTorch-1.8/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/io/gfile.py", line 687, in walk for subitem in walk(ioned subite; tondown.pnerprepsencerop):
[Previous line repeated 1 more time]
File "/home/ma-user/anaconda3/envs/PvTorch-1.8/lib/pvthon3.7/site-packages/tensorboard/compat/tensorflow stub/io/gfile.pv", line 664, in walk
listing = listdir(top)
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/tensorboard/compat/tensorflow stub/io/gfile.py", line 626, in listdir
return get filesystem(dirname).listdir(dirname)
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/io/gfile.py", line 184, in listdir
<pre>entries = os.listdir(compat.as_str_any(dirname))</pre>
PermissionError: [Errno 13] Permission denied: './.lsp symlink/etc/ssl/private'

#### **Possible Causes**

The current directory contains files on which you do not have permission.

#### Solution

Create a folder (for example, **tb\_logs**), place the TensorBoard log file (for example, **tb.events**) in this folder, and run the tensorboard command. The following is an example command:

mkdir -p ./tb\_logs mv tb.events ./tb\_logs tensorboard --logdir ./tb\_logs

(PyTorch-1.8) [ma-user work]\$
(PyTorch-1.8) [ma-user work]\$mkdir -p tb_logs
(PyTorch-1.8) [ma-user work]\$mv tb.events ./tb_logs
(PyTorch-1.8) [ma-user work]\$tensorboardlogdir ./tb_logs
/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/requests/initpy:104: RequestsDependencyWarning: urllib3 (1.26.12) or chardet (5.2.0)/charset_normalizer (2.0.12) doesn't match a support
ted version!
RequestsDependencyWarning)
Tensorflow installation not found - running with reduced feature set.
Serving TensorBoard on localhost; to expose to the network, use a proxy or passbind_all
TensorBoard 2.1.1 at http://localhost:6006/ (Press CTRL+C to quit)

#### 16.4 Training Jobs

#### 16.4.1 OBS Operation Issues

#### 16.4.1.1 Error in File Reading

#### Symptom

- How to read the **json** and **npy** files when creating a training job.
- How the training job uses the cv2 library to read files.
- How to use the torch package in the MXNet environment.
- The following error occurs when the training job reads the file: NotFoundError (see above for traceback): Unsucessful TensorSliceReader constructor: Failed to find any matching files for xxx://xxx

#### **Possible Cause**

In ModelArts, user's data is stored in OBS buckets, but training jobs are running in containers. Therefore, users cannot access files in OBS buckets by accessing local paths.

#### Solution

If an error occurs when you read a file, you can use MoXing to copy data to a container and then access the data in the container. For details, see **1**.

You can also read files based on the file type. For details, see **Reading .json files**, **Reading .npy files**, and **Using the cv2 library to read files**, and **Using the torch package in the MXNet environment**.

- If an error occurs when you read a file, you can use MoXing to copy data to a container and then access the data in the container as follows: import moxing as mox mox.file.make\_dirs('/cache/data\_url') mox.file.copy\_parallel('obs://bucket-name/data\_url', '/cache/data\_url')
- 2. To **read**.json files, run the following code: json.loads(mox.file.read(json\_path, binary=True))
- 3. To use numpy.load to read .npy files, run the following code:
  - Using the MoXing API to read files from OBS np.load(mox.file.read(\_SAMPLE\_PATHS['rgb'], binary=True))
  - Using the file module of MoXing to read and write OBS files with mox.file.File(\_SAMPLE\_PATHS['rgb'], 'rb') as f: np.load(f)
- 4. To **use the cv2 library to read files**, run the following code: cv2.imdecode(np.fromstring(mox.file.read(img\_path), np.uint8), 1)
- To use the torch package in the MXNet environment, run the following code: import os

os.sysytem('pip install torch')

### 16.4.1.2 Error Message Is Displayed Repeatedly When a TensorFlow-1.8 Job Is Connected to OBS

#### Symptom

After a training job is started based on TensorFlow-1.8 and the **tf.gfile** module is used to connect to OBS in code, the following log information is frequently printed:

Connection has been released. Continuing. Found secret key

#### **Possible Cause**

This problem occurs in TensorFlow-1.8. This log is of the INFO level and is not error information. You can set an environment variable to shield logs of the INFO level. The environment variable must be set before the **import tensorflow** or **import moxing** command is executed.

#### Solution

Set the environment variable **TF\_CPP\_MIN\_LOG\_LEVEL** in code to shield logs of the INFO level. Detailed operations are as follows:

import os

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

import tensorflow as tf import moxing.tensorflow as mox

The mapping between TF\_CPP\_MIN\_LOG\_LEVEL and log levels is as follows:

import os os.environ["TF\_CPP\_MIN\_LOG\_LEVEL"]='1' # Default level of logs to be displayed. All information is displayed. os.environ["TF\_CPP\_MIN\_LOG\_LEVEL"]='2' # Only warning and error information is displayed. os.environ["TF\_CPP\_MIN\_LOG\_LEVEL"]='3' # Only error information is displayed.

### 16.4.1.3 TensorFlow Stops Writing TensorBoard to OBS When the Size of Written Data Reaches 5 GB

#### Symptom

The following error message is displayed for a ModelArts training job:

Encountered Unknown Error EntityTooLarge Your proposed upload exceeds the maximum allowed object size.: If the signature check failed. This could be because of a time skew. Attempting to adjust the signer

#### **Possible Cause**

The size of files to be uploaded at a time is limited to 5 GB in OBS. TensorFlow may save the summary file in local cache. Therefore, when flush is triggered each time, the summary file overwrites the original file on OBS. If the size of the file exceeds 5 GB, the file stops being written.

#### Solution

If this problem occurs during the running of a training job, use the following method for troubleshooting.

 You are advised to use the following local cache method: import moxing.tensorflow as mox mox.cache()

### 16.4.1.4 Error "Unable to connect to endpoint" Error Occurs When a Model Is Saved

#### Symptom

An error occurs in the log when a model is saved in a training job. The error details are as follows:

InternalError (see above for traceback): : Unable to connect to endpoint

#### **Possible Cause**

When OBS connections are unstable, the following error may occur: **Unable to connect to endpoint** 

#### Solution

Add code to solve the problem of unstable OBS connections. You can add the following code at the beginning of the existing code so that TensorFlow can read and write ckpt and summary information in local cache mode:

import moxing.tensorflow as mox

mox.cache()

### 16.4.1.5 Error Message "BrokenPipeError: Broken pipe" Displayed When OBS Data Is Copied

#### Symptom

The error message is displayed when MoXing is used to copy data for a training job.

#### Figure 16-19 Error log

readable=readable)
File */home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py*, line 358, in _make_put_request chunkedMode, methodName=methodName, readable=readable)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 390, in _make_request_with_retry
rials of the second and the second a
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 436, in _make_request_internal conn = selfsend_request(connect_server, method, path, header_config, entity, port, scheme, redirect, chunkedMode)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/client.py", line 586, in _send_request entity(util.conn delegate(conn))
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/util.py", line 250, in entity conn.send(chunk)
File "/home/work/anaconda/lib/python3.6/site-packages/moxing/framework/file/src/obs/util.py", line 154, in send self.conn.send(data)
File "/home/work/anaconda/lib/python3.6/http/client.py", line 986, in send self.sock.sendall(data)
File "/home/work/anaconda/lib/python3.6/ssl.py", line 972, in sendall v = self.send(byte_view[count:])
File "/home/work/anaconda/lib/python3.6/ssl.py", line 941, in send return self. sslobj.write(data)
File "/home/work/anaconda/lib/python3.6/ssl.py", line 642, in write
Tetarn sen, ssioojawrite(data)

BrokenPipeError: [Errno 32] Broken pipe

#### **Possible Causes**

The possible causes are as follows:

- In a large-scale distributed job, multiple nodes are concurrently copying files in the same bucket, leading to traffic control in the OBS bucket.
- There is a large number of OBS client connections. During the polling between processes or threads, an OBS client connection timed out if the server does not respond to it within 30 seconds. As a result, the server released the connection.

#### Solution

 If the issue is caused by traffic control, the error code shown in the following figure is displayed. In this case, submit a service ticket. For details about OBS error codes, see Python > Troubleshooting > OBS Server-Side Error Codes in Object Storage Service SDK Reference.

#### Figure 16-20 Error log

[ModelAi	rts Service Log]2021-01-21 11:35:42,178 - file_io.py[line:652] - ERROR: Fail func= <bound <moxing.fram<br="" method="" obsclient.getobjectmetadata="" of="">args=('bucket-816', 'AIRAW_AJ/c00454567/TeleQtj/23_zyl_J_quad_TeleN</bound>
[ModelAr	rts Service Log]2021-01-21 11:35:42,178 - file_io.py[line:658] - ERROR: stat:503 errorCode:None
	errorMessage:None reason:Service Unavailable request-id:000001772302B34C9019B2408F9FF1B2 retry:0

 If the issue is caused by the large number of client connections, especially for files larger than 5 GB, OBS APIs cannot be directly called. In this case, use multiple threads to copy data. The timeout duration set on the OBS server is 30s. Run the following commands to reduce the number of processes: # Configure the number of processes. os.environ['MOX\_FILE\_LARGE\_FILE\_TASK\_NUM']=1 import moxing as mox

# Copy files. mox.file.copy\_parallel(src\_url=your\_src\_dir, dst\_url=your\_target\_dir, threads=0, is\_processing=False)

#### **NOTE**

When creating a training job, you can use the environment variable \_PARTIAL\_MAXIMUM\_SIZE to configure the threshold (in bytes) for downloading large files in multiple parts. If the size of a file exceeds the threshold, the file will be downloaded in multiple parts concurrently.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

#### 16.4.1.6 Error Message "ValueError: Invalid endpoint: obs.xxxx.com" Displayed in Logs

#### Symptom

When TensorBoard is used to directly write data in an OBS path for a training job, an error is displayed.

#### Figure 16-21 Error log

Traceback (most recent call last):

File "/home/work/anaconda/lib/python3.6/threading.py", line 916, in \_bootstrap\_inner self.run()

File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/event\_file\_writer.py", line 219, in run self. record writer.flush()

File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/event\_file\_writer.py", line 69, in flush self.\_py\_recordio\_writer.flush()

File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/record\_writer.py", line 187, in flush self.\_writer.flush()

File "/home/work/anaconda/lib/python3.6/site-packages/tensorboardX/record\_writer.py", line 89, in flush s3 = boto3.client('s3', endpoint\_url=os.environ.get('S3\_ENDPOINT'))

File "/home/work/anaconda/lib/python3.6/site-packages/boto3/\_init\_.py", line 91, in client return\_get\_default\_session).client(\*args, \*\*kwargs)

File "/home/work/anaconda/lib/python3.6/site-packages/boto3/session.py", line 263, in client

aws\_session\_token=aws\_session\_token, config=config) File "/home/work/anaconda/lib/python3.6/site-packages/botocore/session.py", line 835, in create\_client client config=config, api version=api version)

File "/home/work/anaconda/lib/python3.6/site-packages/botocore/client.py", line 85, in create\_client

verify, credentials, scoped\_config, client\_config, endpoint\_bridge) File "/home/work/anaconda/lib/python3.6/site-packages/botocore/client.py", line 287, in \_get\_client\_args

verify, credentials, scoped\_config, client\_config, endpoint\_bridge) File "/home/work/anaconda/lib/python3.6/site-packages/botocore/args.py", line 107, in get\_client\_args

client\_cert=new\_config.client\_cert) File "/home/work/anaconda/lib/python3.6/site-packages/botocore/endpoint.py", line 261, in create\_endpoint raise ValueError("Invalid endpoint: %s" % endpoint\_url)

ValueError: Invalid endpoint: obs.myhuaweicloud.com

#### **Possible Causes**

It is unstable to use TensorBoard to directly write data in OBS.

#### Solution

Locally write data and then copy it back to OBS.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

### 16.4.1.7 Error Message "errorMessage:The specified key does not exist" Displayed in Logs

#### Symptom

When MoXing is used to access an OBS path, the following error is displayed: ERROR:root: stat:404 errorCode:NoSuchKey errorMessage:The specified key does not exist.

#### **Possible Causes**

The possible causes are as follows:

The object is unavailable in the bucket. For details about OBS error codes, see **Python > Troubleshooting > OBS Server-Side Error Codes** in *Object Storage Service SDK Reference*.

#### Solution

- 1. Check whether the OBS path and object are in correct format.
- 2. Use the local PyCharm to remotely access notebook for debugging.

#### **Summary and Suggestions**

Before creating a training job, use a ModelArts development environment to debug training code. This maximally eliminates errors in code migration.

#### **16.4.2 In-Cloud Migration Adaptation Issues**

#### 16.4.2.1 Failed to Import a Module

#### Symptom

The following error occurs in the log when a module is imported to a ModelArts training job:

Traceback (most recent call last):File "project\_dir/main.py", line 1, in <module>from module\_dir import module\_file ImportError: No module named module dir

ImportError: No module named xxx

#### **Possible Cause**

• When a training job is imported to the module, the previous two error messages are displayed in the log. The possible causes are as follows:

Before running code locally, you need to add **project\_dir** to **PYTHONPATH** or install **project\_dir** in **site-package**. However, on ModelArts, you can add **project\_dir** to **sys.path** to solve this problem.

Use **from module\_dir import module\_file** to import a package. The code structure is as follows:

project\_dir |- main.py |- module\_dir | |- \_\_init\_\_.py | |- module\_file.py

When a training job is imported to the module, the error message
 "ImportError: No module named xxx" is displayed in the log. It can be
 determined that the environment does not contain the Python package on
 which the user depends.

#### Solution

- When a training job is imported to the module, the previous two error messages are displayed in the log. The solution is as follows:
  - a. Ensure that the imported module contains \_\_init\_\_.py used for creating module\_dir. Possible Cause provides the code structure.
  - b. Because the location of project\_dir in the container is unknown, use an absolute path by adding project\_dir to sys.path in file main.py, and import the following information: import os import sys # \_file\_ is the absolute path of the main.py script. # os.path.dirname(\_file\_) is the parent directory of main.py, that is, the absolute path of project\_dir. current\_path = os.path.dirname(\_file\_) sys.path.append(current\_path) # Import other modules after sys.path.append is executed. from module\_dir import module\_file
- When a training job is imported to the module, the error message
   "ImportError: No module named xxx" is displayed in the log. Add the
   following code to install the dependency package:
   import os
   os.system('pip install xxx')

#### 16.4.2.2 Error Message "No module named .\*" Displayed in Training Job Logs

Perform the following operations to locate the fault:

- 1. Checking Whether the Dependency Package Is Available
- 2. Checking Whether the Dependency Package Path Can Be Detected
- 3. Checking Whether the Selected Resource Flavor Is Correct
- 4. Summary and Suggestions

#### **Checking Whether the Dependency Package Is Available**

If the dependency package is unavailable, use either of the following methods to install it:

• Method 1 (recommended): When you create an algorithm, place the required file or installation package in the code directory.

The required file varies depending on the dependency package type.

- If the dependency package is an open-source installation package

Create a file named **pip-requirements.txt** in the code directory, and specify the dependency package name and version in the format of *Package name*== *Version* in the file.

For example, the OBS path specified by **Code Directory** contains model files and the **pip-requirements.txt** file. The code directory structure is as follows:

---OBS path to the model boot file

|---model.py # Model boot file

|---pip-requirements.txt # Customized configuration file, which specifies the name and version of the dependency package

The following shows the content of the **pip-requirements.txt** file:

```
alembic==0.8.6
bleach==1.4.3
click==6.6
```

#### If the dependency package is a WHL package

If the training backend does not support the download of open-source installation packages or the use of custom WHL packages, the system cannot automatically download and install the package. In this case, place the WHL package in the code directory, create a file named **pip-requirements.txt**, and specify the name of the WHL package in the file. The dependency package must be in WHL format.

For example, the OBS path specified by **Code Directory** contains model files, the WHL file, and the **pip-requirements.txt** file. The code directory structure is as follows:

---OBS path to the model boot file

|---model.py# Model boot file|---XXX.whl# Dependency package. If multiple dependencies are required, placeall of them here.

|---pip-requirements.txt # Customized configuration file, which specifies the name of the dependency package

The following shows the content of the **pip-requirements.txt** file:

numpy-1.15.4-cp36-cp36m-manylinux1\_x86\_64.whl tensorflow-1.8.0-cp36-cp36m-manylinux1\_x86\_64.whl

• Method 2: Add the following code to the boot file to install the dependency package:

import os os.system('pip install xxx')

In method 1, the dependency package can be downloaded and installed before the training job is started. In method 2, the dependency package is downloaded and installed during the running of the boot file.

#### Checking Whether the Dependency Package Path Can Be Detected

Before executing code locally, add **project\_dir** to **PYTHONPATH** or install **project\_dir** in **site-package**. ModelArts enables you to add **project\_dir** to **sys.path** to resolve this issue.

Run **from module\_dir import module\_file** to import a package. The code structure is as follows:

project\_dir |- main.py |- module\_dir | |- \_\_init\_\_.py | |- module\_file.py

#### **Checking Whether the Selected Resource Flavor Is Correct**

Error message "No module named npu\_bridge.npu\_init" is displayed for a training job.

from npu\_bridge.npu\_init import \* ImportError: No module named npu\_bridge.npu\_init

Check whether the flavor used by the training job supports NPUs. The possible cause is that the job selected a non-NPU flavor, for example, a GPU flavor. As a result, an error occurs when NPUs are used.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

#### 16.4.2.3 Failed to Install a Third-Party Package

#### Symptom

- How to install custom library functions for ModelArts, for example, apex.
- The following error occurs when a third-party package is installed in the ModelArts training environment: xxx.whl is not a supported wheel on this platform

#### **Possible Cause**

Error **xxx.whl is not a supported wheel on this platform** occurs, because the format of the name of the installed file is not supported. For details about the solution, see **2**.

#### Solution

#### 1. Installing the third-party package

- a. For an existing package in **pip**, run the following code to install it: import os os.system('pip install xxx')
- b. For a package that do not exist in **pip**, for example, **apex**, use the following method to upload the installation package to an OBS bucket. In this example, the installation package has been uploaded to **obs://**cnnorth4-test/codes/mox\_benchmarks/apex-master/. Add the following code to the boot file to install the package:

```
try:

import apex

except Exception:

import os

import moxing as mox

mox.file.copy_parallel('obs://cnnorth4-test/codes/mox_benchmarks/apex-master/', '/cache/

apex-master')

os.system('pip --default-timeout=100 install -v --no-cache-dir --global-option="--cpp_ext" --

global-option="--cuda_ext" /cache/apex-master')
```

#### 2. Installation error

If the **xxx.whl** file fails to be installed, perform the following steps to solve the problem:

a. If the **xxx.whl** file fails to be installed, add the following code to the boot file to check the file name and version supported by the **pip** command.

print(pip.pep425tags.get\_supported())

The supported file names and versions are as follows:

[('cp36', 'cp36m', 'manylinux1\_x86\_64'), ('cp36', 'cp36m', 'linux\_x86\_64'), ('cp36', 'abi3', 'manylinux1\_x86\_64'), ('cp36', 'abi3', 'linux\_x86\_64'), ('cp36', 'none', 'manylinux1\_x86\_64'), ('cp35', 'abi3', 'manylinux1\_x86\_64'), ('cp35', 'abi3', 'linux\_x86\_64'), ('cp34', 'abi3', 'linux\_x86\_64'), ('cp33', 'abi3', 'manylinux1\_x86\_64'), ('cp32', 'abi3', 'linux\_x86\_64'), ('cp32', 'abi3', 'manylinux1\_x86\_64'), ('cp32', 'abi3', 'none', 'manylinux1\_x86\_64'), ('cp35', 'abi3', 'linux\_x86\_64'), ('cp35', 'linux

'none', 'any'), ('py34', 'none', 'any'), ('py33', 'none', 'any'), ('py32', 'none', 'any'), ('py31', 'none', 'any')]

b. Change faiss\_gpu-1.5.3-cp36-cp36m-manylinux2010\_x86\_64.whl to faiss\_gpu-1.5.3-cp36-cp36m-manylinux1\_x86\_64.whl, and run the following code to install the package: import moxing as mox import os

mox.file.copy('obs://wolfros-net/zp/AI/code/faiss\_gpu-1.5.3-cp36-cp36mmanylinux2010\_x86\_64.whl','/cache/faiss\_gpu-1.5.3-cp36-cp36m-manylinux1\_x86\_64.whl') os.system('pip install /cache/faiss\_gpu-1.5.3-cp36-cp36m-manylinux1\_x86\_64.whl')

#### 16.4.2.4 Failed to Download the Code Directory

#### Symptom

The code directory fails to be downloaded during training job running, and the following error message is displayed. See **Figure 16-22**.

ERROR: modelarts-downloader.py: Get object key failed: 'Contents'

#### Figure 16-22 Failure of getting content



#### Possible Cause

The code directory specified during training job creation does not exist. As a result, the training fails.

#### Solution

Check whether the code directory specified during training job creation, that is, the OBS bucket path, is correct based on the error cause. There are two methods to check whether it exists.

- Log in to the OBS console using the current account, and search for the OBS buckets, folders, and files in the path to check whether the code directory exists.
- Using APIs to check whether the directory exists: Run the following command in code to check whether the directory exists: import moxing as mox mox.file.exists('obs://obs-test/ModelArts/examples/')

#### 16.4.2.5 Error Message "No such file or directory" Displayed in Training Job Logs

#### Symptom

If a training job failed, error message "No such file or directory" is displayed in logs.

If a training input path is unreachable, error message "No such file or directory" is displayed.

If a training boot file is unavailable, error message "No such file or directory" is displayed.

Figure 16-23 Example log for an unavailable training boot file

	Plationm=Modelarits-Service	
1	3 [2022-08-03T19:51:29+08:00][ModelArts Service Log][task] hang-detect	
14	4 [2022-08-03T19:51:29+08:00][ModelArts Service Log][task] toolkit_hang_detect_pid = 52	
1	5 python: can't open file '/home/ma-user/modelarts/user-job-dir/nlp_classifier_train_daodian_v2_dist.py': [Errno 2]	No such file or directory
1	6 [GIN] 2022/08/03 - 19:51:29   200   44.278µs   127.0.0.1   POST "/scc"	A 60-
1	7 [GIN] 2022/08/03 - 19:51:29   200   25.461µs   127.0.0.1   POST "/scc"	
1	8 [GIN] 2022/08/03 - 19:51:29   200   9 39.358µs   127.0.0.1   POST "/scc"	

#### Possible Causes

- If the training input path is unreachable, the path is incorrect. Perform the following operations to locate the fault:
  - a. Checking Whether the Affected Path Is an OBS Path
  - b. Checking Whether the Affected Path Is Available
- If the training boot file is unavailable, the path to the training job boot command is incorrect. Rectify the fault by referring to Checking the File Boot Path of a Training Job Created Using a Custom Image.
- Multiple processes or workers read and write the same file. If SFS is used, check whether multiple nodes concurrently write the same file. Analyze the code and check whether multiple processes write the same file. It is a good practice to prevent multiple processes or nodes from concurrently reading and writing the same file.

#### Checking Whether the Affected Path Is an OBS Path

When using ModelArts, store data in an OBS bucket. However, the OBS path cannot be used to read data during the execution of the training code.

The reason is as follows:

After a training job is created, the training performance is poor if the running container is directly connected to OBS. To prevent this issue, the system automatically downloads the training data to the local path of the running container. Therefore, an error occurs if an OBS path is used in training code. For example, if the OBS path to the training code is **obs://bucket-A/training/**, the training code will be automatically downloaded to **\${MA\_JOB\_DIR}/training**/.

For example, the OBS path to the training code is **obs://bucket-A/XXX/{training-project}**, where **{training-project}** is the name of the folder where the training code is stored. During training, the system will automatically download the data from OBS **{training-project}** to the local path of the training container (**\$MA\_JOB\_DIR/{training-project}**/).

If the affected path is to the training data, perform the following operations to resolve this issue (see "Parsing Input and Output Paths" for details):

- 1. When creating an algorithm, set the code path parameter, which defaults to **data\_url**, in the input path mapping configuration.
- 2. Add a hyperparameter, which defaults to **data\_url**, to the training code. Use **data\_url** as the local path for inputting the training data.

#### Checking Whether the Affected Path Is Available

The code developed locally needs to be uploaded to the ModelArts backend. It is likely to incorrectly set the path to a dependency file in training code.

You are suggested to use the following general solution to obtain the absolute path to a dependency file through the OS API.

Example:

project_root	# Root directory for code
BootfileDirectory	# Directory where the boot file is located
bootfile.py	# Boot file
otherfileDirectory	# Directory where other dependency files are located
otherfile.py	# Other dependency files

Do as follows to obtain the path to a dependency file, **otherfile\_path** in this example, in the boot file:

```
import os
```

current\_path = os.path.dirname(os.path.realpath(\_\_file\_\_)) # Directory where the boot file is located project\_root = os.path.dirname(current\_path) # Root directory of the project, which is the code directory set on the ModelArts training console otherfile\_path = os.path.join(project\_root, "otherfileDirectory", "otherfile.py")

#### Checking the File Boot Path of a Training Job Created Using a Custom Image

Take OBS path **obs://obs-bucket/training-test/demo-code** as an example. The training code in this path will be automatically downloaded to **\${MA\_JOB\_DIR}/ demo-code** in the training container, where **demo-code** is the last-level directory of the OBS path and can be customized.

If you use a custom image to create a training job, the system will automatically run the image boot command after the code directory is downloaded. The boot command must comply with the following rules:

- If the training startup script is a .py file, **train.py** for example, the boot command can be **python \${MA\_JOB\_DIR}/demo-code/train.py**.
- If the training startup script is an .sh file, **main.sh** for example, the boot command can be **bash \${MA\_JOB\_DIR}/demo-code/main.sh**,

where **demo-code** is the last-level directory of the OBS path and can be customized.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

#### 16.4.2.6 Failed to Find the .so File During Training

#### Symptom

During the execution of a ModelArts training job, the following error message is displayed in the log and the training failed:

libcudart.so.9.0 cannot open shared object file no such file or directory

#### **Possible Cause**

The CUDA version of the .so file generated during compilation is different from that of the training job.

#### Solution

If the CUDA version in the compilation environment is different from that in the training environment, an error will occur when a training job runs. For example, this error occurs if the .so file generated in the TensorFlow 1.13 development environment of CUDA version 10 is used in the TensorFlow 1.12 training environment of CUDA version 9.0.

To resolve this issue, perform the following operations:

- Add the following command before executing a training job to check whether the .so file is available. If the .so file is available, go to 2. Otherwise, go to 3. import os; os.system(find /usr -name \*libcudart.so\*);
- 2. Configure the environment variable **LD\_LIBRARY\_PATH** and issue the training job again.

For example, if the path for storing the .so file is **/use/local/cuda/lib64**, configure **LD\_LIBRARY\_PATH** as follows: export LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64

- 3. Run the following command to check whether the CUDA version of the training environment supports the .so file: os.system("cat /usr/local/cuda/version.txt")
  - a. If so, import an external .so file (download it from the browser) and configure LD\_LIBRARY\_PATH in 2.
  - b. If not, replace the engine and issue the training job again. Alternatively, use a custom image to create a job. For details, see Using a Custom Image to Train Models.

### 16.4.2.7 ModelArts Training Job Failed to Parse Parameters and an Error Is Displayed in the Log

#### Symptom

The ModelArts training job failed to parse parameters, and the following error occurs:

error: unrecognized arguments: --data\_url=xxx://xxx/xxx error: unrecognized arguments: --init\_method=tcp://job absl.flags.\_exceptions.UnrecognizedFlagError:Unknown command line flag 'task\_index'

#### **Possible Cause**

- The parameters are not defined.
- In the training environment, the system may input parameters that are not defined in the Python script. As a result, the parameters cannot be parsed, and an error is displayed in the log.
## Solution

- 1. Define the parameters. The following is a code sample for reference: parser.add\_argument('--init\_method', default='tcp://xxx',help="init-method")
- 2. Replace args = parser.parse\_args() with args, unparsed = parser.parse\_known\_args(). The following is a code sample: import argparse parser = argparse.ArgumentParser() parser.add\_argument('--data\_url', type=str, default=None, help='obs path of dataset') args, unparsed = parser.parse\_known\_args()

## 16.4.2.8 Training Output Path Is Used by Another Job

#### Symptom

The following error message is displayed when a training job is created: Operation failed. Other running job contain train\_url: /bucket-20181114/code\_hxm/

## **Possible Cause**

According to the error information, the same training output path is used by another job when a training job is created.

## Solution

A training output path can be used by only one job in the running, queuing, or initializing state.

If this error occurs, check and re-set the training output path of the training job to avoid the job creation failure.

## 16.4.2.9 Error Message "RuntimeError: std::exception" Displayed for a PyTorch 1.0 Engine

#### Symptom

When a PyTorch 1.0 image is used, the following error message is displayed: "RuntimeError: std::exception"

#### **Possible Causes**

The soft link of libmkldnn in the PyTorch 1.0 image conflicts with that of the native Torch. For details, see **conv1d fails in PyTorch 1.0**.

#### Solution

- This issue is caused by library conflict in the environment. To resolve this issue, add the following code at the very beginning of the boot script: import os os.system("rm /home/work/anaconda3/lib/libmkldnn.so") os.system("rm /home/work/anaconda3/lib/libmkldnn.so.0")
- 2. Use the local PyCharm to remotely access notebook for debugging.

## Summary and Suggestions

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.2.10 Error Message "retCode=0x91, [the model stream execute failed]" Displayed in MindSpore Logs

### Symptom

When MindSpore is used for training, the following error message is displayed: [ERROR] RUNTIME(3002)model execute error, retCode=0x91, [the model stream execute failed]

#### **Possible Causes**

The speed of reading data cannot keep up with the model iteration speed.

#### Solution

- 1. Reduce shuffle operations during preprocessing. dataset = dataset.shuffle(buffer\_size=x)
- 2. Disable data preprocessing, which may affect system performance. NPURunConfig(enable\_data\_pre\_proc=Flase)

### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.2.11 Error Occurred When Pandas Reads Data from an OBS File If MoXing Is Used to Adapt to an OBS Path

#### Symptom

If MoXing is used to adapt to an OBS path, an error occurs when pandas of a later version reads data from an OBS file.

- 1. 'can't decode byte xxx in position xxx'
- 2. 'OSError:File isn't open for writing'

#### **Possible Causes**

MoXing does not support Pandas of a later version.

#### Solution

 After the OBS path is adapted, change the file access mode from r to rb and change the \_write\_check\_passed value in mox.file.File to True, as shown in the following is sample code: import pandas as pd import moxing as mox

mox.file.shift('os', 'mox') # Replace the open operation of the operating system with the operation for adapting the **mox.file.File** to the OBS path.

```
param = {'encoding': 'utf-8'}
path = 'xxx.csv'
with open(path, 'rb') as f:
    f._wirte_check_passed = True
    df = pd.read_csv(ff, **param)
```

2. Use the local PyCharm to remotely access notebook for debugging.

### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.2.12 Error Message "Please upgrade numpy to >= xxx to use this pandas version" Displayed in Logs

#### Symptom

Dependency conflicts occur when other packages are installed. There are special requirements on the NumPy library. However, NumPy cannot be uninstalled. The error message similar to the following is displayed: your numpy version is 1.14.5.Please upgrade numpy to >= 1.15.4 to use this pandas version

#### **Possible Causes**

Both Conda and pip packages are installed. Some packages cannot be uninstalled.

#### Solution

Perform the following operations to resolve this issue:

- 1. Uninstall the components that can be uninstalled in NumPy.
- 2. Delete the NumPy folder in the **site-packages** directory.
- Install the required version again. import os os.system("pip uninstall -y numpy") os.system('rm -rf /home/work/anaconda/lib/python3.6/site-packages/numpy/") os.system("pip install numpy==1.15.4")

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.2.13 Reinstalled CUDA Version Does Not Match the One in the Target Image

#### Symptom

An error occurs after the engine version is reinstalled or a new CUDA package is compiled based on the existing image. 1. "RuntimeError: cuda runtime error (11) : invalid argument at /pytorch/aten/src/THC/ THCCachingHostAllocator.cpp:278" 2. "libcudart.so.9.0 cannot open shared object file no such file or directory" 3. "Make sure the device specification refers to a valid device. The requested device appears to be a GPU,but CUDA is not enabled"

#### Possible Causes

The possible cause is as follows:

The CUDA version of the newly installed package does not match the CUDA version in the image.

#### Solution

Use the local PyCharm to remotely access notebook for debugging and installation.

- 1. Remotely log in to the selected image and run **nvcc** -**V** to obtain the CUDA version of the image.
- 2. Reinstall Torch. Ensure that the version matches the one obtained in the previous step.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.2.14 Error ModelArts.2763 Occurred During Training Job Creation

#### Symptom

When a training job is created, error code ModelArts.2763 is displayed, indicating that the selected instance is invalid.

#### **Possible Causes**

The selected training flavor does not match the algorithm.

For example, the algorithm supports GPUs, but Ascend flavor is selected for creating the training job.

### Solution

- 1. Check the training resource flavor configured in the algorithm code.
- 2. Check whether the resource flavor selected during training job creation is correct. If not, create a training job with the correct resource flavor.

## 16.4.2.15 Error Message "AttributeError: module '\*\*\*' has no attribute '\*\*\*'" Displayed Training Job Logs

#### Symptom

Error message "AttributeError: module '\*\*\*' has no attribute '\*\*\*'" is displayed in the logs of a training job, for example, "AttributeError: module 'torch' has no attribute 'concat'".

### **Possible Causes**

The possible causes are as follows:

- The Python package is incorrectly used. There is no required variable or method in the Python package.
- The Python package version in the third-party pip source has been updated. As a result, the version of the Python package installed in the training job may also change. If a training job ran properly originally, but this issue occurs in the training job later, consider this cause.

#### Solution

- Use notebook for debugging.
- Specify a version for installation, for example, **pip install xxx==** *1.x.x.*
- The third-party pip source may be updated at any time. To prevent this issue from occurring, create a custom image. For details, see Using a Custom Image.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.2.16 System Container Exits Unexpectedly

#### Symptom

After a training job is created, the system container exits unexpectedly.

#### Figure 16-24 Error logs

34	[ModelArts Service Log]2022-10-11 19:17:35,178 - file_io.py[line:728] - WARNING: Retry=4, Wait=3.2, Timestamp=1665487055.178172, Function=getObject, args=
	('modelarts-cn-north-4-test', 'modelarts/code-test'), kwargs={loadStreamInMemory:False, cache:False, }
35	[ModelArts Service Log]2022-10-11 19:17:38,405 - file_io.py[line:728] - WARNING: Retry=3, Wait=6.4, Timestamp=1665487058.4054542, Function=getObject, args=
	('modelarts-cn-north-4-test', 'modelarts/code-test'), kwargs={loadStreamInMemory:False, cache:False, }
36	[ModelArts Service Log]2022-10-11 19:17:44,832 - file_io.py[line:728] - WARNING: Retry=2, Wait=12.8, Timestamp=1665487064.8322, Function=getObject, args=
	('modelarts-cn-north-4-test', 'modelarts/code-test'), kwargs={loadStreamInMemory:False, cache:False, }
37	[ModelArts Service Log]2022-10-11 19:17:57,663 - file_io.py[line:728] - WARNING: Retry=1, Wait=25.6, Timestamp=1665487077.6639552, Function=getObject,
	args=('modelarts-cn-north-4-test', 'modelarts/code-test'), kwargs={loadStreamInMemory:False, cache:False, }
38	[Mode]Arts Service Log]2022-10-11 19:18:23,266 - file_io.py[line:741] - ERROR: Failed to call:
39	func= <bound 0x7fb332a2a910="" <moxing.framework.file.src.obs.client.obsclient="" at="" method="" object="" obsclient.getobject="" of="">&gt;</bound>
40	args=('modelarts-cn-north-4-test', 'modelarts/code-test')
41	<pre>kwargs={loadStreamInMemory:False, cache:False, }</pre>
42	[ModelArts Service Log]2022-10-11 19:18:23,267 - file_io.py[line:748] - ERROR:
43	stat:404
44	errorCode:NoSuchKey
45	errorMessage:The specified key does not exist.
46	reason:Not Found
47	request-id:00000183C6C4010C66D399E000C0E366
48	retry:0
49	[ModelArts Service Log]2022-10-11 19:18:23,267 - modelarts-downloader.py[line:90] - ERROR: modelarts-downloader.py: Download directory failed: [Errno
	{'status': 404, 'reason': 'Not Found', 'errorCode': 'NoSuchKey', 'errorMessage': 'The specified key does not exist.', 'body': None, 'requestId':
	'00000183C6C4010C66D399E000C0E366', 'hostId': 'tMe7zkB1evwTBxJLL7XbKfb681gFdhz/M+xo5tgabLcrgw90DRbnKiHVYoJmYHPT', 'header': [('x-reserved', 'amazon, aws
	and amazon web services are trademarks or registered trademarks of Amazon Technologies, Inc'), ('request-id', '00000183C6C4010C66D399E000C0E366'), ('id-2',
	'32AAAQAAEAABAAAQAAEAABAAAQAAEAABAAAQAAEAABCSdfbsDDEx4QVqrcVsoq5C/Io8A16gNg'), ('content-type', 'application/xml'), ('date', 'Tue, 11 Oct 2022 11:17:57 GMT'),
	('content-length', '310')]}] file or directory or bucket not found.

#### **Possible Causes**

The possible causes are as follows:

- 1. An error occurred in OBS.
  - a. Unavailable file: The specified key does not exist.
  - b. Insufficient OBS permissions
  - c. OBS traffic limiting

- d. Others
- 2. The disk space is insufficient.

### Solution

- 1. For an OBS error:
  - a. Unavailable file: The specified key does not exist.
    - For details, see Error Message "errorMessage:The specified key does not exist" Displayed in Logs.
  - b. Insufficient OBS permissions
    - For details, see What Should I Do If Error "stat:403 reason:Forbidden" Is Displayed in Logs When a Training Job Accesses OBS.
  - OBS traffic limiting
     For details, see Error Message "BrokenPipeError: Broken pipe"
     Displayed When OBS Data Is Copied.
  - Others
     For details, see . Alternatively, obtain the request ID and contact OBS customer service.
- 2. For insufficient disk space:

For details, see Common Issues Related to Insufficient Disk Space and Solutions.

## 16.4.3 Hard Faults Due to Space Limit

## 16.4.3.1 Downloading Files Timed Out or No Space Left for Reading Data

#### Symptom

When data, code, or model is copied during training, the error message "No space left on device" is displayed.

Figure 16-25 Error log



## Possible Causes

The possible causes are as follows:

- The disk space is insufficient.
- When a distributed job is executed, the **docker base size** configuration does not take effect on certain nodes. As a result, the storage space of the / root directory in the container is only the default value of 10 GB, which should be 50 GB, leading to the job training failure.
- The storage space is sufficient, but the error message "No Space left on device" is still displayed.

If there are a large number of files in the same directory, the kernel creates an index table to accelerate file retrieval. If a large number of files are created in a short period of time, the number of indexes reaches the upper limit, and an error occurs.

#### **NOTE**

The issue occurs depending on the following factors:

- A longer file name leads to a smaller upper limit for the number of files.
- A smaller block size leads to a smaller upper limit for the number of files. (There are three block sizes, 1024 bytes, 2048 bytes, and 4096 bytes. The default size is 4096 bytes.)
- The issue is more likely to occur if files are created in a shorter period of time. The reason is as follows: There is a cache, the size of which is determined based on the preceding two factors. When the number of files in the directory is large, the cache is enabled. The resources are released if they are not used.

## Solution

- 1. Rectify the fault by following the operations described in Error Message "write line error" Displayed in Logs.
- 2. If the issue occurs only on certain nodes used by the distributed job, submit a service ticket to isolate the faulty nodes.
- 3. If the issue is caused by EulerOS restrictions, take the following measures:
  - Reduce the number of files in a single directory.
  - Slow down the file creation speed.
  - Disable the dir\_index attribute of the Ext4 file system, which may affect the file retrieval performance. For details, see https://access.redhat.com/ solutions/29894.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.3.2 Insufficient Container Space for Copying Data

#### Symptom

When a ModelArts training job was running, the error below was printed in the log. As a result, data failed to be copied to the container.

OSError:[Errno 28] No space left on device

### **Possible Causes**

The container space is insufficient for downloading data.

#### Solution

- 1. Check if data is downloaded to the **/cache** directory. Each GPU node has a **/ cache** directory with 4 TB of storage. Check if the directory is experiencing an excessive creation of files simultaneously, which will run out of inodes, leading to a shortage of space.
- 2. Check whether GPU resources are used. If CPU resources are used, **/cache** and the code directory share 10 GB of memory. As a result, the memory is insufficient. In this case, use GPU resources instead.
- Add the following environment variable to the code: import os os.system('export TMPDIR=/cache')

## 16.4.3.3 Error Message "No space left" Displayed When a TensorFlow Multinode Job Downloads Data to /cache

#### Symptom

During training job creation, error message "No space left" is displayed when a TensorFlow multi-node job downloads data to **/cache**.

#### **Possible Cause**

In a TensorFlow multi-node job, the **parameter server** (ps) and **worker** roles are started. The **ps** and **worker** roles are scheduled to the same machine. Training data is useless for **ps**. Therefore, the ps-related logic in code does not need to download the training data. If **ps** also downloads data to **/cache**, the actually downloaded data will be doubled. For example, if only 2.5 TB data is downloaded, the program displays a message indicating that space is insufficient because the **/ cache** has only 4 TB available space.

## Solution

When a TensorFlow multi-node job is used to download data, the correct download logic is as follows:

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument("--job_name", type=str, default="")
args = parser.parse_known_args()
```

```
if args[0].job_name != "ps":
copy.....
```

## 16.4.3.4 Size of the Log File Has Reached the Limit

#### Symptom

An error occurs during the running of a ModelArts training job, indicating that the size of the log file has reached the limit.

modelarts-pope: log length overflow(max:1073741824; already: 107341771; new:90), process will continue running silently

### Possible Cause

Error information indicates that the size of the log file has reached the limit. After this error occurs, the volume of logs does not increase and the background continues to run.

## Solution

Reduce unnecessary log output from the boot file.

## 16.4.3.5 Error Message "write line error" Displayed in Logs

#### Symptom

During program running, a large number of error messages "write line error" are generated. This issue recurs each time the program runs at a specific progress.

#### Figure 16-26 Error log

[IVIOUEIAITS SERVICE LOG][ITIOUEIaITS-pipe, write line error [ModelArts Service Log]modelarts-pipe: write line error

#### **Possible Causes**

The possible causes are as follows:

• Core files are generated during the program running and exhaust the storage space in the / root directory.

• The 3.5 TB of storage space in the **/cache** directory is used up by the local data and files stored in it.

#### **NOTE**

The disk space for in-cloud training consists of the space from the following directories:

- 1. The / root directory, which is specified by **base size** in Docker. The default value is 10 GB. On the cloud, the value has been changed to 50 GB.
- 2. The /cache directory, which is 3.5 TB typically.

## Solution

 If core files are generated in the training job's work directory, add the code below at the beginning of the boot script to disable the generation of the core files.
 import os

os.system("ulimit -c 0")

- 2. Check whether the dataset and checkpoint file have used up the storage space of the **/cache** directory.
- 3. Use the local PyCharm to remotely access notebook for debugging.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.3.6 Error Message "No space left on device" Displayed in Logs

#### Symptom

When data, code, or model is copied during training, the error message "No space left on device" is displayed.

NF0:root:RawImageIterAsync: loading image list
raceback (most recent catt tast):
val not area bath area in allocates
Val path, ang. Jact 1227
valima listerual liet
val_img_list-val_tist/ File "/home/mind/tf-models/moving/build/moving/mynet/data/data factory py" line 134 in get data iter
File "Home/mind/ff-models/moving/multid/moving/multic/data/imageraw dataset asyn ruy" line 405 in get data iter
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw dataset async.py", the 184, ininit
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw dataset async.py", line 184, in <listcomp></listcomp>
File "/home/work/anaconda3/lib/python3.6/multiprocessing/context.py", line 129, in RawArray
return RawArray(typecode or type, size or initializer)
File "/home/work/anaconda37lib/python3.6/multiprocessing/sharedctypes.py", line 60, in RawArray
obj = _new value(type )
File "/home/work/anaconda3/lib/python3.6/multiprocessing/sharedctypes.py", line 40, in _new_value
wrapper = heap.BufferWrapper(size)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 248, ininit
block = BufferWrapperheap.malloc(size)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 230, in malloc
(arena, start, stop) = selfmalloc(size)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 128, in _malloc
arena = Arena(length)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 77, ininit
t.write(zeros)
SError: [Errno 28] No space Left on device
xception ignored in: -bound method RawimageiterAsyncdet of <moxing.mxnet.data.imageraw_dataset_async.kawimageiterasync 0x="" at="" fa185881960="" object="">&gt; reserved.for the second s</moxing.mxnet.data.imageraw_dataset_async.kawimageiterasync>
raceback (most recent cart tast):

#### Figure 16-27 Error log

#### **Possible Causes**

The possible causes are as follows:

- The disk space is insufficient.
- When a distributed job is executed, the **docker base size** configuration does not take effect on certain nodes. As a result, the storage space of the / root

directory in the container is only the default value of 10 GB, which should be 50 GB, leading to the job training failure.

• The storage space is sufficient, but the error message "No Space left on device" is still displayed.

If there are a large number of files in the same directory, the kernel creates an index table to accelerate file retrieval. If a large number of files are created in a short period of time, the number of indexes reaches the upper limit, and an error occurs.

#### **NOTE**

The issue occurs depending on the following factors:

- A longer file name leads to a smaller upper limit for the number of files.
- A smaller block size leads to a smaller upper limit for the number of files. The block size can be 1024 bytes, 2048 bytes, or 4096 bytes, and it defaults to 4096 bytes.
- The issue is more likely to occur if files are created in a shorter period of time. The reason is as follows: There is a cache, the size of which is determined based on the preceding two factors. When the number of files in the directory is large, the cache is enabled. The resources are released if they are not used.

#### Solution

- 1. Rectify the fault by following the operations described in Error Message "write line error" Displayed in Logs.
- 2. If the issue occurs only on certain nodes used by the distributed job, submit a service ticket to isolate the faulty nodes.
- 3. If the issue is caused by EulerOS restrictions, take the following measures:
  - Reduce the number of files in a single directory.
  - Slow down the file creation speed.
  - Disable the dir\_index attribute of the Ext4 file system, which may affect the file retrieval performance. For details, see https://access.redhat.com/ solutions/29894.

#### Summary and Suggestions

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

#### 16.4.3.7 Training Job Failed Due to OOM

#### Symptom

If a training job failed due to out of memory (OOM), possible symptoms as as follows:

- 1. Error code 137 is returned.
- 2. The log file contains error information with keyword killed.

#### Figure 16-28 Error log

Traceback (most recent call last):

- File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 261, in <module> main()
- File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 251, in main
- loss,acc = train\_and\_test(e, opt.alpha\_start)
- File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 243, in train\_and\_test acc = test(epoch, alpha start, False)
- File "/home/ma-user/modelarts/user-job-dir/addernet-firstlast/main-imgnet.py", line 222, in test
- output = net(images, epoch, alpha start) File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/modules/module.py", line 541, in call result = self.forward(\*input, \*\*kwargs)
- File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/parallel/data\_parallel.py", line 152, in forward outputs = self.parallel apply(replicas, inputs, kwargs)
- File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/parallel/data\_parallel.py", line 162, in parallel\_apply
- return parallel\_apply(replicas, inputs, kwargs, self.device\_ids(:len(replicas))) File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/nn/parallel/parallel\_apply.py", line 75, in parallel\_apply thread.start()
- File "/home/ma-user/anaconda/lib/python3.6/threading.py", line 851, in start
- self. started.wait()
- File "/home/ma-user/anaconda/lib/python3.6/threading.py", line 551, in wait signaled = self.\_cond.wait(timeout)

File "/home/ma-user/anaconda/lib/python3.6/threading.py", line 295, in wait

waiter.acquire()

File "/home/ma-user/anaconda/lib/python3.6/site-packages/torch/utils/data/\_utils/signal\_handling.py", line 66, in handler error if any worker fails()

- RuntimeError: DataLoader worker (pid 38077) is killed by signal: Killed.
- Error message "RuntimeError: CUDA out of memory." is displayed in logs. 3.

#### Figure 16-29 Error log

Traceback (most recent call last):
<pre>File "memory_test.py", line 47, in <module></module></pre>
<pre>tmp_tensor = torch.empty(int(total_memory * 0.45), dtype=torch.int8, device='cuda')</pre>
RuntimeError: CUDA out of memory. Tried to allocate 14.29 GiB (GPU 0; 14.29 GiB total capacity; 0 bytes
already allocated; 14.29 GiB free; 0 bytes reserved in total by PyTorch)

4. Error message "Dst tensor is not initialized" is displayed in TensorFlow logs.

#### **Possible Causes**

The possible causes are as follows:

- GPU memory is insufficient.
- OOM occurred on certain nodes. This issue is typically caused by the node fault.

#### Solution

- Modify hyperparameter settings to release unnecessary tensors. 1.
  - Modify network parameters, such as **batch\_size**, **hide\_layer**, and a. cell\_nums.
  - b. Release unnecessary tensors. del tmp\_tensor torch.cuda.empty\_cache()
- Use the local PyCharm to remotely access notebook for debugging. 2.
- If the fault persists, submit a service ticket to locate the fault or even isolate 3. the affected node.

#### Summary and Suggestions

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.3.8 Common Issues Related to Insufficient Disk Space and Solutions

This section centrally describes common issues related to insufficient disk space and solutions to these issues.

## Symptom

When data, code, or model is copied during training, error message "No space left on device" is displayed.

#### Figure 16-30 Error log

ANTO LOCE ON ELONGINE TO ELONGE PARENCE DE MONACH INTINEADO EO O CONSCIENCIÓN DE MONO EN COMO EN COMO EL CONSCI
INF0:root:RawImageIterAsync: loading image list
raceback (most recent call last):
File "test.py", line 142, in <module></module>
val_path, args.batch_size)
File "test.py", line 59, in get_data
val_img_list=val_list)
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/data_factory.py", line 134, in get_data_iter
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw dataset async.py", line 405, in get data iter
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw_dataset_async.py", line 184, inint
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw dataset async.py", line 184, in <listcomp></listcomp>
File "/home/work/anaconda3/lib/python3.6/multiprocessing/context.py", line 129, in RawArray
return RawArray(typecode or type, size or initializer)
File "/home/work/anaconda37lib/python3.67multiprocessing/sharedctypes.py", line 60, in RawArray
obj = new value(type)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/sharedctypes.py", line 40, in _new value
wrapper = heap.BufferWrapper(size)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 248, ininit
block = BufferWrapper. heap.malloc(size)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 230, in malloc
(arena, start, stop) = selfmalloc(size)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 128, in _malloc
arena = Arena(length)
File "/home/work/anaconda3/lib/python3.6/multiprocessing/heap.py", line 77, ininit
f.write(zeros)
DSError: [Errno 28] No space left on device
exception ignored in: <bound_method_rawimageiterasyncdel <moxing.mxnet.data.imageraw_dataset_async.rawimageiterasync_object_at_0x7fa18588f9b0="" of="">&gt;</bound_method_rawimageiterasyncdel>
Fraceback (most recent call last):
File "/home/mind/tf-models/moxing/build/moxing/mxnet/data/imageraw dataset async.py", line 222, in del

## Possible Causes

The possible causes are as follows:

- The storage space in the **/cache** directory is used up by the local data and files stored in it.
- Data is decompressed when being processed. As a result, the data volume increases, and finally the storage space in the **/cache** directory is used up.
- Data is not saved in /cache or /home/ma-user/ (/cache will be softly connected to /home/ma-user/). As a result, the system directory is fully occupied. The system directory supports only basic running of system functions. It cannot be used to store large volumes of data.
- During the training of certain jobs, checkpoint files will be generated and updated. If historical checkpoint files are not deleted after an update, the / cache directory will be exhausted.
- The storage space is sufficient, but the error message "No Space left on device" is still displayed. This may be triggered by insufficient inodes or an error in the file index cache of the operating system. As a result, no file can be created in the system disk, and finally data disks are used up.

#### **NOTE**

The conditions for triggering an error in the file index cache are as follows:

- A longer file name leads to a smaller upper limit for the number of files.
- A smaller block size leads to a smaller upper limit for the number of files. (There are three block sizes, 1024 bytes, 2048 bytes, and 4096 bytes. The default size is 4096 bytes.)
- This issue is more likely to occur if files are created in a shorter period of time. The reason is as follows: There is a cache, the size of which is determined based on the preceding two factors. When the number of files in the directory is large, the cache will be enabled and released with the files.

• Core files are generated during the program running and exhaust the storage space in the / root directory.

## Solution

- 1. Obtain the sizes of the dataset, decompressed dataset, and checkpoint file and check whether they have exhausted the disk space.
- 2. If the volume of data exceeds the **/cache** size, use SFS to attach more data disks for expanding the storage size.
- 3. Save the data and checkpoint in /cache or /home/ma-user/.
- 4. Check the checkpoint logic and ensure that historical checkpoints are deleted so that they will not use up the storage space in **/cache**.
- 5. If the file size is smaller than the **/cache** size, and the number of files exceeds 500,000, the issue may be caused by insufficient inodes or an error in the file index cache of the operating system. In this case, do as follows to resolve this issue:
  - Reduce the number of files in a single directory.
  - Slow down the file creation speed. For example, during data decompression, add a sleep period of 5s before decompressing the next piece of data.
- If core files are generated in the training job's work directory, add the code below at the beginning of the boot script to disable the generation of the core files. (debug code in a development environment before adding the code): import os os.system("ulimit -c 0")

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## **16.4.4 Internet Access Issues**

## 16.4.4.1 Error Message "Network is unreachable" Displayed in Logs

#### Symptom

When PyTorch is used, the following error message will be displayed in logs after **pretrained** in **torchvision.models** is set to **True**: 'OSError: [Errno 101] Network is unreachable'

#### **Possible Causes**

For security purposes, ModelArts internal training nodes are not allowed to access the Internet.

## Solution

1. Change the **pretrained** value to **False**, download the pre-trained model, and load the path to this model.

import torch import torchvision.models as models model1 = models.resnet34(pretrained=False, progress=True) checkpoint = '/xxx/resnet34-333f7ec4.pth' state\_dict = torch.load(checkpoint) model1.load\_state\_dict(state\_dict)

2. Use the local PyCharm to remotely access notebook for debugging.

### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.4.2 URL Connection Timed Out in a Running Training Job

#### Symptom

In a running training job, a URL connection timeout error occurs.

urllib.error.URLERROR:<urlopen error [Errno 110] Connection timed out>

#### **Possible Causes**

For security purposes, ModelArts is not allowed to access the Internet to download data.

#### Solution

Download the required data to a local directory and upload it to OBS. Then, access the OBS path from ModelArts to obtain the data.

## 16.4.5 Permission Issues

# 16.4.5.1 What Should I Do If Error "stat:403 reason:Forbidden" Is Displayed in Logs When a Training Job Accesses OBS

#### Symptom

When a training job accesses OBS, an error occurs.

#### Figure 16-31 Error log

ERROR:root:Failed to call:
func= <bound 0x7fddb4ad06d0="" <moxing.framework.file.src.obs.client.obsclient="" at="" method="" object="" obsclient.getobjectmetadata="" of="">&gt;</bound>
args=('bucket-cv-competition-bj4', 'fangjiemin/output/')
kwargs={}
ERROR:root:
stat:403
errorCode:None
errorMessage:None
reason:Forbidden
request-id:00000179D5ACCAC445CAA1A71019C9D0
retry:0

## **Possible Causes**

The possible causes are as follows:

• The OBS permission is incorrect. As a result, data cannot be read.

### Solution

Verify that OBS permissions are correctly assigned. If the problem persists, troubleshoot by following the instructions provided in "Why Can't I Access OBS (403 AccessDenied) After Being Granted with the OBS Access Permission?".

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

 If an error occurred in OBS, identify the cause based on the error information, including the error code and message. For details about OBS error codes, see Python > Troubleshooting > OBS Server-Side Error Codes in Object Storage Service SDK Reference.

## 16.4.5.2 Error Message "Permission denied" Displayed in Logs

#### Symptom

When a training job accesses the attached EFS disks or executes the .sh boot script, an error occurs.

• [Errno 13]Permission denied: '/xxx/xxxx'

#### Figure 16-32 Error log

Traceback (most recent call last): File "codes/prepare\_listdir.py", line 11, in <module> rec\_file\_list = os.listdir(recurrent path) OSError: [Errno 13] Permission denied: '/data/recurrent'

- bash: /bin/ln: Permission denied
- bash:/home/ma-user/.pip/pip.conf: Permission Denied (in a custom image)
- tee: /xxx/xxxx: Permission denied cp: cannot stat " No such file or directory (in a custom image)

## **Possible Causes**

The possible causes are as follows:

- [Errno 13]Permission denied: '/xxx/xxxx'
  - When data is uploaded, the ownership and permissions to the file are not changed. As a result, the work user group does not have the permission to access the training job.

- After the .sh file in the code directory is copied to the container, the execution permission is not granted for the file.
- bash: /bin/ln: Permission denied
- For security purposes, the ln command is not supported.
- bash:/home/ma-user/.pip/pip.conf: Permission Denied
   After the version of training jobs is switched from V1 to V2, the UID of the ma-user user is still 1102.
- tee: /xxx/xxxx: Permission denied cp: cannot stat ": No such file or directory The used startup script is **run\_train.sh** of an earlier version. Some environment variables in the script are unavailable in the training jobs of the new version.
- The APIs using the Python file concurrently read and write the same file.

#### Solution

1. Add permissions to access the attached EFS disks so that the permissions are the same as those of user group (1000) used in the training container. For example, if the **/nas** disk is attached, run the following command: chown -R 1000: 1000 /nas Or

chmod 777 -R /nas

- 2. If the execution permission has not been granted for the .sh file used by the custom image, run **chmod** +x xxx.sh to grant the permission before starting the script.
- 3. On the ModelArts console, if a training job is created using a custom image, a V2 container image is started using UID 1000 by default. In this case, change the UID of the **ma-user** user from 1102 to 1000. To obtain the sudo permission, comment out the sudoers line.

```
FROM {your-v1-custom-docker-image or other docker-image}
USER root
# prepare moxing_framework and seccomponent package
# and chmod/chown moxing framework package to the right permission or owner (ma-user)
RUN groupadd ma-group -g 1000 && \
   useradd -d /home/ma-user -m -u 1000 -g 1000 -s /bin/bash ma-user && \
   chmod 770 /home/ma-user && \
   # usermod -a -G work ma-user && \
   # alien -i seccomponent-1.0.2-2.0.release.x86 64.rpm && \
   chmod 770 /root && \
   # or silver bullet of files permission
   # chmod -R 777 /root && \
   usermod -a -G root ma-user
# ENV LD LIBRARY PATH=/usr/local/seccomponent/lib:$LD LIBRARY PATH
# RUN echo "ma-user ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
# RUN pip install moxing_framework-2.0.0.rc2.4b57a67b-py2.py3-none-any.whl
USER ma-user
WORKDIR /home/ma-user
```

- 4. Migrate environment variables from V1 training jobs to V2 training jobs.
  - Use V2 MA\_NUM\_HOSTS (the number of selected training nodes) to replace V1 DLS\_TASK\_NUMBER.
  - Use V2 VC\_TASK\_INDEX (or MA\_TASK\_INDEX that will be available later) to replace V1 DLS\_TASK\_INDEX. Obtain the environment variable using the method provided in the demo script for compatibility.
  - Use V2 \${MA\_VJ\_NAME}-\${MA\_TASK\_NAME}-0.\${MA\_VJ\_NAME}:6666 to replace V1 BATCH\_CUSTOM0\_HOSTS.
  - Use V2 \${MA\_VJ\_NAME}-\${MA\_TASK\_NAME}-{N}.\$ {MA\_VJ\_NAME}:6666 to replace V1 BATCH\_CUSTOM{N}\_HOSTS generally.
- 5. Check whether there are settings that allow concurrent reading and writing of the same file in the code. If so, modify the settings to forbid this operation.

If a job uses multiple cards, the same code for reading and writing data may be available on each card. In this case, do as follows to modify the code:

```
import moxing as mox
from mindspore.communication import init, get_rank, get_group_size
init()
rank_id = get_rank()
# Enable only card 0 to download data.
if rank_id % 8 == 0:
    mox.file.copy_parallel('obs://bucket-name/dir1/dir2/', '/cache')
```

## **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.6 GPU Issues

# 16.4.6.1 Error Message "No CUDA-capable device is detected" Displayed in Logs

#### Symptom

An error similar to the following occurs during the running of the program: 1. 'failed call to culnit: CUDA\_ERROR\_NO\_DEVICE: no CUDA-capable device is detected' 2. 'No CUDA-capable device is detected although requirements are installed'

## Possible Causes

The possible causes are as follows:

- CUDA\_VISIBLE\_DEVICES has been incorrectly set.
- CUDA operations are performed on GPUs with IDs that are not specified by CUDA\_VISIBLE\_DEVICES.

#### Solution

1. Do not change the **CUDA\_VISIBLE\_DEVICES** value in the code. Use its default value.

- 2. Ensure that the specified GPU IDs are within the available GPU IDs.
- If the error persists, print the CUDA\_VISIBLE\_DEVICES value and debug it in the notebook, or run the following commands to check whether the returned result is True: import torch torch.cuda.is\_available()

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.6.2 Error Message "RuntimeError: connect() timed out" Displayed in Logs

#### Symptom

When PyTorch is used for distributed training, the following error occurs.

#### Figure 16-33 Error log

INFO - 03/23/21 17:20:50 - 0:00:04 - Building data done with 1331166 images loaded. Traceback (most recent call last): File "swav-master/main\_swav.py", line 500, in <module> main() File "swav-master/main\_swav.py", line 191, in main mp.spawn(main\_worker, nprocs=args.ngpu, args=()) File "/home/work/anaconda/lib/python3.6/site-packages/torch/multiprocessing/spawn.py", line 171, in spawn while not spawn\_context.join(): File "/home/work/anaconda/lib/python3.6/site-packages/torch/multiprocessing/spawn.py", line 118, in join raise Exception(msg) Exception: Process 2 terminated with the following error: Traceback (most recent call last): File "/home/work/anaconda/lib/python3.6/site-packages/torch/multiprocessing/spawn.py", line 19, in \_wrap fn(i, \*args) File "/cache/user-job-dir/swav-master/main\_swav.py", line 231, in main\_worker rank=args.rank) File "/home/work/anaconda/lib/python3.6/site-packages/torch/distributed/distributed\_c10d.py", line 397, in init\_process\_group store, rank, world\_size = next(rendezvous\_iterator) File \*/home/work/anaconda/lib/python3.6/site-packages/torch/distributed/rendezvous.py\*, line 168, in \_env\_rendezvous\_handler store = TCPStore(master\_addr, master\_port, world\_size, start\_daemon) RuntimeError: connect() timed out.

#### **Possible Causes**

If data is copied before this issue occurs, data copy on all nodes is not complete at the same time. If you perform **torch.distributed.init\_process\_group()** when data copy is still in progress on certain nodes, the connection timed out.

#### Solution

If the issue is caused by asynchronous data copy between nodes and no barrier occurs, perform **torch.distributed.init\_process\_group()** before copying data, copy data based on **local\_rank()==0**, call **torch.distributed.barrier()**, and wait until data copy is complete on all nodes. For details, see the following code: import moxing as mox import torch

```
torch.distributed.init_process_group()
if local_rank == 0:
    mox.file.copy_parallel(src,dst)
```

torch.distributed.barrier()

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.6.3 Error Message "cuda runtime error (10) : invalid device ordinal at xxx" Displayed in Logs

#### Symptom

A training job failed, and the following error is displayed in logs.

#### Figure 16-34 Error log

main()
File "train.py", line 278, in main
torch.cuda.set_device(args.local_rank)
File "/home/work/anaconda/lib/python3.6/site-packages/torch/cuda/_initpy", line 300, in set_device
torch. C. cuda setDevice(device)
RuntimeError: cuda runtime error (10) : invalid device ordinal at /pytorch/torch/csrc/cuda/Module.cpp:37

## Possible Causes

The possible causes are as follows:

- The CUDA\_VISIBLE\_DEVICES setting does not comply with job specifications. For example, you select a job with four GPUs, and the IDs of available GPUs are 0, 1, 2, and 3. However, when you perform CUDA operations, for example tensor.to(device="cuda:7"), tensors are specified to run on GPU 7, which is beyond the available GPU IDs.
- GPUs are damaged on resource nodes if CUDA operations are performed on a GPU with a specified ID. As a result, the number of GPUs that can be detected is less than the selected specifications.

## Solution

- 1. Perform CUDA operations on the GPUs with IDs specified by CUDA\_VISIBLE\_DEVICES.
- 2. If a GPU on a resource node is damaged, contact technical support.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.6.4 Error Message "RuntimeError: Cannot re-initialize CUDA in forked subprocess" Displayed in Logs

## Symptom

When PyTorch is used to start multiple processes, the following error message is displayed: RuntimeError: Cannot re-initialize CUDA in forked subprocess

#### **Possible Causes**

The multi-processing startup mode is incorrect.

### Solution

For details, see Writing Distributed Applications with PyTorch. """run.py:" #!/usr/bin/env python import os import torch import torch.distributed as dist import torch.multiprocessing as mp def run(rank, size): """ Distributed function to be implemented later. """ pass def init\_process(rank, size, fn, backend='gloo'): """ Initialize the distributed environment. """ os.environ['MASTER\_ADDR'] = '127.0.0.1' os.environ['MASTER\_PORT'] = '29500' dist.init\_process\_group(backend, rank=rank, world\_size=size) fn(rank, size) if \_\_name\_\_ == "\_\_main\_\_": size = 2processes = [] mp.set\_start\_method("spawn") for rank in range(size): p = mp.Process(target=init\_process, args=(rank, size, run)) p.start() processes.append(p) for p in processes: p.join()

## **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.6.5 No GPU Is Found for a Training Job

#### Symptom

The following error message is displayed during the running of a ModelArts training job:

failed call to cuInit: CUDA\_ERROR\_NO\_DEVICE: no CUDA-capable device is detected

## **Possible Cause**

According to error information, the error cause is that the training job running program cannot read the GPU.

## Solution

Check whether the following configuration information is added to code and set the GPU visible to the program based on the error message:

os.environ['CUDA\_VISIBLE\_DEVICES'] = '0,1,2,3,4,5,6,7'

In the preceding information, **0** is a GPU ID of the server. The GPU ID can be 0, 1, 2, 3, or the like, indicating a GPU ID visible to the program. If the configuration information is not added, the GPU corresponding to the ID is unavailable.

## **16.4.7 Service Code Issues**

# 16.4.7.1 Error Message "pandas.errors.ParserError: Error tokenizing data. C error: Expected .\* fields" Displayed in Logs

## Symptom

When pandas is used to read CSV data, the following error is displayed in logs, and the training job failed: pandas.errors.ParserError: Error tokenizing data. C error: Expected 4 field

#### Possible Causes

The number of columns in each row of the CSV file is different.

### Solution

Use either of the following methods to resolve this issue:

- Check the CSV file and delete the lines with extra columns.
- Run the following commands to ignore the lines with extra columns: import pandas as pd pd.read csv(filePath,error\_bad lines=False)

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.7.2 Error Message "max\_pool2d\_with\_indices\_out\_cuda\_frame failed with error code 0" Displayed in Logs

#### Symptom

After PyTorch 1.3 is upgraded to 1.4, the following error message is displayed: "RuntimeError:max\_pool2d\_with\_indices\_out\_cuda\_frame failed with error code 0"

## **Possible Causes**

The PyTorch 1.4 engine is incompatible with that of PyTorch 1.3.

### Solution

- Run the following commands to add contiguous data: images = images.cuda() pred = model(images.permute(0, 3, 1, 2).contigous())
- 2. Roll back to PyTorch 1.3.
- 3. Use the local PyCharm to remotely access notebook for debugging.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.7.3 Training Job Failed with Error Code 139

#### Symptom

The training job failed, and error code 139 is returned.

#### Possible Causes

The possible causes are as follows:

- Certain pip packages in the pip source have been updated, leading to data incompatibility. For example, an error occurs when the transformers package is imported after the package update.
- The user code has a bug, leading to memory overwriting or unauthorized memory access.
- An unknown system error occurs. In this case, create the training job again. If the fault persists, submit a service ticket.

#### Solution

1. If the training job succeeded before and no modification has been made, compare the logs in the two cases and check whether any dependency package has been updated in the pip source.

#### Figure 16-35 Log comparison



- 2. Use the local PyCharm to remotely access notebook for debugging.
- 3. If the fault persists, contact technical support engineers.

## Summary and Suggestions

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.7.4 Debugging Training Code in the Cloud Environment If a Training Job Failed

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.7.5 Error Message "'(slice(0, 13184, None), slice(None, None, None))' is an invalid key" Displayed in Logs

#### Symptom

The following error message is displayed during training: TypeError: '(slice(0, 13184, None), slice(None, None, None))' is an invalid key

#### **Possible Causes**

The data selected for segmentation is incorrect.

#### Solution

Run the following command to resolve the issue: X = dataset.iloc[:,:-1].values

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.7.6 Error Message "DataFrame.dtypes for data must be int, float or bool" Displayed in Logs

#### Symptom

The following error message is displayed during training: DataFrame.dtypes for data must be int, float or bool

#### **Possible Causes**

The possible cause is as follows:

The training data is not of the int, float, or bool type.

## Solution

Run the following commands to convert the error column: from sklearn import preprocessing lbl = preprocessing.LabelEncoder() train\_x['acc\_id1'] = lbl.fit\_transform(train\_x['acc\_id1'].astype(str))

## **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

# 16.4.7.7 Error Message "CUDNN\_STATUS\_NOT\_SUPPORTED" Displayed in Logs

#### Symptom

The following error message is displayed during PyTorch training: RuntimeError: cuDNN error: CUDNN\_STATUS\_NOT\_SUPPORTED. This error may appear if you passed in a non-contiguous input.

#### **Possible Causes**

The input data is not of contiguous type, which is not supported by cuDNN.

#### Solution

- 1. Disable cuDNN before training. torch.backends.cudnn.enabled = False
- Convert the input data into contiguous data. images = images.cuda() images = images.permute(0, 3, 1, 2).contigous()

#### Summary and Suggestions

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.7.8 Error Message "Out of bounds nanosecond timestamp" Displayed in Logs

#### Symptom

When pandas.to\_datetime is used to convert time, the following error message is displayed:

pandas.\_libs.tslibs.np\_datetime.OutOfBoundsDatetime: Out of bounds nanosecond timestamp: 1-01-02 13:20:00

### Possible Causes

The time is out of the permitted range. For details, see the official document.

## Solution

Check the time. Timestamps in pandas are in the unit of nanosecond. Ensure that the time is within the following permitted range:

- Earliest time: 1677-09-22 00:12:43.145225
- Latest time: 2262-04-11 23:47:16.854775807

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.7.9 Error Message "Unexpected keyword argument passed to optimizer" Displayed in Logs

#### Symptom

After Keras is upgraded to 2.3.0 or later, the following error message is displayed: TypeError: Unexpected keyword argument passed to optimizer: learning\_rate

### **Possible Causes**

Certain parameters have been renamed in Keras. For details, see Keras 2.3.0.

#### Figure 16-36 API changes

Rename 1r to learning\_rate for all optimizers.

#### Solution

Rename learning\_rate lr.

#### **Summary and Suggestions**

Before creating a training job, use the ModelArts development environment to debug the training code to maximally eliminate errors in code migration.

## 16.4.7.10 Error Message "no socket interface found" Displayed in Logs

#### Symptom

An NCCL debug log level is set in a distributed job executed using a PyTorch image. import os os.environ["NCCL DEBUG"] = "INFO"

The following error message is displayed.

#### Figure 16-37 Error log

```
job0879f61e-job-base-pda-2-0:712:712 [0] bootstrap.cc:37 NCCL WARN Bootstrap : no socket interface found
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO init.cc:128 -> 3
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO bootstrap.cc:76 -> 3
job0879f61e-job-base-pda-2-0:712:712 [0] NCCL INFO bootstrap.cc:265 -> 3
Traceback (most recent call last):
File 'train_net.py', line 1923, in <module>
main_worker(args)
File 'train_net.py', line 355, in min_worker
network = torch.nn.parallel.DistributedDataParallel(network, device_ids=device_ids, find_unused_parameters=True)
File 'rhome/work/anaconda/lib/python3.6/site-packages/torch/nn/parallel/distributed.py", line 298, in __init__
self.broadcast_bucket_size)
File 'rhome/work/anaconda/lib/python3.6/site-packages/torch/nn/parallel/distributed.py", line 480, in _distributed_broadcast_coalesced
dist_broadcast_bucket_size)
RuntimeError: NCCL error in: /pytorch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch/torch
```

#### Possible Causes

The environment variables NCCL\_IB\_TC, NCCL\_IB\_GID\_INDEX, and NCCL\_IB\_TIMEOUT are not configured. As a result, the communication is slow and unstable, and the IB communication is interrupted.

### Solution

Add environment variables to the code.

```
import os
os.environ["NCCL_IB_TC"] = "128"
os.environ["NCCL_IB_GID_INDEX"] = "3"
os.environ["NCCL_IB_TIMEOUT"] = "22"
```

# 16.4.7.11 Error Message "Runtimeerror: Dataloader worker (pid 46212) is killed by signal: Killed BP" Displayed in Logs

#### Symptom

During the running of a training job, error message "Runtimeerror: Dataloader worker (pid 46212) is killed by signal: Killed BP" is displayed in logs.

#### **Possible Causes**

The Dataloader process exits because the batch size is too large.

#### Solution

Decrease the batch size.

# 16.4.7.12 Error Message "AttributeError: 'NoneType' object has no attribute 'dtype'" Displayed in Logs

#### Symptom

Code can run properly in the notebook Keras image. When tensorflow.keras is used for training, error message "AttributeError: 'NoneType' object has no attribute 'dtype'" is displayed.

## **Possible Causes**

The NumPy version of the training image is different from that in the notebook instance.

### Solution

Print the NumPy version in the code and check whether the version is 1.18.5. If the version is not 1.18.5, run the following command at the beginning of the code:

import os os.system('pip install numpy==1.18.5')

If the error persists, modify the preceding code as follows:

```
import os
os.system('pip install numpy==1.18.5')
os.system('pip install keras==2.6.0')
os.system('pip install tensorflow==2.6.0')
```

## 16.4.7.13 Error Message "No module name 'unidecode'" Displayed in Logs

#### Symptom

After the configuration file of the Tacotron 2 model downloaded from the master branch of MindSpore open-source Gitee is modified and then uploaded to ModelArts for training, error message "No module name 'unidecode'" is displayed in logs.

#### **Possible Causes**

The Unidecode name of the **requirements.txt** file is incorrect, where **U** should be lowercase. As a result, the Unidecode module is not installed in the training job environment.

## Solution

Change **Unidecode** in **requirements.txt** to **unidecode**.

#### **Summary and Suggestions**

Add the following line to the training code:

os.system('pip list')

Run the training job and check whether the required module is available in logs.

#### 16.4.7.14 Distributed Tensorflow Cannot Use tf.variable

#### Symptom

The following error occurs when **tf.variable** is used across multiple machines and multiple GPUs: **WARNING:tensorflow:Gradient is None for variable:v0/tower\_0/UNET\_v7/sub\_pixel/Variable:0.Make sure this variable is used in loss computation** 

#### Figure 16-38 Distributed Tensorflow unavailable

WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_0/UNET\_v7/sub\_pixel/Variable:0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_1/UNET\_v7/sub\_pixel/Variable\_1:0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_1/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_1/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_1/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_2/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_2/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_2/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_3/UNET\_v7/sub\_pixel/Variable\_1.0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_4/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_f/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_f/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_f/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tensorflow:Gradient is None for varaible: v0\_ftower\_f/UNET\_v7/sub\_pixel/Variable\_0. Make sure this variable is used in loss computation. WARNING:tens

#### Possible Cause

Distributed TensorFlow needs to use tf.get\_variable instead of tf.variable.

#### Solution

Replace **tf.variable** in the boot file with **tf.get\_variable**.

## 16.4.7.15 When MXNet Creates kvstore, the Program Is Blocked and No Error Is Reported

#### Symptom

When **kv\_store = mxnet.kv.create('dist\_async')** is used to create **kvstore**, the program is blocked. For example, run the following code. If **end** is not displayed, the program is blocked.

```
print('start')
kv_store = mxnet.kv.create('dist_async')
print('end')
```

#### **Possible Cause**

The possible cause of a worker block is that the server cannot be connected.

### Solution

Place the following code before **import mxnet** in **Boot File** to check the communication status between nodes. In addition, ps can be resent.

```
import os
os.environ['PS_VERBOSE'] = '2'
os.environ['PS_RESEND'] = '1'
```

In the preceding code, **os.environ['PS\_VERBOSE']** = '2' indicates that all communication information is printed. **os.environ['PS\_RESEND']** = '1' indicates that the Van instance resends the message if it does not receive the ACK message within the milliseconds set by **PS\_RESEND\_TIMEOUT**.

## 16.4.7.16 ECC Error Occurs in the Log, Causing Training Job Failure

## Symptom

The following error occurs during the running of the training job log: **RuntimeError: CUDA error: uncorrectable ECC error encountered** 

## **Possible Cause**

ECC errors

## Solution

If there are more than 64 ECC errors, the system automatically isolates the faulty nodes. After the isolation, restart the training job to check whether the fault is rectified. If the training job fails again or is suspended due to an unisolated node, contact technical support.

# 16.4.7.17 Training Job Failed Because the Maximum Recursion Depth Is Exceeded

## Symptom

An error occurs for a ModelArts training job.

RuntimeError: maximum recursion depth exceeded in \_\_instancecheck\_\_

#### Possible Causes

The training failed because the recursion depth exceeded the default recursion depth of Python.

## Solution

If the maximum recursion depth is exceeded, increase the recursion depth in the boot file as follows:

import sys sys.setrecursionlimit(1000000)

## 16.4.7.18 Training Using a Built-in Algorithm Failed Due to a bndbox Error

#### Symptom

When a training job is created using a built-in algorithm, the training failed with the following error message in the log:

KeyError: 'bndbox'

### Possible Causes

Non-rectangles are used for labeling training sets. However, the built-in algorithm does not support datasets labeled by a non-rectangle.

## Solution

This issue can be resolved in either of the following ways:

- Method 1: Use a common framework to develop a model that supports polygon-labeled datasets.
- Method 2: Use rectangles to label the datasets. Then, start the training job again.

## 16.4.7.19 Training Job Status Is Reviewing Job Initialization

## Symptom

When **Algorithm Source** is set to **Custom** during training job creation, the training job status is **Reviewing Job Initialization**.

#### **Possible Cause**

When a custom image is running for the first time, the image needs to be reviewed first. After the image is reviewed, you can create a job. That is, the current status is **Reviewing Job Initialization**.

## 16.4.7.20 Training Job Process Exits Unexpectedly

#### Symptom

Running a training job failed, and error information similar to the following is displayed in logs:

[Modelarts Service Log]Training end with return code: 137

## Possible Causes

According to the log, the exit code of the training job is 137. The training process starts after the user code is executed. Therefore, the exit code mentioned in this section is generated after the code for training job is executed. Common error codes include codes 247 and 139.

• Exit code: 137 or 247

The possible cause is that the memory overflows. To resolve this issue, you can reduce the data volume, decrease the **batch\_size** value, optimize the code, or aggregate and replicate the data.

#### **NOTE**

The size of data files is not equal to the memory usage. Therefore, evaluate the memory usage.

• Exit code: 139

Check the version of the installation package. There may be a package conflict.

## Troubleshooting

According to the error information, the error is caused by the user code.

You can use either of the following methods to locate the fault:

- Debug the code online (only available for the non-distributed code).
  - a. Apply for a development environment instance with the same specifications in the development environment (notebook).
  - b. Debug the user code in the notebook and find the improper code snippet.
  - c. Find a solution by searching the key code snippet and exit code in a search engine.
- Locate the fault based on the training logs.
  - a. Identify the improper code snippet based on the logs.
  - b. Print the improper code snippet to obtain more detailed log information.
  - c. Run the training job again to locate the improper code snippet.

#### 16.4.7.21 Stopped Training Job Process

#### Symptom

The training job process is stopped and the logs are interrupted.

#### **Possible Causes**

CPU soft lock

The decompression of a large number of files may cause CPU soft lock and node restart. You can suspend the decompression for the specified amount of time by invoking sleep method when decompressing a large number of files. For example, every time 10,000 files are decompressed, the decompression stops for 1 second.

• Storage limitation

Use data disks based on specifications. For details about a data disk size, see

CPU overload
 Reduce the number of threads.

#### Troubleshooting

According to the error information, the error is caused by the user code.

You can use either of the following methods to locate the fault:

- Debug the code online (only available for the non-distributed code).
  - a. Apply for a development environment instance with the same specifications in the development environment (notebook).
  - b. Debug the user code in the notebook and find the improper code snippet.
  - c. Find a solution by searching the key code snippet and exit code in a search engine.
- Locate the fault based on the training logs.
  - a. Identify the improper code snippet based on the logs.
  - b. Print the improper code snippet to obtain more detailed log information.

c. Run the training job again to locate the improper code snippet.

## 16.4.8 Training Job Suspended

## 16.4.8.1 Data Replication Suspension

### Symptom

The system stops responding when **mox.file.copy\_parallel** is called to copy data.

## Solution

- Run the following commands to copy files or folders: import moxing as mox mox.file.set\_auth(is\_secure=False)
- Run the following command to copy a single file that is greater than 5 GB: from moxing.framework.file import file\_io

Run **file\_io.\_LARGE\_FILE\_METHOD** to check the version of the MoXing API. Output value **1** indicates V1 and **2** indicates V2.

Run file\_io.\_NUMBER\_OF\_PROCESSES=1 to resolve the issue for the V1 API.

To resolve the issue for the V2 API, run **file\_io.\_LARGE\_FILE\_METHOD = 1** to switch to V1 and perform operations required in V1. Alternatively, run **file\_io.\_LARGE\_FILE\_TASK\_NUM=1** to resolve this issue.

• Run the following command to copy a folder: mox.file.copy\_parallel(threads=0,is\_processing=False)

## 16.4.8.2 Suspension Before Training

If a job is trained on multiple nodes and suspension occurs before the job starts, add **os.environ["NCCL\_DEBUG"] = "INFO"** to the code to view the NCCL debugging information.

## Symptom 1

The job is suspended before the NCCL debugging information is displayed in logs.

## Solution 1

Check the code for parameters such as **master\_ip** and **rank**. Ensure that these parameters are specified.

## Symptom 2

The GDR information is displayed only on certain nodes of a multi-node training job.

21 NCCL INFO Channel 01 : 11(55000) -> 2(55000) [receive] via NET/IB/0/GDRDMA 21 NCCL INFO Channel 01 : 2(55000) -> 0(24000) via P2P/IPC 71 NCCL INFO Channel 01 : 2(55000) -> 0(255000) via P2P/IPC 91 NCCL INFO Channel 01 : 15(5000) -> 10(555000) [send) via NET/IB/0/GDRDMA

The possible cause of the suspension is GDR.

nnel 00 ; 11[51000] -> 15[e9000] via P2P/IPC nnel 00 : 13[be000] -> 9[32000] via P2P/IPC nnel 00 : 3[51000] -> 8[2d000] [receive] via NET/IB/0 nnel 00 : 9[32000] -> 2[5b000] [send] via NET/IB/0 nnel 00 : 9[32000] -> 10[5b000] via P2P/IPC

## Solution 2

Set **os.environ["NCCL\_NET\_GDR\_LEVEL"] = '0'** at the beginning of the program or ask the O&M personnel to add the GDR information to the affected nodes.

#### Symptom 3

Communication information such as "Got completion with error 12, opcode 1, len 32478, vendor err 129" is displayed. The current network is unstable.

### Solution 3

Add the following environment variables:

- NCCL\_IB\_GID\_INDEX=3: enables RoCEv2. RoCEv1 is enabled by default. However, RoCEv1 does not support congestion control on switches, which may lead to packet loss. In addition, later-version switches do not support RoCEv1, leading to a RoCEv1 failure.
- NCCL\_IB\_TC=128: enables data packets to be transmitted through the queue 4 of switches, which is RoCE-compliant.
- NCCL\_IB\_TIMEOUT=22: enables a longer timeout interval. Generally, there is a network interruption lasting about 5s if the network is unstable and then the timeout message is returned. Change the timeout interval to 22s, indicating that the timeout message will be returned in about 20s (4.096 µs x 2 ^ timeout).

## **16.4.8.3 Suspension During Training**

#### Symptom 1

According to the logs of the nodes on which a training job runs, an error occurred on a node but the job did not exit, leading to the job suspension.

## Solution 1

Check the error cause and rectify the fault.

#### Symptom 2

The job is stuck in sync-batch-norm or the training speed is lowered down. If syncbatch-norm is enabled for PyTorch, the training speed is lowered down because all node data must be synchronized on each batch normalization layer in every iteration, which leads to heavy communication traffic.

from sync\_batchnorm import SynchronizedBatchNorm1d, DataParallelWithCallback sync\_bn = SynchronizedBatchNorm1d(10, eps=1e-5, affine=False) sync\_bn = DataParallelWithCallback(sync\_bn, device\_ids=[0, 1])

## Solution 2

Disable sync-batch-norm, or upgrade the PyTorch version to 1.10.

#### Symptom 3

The job is stuck in TensorBoard.

```
writer = SummaryWriter('./path/to/log')
```

#### Solution 3

Set a local path for storage, for example, **cache/tensorboard**. Do not store data in OBS.

#### Symptom 4

When PyTorch dataloader is used to read data, the job is stuck in data reading, and logs stop to update.

```
(5/16 12:01:54)[INFO] logging.py: 95; joon stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:25510;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:25510;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:25540;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:25510;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:25510;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:255119;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "161", "eta": "805:50", "split": "test_iter", "time_diff": 38:25519;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "162", "eta": "805:50", "split": "test_iter", "time_diff": 38:25519;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": "162", "eta": "805:50", "split": "test_iter", "time_diff: 0:53575;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53556;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53556;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53556;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53556;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53558;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53558;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 162", "eta": "006:47", "split": "test_iter", "time_diff: 0:53558;
INFO:timesformer.utils.logging.joon_stats: ("cur_iter": 16
```

## Solution 4

When using dataloader to read data, set num\_work to a small value.



## 16.4.8.4 Suspension in the Last Training Epoch

#### Symptom

Logs showed that an error occurred in split data. As a result, processes are in different epochs, and uncompleted processes are suspended because they do not receive response from other processes. As shown in the following figure, some processes are in epoch 48 while others are in epoch 49 at the same time.

loss exit lane:0.12314446270465851 step loss is 0.29470521211624146 [2022-04-26 13:57:20,757][INFO][train\_epoch]:Rank:2 Epoch:[48][20384/all] Data Time 0.000(0.000) Net Time 0.705(0.890) Loss 0.3403(0.3792)LR 0.00021887 [2022-04-26 13:57:20,757][INFO][train\_epoch]:Rank:1 Epoch:[48][20384/all] Data Time 0.000(0.000) Net Time 0.705(0.891) Loss 0.3028(0.3466) LR 0.00021887 [2022-04-26 13:57:20,757][INFO][train\_epoch]:Rank:4 Epoch:[49][20384/all] Data Time 0.000(0.147) Net Time 0.705(0.709) Loss 0.3364(0.3414)LR 0.00021887 [2022-04-26 13:57:20,758][INFO][train\_epoch]:Rank:3 Epoch:[49][20384/all] Data Time 0.000 (0.115) Net Time 0.706(0.814) Loss 0.3345(0.3418) LR 0.00021887 [2022-04-26 13:57:20,758][INFO][train\_epoch]:Rank:0 Epoch:[49][20384/all] Data Time 0.000(0.006) Net Time 0.704(0.885) Loss 0.2947(0.3566) LR 0.00021887 [2022-04-26 13:57:20,758][INFO][train\_epoch]:Rank:7 Epoch:[49][20384/all] Data Time 0.001 (0.000) Net Time 0.706 (0.891) Loss 0.3782(0.3614) LR 0.00021887 [2022-04-26 13:57:20,759][INFO][train\_epoch]:Rank:5 Epoch:[48][20384/all] Data Time 0.000(0.000) Net Time 0.706 (0.891) Loss 0.5471(0.3642) LR 0.00021887 [2022-04-26 13:57:20,763][INFO][train\_epoch]:Rank:6 Epoch:[49][20384/all] Data Time 0.000(0.000) Net Time 0.706(0.891) Loss 0.5471(0.3642) LR 0.00021887 [2022-04-26 13:57:20,763][INFO][train\_epoch]:Rank:6 Epoch:[49][20384/all] Data Time 0.000(0.000) Net Time 0.704(0.891) Loss 0.2643(0.3390)LR 0.00021887 stage 1 loss 0.4600560665130615 mul\_cls\_loss loss:0.01245919056236744 mul\_offset\_loss 0.44759687781333923 origin stage2\_loss 0.048592399805784225 loss exit lane:0.10233864188194275

#### Solution

Split tensors to align data.

## **16.4.9 Running a Training Job Failed**

## 16.4.9.1 Troubleshooting a Training Job Failure

#### Symptom

A training job is in **Failed** state.

#### **Cause Analysis and Solution**

- The error "MoxFileNotExistsException(resp, 'file or directory or bucket not found.')" is displayed in the training logs.
  - Cause: The train\_data\_obs directory is not found when MoXing copies files.
  - Solution: Correct the address of the train\_data\_obs directory and restart the training job.

#### NOTICE

Do not delete any objects from the OBS directory while MoXing is downloading them. This will cause the download to fail.

- The error NVIDIA A30 with CUDA capability sm\_80 is not compatible with the current PyTorch installation. The current PyTorch install supports CUDA capabilities sm\_37 sm\_50 sm\_60 sm\_70' is displayed in the training logs.
  - Cause: The CUDA version of the image used by the training job supports only the sm\_37, sm\_50, sm\_60, and sm\_70 accelerator cards. The sm\_80 accelerator card is not supported.
  - Solution: Use a custom image to create a training job and install the target CUDA and PyTorch versions.
- The error "ERROR:root:label\_map.pbtxt cannot be found. It will take a long time to open every annotation files to generate a tmp label\_map.pbtxt." is displayed in the training logs.
- If you use an algorithm that you subscribed to from AI Gallery, make sure the data label is accurate.
- If you use an object detection algorithm, make sure the label box of the data is non-rectangular.

D NOTE

Object detection algorithms support only rectangular label boxes.

- The error "RuntimeError: The server socket has failed to listen on any local network address. The server socket has failed to bind to [::]:29500 (errno: 98 Address already in use). The server socket has failed to bind to 0.0.0.0:29500 (errno: 98 Address already in use)." is displayed in the training logs.
  - Cause: The port number of the training job is not unique.
  - Solution: Change the port number in the code and restart the training job.
- The error "WARNING: root: Retry=7, Wait=0.4, Times tamp=1697620658.6282516" is displayed in the training logs.
  - Cause: The MoXing version is too old.
  - Solution: Contact technical support engineers to upgrade MoXing to 2.1.6 or later.

## 16.4.9.2 An NCCL Error Occurs When a Training Job Fails to Be Executed

#### Symptom

The training job fails to be executed. The training job logs contain NCCL-related errors, such as "NCCL timeout", "RuntimeError: NCCL communicator was aborted on rank 7", "NCCL WARN Bootstrap: no socket interface found", and "NCCL INFO Call to connect returned Connection refused, retrying".

#### **Possible Causes**

NCCL is a library that provides primitives for communication between GPUs. It implements collective communication and point-to-point send/receive primitives. If a training job reports an NCCL error, you can adjust the NCCL environment variables to solve the problem.

### Solution

- 1. Go to the details page of the training job, click the **Logs** tab, and view the NCCL error.
  - If the error message NCCL timeout or RuntimeError: NCCL communicator was aborted on rank 7 is displayed, InfiniBand Verbs times out. Click Rebuild in the upper right corner to create a training job again. Set the environment variable NCCL\_IB\_TIMEOUT to 22. Submit the training job and wait until the job is completed.
  - If the error message NCCL WARN Bootstrap : no socket interface found or NCCL INFO Call to connect returned Connection refused, retrying is displayed, NCCL cannot find the communication network adapter or access the IP address. Check whether the NCCL\_SOCKET\_IFNAME environment variable is set in the training code. This environment

variable is automatically injected by the system and does not need to be set in the training code. After the **NCCL\_SOCKET\_IFNAME** environment variable is removed from the training code, click **Rebuild** in the upper right corner to create a training job again. After the training job is submitted, wait until the job is completed.

- 2. Wait and check whether the status of the training job changes to **Completed**.
  - If yes, no further action is required.
  - If no, contact technical support to check the node status.

### Summary and Suggestions

- The NCCL\_SOCKET\_IFNAME environment variable is used to specify the name of the network adapter for communication.
   NCCL\_SOCKET\_IFNAME=eth0 means that only the eth0 network adapter is used for communication. This environment variable is automatically injected by the system. Because the name of the communication network adapter is not fixed, this environment variable should not be set by default in the training code.
- The NCCL\_IB\_TIMEOUT environment variable is used to control InfiniBand Verbs timeout. The default value used by NCCL is **18**. The value ranges from 1 to 22.

## 16.4.9.3 A Training Job Created Using a Custom Image Is Always in the Running State

### Symptom

A training job created using a custom image is always in the running state.

### **Cause Analysis and Solution**

The log message below indicates that the CPU architecture of the custom image does not match that of the resource pool node.

standard\_init\_linux.go:215: exec user process caused "**exec format error**" libcontainer: container start initialization failed: standard\_init\_linux.go:215: exec user process caused "**exec format error**"

This usually happens when the resource type and specifications are incorrectly set during job creation. For example, a custom image that uses the Arm CPU architecture should have NPU specifications, but x86 CPU or x86 GPU specifications are chosen instead.

## 16.4.9.4 Running a Job Failed Due to Persistently Rising Memory Usage

### Symptom

A training job is in the **Failed** state.

### **Possible Causes**

The memory usage continues to rise, leading to the training job failure.

## Solution

- 1. View the logs and monitoring data of the training job to check whether there are any OOM errors.
  - If yes, go to 2.
  - If there are no OOM errors but the monitoring metrics show anomalies, go to 3.
- 2. Check whether there is any code in the training script that keeps using resources and prevents them from being allocated efficiently.
  - If yes, optimize the code and wait until the job runs properly.
  - If no, either upgrade the resource specifications allocated to the training job or contact technical support.
- 3. Restart the training job. Use CloudShell to log in to the training container to check the memory metrics and see if the memory usage spikes.
  - If yes, check the training job logs generated when the memory usage spikes and improve the relevant code logic to lower the memory consumption.
  - If no, either upgrade the resource specifications allocated to the training job or contact technical support.

## 16.4.10 Training Jobs Created in a Dedicated Resource Pool

# 16.4.10.1 No Cloud Storage Name or Mount Path Displayed on the Page for Creating a Training Job

#### Symptom

On the page for creating a training job, there is no option for the cloud storage and mount path.

#### **Possible Causes**

The network of the target dedicated resource pool is not connected, or no SFS has been created.

#### Solution

In the dedicated resource pool list, click the ID or name of the target resource pool to go to its details page. Click **Configure NAS VPC** in the upper right corner to check whether NAS VPC has been enabled. If the NAS VPC name and NAS subnet ID on the details page are left blank, NAS VPC is not enabled. In this case, enable NAS VPC.

If an error message is displayed after you attempt to enable it, the possible cause is that a VPC peering connection has been created for the VPC. In this case, delete the VPC peering connection and try again.

## 16.4.10.2 Storage Volume Failed to Be Mounted to the Pod During Training Job Creation

## Symptom

The training job remains in the **Creating** state. When you check the events of the training job, error message "Unable to mount volumes for pod xxx ... list of unmounted volumes=[nfs-x]" is displayed.

## Possible Cause

For your SFS Turbo file system to function correctly, it must reside within a VPC network that is interconnected with the network of the dedicated resource pool. This connection is essential to ensure that the SFS can be successfully mounted to any training job executed within the dedicated resource pool. Disconnected network may lead to mounting failure.

## Procedure

1. Go to the training job details page and obtain the SFS Turbo name.

#### Compute Nodes 1 Dedicated resource pool pool Specifications CPU: 8 vCPUs 32GB

sfs-turbo- 4 4ab556ł c0 SFS Turbo 9da8779c.sfsturbo.internal:/ /temp

- Log in to the SFS console, locate the SFS Turbo mounted to the training job, 2. and click it to go to the details page. Obtain the VPC, security group, and endpoint information.
  - VPC: value of VPC
  - Security group: value of Security Group
  - Endpoint: value of **Shared Path** excludes ":/", for example, the shared path is 4ab556b5-d689-44f1-9302-24c09daxxxxc.sfsturbo.internal:/, then the SFS Turbo endpoint is 4ab556b5d689-44f1-9302-24c09daxxxxc.sfsturbo.internal.
- Check whether the VPC CIDR block meets the following requirements: 3.

Requirement 1: To prevent CIDR block conflicts with the dedicated resource pool, the SFS Turbo CIDR block cannot overlap with 192.168.20.0/24 (default CIDR block of the dedicated resource pool). Go to the resource pool details page and check **Network** to obtain the actual CIDR block of the dedicated resource pool.

Requirement 2: To prevent network conflicts with the container, the SFS Turbo CIDR block cannot overlap with 172 CIDR block (used by the container network).

## Figure 16-39 Obtaining SFS Turbo name

- If the requirements are not met, modify the VPC CIDR block of SFS Turbo. The recommended value is 10.*X.X.X*.
- If the requirements are met, go to the next step.
- 4. Check whether the VPC CIDR block of SFS Turbo is limited by a security group rule.

Create a training job in the selected dedicated resource pool without mounting SFS Turbo. Once the job is in the **Running** state, access the **worker-0** instance via Cloud Shell. Execute the command **curl {sfs-turboendpoint}:{port}** to verify if the ports are open. The ports that SFS Turbo requires for inbound traffic are 111, 445, 2049, 2051, 2052, and 20048.

- If yes, modify the security group configurations.
- If there is no such a security group rule, perform the following steps.
- 5. Check whether SFS Turbo is normal.

Create an ECS that uses the same CIDR block as SFS Turbo and mount the SFS Turbo to the ECS. If mounting failed, SFS Turbo is abnormal.

- a. If SFS Turbo is abnormal, contact SFS technical support.
- b. If SFS Turbo is normal, contact ModelArts technical support.

## **16.4.11 Training Performance Issues**

### 16.4.11.1 Training Performance Deteriorated

### Symptom

When a ModelArts algorithm is used for training, it will take more time than expected for training.

#### **Possible Causes**

The possible causes are as follows:

- 1. The job code or training parameters have been modified.
- 2. The GPU hardware for training malfunctions.

#### Solution

- 1. Check whether the training code and parameters have been modified.
- 2. Check whether the allocation of the CPU, memory, GPU, snt9, or Infiniband resources complies with the expectation.
- 3. Use CloudShell to log in to the Linux and check the GPU working status.
  - Run the **nvidia-smi** command to check whether the GPU is working properly.
  - Run the **nvidia-smi -q -d TEMPERATURE** command to check the temperature. If the temperature is too high, the training performance deteriorates.

## **16.5 Inference Deployment**

## **16.5.1 AI Application Management**

## 16.5.1.1 Creating an AI Application Failed

## Fault Locating and Troubleshooting

There are two cases of an AI application creation failure: An error occurred during the AI application creation or API calling; the command for creating an AI application was successfully issued, but the creation failed.

- 1. For case 1, the issue is generally caused by invalid input parameters. In this case, rectify the fault as prompted.
- 2. For case 2, do as follows to rectify the fault:
  - On the AI application details page, view the events on the **Events** tab page. Analyze the failure cause based on the events and rectify the fault.
  - If the AI application is in the state of a building failure, click View Model Building Log on the Events tab page on the AI application details page. The building log provides details about the failure. Rectify the fault based on the cause.

#### Figure 16-40 View Model Building Log

Parameter Configu	ration Runtime Dependency Events Constraint Associated	Services
<ol> <li>Note: Events are sa</li> </ol>	aved for three months and will be automatically cleared thereafter, View Model Building Log	Oct,21,2022 14:40:07 - Oct,22,2022 14:40:07
Event Type	Event Message	First Occurred On ↓Ξ
🙆 Abnormal	Falled to build the image. For details, view the building log.	Oct 21, 2022 14:26:32 GMT+08:00
🔕 Abnormal	The status of the image building task is ERROR.	Oct 21, 2022 14:26:32 GMT+08:00
Normal	Start the image building task.	Oct 21, 2022 14:26:05 GMT+08:00
🕑 Normal	Model imported successfully.	Oct 21, 2022 14:26:05 GMT+08:00

## **Common Issues**

1. Dockerfiles are not allowed in a model file directory.

According to model building logs, "Not only a Dockerfile in your OBS path, please make sure, The dockerfile list" is displayed, indicating that the file directory is incorrect and that the file should be removed from the directory.

#### Figure 16-41 Error message for an incorrect Dockerfile directory

	Enter a keyword.
download obs file	
uccessfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef	-a8a43238ae78/Dockerfile from OB5
successfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef 185	-a8a43238ae78/model/Dockerfile from
Successfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef 185	-a8a43238ae78/model/config.json from
Successfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef	-
18a43238ae78/model/customize_service.py from OBS	
Successfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef	-a8a43238ae78/model/saved_model.pb
rom OBS	
Successfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef	-
.8a43238ae78/model/variables/variables.data-00000-of-00001 from OBS	
Successfully to download file cnnorth7-infer-model0/7b8e1fee-992b-475f-a8ef	-
18a43238ae78/model/variables/variables.index from OBS	
Download OBS file successfully!	
docker_login	
uccessful to login the SWR, current time is 2022-10-21-14-26, region name	is cn-north-7
Successful to login the SWR, current time is 2022-10-21-14-26, region name	is cn-north-7
Successful to login the SWR, current time is 2022-10-21-14-26, region name	is cn-north-7
======================================	
lot only a Dockerfile in your OBS path, please make sure, The dockerfile li	st:
/7b8e1fee-992b-475f-a8ef-a8a43238ae78/model/Dockerfile ./7b8e1fee-992b-475	f-a8ef-a8a43238ae78/Dockerfile

2. The pip software package version is different from the version recorded in logs.

#### Figure 16-42 Incorrect pip software package version

Model Building Log		5
	Enter a keyword.	Q
<pre>CI91m WARNING: The scripts pip, pip2 and pip2.7 are installed in '/home/modelarts/.1 PATH. Consider adding this directory to PATH or, if you prefer to suppress this warning, u location. I[OmSuccessfully installed pip-20.3.4 Removing intermediate container 22a58ad6fad4&gt; 11b9323899e Step 3/3 : RUN pip installuser -i xxx&gt; Running in 40f0afcf6dac CI91mWARNING: pip is being invoked by an old script wrapper. This will fail in a fut Please see xx To avoid this problem you can invoke Python with '-m pip' instead of running pip dir C[0mE[91mDEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. P as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in about Python 2 support in pip can be found at xxx C[0mLooking in indexes: xxx C[91mERROR: Could not find a version that satisfies the requirement Pillow==10.2.0 ( 1.2. 1.3. 1.4. 1.5. 1.6. 1.7.0, 1.7.1, 1.7.2, 1.7.3, 1.7.4, 1.7.5, 1.7.6, 1.7.7, 1.7 2.2.1, 2.2.2, 2.3.0, 2.3.1, 2.3.2, 2.4.0, 2.5.0, 2.5.1, 2.5.2, 2.5.3, 2.6.0, 2.6.1, 2.8.2, 2.9.0, 3.0.0, 3.1.0rc1, 3.1.0, 3.1.1, 3.1.2, 3.2.0, 3.3.0, 3.3.1, 3.3.2, 3.3. (.0.0, 4.1.0, 4.1.1, 4.2.0, 4.2.1, 4.3.0, 5.0.0, 5.1.0, 5.2.0, 5.3.0, 5.4.1), </pre>	cal/bin' which is no seeno-warn-script- ure version of pip. ectly. lease upgrade your Py January 2021. More de from versions: 1.0, 1 .8, 2.0.0, 2.1.0, 2.2 2.6.2, 2.7.0, 2.8.0, 3 3, 3.4.0, 3.4.1, 3.4.1 6.0.0, 6.1.0, 6.2.0, 0	t on thon tails .1, .0, 2.8.1, 2, 6.2.1,
<pre>6.2.2) [JOm:[GIMERROR: No matching distribution found for Pillow==10.2.0 [[OmThe command '/bin/sh -c pip installuser -i xxx Failed to build acc53770-95bf-443a-8431-1b5a151fe7e3:0.0.1 image after 1th attempt</pre>	***	
Sending build context to Docker daemon 175.8MB Step 1/3 : FROM swr.cn-north-7.myhuaweicloud.com/op_svc_modelarts_container2/tfservi	ng-model-	

3. Error message "exec /usr/bin/sh: exec format error" is displayed in model building logs.

This issue is generally due to the inconsistency between the used system engine and the system engine for creating the image. For example, an x86 image is used but it is displayed as Arm. View the configured system engine on the AI application details page.

# 16.5.1.2 Failed to Build an Image or Import a File When an IAM user Creates an AI Application

## Symptom

• When an IAM user creates an AI application, creating an image failed. The failure log indicates that downloading the OBS file failed.

 When an IAM user creates an AI application, either of the following prompts are displayed: Failed to copy model file due to obs exception. Please Check your obs access right. and User %s does not have obs:object:PutObjectAcl permission. The AI application fails to be created due to OBS import exceptions or permission issues.

## **Possible Causes**

Using ModelArts requires OBS authorization. ModelArts users require OBS system permissions. The IAM permissions of an IAM user are configured by their tenants. If a tenant does not grant the OBS **putObjectAcl** permission to their IAM users, this issue occurs.

## Solution

 Log in to the IAM console, choose Permissions > Policies/Roles, and click Create Custom Policy in the upper right corner to create a custom policy.

IAM	Policies/Roles ③				Feedback     Create Custom Policy
Users	Delete Custom policies available for creation: 27			All policies/toles	Enter a policy name, role name, or description.     Q
User Groups	Policy/Role Name	Type	Description		Operation
Permissions .		Custom policy			Modify Delete
Pairies/Bales		Custom policy			Modily Delete
Projects		Custom policy			Modity Delete
Agencies		Custom policy	-		Modity Delete
Identity Providers		Custom policy			Modify Delete
Security Settings		Custom policy			Modily Delete
		Custom policy			Modify Delete

<pre>* Policy Name obs_custom Policy View Vasual editor JSON * Policy Content</pre>	icies/Roles / Create Ci	ustom Policy n policies to supplement system-defined policies for fine-grained permissions management. Learn more
Policy View Visual editor JSON * Policy Content * Policy Content * Doisy Octive * Doisy Octive * Constructed * Co	* Policy Name	obs_custom
* Policy Content	Policy View	Visual editor JSON
	* Policy Content	<pre>1 '( "Version": "1.1", 3 '( "Statement": [</pre>

Figure 16-44 Creating a custom policy

An example custom policy is as follows:



2. Assign custom policy permissions to the user group to which the IAM user belongs.

Figure 16-45 Assigning permissions to an IAM user

IAM	User Groups / beijing4-test						Delete
Users User Groups	Name beijing4-test ≟	2 Oroup ID	5389699918744504803346982050042				
Permissions •	Description 🖉	Created	Apr 07, 2022 07:28:07 GM1+08:00				
Projects							
Agencies	Permissions Users						Go to Old Edition
identity Providers	Dalata Authorize	Authorization records (IAM projects): 1; (enterprise p	rojects): 0	User group name: beijing 🔘	Search by policy/tole name.	Q By IAM Project	By Enterprise Project
Security Settings	Policy/Role	Policy/Role Description	Project [Region]	Principal	Principal Description	Principal Type	Operation
	obs_custom		All resources (Existing and future projects)	beijing4-test		User Group	Delete

# 16.5.1.3 Obtaining the Directory Structure in the Target Image When Importing an AI Application Through OBS

## Symptom

When I create an AI application, customized files and folders are stored in the OBS directory specified by a meta model source, and these files and folders will be copied to the target image. What is the path to the copied files and folders?

When an AI application is imported through OBS, ModelArts copies all files and folders in the specified OBS directory to a path specified in the image. You can obtain the path in the image by using **self.model\_path**.

## Solution

For details about how to obtain the path in an image, see **Specifications for Model Inference Coding**.

## 16.5.1.4 Failed to Obtain Certain Logs on the ModelArts Log Query Page

#### Symptom

I used a base image to import AI applications through OBS and wrote some inference code for implementing the inference logic. After an error occurred, I attempted to use the fault logs to locate the fault. However, certain logs were not displayed on the log query page in ModelArts.

#### Possible Causes

To display the logs of an inference service, print the logs on the console through coding. Python logging used by inference base images allows the display of only warning logs. To display INFO logs, set the log level to INFO in the code.

#### Solution

In the PY file for the inference code, set the default level of logs output to the console to **INFO**. The example code is as follows:

import logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(name)s - %(levelname)s - %(message)s')

# 16.5.1.5 Failed to Download a pip Package When an AI Application Is Created Using OBS

### Symptom

Creating an AI application using OBS failed. Logs showed that downloading the pip package failed, for example, downloading the NumPy 1.16 package failed.

#### **Possible Causes**

Possible causes are as follows:

- 1. The package is not available in the pip source. The default pip source is pypi.org. Check whether the package of the target version is available in pypi.org and check the package installation restrictions.
- 2. The downloaded package does not match the architecture in the base image. For example, an x86 package is downloaded for Arm, or a Python 3 package is downloaded for Python 2. For details about the runtime environment of a base image, see **Available Inference Base Images**.

3. The sequence of configuring package dependencies is incorrect.

## Solution

- 1. Log in to pypi.org and check whether the required installation package is available. If the package is unavailable, use the WHL package and place it into the OBS directory where the model is stored.
- 2. Check whether the installation restrictions and dependencies of the package are met.
- 3. If there are package dependencies, configure the dependencies in a correct sequence. For details, see How Do I Edit the Installation Package Dependency Parameters in a Model Configuration File When Importing a Model?

## 16.5.1.6 Failed to Use a Custom Image to Create an AI application

## Symptom

When I used a custom image to create an AI application, the creation failed.

## Possible Causes

Possible causes are as follows:

- The URL of the image used for importing the AI application is invalid or the image is unavailable.
- SWR operation permissions are not included in the agency authorization configured on ModelArts.
- The IAM user does not obtain SWR operation permissions from the tenant.
- The image used is from another account.
- The image used is a public image.

### Solution

- 1. Go to the SWR console and check whether the target image is available and whether the URL of the image is the same as the actual one, including the spelling and letter cases in the URL.
- Check whether SWR operation permissions are included in the agency authorization configured on ModelArts. To do so, go to the Global Configuration page on ModelArts and view the authorization details. If no SWR operation permissions are configured, go to the IAM console and grant the permissions to the target agency.

Figure 16-46 Global Configuration

Authorized To 👙	Authorized 👙	Authorizatio ≑	Authorization Content 👙	Creation Time 🍦	Operation
_EI	IAM user	Agency	modelarts	Dec 28, 2023 09:31:54 GMT+08:00	View Permissions Delete

-				
View Permissio	sions			
Username	baisha			
Agency Name	modelarts_agency			
Agency Permission	4 permissions Modify permissions in IAM			
	Name	Туре	Description	
	ModelArts CommonOperations	System-defined policy	Common permissions of ModelArts service, except create, update, dele	
	SWR Admin	System-defined role	Software Repository Admin	
	OBS OperateAccess	System-defined policy	Basic operation permissions to view the bucket list, obtain bucket me	
	Tenant Administrator	System-defined role	Tenant Administrator (Exclude IAM)	
		OK Cancel		

Figure 16-47 Entrance to permissions modification in IAM

#### Figure 16-48 Authorizing an agency

IAM	Agencies / modelants_agency						
Users User Groups	Basic Information Permissions Delete Authorization	records (IAM projects): 4			Agency name: modelarits	Search by policy/ro	Go to Old Edition le name. Q
Permissuris •	Policy/Role	Policy/Role Description	Project [Region]	Principal	Principal Description	Principal Type	Operation
Projects	ModelArts CommonOperations	Common permissions of ModelArts service,ex	All resources (Existing and future projects)	modelarts_agency	Created by ModelArts service.	Agency	Delete
Identity Providers	SWR Admin	Software Repository Admin	All resources (Existing and future projects)	modelarts_agency	Created by ModelArts service.	Agency	Delete
Security Settings	OBS OperateAccess	Basic operation permissions to view the buck	All resources (Existing and future projects)	modelarts_agency	Created by ModelArts service.	Agency	Delete
	Tenant Administrator	Tenant Administrator (Exclude IAM)	All resources [Existing and future projects]	modelarts_agency	Created by ModelArts service.	Agency	Delete

#### 3. Set a private image

Log in to SWR, choose **My Images** in the navigation pane on the left to view image details. Click **Edit** in the upper right corner and set **Type** to **Private**.

Figure 16-49 Changing the image type to private

SWR	My Images / save-notebook		Set Auto Sync Upbaad Image Add Trigger Est Detete
Dashboard My Images Image Resources Organizations Experience Center	Norm Sale Hittopik Type Private Type 2 Dispose Used 9.5 GB	Edit Image Ogarcation worket Name seve-debtook	×
	Tags Description Pull/Push Permissions St Sync Doke Enter a keyword	Type Public Private  Culegory Cither   Description	α ] [C] [Ø]
	Trag     2     1     1     Total Records: 2 < 1 >	938,96 Cance	Updated ©         Operation           even.         May 27, 3203 195 118 007-01 60         Exam Speci_contraved Command: Vie           even.         May 25, 3203 17:30 46 007-01 60         Scien: Spec_contraved Command: Vie

# 16.5.1.7 Insufficient Disk Space Is Displayed When a Service Is Deployed After an AI Application Is Imported

### Symptom

After an AI application is imported, message "No space left on device" is displayed during service deployment.

ModelArts uses containers to deploy services. There are size limitations for containers to run. If the size of your model file, custom file, or system file exceeds the Docker size, a message will be displayed, indicating that the image space is insufficient.

## Solution

The maximum Docker size for a container in a public resource pool is 10 GB, and that for a container in a dedicated resource pool is 30 GB.

If the AI application is imported from OBS or a training job, the total size of the base image, model files, code, data files, and software packages cannot exceed the limit.

If the AI application is imported from a custom image, the total size of the decompressed image and image dependencies cannot exceed the limit.

# 16.5.1.8 Error Occurred When a Created AI Application Is Deployed as a Service

## Symptom

After an AI application is created, an error occurred when it is deployed as a service.

### **Possible Causes**

When an AI application is imported using a custom or base image, many service logics are customized. Any error in the logics will result in a service deployment or prediction failure.

### Solution

1. After deploying a service failed, go to the service details page and view deployment logs to identify the failure cause. (Ensure that standard input and output functions are used for code output. Otherwise, the output will not be displayed on the ModelArts console.) Find the code based on the error in the logs to locate the fault.

# 16.5.1.9 Invalid Runtime Dependency Configured in an Imported Custom Image

### Symptom

When a custom image is imported through an API to create an AI application, the runtime dependency is configured, but the pip dependency package is not properly installed.

An imported custom image does not support the runtime dependency. The system does not automatically install the required pip dependency package.

## Solution

Create a custom image again.

Install the pip dependency package (for example, the Flask dependency package) in the Dockerfile file that is used to create the image.

## 16.5.1.10 Garbled Characters Displayed in an AI Application Name Returned When AI Application Details Are Obtained Through an API

### Symptom

When details about an AI application are obtained through an API, garbled characters are displayed in a returned AI application name (**model\_name**). For example, the AI application name (**model\_name**) is **query\_vec\_recall\_model**, but the name returned from the API is **query\_vec\_recall\_model\_b**.

#### Figure 16-50 Garbled characters in an AI application name

[2022/08/12 00:03:25 GMT+0800] [INFO] ====================================
[2022/08/12 00:03:25 GMT+0800] [INFO] ====================================
[2022/08/12 00:03:25 GMT+0800] [INFO] ====================================
[2022/08/12 00.03:25 GMT+0800] [INFO] Execute user name is cbc. op. user1, user id is 04ef6da7140025321115c01f1d1c5ed6, job id is 6AB
[2022/08/12 00:03:25 GMT+0800] [INFC] Request url is https://modelarts.cn-north-4.myhuaweicloud.com/v1/88
[2022/08/12 00.03:25 GMT+0800] [INFO] Request query param is null
[2022/08/12 00:03:25 GMT+0800] [INFO] Request method is GET
[2022/08/12 00:03:25 GMT+0800] [INFO] Request header is [REST_API_MARK=REST_API_MARK, User-Agent=Dayu]
[2022/08/12 00:03:26 GMT+0800] [INFO] Response body: ['count'\3,"total_count'\0,"models"]['model id:":ca12cbdb-e7eb-4084-9ea3-36c0bd6a83a3","model name";"guery vec recall model b","model version";"0.0.1] "model_type";"TensorFlow","model
lshed", "lenant" "797c8tc3c4dd4f6985d
time]_model_source``custom`_tunab
2", "model_type". "TensorFlow", "model_1 peate_at"-166014739(
efault-ai-project","Install_type"["real-tim
me""query_vec_recall_model_*model_
[*:1660061011767,"workspace_id":"0","
12022/08/12 00:03:26 GMT+08001 [INFO] Job exited successfully.

## **Possible Causes**

If an AI application name contains underscores (\_), these characters must be escaped.

### Solution

Add the **exact\_match** parameter to the request and set the parameter value to **true** to ensure that the returned value of **model\_name** is correct.

# 16.5.1.11 The Model or Image Exceeded the Size Limit for AI Application Import

### Symptom

When an AI application is imported, a prompt says that the model or image exceeds the limit.

If the AI application is imported using OBS or training, the total size of the basic image, model files, code, data files, and downloaded software packages exceeds the limit.

If the AI application is imported using a custom image, the total size of the decompressed image and image dependencies exceeds the limit.

## Solution

Downsize the model or image and import the AI application again.

# 16.5.1.12 A Single Model File Exceeded the Size Limit (5 GB) for AI Application Import

#### Symptom

When an AI application is imported, a prompt says that a single model file exceeded the size limit (5 GB).

#### **Possible Causes**

If dynamic loading is not used, a single model file cannot exceed 5 GB. Otherwise, the AI application fails to be imported.

#### Solution

- Downsize the model file and import the AI application again.
- Use the dynamic loading function to import the AI application.

#### Figure 16-51 Using dynamic loading

## 16.5.1.13 Creating an AI Application Failed Due to Image Building Timeout

#### Symptom

The AI application fails to be created. A message is displayed showing "Model image build task timed out", and no detailed build log is generated.

#### Figure 16-52 Building the model image timed out

Model Building Log					
	Enter a keyword.	Q	]		
Model image build task timed out					

#### Possible Cause

ImagePacker has a timeout limit when building images. The default value is 30 minutes (which may vary in different regions). If building a model image times out, the building task will fail. In this case, the message "Model image build task timed out" is displayed, and no detailed build log is generated.

#### Solution

- Prepare the dependency packages to be downloaded and built beforehand to save time. You can install the running environment dependency using an offline wheel package. When installing the offline wheel package, ensure that the wheel package and model file are stored in the same directory.
- Optimize the model code to improve the efficiency of building model images.

## **16.5.2 Service Deployment**

## 16.5.2.1 Error Occurred When a Custom Image Model Is Deployed as a Real-Time Service

#### Symptom

A model fails to be deployed as a real-time service. On the real-time service details page, the message "failed to pull image, retry later" is displayed on the **Events** tab page while no information is displayed on the **Logs** tab page.

## Solution

This fault is typically caused by the excessive size of the model you have deployed. Do the following:

- Simplify the model, re-import it, and deploy it as a real-time service.
- Purchase a dedicated resource pool and use it to deploy the model as a realtime service.

### 16.5.2.2 Alarm Status of a Deployed Real-Time Service

### Symptom

A deployed real-time service is in the **Alarm** state.

## Solution

The prediction using a real-time service that is in the **Alarm** state may fail. Perform the following operations to locate the fault and deploy the service again:

1. Check whether there are too many prediction requests on the backend.

If you call APIs for prediction, check whether there are too many prediction requests. A large number of prediction requests lead to the alarm state of the real-time service.

2. Check whether the service memory is functional.

Check whether memory overflow or leakage occurs in the inference code.

- Check whether the model is running properly.
   If the model fails, for example, the associated resources are faulty, check inference loas.
- 4. Check whether there is an abnormal amount of instance pods.

If O&M engineers have deleted abnormal instance pods, the alarm "Service error. There are *XXX* abnormal instances." may occur in the event. Once the alarm is displayed, the service automatically starts a new normal instance to restore to the normal state. The process may take a while.

## 16.5.2.3 Failed to Start a Service

## Symptom

After a service is started, the system displays a message, indicating a container startup failure.

Figure 16-53 Service startup failure

Abnormal Service service-fe44-cmy started failed.

### **Possible Causes**

Possible causes are as follows:

- The AI application is faulty and cannot be started.
- The port configured in the image is incorrect.
- The health check is incorrectly configured.
- The model inference code customize\_service.py is incorrectly edited.
- The image fails to be pulled.
- Scheduling failed due to insufficient resources.

### Faulty AI Application

If the image used for creating an AI application is faulty, recreate the image by following the instructions provided in **Creating a Custom Image and Using It to Create an AI Application**. Ensure the image can be started properly and the expected data can be returned through curl on the local host.

## Incorrect Port in the Image

The port enabled in the image is not 8080, or the port enabled in the image is different from the port configured during AI application creation. As a result, the register-agent cannot communicate with the AI application during service deployment. After a certain period of time (20 minutes at most), it is considered that starting the AI application failed.

If this fault occurs, check the port enabled in the custom image code and the port configured during AI application creation. Ensure that the two ports are the same. If you do not specify a port during AI application creation, ModelArts will listen to port 8080 by default. In this case, the port enabled in the custom image code must be 8080.

#### Figure 16-54 Port enabled in the custom image code

```
# host must be "0.0.0.0", port must be 8080
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=8080)
```

#### Figure 16-55 Port configured during AI application creation



### **Incorrect Health Check Configuration**

If health check is enabled in the image, perform the following operations to locate the fault:

• Check whether the health check port runs properly.

If health check is enabled in a custom image, check whether the health check API is functional during image test. For details about how to test an image locally, see **Building a Custom Image and Using It to Create an AI Application**.

• Check whether the health check address configured during AI application creation is the same as the actual one.

If the AI application is created using a base image provided by ModelArts, the health check URL must be **/health** by default.

rigule 10-50 Com	inguining the neatting	
Health Check		
	* Check Mode	HTTP request O Command
	★ Health Check URL	/health
	* Health Check Period	
	* Delay( seconds )	
	* Maximum Failures	

### Figure 16-56 Configuring the health check URL

## Incorrect customize\_service.py

Check service runtime logs to locate the fault and rectify it.

## Pulling an Image Failed

If the service fails to be started and a message is displayed indicating that the image fails to be pulled, see What Do I Do If an Image Fails to Be Pulled When a Service Is Deployed, Started, Upgraded, or Modified?

## Scheduling Failed Due To Insufficient Resources

The service fails to be started, and a message is displayed indicating that resources are insufficient and service scheduling fails. For details, see **What Do I Do If Resources Are Insufficient When a Service Is Deployed, Started, Upgraded, or Modified?**.

### **Insufficient Memory**

The service fails to be started, and a message is displayed indicating that the memory is insufficient. For details, see **What Can I Do if the Memory Is Insufficient?**.

# 16.5.2.4 What Do I Do If an Image Fails to Be Pulled When a Service Is Deployed, Started, Upgraded, or Modified?

#### Possible Causes

The available disk space of the node is smaller than the image size.

### Solution

- 1. Reduce the image size.
- 2. If the problem persists after the image size is reduced, contact the system administrator.

# 16.5.2.5 What Do I Do If an Image Restarts Repeatedly When a Service Is Deployed, Started, Upgraded, or Modified?

## Possible Causes

There is a bug in the container image code.

## Solution

Debug the container image code based on container logs, create the AI application again, and deploy the application as a real-time service.

# 16.5.2.6 What Do I Do If a Container Health Check Fails When a Service Is Deployed, Started, Upgraded, or Modified?

## **Possible Causes**

Calling the container health check API failed. The possible causes are as follows:

- The health check is incorrectly configured for the image.
- The health check is incorrectly configured for the AI application.

### Solution

Check container logs for the cause of the health check failure.

- If the health check is incorrectly configured for the image, debug the code, create an image again and then the AI application, and use the new AI application to deploy the service. For details about how to configure the image health API for an image, see parameter **health** in Specifications for Writing the Model Configuration File.
- If the health check is incorrectly configured for the AI application, create a new AI application or create a version of the existing AI application, correctly configure the health check, and use the new AI application or version to deploy the service. For details about the AI application health check, see parameter **Health Check** in Creating and Importing a Model Image.

# 16.5.2.7 What Do I Do If Resources Are Insufficient When a Service Is Deployed, Started, Upgraded, or Modified?

## Symptom

The service fails to be started, and an error message is displayed, indicating that resources are insufficient and service scheduling fails. ("Schedule failed due to insufficient resources. Retry later." or "ModelArts.3976: No resources are available for the selected specification.")

Usage Guides	Prediction	Configuration Updates	Monitoring	Events 🕗	Logs	Tags	
<ol> <li>Note: Events are</li> </ol>	saved for one mo	nth and will be automatically clear	red thereafter.				
Event Type	Eve	nt Message					Occurrences
Abnormal	Faib	ed to update service, rollback it.					1
O Abnormal	[mc	del-385b 0.0.1] [pool-t4-video-infe	er Schedule failed du	ue to insufficient re	sources. Reti	y later. 0/1 nodes are available: 1	99
<ul> <li>Normal</li> </ul>	Pre	paring environment.					1
Normal	Upo	lating service.					1

- The configured instance specifications are beyond the remaining CPU or memory resources. ("insufficient CPU" / "insufficient memory")
- The disk capacity cannot meet the requirements of the model. ("x node(s) had taint {node.kubernetes.io/disk-pressure: }" / "No space")

#### Solution

When resources are insufficient, ModelArts retries for three times. If resources are released during these retries, the service can be successfully deployed.

If resources are still insufficient after three retries, the service deployment fails. In this case, perform the following operations to resolve this issue:

- If the service is to be deployed in a public resource pool, wait until other users release resources.
- If the service is to be deployed in a dedicated resource pool, select lower container specifications or custom specifications to deploy the service on the premise that the model requirements are met.
- Expand the capacity of the current resource pool before deploying the service. To expand the capacity of the public resource pool, contact the system administrator. To expand the capacity of the dedicated resource pool, refer to **Resizing a Resource Pool**.
- If the disk space is insufficient, try again to schedule the instance to another node. If the disk space of a single instance is still insufficient, contact the system administrator to use proper specifications.

#### **NOTE**

If an AI application imported though a large model is used to deploy the service, ensure that the disk space of the dedicated resource pool is greater than 1 TB (1000 GB).

## 16.5.2.8 Error Occurred When a CV2 Model Package Is Used to Deploy a Real-Time Service

#### Symptom

An error occurred when a CV2 model package is used to deploy a real-time service.

When a meta model is imported from OBS, the service base image is used. However, the base image does not provide the SO data on which CV2 depends. Therefore, ModelArts does not support the import of CV2 model packages from OBS.

## Solution

Use the CV2 model package to create a custom image, upload the custom image to SWR, import a meta model from the container image, and deploy a real-time service. For details about how to create a custom image, see **Creating a Custom Image and Using It to Create an AI Application**.

## 16.5.2.9 Service Is Consistently Being Deployed

### Symptom

A service retains in the **Deploying** state. No obvious error is found in AI application logs.

## **Possible Causes**

The AI application port is typically incorrect. Check whether the port for creating the AI application is correct.

### Solution

Check the AI application port. If it is not configured, the default port 8080 is used. If you have changed the port number in the configuration file of the custom image, configure the correct port number when deploying the AI application.

For details, see **How Do I Change the Default Port to Create a Real-Time Service Using a Custom Image?** 

## 16.5.2.10 A Started Service Is Intermittently in the Alarm State

## Symptom

The traffic for prediction is not heavy, but the following error frequently occurs:

- Backend service internal error. Backend service read timed out
- Send the request from gateway to the service failed due to connection refused, please confirm your service is connectable
- Send the request from gateway to the service failed due to connection timeout, please confirm your service is able to process the new request

## Possible Causes

After a prediction request is sent, the service stops and then starts.

## Solution

Check the image used by the service, identify the cause of the service stop, and rectify the fault. Re-create the AI application and use it to deploy a service.

# 16.5.2.11 Failed to Deploy a Service and Error "No Module named XXX" Occurred

#### Symptom

Deploying a service failed. The system displays error message "No Module named XXX".

## **Possible Causes**

"No Module named XXX" indicates that the dependency module is not imported to the model.

## Solution

Import the required dependency module to the model through inference code.

For example, when you attempt to deploy a PyTorch AI application as a real-time service, the system displays error message "ModuleNotFoundError: No module named 'model\_service.tfserving\_model\_service'". In this case, configure "from model\_service.pytorch\_model\_service import PTServingBaseService" in **customize\_service.py**. Example code:

import log
from model\_service.pytorch\_model\_service import PTServingBaseService

# 16.5.2.12 Insufficient Permission to or Unavailable Input/Output OBS Path of a Batch Service

#### Symptom

- An input/output path is unavailable, and the following error message is displayed: "error\_code": "ModelArts.3551", "error\_msg": "OBS path xxxx does not exist."
- When the access to an input/output path is denied, the following error message is displayed: "error\_code": "ModelArts.3567", "error\_msg": "OBS error occurs because Access Denied."

#### **Possible Causes**

ModelArts.3551: The OBS path for data input or output does not exist.

ModelArts.3567: The OBS path for data input or output is available, but the current account does not have the permission to access the path.

## Solution

ModelArts.3551: Check whether the data input path is available in OBS. If not, create an OBS path as required. If the path is available but the error persists, submit a service ticket to apply for technical support.

ModelArts.3567: You can access only the OBS path under your own account. To read the OBS data of other users through ModelArts, configure an agency. Otherwise, the access is denied.

Log in to the ModelArts management console. In the navigation pane, choose **Settings**. Click **View Permissions** to check whether the OBS agency permission is configured.

Figure 16-58 Viewing permissions

/iew Permissio	ons			>
Authorized To				
Igency Name	modelarts_agency			
gency Permission	9 permissions Modify permissions in IAM			
	Name	Туре	Description	
	CTS FullAccess	System-defined policy	Full permissions for Cloud Trace Service	
	OBS Administrator	System-defined policy	Object Storage Service Administrator	
	Tenant Administrator	System-defined role	Tenant Administrator (Exclude IAM)	
	IAM AgencyFullAccess	System-defined policy	Full permissions required to manage age	ncies in Identity and Access
	ModelArts CommonOperations	System-defined policy	Common permissions of ModelArts service	e,except create,update,del
	10 • Total Records: 9 < 1 >			
		OK Cancel		

If an agency already exists but the error persists, submit a service ticket for technical support.

## 16.5.2.13 What Can I Do if the Memory Is Insufficient?

#### Symptom

• The deployment or upgrade of a real-time service fails and information similar to the following is displayed in the event.

Figure 16-59 Example 1 of a message indicating insufficient memory



• An alarm is generated for a running service, and the following suggestion is displayed in the event: "Insufficient memory, please increase memory."



Figure 16-60 Example 2 of a message indicating insufficient memory

- If this message is displayed during deployment or upgrade, the memory size of the chosen compute node is insufficient for the application deployment, and you need to increase the memory.
- If an alarm is generated for a running service, memory overflow occurs due to code problems, or the service usage is too large so the memory requirement increases.

## Solution

• When deploying or upgrading a real-time service, select a compute node with larger memory.

#### Figure 16-61 Compute node specifications

* Al Application and Configuration				
	AI Application Source	My AI Applications	My Subscriptions	
	AI Application and Version		∨ 0.0.1	✓ C
	Specifications		~	

• If an alarm is generated for a running service, check whether memory overflow occurs due to code problems, or more memory is required due to heavy service usage. If more memory is required, upgrade the real-time service and select a compute node with larger memory.

## **16.5.3 Service Prediction**

## 16.5.3.1 Service Prediction Failed

### Symptom

After a real-time service is deployed and running, an inference request is sent to the service, but the inference failed.

## **Cause Analysis and Solution**

Service prediction involves multiple phases, including the client, Internet, APIG, dispatcher, and model service. A fault in any phase may lead to a prediction failure.

#### Figure 16-62 Prediction process



1. If an "APIG.XXXX" error occurs, the request is intercepted on API Gateway due to a fault.

Rectify the fault by referring to **Error "APIG.XXXX" Occurred in a Prediction Failure**.

The following shows the other cases in which a request is intercepted on API Gateway:

- Method Not Allowed
- Request Timed Out
- 2. If a "ModelArts.XXXXX" error occurs, the request is intercepted on the dispatcher due to a fault.

Rectify the fault by referring to the methods provided in the following typical cases:

- Error ModelArts.4302 Occurred in Real-Time Service Prediction
- Error ModelArts.4302 Occurred in Real-Time Service Prediction
- Error ModelArts.4503 Occurred in Real-Time Service Prediction
- 3. If an inference image is used and an "MR.XXXX" error occurs, the request has been sent to the model service, and the fault is generally due to a bug in model inference code.

Identify the cause of the prediction failure based on the error information in logs, debug the model inference code, and import the model again for prediction.

Rectify the fault by referring to **Error MR.0105 Occurred in Real-Time Service Prediction**.

- 4. In other cases, check whether the client and the Internet are accessible.
- 5. If the fault persists, contact the system administrator.

### 16.5.3.2 Error "APIG.XXXX" Occurred in a Prediction Failure

A request is intercepted on API Gateway due to a fault, and error "APIG.XXXX" occurs.

Rectify the fault by referring to the methods provided in the following typical cases:

- APIG.0101 Incorrect Prediction URL
- APIG.0201 Request Body Oversized
- APIG.0301 Authentication Failed

For more details about API Gateway error codes and solutions, see .

## APIG.0101 Incorrect Prediction URL

If the prediction URL is incorrect, API Gateway intercepts the request and reports error message "APIG.0101:The API does not exist or has not been published in the environment". In this case, go to the real-time service details page and obtain the correct API address on the **Usage Guides** tab page.

**NOTE** 

If you have specified a custom path in the configuration file, add this path to the called API path. For example, if you have specified custom path **/predictions/poetry**, the called API path will be *{API address}***/predictions/poetry**.

Figure 16-63 Obtaining an API address

API URL https://!

## APIG.0201 Request Body Oversized

If a request body is oversized, API Gateway intercepts the request and reports error message "APIG.0201:Request entity too large". Reduce the prediction request body and try again.

If you perform prediction by calling an API address, the maximum size of the request body is 12 MB. If the size of the request body exceeds 12 MB, the request will be intercepted.

If you perform prediction on the **Prediction** tab of the service details page, the maximum size of the request body is 8 MB. The size limit varies between the two tab pages because they use different network links.

Figure 16-64 Request error APIG.0201



## **APIG.0301** Authentication Failed

If an API is called for service prediction or a token is used for application authentication, a correct token must be obtained. If the token is invalid, API Gateway intercepts the request and reports error message "APIG.0301:Incorrect IAM authentication information: decrypt token fail". Obtain the correct token and enter it in **X-Auth-Token** for prediction.

To obtain a token in a region, obtain the endpoint for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API that is used to obtain a user token. Then, construct the URL as follows:

https://*{iam-endpoint}*/v3/auth/tokens

## 16.5.3.3 Error ModelArts.4206 Occurred in Real-Time Service Prediction

## Symptom

After a real-time service is deployed and running, an inference request is sent to the service, but error ModelArts.4206 occurred.

## **Possible Causes**

ModelArts.4206 indicates that the request traffic on an API exceeded the preset threshold. To ensure stable service running, ModelArts limits the inference request traffic on a single API.

## Solution

Reduce the inference request traffic on an API. If ultra-high concurrency is required, submit a service ticket.

## 16.5.3.4 Error ModelArts.4302 Occurred in Real-Time Service Prediction

### Symptom

After a real-time service is deployed and running, an inference request is sent to the service, but error ModelArts.4302 occurred.

## **Cause Analysis and Solution**

Error ModelArts.4302 may occur in multiple scenarios. The following describes two typical scenarios:

1. "error\_msg": "Gateway forwarding error. Failed to invoke backend service due to connection refused. "

This error occurs in either of the following cases:

- The traffic exceeded the threshold that can be processed by the model. In this case, reduce the traffic or increase the number of model instances.
- The image is faulty. In this case, separately run the image and check whether it is functional.
- 2. "error\_msg": "Due to self protection, the backend service is disconnected, please wait moment."

This error occurs due to excessive number of model errors. A large number of model errors trigger dispatcher circuit breaker, leading to a prediction failure. In this case, check the result returned by the model and handle these errors. Adjust request parameters or reduce the request traffic for higher model calling success rate.

## 16.5.3.5 Error ModelArts.4503 Occurred in Real-Time Service Prediction

### Symptom

After a real-time service is deployed and running, an inference request is sent to the service, but error ModelArts.4503 occurred.

## **Cause Analysis and Solution**

Error ModelArts.4503 may occur in multiple scenarios. The following describes typical scenarios:

1. Communication error

Request error: {"error\_code":"ModelArts.4503","error\_msg":"Failed to respond due to backend service not found or failed to respond"}

To ensure high performance, ModelArts reuses the connections to the same model service. According to the TCP protocol, a disconnection can be initiated either by the client or server of a connection. Disconnecting a connection requires a four-way handshake. If the model service (server) initiates a disconnection, but the connection is being used by ModelArts (client), a communication error occurs and this error code is returned.

If your model is imported from a custom image, set **keep-alive** of the web server used by the custom image to a larger value. This prevents a disconnection request initiated from the server. If you use Gunicorn as the web server, configure the **keep-alive** value by running the **Gunicorn** command. Models imported from other sources have been configured in the service.

2. Protocol error

Request error: {"error\_code":"ModelArts.4503", "error\_msg":"Failed to find backend service because SSL error in the backend service, please check the service is https"}

If the model used for deploying a real-time service is imported from a container image, this error occurs when the protocol used by the container API is incorrectly configured.

For security purposes, all ModelArts inference requests are HTTPS-compliant. When you import a model from a container image, ModelArts allows the image to use HTTPS or HTTP. However, you must specify the protocol used by the image in **Container API**.

### Figure 16-65 Container API

* Meta Model Source	Tr	aining Job	OBS	Container Image	Template		
	1. A mo service client o 2. If the Applica	odel imported fr deployment, Mo r browser . Para e meta model is tion.	om a containe odelArts uses t meters . from a custon	r Image is of the Image type. he image to deploy inference n image, ensure the size of th	Ensure the Ima services. Learn e meta model (	ge can be properly started ar more about image specificat complies with Restrictions on	id provides inference APIs. During ions . Upload an image through a the Image Size for Importing an AI
	*	Container Im	age Path	Select the container Image st	orage path.	6	
	*	Container AP	0		:// {host} :	Port Number	
		Image Replica	ation De	HTTPS HTTP 2 created quickly, but you can rvice deployment.	applications ervice deploy modify or dele	can be created quickly, <b>but n</b> ment. When this function is e te images in the source direct	nodifying or deleting images in nabled, AI applications cannot tory as that would not affect
		Health Check	0				
	0	Add Al App	ication Descrip	otion			

If the **Container API** value is inconsistent with the value provided by your image, for example, **Container API** is set to **HTTPS** but your image actually uses HTTP, the preceding error occurs.

To resolve this issue, create an AI application version, select the correct protocol (HTTP or HTTPS), and deploy a real-time service again or update the existing real-time service.

#### 3. Long prediction time

The following error is reported: {"error\_code": "ModelArts.4503", "error\_msg": "Failed to find backend service because response timed out, please confirm your service is able to process the request without timeout. "}

Due to the limitation of API Gateway, the prediction duration of each request does not exceed 40 seconds. A prediction is successful if the entire process takes a time not longer than the time limit. The process involves sending data to ModelArts, performing prediction, and sending the prediction result back. If a prediction takes a time longer than the time limit or ModelArts cannot respond to frequent prediction requests, this error occurs.

Take the following measures to resolve this issue:

- If a prediction request is oversized, the request times out due to slow data processing. In this case, optimize the prediction code to shorten the prediction time.
- A complex model leads to slow inference. Optimize the model to shorten the prediction time.
- Increase the number of instances or select a compute node flavor with better performance. For example, use GPUs instead of CPUs to improve the service processing performance.
- 4. Service error

The following error is reported: {"error\_code": "ModelArts.4503","error\_msg": "Backend service respond timeout, please confirm your service is able to process the request without timeout. "}

Service logs are as follows:

[2022-10-24 11:37:31 +0000] [897] [INFO] Booting worker with pid: 897 [2022-10-24 11:41:47 +0000] [1997] [INFO] Booting worker with pid: 1997 [2022-10-24 11:41:22 +0000] [1897] [INFO] Booting worker with pid: 1897 [2022-10-24 11:37:54 +0000] [997] [INFO] Booting worker with pid: 997

The service malfunctions and restarts repeatedly. As a result, prediction requests cannot be sent to the service instance.

Take the following measures to resolve this issue:

- Reduce the number of prediction requests and check whether the fault is resolved. If the fault does not recur, the service process exits due to heavy load. In this case, increase the number of instances or improve the instance specifications.
- The inference code is defective. Debug the code to rectify the fault.

## 16.5.3.6 Error MR.0105 Occurred in Real-Time Service Prediction

### Symptom

During the prediction in a running real-time service, error { "erno": "MR.0105", "msg": "Recognition failed", "words\_result": {}} occurred.

iguic					
Status	Running(52 minutes until stop)	Source	My Deployment	4 Failed to obtain the inference result.	×
Failed Calls/Total Calls	1/1 View Details	Description	2		
Custom Settings	0	Traffic Limit			
Usage Guides Pre	liction Configuration Updates Monitoring Events Logs				
Request Path /	Image File     Upload     Predict				
Test Image Preview		Test Result Clear Letter, autor Check the I Check the	<pre>ml_upredict_failure ogs or contact technical support_Predict "= "Mexist", "Recognition failed", =_result": []</pre>		e G

#### Figure 16-66 Prediction failed

## **Possible Causes**

Locate the fault by analyzing the error log on the **Logs** tab of the real-time service details page.

#### Figure 16-67 Error log

model-78ac_0.0.1   All nodes	Support exact search and fu Q
2022-09-06 03:37:29.843491: I tensorflow_serving/core/loader_harness.cc:86] Successfully loaded servable version {name: serve version	n: 1}
2022-09-06 03:37:29.845570: I tensorflow_serving/model_servers/main.cc:323] Running ModelServer at 0.0.0.018500	
2022-09-06 03:37:30 UTC [MainThread ] - /home/mind/model_service/tfserving_model_service.py[line:86] - INFO: Connecting to TensorFic	/ server 127.0.0.1:8500 []
/home/modelarts/.local/iib/python2.7/site-packages/tensorflow_serving/apis/prediction_service_pb2.py:131: Deprecationwarning: Beta_t	Seate_PredictionService_stub() method is deprecated. This method will be removed in near future versions of
IF Serving, Please switch to GA gMPC API in prediction_service_p02_grpc.	
prediction: Service_poz_proc., vepretationwerning/	
2022-09-06 OSIS9121 OC (Maintered ) = //om//mio/app.py/inercos/ = Except Argoniche Chashedi [C/200859898-004210-06/06/06541356]	
Externation of the second sec second second sec	
ras = Vade/Manager model service information dist	
File "/home/mind/model service/model service.mv". line 93. in inference	
data = selfpreprocess(data)	
File "/home/mind/model/1/custom_service.pv". line 17. in _preprocess	
inages, append()	
TypeError: append() takes exactly one argument (0 given)	
[c720d8398943d4d2fd4b7bb7de5a1a38]	
2022-09-06 03:40:12 UTC [MainThread ] - /home/mind/app.py[line:109] - ERROR: ATgorithm crashed! [0a878b87343b84e11df6c886197a577e]	
2022-09-06 03:40:12 UTC [MainThread ] - /home/mind/app.py[line:110] - ERROR: Traceback (most recent call last):	
File "/home/mind/app.py", line 100, in inference_task	
res = ModelManager.model_service.inference(rec_dict)	
File "/home/mind/model_service/model_service.py", line 93, in inference	
data = selfpreprocess(data)	
File "/home/mind/model/1/custom_service.py", line 17, in _preprocess	
inages.append()	
TypeError: append() takes exactly one argument (0 given)	
[0a878b87343b84e11df6c886197a577e]	

According to the error log shown in the preceding figure, the prediction failure is caused by the model inference code.

## Solution

According to the error log, mandatory parameters are missing in the append() method. To rectify the fault, modify the code in the model inference code file **customize\_service.py** to transfer proper parameters to the append() method.

## 16.5.3.7 Method Not Allowed

#### Symptom

Error message "Method Not Allowed" is displayed during service prediction.

The APIs registered by default for service prediction must be called using POST. If you use GET, API Gateway will intercept the request.

### Solution

Use POST to call the API.

### 16.5.3.8 Request Timed Out

#### Symptom

The prediction request times out, and the error {"error\_code": "ModelArts.4205","error\_msg":"Connection time out."} is reported.

## **Possible Causes**

If a request times out, there is a high probability that the request is intercepted by API Gateway. Check the API Gateway and model.

#### Solution

- 1. Run the **:curl -kv** {*Prediction address*} command on the local host to check whether the API Gateway is reachable. If the request timed out, check the local firewall, proxy, and network configurations.
- 2. Check whether the model is started or the duration for the model to process a single request. Due to the limitation of API Gateway, the duration of a single prediction cannot exceed 40s. If the duration exceeds 40s, the system will return a timeout error by default.

## 16.5.3.9 Error Occurred When an API Is Called for Deploying a Model Created Using a Custom Image

If an error occurs when an API is called for service deployment, check the following items:

- 1. Check whether POST is used in the configuration file for the model API.
- 2. Check whether the URL in the configuration file contains a customized path, for example, **/predictions/poetry** (the default path is **/**).
- 3. Check whether the called path in the body of the API request contains a customized path, for example, **{API address}/predictions/poetry**.

## 16.6 MoXing

## 16.6.1 Error Occurs When MoXing Is Used to Copy Data

## Symptom

- 1. When you call **moxing.file.copy\_parallel()** to copy a file from the OBS bucket for a development environment to another bucket, the file is not visible in the target bucket.
- 2. An error occurs when MoXing is used to copy data. Example:
  - The following error occurs when MoXing is used to copy OBS data in the ModelArts development environment: keyError: 'request-id'
  - The error No files to copy occurs when ModelArts uses MoXing to copy data.
  - socket.gaierror: [Errno -2] Name or service not known
  - ERROR:root:Failed to call:

func=<bound method ObsClient.getObject of <obs.client.ObsClient object
at 0x7fd705939710>>

args=('bucket', 'data/TFRecord/HY\_all\_inside/ no\_adjust\_light\_3/09\_06\_6x128x128\_0000000212.tfrecord')

- 3. When MoXing is used to copy data, an error message is displayed, indicating that the operation timed out. Example:
  - TimeoutError: [Errno 110] Connection timed out
  - WARNING:root:Retry=9,Wait=0.1, Timestamp = 1567152567.5327423

### **Possible Cause**

The possible causes are as follows:

- The source file does not exist.
- The target OBS path is incorrect or the two OBS paths are not in the same region.
- Space of the training job is insufficient.

## Solution

Check the following items based on the error message:

- 1. Check whether the first parameter of **moxing.file.copy\_parallel()** contains a file. If it contains no file, the error message "No files to copy" is displayed.
  - If the file exists, go to **2**.
  - If the file does not exist, ignore the error and proceed with subsequent operations.
- 2. Check whether the OBS path where data is copied is in the same region as the development environment or training job.

Log in to the ModelArts management console, and view the region where ModelArts resides. Log in to OBS Console, and view the region where the OBS bucket resides. Check whether they are in the same region.

- If they are in the same region, go to step **3**.

- If they are not in the same region, create a bucket and a folder in OBS that is in the same region as ModelArts, and upload data to the bucket.
- 3. Check whether the OBS path is **obs://xxx**. You can check whether the OBS path exists as follows:

#### mox.file.exists('obs://bucket\_name/sub\_dir\_0/sub\_dir\_1')

- If the path exists, go to 4.
- If the path does not exist, change it to an available OBS path.
- 4. Check whether the used resource is a CPU. The **/cache** directory of the CPU and the code directory share 10 GB. The possible cause is insufficient space. You can run the following command in code to check the disk size:

#### os.system('df -hT')

- If disk space is sufficient, go to 5.
- If disk space is insufficient, use GPU resources.
- 5. If data fails to be copied using MoXing in a notebook instance, run the **df -hT** command on the **Terminal** page to check the space size and check whether the failure cause is insufficient space. You can use EVS to attach disks when creating a notebook instance.

If code is correct but the problem persists, submit a service ticket to get professional technical support.

## 16.6.2 How Do I Disable the Warmup Function of the Mox?

#### Symptom

When the TensorFlow version of the training job Mox is running, 50 steps are executed for four times before the job is formally running.

Warmup indicates a process of using a small learning rate to train several epochs first. Network parameters are randomly initialized. If a large learning rate is used at the beginning, the value may be unstable. This is why warmup is used. After the training process is basically stable, the originally set initial learning rate can be used for training.

### **Possible Cause**

There are multiple execution modes for distributed TensorFlow. Mox executes 50 steps for four times to record the execution time, and selects the model with the minimum execution time.

## Solution

When creating a training job, add **variable\_update=parameter\_server** in **Running Parameter** to disable the warmup function of Mox.

## 16.6.3 Pytorch Mox Logs Are Repeatedly Generated

### Symptom

The Pytorch engine of a frequently-used framework is used as an algorithm source of a ModelArts training job. During the running of the training job, Mox versions

for each epoch will be printed in the Pytorch Mox log. The log details are as follows:

INFO:root:Using MoXing-v1.13.0-de803ac9 INFO:root:Using OBS-Python-SDK-3.1.2 INFO:root:Using MoXing-v1.13.0-de803ac9 INFO:root:Using OBS-Python-SDK-3.1.2

## **Possible Cause**

Pytorch creates multiple processes in spawn mode. Each process invokes the Mox to download data in multi-process mode. In this case, subprocesses are destroyed and recreated repeatedly, and Mox is imported repeatedly. As a result, a large amount of Mox version information is printed.

## Solution

To avoid repeated output of the Pytorch Mox logs of the training job, you need to add the following code to the boot file. When **MOX\_SILENT\_MODE = "1"**, Mox version information can be blocked in the log.

```
import os
os.environ["MOX_SILENT_MODE"] = "1"
```

## 16.6.4 Does moxing.tensorflow Contain the Entire TensorFlow? How Do I Perform Local Fine Tune on the Generated Checkpoint?

### Symptom

When MoXing is used to train a model, **global\_step** is placed in the Adam name range. The non-MoXing code does not contain the Adam name range. See Figure **16-68**. In the figure, **1** indicates MoXing code, and **2** indicates non-MoXing code.

Figure 16-68 Sample code



## Solution

Fine tune is a process of using a model that is trained by others and your own data to train a new model. It is equivalent to using the several top layers of a

model trained by others to extract shallow features, and then making the features fall into our own classification.

Generally, the accuracy of a newly trained model increases gradually from a very low value. However, fine tune allows you to obtain a better effect after a relatively small number of iterations. The advantage of fine tune is that it prevents you from training a model from scratch and improves training efficiency. Fine tune is a good choice when the data volume is not large.

All APIs contained in **moxing.tensorflow** have been optimized for TensorFlow. The actual APIs inside are the native APIs of TensorFlow.

If non-MoXing code does not contain the Adam name range, add the following content to non-MoXing code:

with tf.variable\_scope("Adam"):

When adding code, you are advised to use **tf.train.get\_or\_create\_global\_step()** instead of **global\_step**.

## 16.6.5 Copying Data Using MoXing Is Slow and the Log Is Repeatedly Printed in a Training Job

### Symptom

- Copying data using MoXing is slow in a ModelArts training job.
- The log INFO:root:Listing OBS is repeatedly printed.

#### Figure 16-69 Repeated log printing

INFO:root:Listing	OBS:	77000
INFO:root:Listing	OBS:	78000
INFO:root:Listing	OBS:	79000
INFO:root:Listing	OBS:	80000
INFO:root:Listing	OBS:	81000
INFO:root:Listing	OBS:	82000
INFO:root:Listing	OBS:	83000
INFO:root:Listing	OBS:	84000
INFO:root:Listing	OBS:	85000
INFO:root:Listing	OBS:	86000
INFO:root:Listing	OBS:	87000
INFO:root:Listing	OBS:	88000
INFO:root:Listing	OBS:	89000

### Possible Cause

- 1. The possible causes for slow data copying are as follows:
  - Reading data from OBS will make data reading become a training bottleneck, resulting in slow iteration.
  - Data fails to be read from OBS due to environment or network issues. As a result, the job fails.
- 2. The log is printed repeatedly. The log indicates that the file is being read from the remote end. After the file list is read, data starts to be downloaded. If there are many files, this process takes a long time.
# Solution

When creating a training job, you can save data to OBS. You are advised not to use the OBS APIs of TensorFlow, MXNet, and PyTorch to directly read data from OBS.

- If the file is small, you can save data on OBS as a **.tar** package. When starting the training, download the package from OBS to the **/cache** directory and decompress the package.
- If the file is large, save data as multiple **.tar** packages and invoke multiple processes in the entry script to decompress data in parallel. You are advised not to save discrete files to OBS. Otherwise, data download will be slow.
- In a training job, use the following code to decompress the .tar package: import moxing as mox import os mox.file.copy\_parallel("obs://donotdel-modelarts-test/AI/data/PyTorch-1.0.1/tiny-imagenet-200.tar", '/ cache/tiny-imagenet-200.tar') os.system('cd /cache; tar -xvf tiny-imagenet-200.tar > /dev/null 2>&1')

# 16.6.6 Failed to Access a Folder Using MoXing and Read the Folder Size Using get\_size

# Symptom

- The folder cannot be accessed using MoXing.
- The folder size read by using **get\_size** of MoXing is 0.

### Possible Cause

To use MoXing to access a folder, you need to add the **recursive=True** parameter. The default value is **False**.

### Solution

Obtain the size of an OBS folder.

mox.file.get\_size('obs://bucket\_name/sub\_dir\_0/sub\_dir\_1', recursive=True)

Obtain the size of an OBS file.

mox.file.get\_size('obs://bucket\_name/obs\_file.txt')

# 16.7 APIs or SDKs

# 16.7.1 "ERROR: Could not install packages due to an OSError" Occurred During ModelArts SDK Installation

### Symptom

When ModelArts SDKs are installed, the following error message is displayed: "ERROR: Could not install packages due to an OSError: [WinError 2] The system cannot find the file specified: 'c:\python39\Scripts\ephemeral-port-reserve.exe' -> 'c:\python39\Scripts\ephemeral-port-reserve.exe.deleteme".

# **Possible Causes**

The role of the login user is incorrect.

# Solution

Log in to the system as the administrator, press **Windows+R**, enter **cmd**, and run the following command:

python -m pip install --upgrade pip

# 16.7.2 Error Occurred During Service Deployment After the Target Path to a File Downloaded Through a ModelArts SDK Is Set to a File Name

# Symptom

A ModelArts SDK was used to download a file from OBS, and the target path was set to the file name. No error was reported in the local IDE, but an error occurred when the target AI application was deployed as a real-time service.

Sample code:

session.obs.download\_file (obs\_path, local\_path)

The error message is as follows:

2022-07-06 16:22:36 CST [ThreadPoolEx] - /home/work/predict/model/customize\_service.py[line:184] - WARNING: 4 try: IsADirectoryError(21, 'ls a directory'). update products failed!

### **Possible Causes**

The target path (**local\_path**) was incorrectly set in code.

### Solution

Set **local\_path** to a folder and ensure the folder name extension ends with a slash (/).

# 16.7.3 A Training Job Created Using an API Is Abnormal

# Symptom

When you call an API to create a training job (CPU specifications for the dedicated resource pool), the training job status changes from **Creating** to **Abnormal**, and specifications information on the training job details page is --.

### Possible Causes

A parameter that is not supported by dedicated resource pools of CPU specifications is used in the API call.

# Solution

Make sure that the API request body does not contain **flavor\_id** because this parameter is not supported by dedicated resource pools of CPU specifications

# 16.8 Change History

Released On	Description
2024-01-18	Added: • A Training Job Created Using a Custom Image Is Always in the Running State
	Troubleshooting a Training Job Failure
	• A Training Job Created Using an API Is Abnormal
	Running a Job Failed Due to Persistently Rising     Memory Usage
	Added An NCCL Error Occurs When a Training Job Fails to Be Executed.
2023-11-23	Added An NCCL Error Occurs When a Training Job Fails to Be Executed.
2023-11-08	Added:
	<ul> <li>Failed to Create a Notebook Instance and JupyterProcessKilled Is Displayed in Events</li> </ul>
	• Storage Volume Failed to Be Mounted to the Pod During Training Job Creation
2023-09-07	Added The Model or Image Exceeded the Size Limit for AI Application Import.
	Added A Single Model File Exceeded the Size Limit (5 GB) for AI Application Import.
	Added What Do I Do If an Image Fails to Be Pulled When a Service Is Deployed, Started, Upgraded, or Modified?.
	Added What Do I Do If an Image Restarts Repeatedly When a Service Is Deployed, Started, Upgraded, or Modified?.
	Added What Do I Do If a Container Health Check Fails When a Service Is Deployed, Started, Upgraded, or Modified?.
	Added What Do I Do If Resources Are Insufficient When a Service Is Deployed, Started, Upgraded, or Modified?.
2023-08-31	Deleted "DevEnviron (Notebook of Old Version)".

Released On	Description
2023-08-30	Deleted "OBS Operation Issues" in <i>DevEnviron (New Notebook)</i> and "Why Error: 403 Forbidden Is Displayed When I Perform Operations on OBS?" in <i>General Issues</i> . Moved OBS documentation into General Issues > Incorrect OBS Path on ModelArts.
2022-11-01	Modified the document structure. Added cases related to AI application management. Added <b>service prediction failure</b> cases.
2022-08-31	Added case Error MR.0105 Occurred in Real-Time Service Prediction.
2022-08-26	Added a general OBS case: Incorrect OBS Path on ModelArts
2022-08-15	Added cases related to training job suspension.
2022-01-04	Added OBS download permission cases.
2021-12-15	Added case Error ModelArts.2763 Occurred During Training Job Creation.
2021-09-15	Added training job troubleshooting cases.
2021-07-16	Revised the contents of training job. Deleted an outdated item of troubleshooting from training jobs. Added content of troubleshooting to training jobs. Training Job Process Exits Unexpectedly
2020-12-10	Added the troubleshooting guide for ExeML. Failed to Publish a Dataset Version Invalid Dataset Version Failed to Create an ExeML-powered Training Job ExeML-powered Training Job Failed Failed to Submit the Model Publishing Task Failed to Publish a Model Failed to Submit the Real-time Service Deployment Task Failed to Deploy a Real-time Service
2019-11-25	inis is the first official release.

# **17** Change History

Released On	Description
2024-04-30	This is the first official release.