MindX DL
V100R020C20
User Guide

Issue	02
Date	2021-03-22

HUAWEI TECHNOLOGIES CO., LTD.

Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Product Description....1
1.1 Product Introduction....1
1.1.1 Product Positioning....1
1.1.2 Functions....2
1.2 Application Scenarios....3
1.3 System Architecture....3
1.4 Software Version Mapping....4
1.5 Training Job Model Description....5
2 Installation and Deployment....6
2.1 Before You Start....6
2.1.1 Disclaimer....6
2.1.2 Constraints....6
2.2 Installation Overview....7
2.2.1 Environment Dependencies....7
2.2.2 Networking Schemes....9
2.2.2.1 Logical Networking Scheme....9
2.2.2.2 Typical Physical Networking Scheme....10
2.2.3 Installation Scenarios....10
2.3 MindX DL Installation....11
2.3.1 MindX DL Online Installation....11
2.3.1.1 Preparations for Installation....11
2.3.1.2 Online Installation....13
2.3.2 MindX DL Offline Installation....14
2.3.2.1 Preparations for Installation....14
2.3.2.2 Offline Installation....35
2.3.3 MindX DL Manual Installation....37
2.3.3.1 Preparations for Installation....37
2.3.3.2 Manual Installation (Using Images Downloaded from Ascend Hub)....42
2.3.3.3 Manual Installation (Using Manually Built Images)....45
2.4 Environment Check....49
2.4.1 Checking the Environment Manually....49
2.4.2 Checking the Environment Using a Script....51
2.5 MindX DL Uninstallation....56

 Copyright © Huawei Technologies Co., Ltd.

2.5.1 Automatic Uninstallation....56
2.5.2 Manual Uninstallation....58
2.5.2.1 Clearing Running Resources....58
2.5.2.2 Deleting Component Logs....59
2.5.2.3 Removing a Node from a Cluster....60
2.6 MindX DL Upgrade....60
2.6.1 Preparing for the Upgrade....60
2.6.2 Upgrading MindX DL....62
2.7 Security Hardening....64
2.7.1 Hardening OS Security....64
2.7.2 Hardening Container Security....64
2.7.3 Security Hardening for Ownerless Files....67
2.7.4 Hardening the cAdvisor Monitoring Port....67
2.8 Common Operations....67
2.8.1 Checking the Python and Ansible Versions....68
2.8.2 Installing Python and Ansible....70
2.8.2.1 Installing Python and Ansible Online....70
2.8.2.2 Installing Python and Ansible Offline....72
2.8.3 Configuring Ansible Host Information....85
2.8.4 Obtaining MindX DL Images....90
2.8.5 Building MindX DL Images....95
2.8.6 Modify the Permission of /etc/passwd....99
2.8.7 Installing the NFS....99
2.8.7.1 Ubuntu....99
2.8.7.2 CentOS....101
2.9 User Information....103

3 Usage Guidelines....104

3.1 Instructions....104
3.2 Interconnection Programming Guide....104
3.3 Scheduling Configuration....106
3.4 ResNet-50 Model Use Examples....109
3.4.1 TensorFlow....109
3.4.1.1 Atlas 800 Training Server....109
3.4.1.1.1 Preparing the NPU Training Environment....109
3.4.1.1.2 Creating a YAML File....116
3.4.1.1.3 Preparing for Running a Training Job....121
3.4.1.1.4 Delivering a Training Job....123
3.4.1.1.5 Checking the Running Status....124
3.4.1.1.6 Viewing the Running Result....128
3.4.1.1.7 Deleting a Training Job....129
3.4.1.2 Server (with Atlas 300T Training Cards)....129
3.4.1.2.1 Preparing the NPU Training Environment....129

 Copyright © Huawei Technologies Co., Ltd.

3.4.1.2.2 Creating a YAML File....136
3.4.1.2.3 Preparing for Running a Training Job....141
3.4.1.2.4 Delivering a Training Job....143
3.4.1.2.5 Checking the Running Status....144
3.4.1.2.6 Viewing the Running Result....148
3.4.1.2.7 Deleting a Training Job....149
3.4.2 PyTorch....149
3.4.2.1 Atlas 800 Training Server....149
3.4.2.1.1 Preparing the NPU Training Environment....149
3.4.2.1.2 Creating a YAML File....151
3.4.2.1.3 Preparing for Running a Training Job....153
3.4.2.1.4 Delivering a Training Job....158
3.4.2.1.5 Checking the Running Status....159
3.4.2.1.6 Viewing the Running Result....163
3.4.2.1.7 Deleting a Training Job....164
3.4.2.2 Server (with Atlas 300T Training Cards)....164
3.4.2.2.1 Preparing the NPU Training Environment....164
3.4.2.2.2 Creating a YAML File....166
3.4.2.2.3 Preparing for Running a Training Job....168
3.4.2.2.4 Delivering a Training Job....173
3.4.2.2.5 Checking the Running Status....174
3.4.2.2.6 Viewing the Running Result....178
3.4.2.2.7 Deleting a Training Job....179
3.4.3 MindSpore....179
3.4.3.1 Preparing the NPU Training Environment....179
3.4.3.2 Creating a YAML File....181
3.4.3.3 Preparing for Running a Training Job....185
3.4.3.4 Delivering a Training Job....187
3.4.3.5 Checking the Running Status....187
3.4.3.6 Viewing the Running Result....191
3.4.3.7 Deleting a Training Job....192
3.4.4 Inference Job....192
3.4.4.1 Preparing the NPU Inference Environment....192
3.4.4.2 Creating a YAML File....193
3.4.4.3 Delivering Inference Jobs....194
3.4.4.4 Checking the Running Status....194
3.4.4.5 Deleting an Inference Job....194
3.5 Log Collection....194
3.6 Common Operations....198
3.6.1 Creating an NPU Training Script (MindSpore)....198
3.6.2 Creating a Container Image Using a Dockerfile (TensorFlow)....202
3.6.3 Creating a Container Image Using a Dockerfile (PyTorch)....212

 Copyright © Huawei Technologies Co., Ltd.

3.6.4 Creating a Container Image Using a Dockerfile (MindSpore) 220
3.6.5 Creating an Inference Image Using a Dockerfile 229
3.6.6 Creating the WHL Package of the TensorFlow Framework 232
4 API Reference 236
4.1 Overview 236
4.2 Description 236
4.2.1 API Communication Protocols 236
4.2.2 Encoding Format 236
4.2.3 URLs 237
4.2.4 API Authentication 238
4.2.5 Requests 238
4.2.6 Responses 239
4.2.7 Status Codes 239
4.3 API Reference 241
4.3.1 Data Structure 242
4.3.2 Volcano Job 387
4.3.2.1 Reading All Volcano Jobs Under a Namespace 388
4.3.2.2 Creating a Volcano Job 392
4.3.2.3 Deleting All Volcano Jobs Under a Namespace 397
4.3.2.4 Reading the Details of a Volcano Job 401
4.3.2.5 Deleting a Volcano Job 404
4.3.3 cAdvisor 406
4.3.3.1 Obtaining Ascend AI Processor Monitoring Information 406
4.3.3.2 cAdvisor Prometheus Metrics API 408
4.3.3.3 Other cAdvisor APIs 410
5 FAQs 411
5.1 Pod Remains in the Terminating State After vcjob Is Manually Deleted 411
5.2 The Training Task Is in the Pending State Because "nodes are unavailable" 413
5.3 A Job Was Pending Due to Insufficient Volcano Resources 415
5.4 Failed to Generate the hcl.json File 416
5.5 Calico Network Plugin Not Ready 417
5.6 Error Message "admission webhook "validatejob.volcano.sh" denied the request" Is Displayed When a YAML Job Is Running 418
5.7 Kubernetes Fails to Be Restarted After the Server Is Restarted 419
5.8 Message "certificate signed by unknown authority" Is Displayed When a kubectl Command Is Run 421
6 Communication Matrix 422
A Change History 424

 Copyright © Huawei Technologies Co., Ltd.

1 Product Description

1.1 Product Introduction

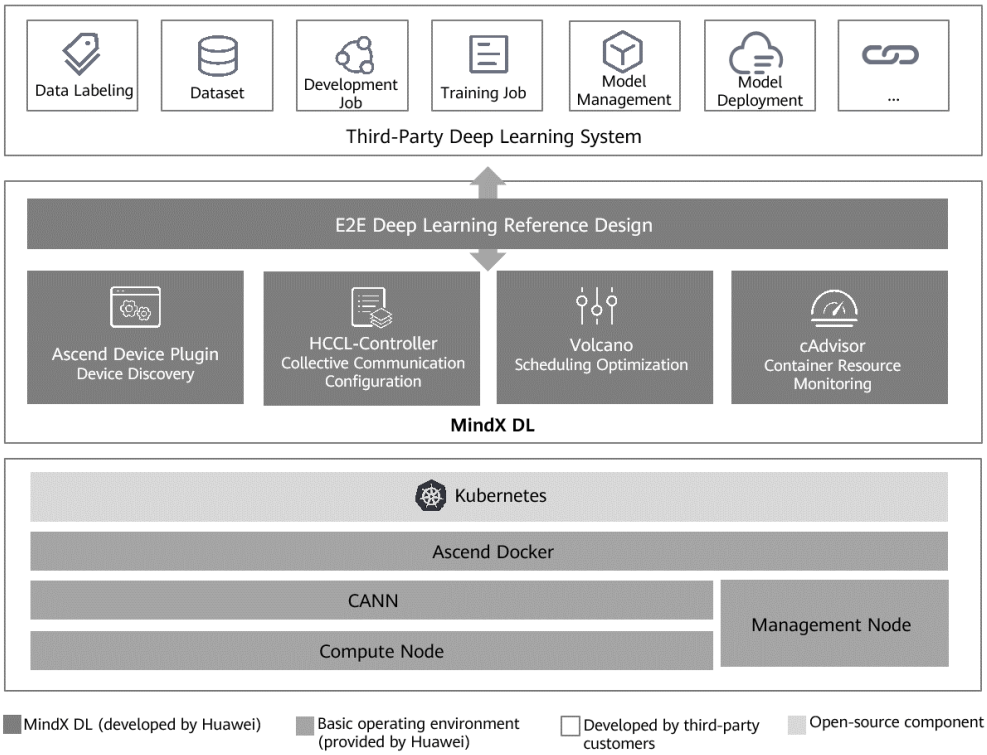
1.1.1 Product Positioning

MindX DL (Ascend deep learning component) is a deep learning component reference design powered by Atlas 800 training servers, Atlas 800 inference servers, servers (with Atlas 300T training cards), and GPU servers. It manages resources and optimizes scheduling for Ascend AI Processors, and generates distributed training collective communication configurations, enabling partners to quickly develop deep learning systems. To obtain the source code and documents, visit **Ascend Developer Community**.

MindX DL components include Ascend Device Plugin, Huawei Collective Communication Library (HCCL)-Controller, Volcano, and Container Advisor (cAdvisor), as shown in **Figure 1-1**.

 Copyright © Huawei Technologies Co., Ltd.

Figure 1-1 Product positioning



1.1.2 Functions

Ascend Device Plugin: Based on the Kubernetes device plug-in mechanism, the device discovery, device allocation, and device health status reporting functions of Ascend AI Processors are added so that the Kubernetes can manage Ascend AI Processor resources.

HCCL-Controller: a Huawei-developed component used for NPU training jobs. It uses the Kubernetes (K8s for short) informer mechanism to continuously monitor NPU training jobs and various events of pods, read the NPU information of pods, and generate the corresponding ConfigMap. ConfigMap contains the HCCL configuration on which training jobs depend, facilitating better collaboration and scheduling of the underlying NPUs. No manual configuration is required.

Volcano: Based on the open-source Volcano cluster scheduling, the affinity scheduling based on the physical topology of the Ascend training card is enhanced to maximize the computing performance of Ascend AI Processors.

cAdvisor: monitors the resource usage and performance characteristics of running containers. In addition to the powerful container resource monitoring function of the open-source cAdvisor, this component can monitor NPU resources in real time and provide APIs and metrics interfaces for you to use with other monitoring software to obtain information such as the Ascend AI Processor usage, frequency, temperature, voltage, and memory in real time.

MindX DL generally consists of the management node, compute nodes, and storage node. The functions of these nodes are as follows:

 Copyright © Huawei Technologies Co., Ltd.

- Management node (Master node): manages clusters, distributes training or inference jobs to each compute node for execution, and implements deep learning functions such as data management, job management, model management, and log monitoring.
- Compute node (Worker node): performs training and inference jobs.
- Storage node: stores datasets and training output models.

Table 1-1 describes the MindX DL component deployment on each node.

Table 1-1 Components deployed on each node

Node	Component	Function
Management node	HCCL-Controller	Plugin developed based on the Kubernetes controller mechanism, which is used to generate the ranktable information of the cluster HCCL.
	Volcano	Enhances the affinity scheduling function of Ascend 910 AI Processors based on the open-source Volcano cluster scheduling.
Compute node	Ascend Device Plugin	Provides the common device plug-in mechanism and standard device APIs for Kubernetes to use devices.
	cAdvisor	Enhances open-source cAdvisor, which monitors Ascend AI Processors.

NOTE

If the management node is also a compute node and has Ascend processors, Ascend Device Plugin and cAdvisor must be installed on the management node.

1.2 Application Scenarios

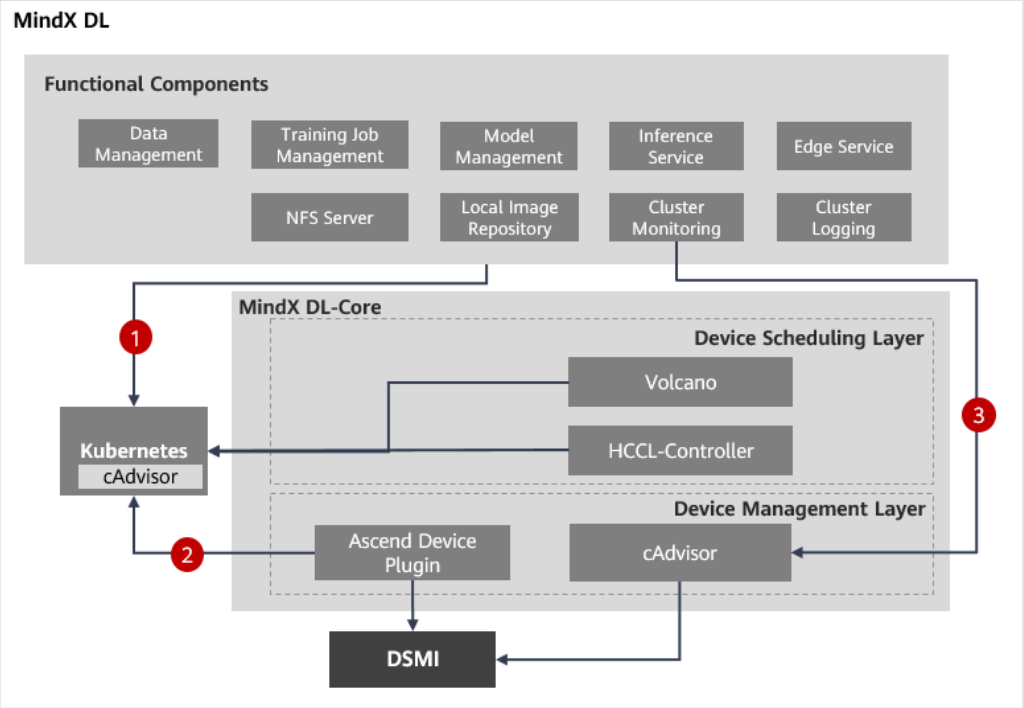
You can use MindX DL components to quickly create NPU training and inference jobs, and build your own deep learning systems based on the basic components.

1.3 System Architecture

Figure 1-2 shows the system architecture of MindX DL.

 Copyright © Huawei Technologies Co., Ltd.

Figure 1-2 System architecture



Based on the Kubernetes ecosystem, MindX DL provides device management and scheduling functions for ISVs to integrate into their own systems.

As shown in **Figure 1-2**, MindX DL provides the following external APIs:

1. vcjob: An external API provided after Volcano is registered with Kubernetes. It is used to add, delete, and query vcjobs. If Volcano is used as the job scheduler, you need to create jobs of the vcjob type.
2. Ascend Device Plugin: A standard API for Kubernetes device plugin to manage Ascend devices. It provides the Register, ListAndWatch, and Allocate APIs.
3. cAdvisor: In compliance with cAdvisor standard API specifications, this API provides the function of obtaining the Ascend AI Processor status.

1.4 Software Version Mapping

Table 1-2 Software Version Mapping

Dependency	Version	Description
Kubernetes	1.17.x	Select the latest bugfix version.
Docker-ce	18.06.3	The Docker version depends on the Kubernetes requirement.

 Copyright © Huawei Technologies Co., Ltd.

Dependency	Version	Description
OS	• Ubuntu 18.04 • CentOS 7.6 • EulerOS 2.8	-
NPU driver	For details, see the **version mapping**.	-
MindX DL	v20.2.0	Component versions: • Volcano: v1.0.1-r40 • HCCL-Controller: v20.2.0 • Ascend Device Plugin: v20.2.0 • cAdvisor: v0.34.0-r40

1.5 Training Job Model Description

Based on the service model design, the training job constraints are as follows:

- Atlas 800 training server
  - The number of NPUs on a training node cannot exceed 8.
  - The number of NPUs applied for a training job is 1, 2, 4, 8, or a multiple of 8.
  - If the number of Ascend 910 AI Processors applied for a training job is less than or equal to 8, only one pod can be applied for. If the number is greater than 8, each pod has eight Ascend 910 AI Processors.
- servers (with Atlas 300T training cards)
  - The number of NPUs on a training node cannot exceed 2.
  - The number of pods that can be applied for is not limited in a distributed training job.
  - Only x86 servers are supported in this version.
  - Only Ubuntu is supported in this version.

Copyright © Huawei Technologies Co., Ltd.

2 Installation and Deployment

2.1 Before You Start

2.1.1 Disclaimer

- This document may include the third-party information covering products, services, software, components, and data. Huawei does not control and assumes no responsibility for the third-party content, including but not limited to the content's accuracy, compatibility, reliability, availability, legitimacy, appropriateness, performance, non-infringement, and status update, unless otherwise specified in this document. Huawei does not provide any guarantee or authorization for the third-party content mentioned or referenced in this document.
- If you need a third-party license, obtain it in an authorized or legal way, unless otherwise specified in this document.

2.1.2 Constraints

- If the space usage of the root directory is higher than 85%, the Kubelet image garbage collection mechanism is triggered and the service is unavailable. Ensure that the root directory has sufficient space. For details about the image garbage collection policy, see the **Kubernetes document**.
- Ensure that the UIDs and GIDs of users **HwHiAiUser** and **hwMindX** on all physical machines (management nodes and compute nodes) and containers are not occupied. If the UIDs and GIDs are occupied, services may be unavailable.
  - The UID and GID of **HwHiAiUser** are both **1000**.
  - The UID and GID of **hwMindX** are both **9000**.
- The default validity period of the Kubernetes certificate is 365 days. Update the certificate before it expires.
- MindX DL images used in the ARM architecture and x86 architecture are incompatible.

 Copyright © Huawei Technologies Co., Ltd.

2.2 Installation Overview

2.2.1 Environment Dependencies

To ensure that MindX DL can be successfully installed, the software and hardware environments must meet certain requirements.

Hardware Requirements

Table 2-1 Hardware requirements

Item	Configuration Reference
Server (single-node)	• ARM: Atlas 800 training server (model 9000) • x86: Atlas 800 training server (model 9010) and servers (with Atlas 300T training cards)
Server (cluster)	Management node: • ARM: TaiShan 200 server (model 2280) • x86: FusionServer Pro 2288H V5
	Compute node: • ARM: Atlas 800 training server (model 9000) and Atlas 800 inference server (model 3000) • x86: Atlas 800 training server (model 9010), Atlas 800 inference server (model 3010), and servers (with Atlas 300T training cards)
	Storage node: storage server
Memory	Management node memory > 64 GB
Storage	> 1 TB For details about the drive space plan, see **Table 2-3**.
Network	• Out-of-band management (BMC): > 1 Gbit/s • In-band management (SSH): > 1 Gbit/s • Service plane: > 10 Gbit/s • Storage plane: > 25 Gbit/s • Parameter plane: 100 Gbit/s

Software

Before installing MindX DL, install the software listed in **Table 2-2**.

Copyright © Huawei Technologies Co., Ltd.

NOTICE

The dependencies of ARM are different from those of x86. Select dependencies based on the system architecture.

Table 2-2 Software environment

Software Package	Version	Installation Position	How to Obtain
OS	• Ubuntu 18.04 • CentOS 7.6 • EulerOS 2.8 **NOTE** • EulerOS 2.8 supports only manual installation of MindX DL. • The servers (with Atlas 300T training cards) supports only Ubuntu 18.04.	All nodes	• Ubuntu and CentOS Log in to the **download address** to obtain the operation guide of the corresponding version. • EulerOS Log in to the **download address** to obtain the operation guide of the corresponding version.
NPU driver	For details, see the **version mapping**.	Compute node	For details, see the ***Driver and Firmware Installation and Upgrade Guides*** of hardware products to obtain the guide of the corresponding version.

After the software is installed, perform the following operation:

Run the following command on a compute node to check the NPU driver version:

/usr/local/Ascend/driver/tools/upgrade-tool --device_index -1 --system_version

For example, if the driver version is 20.2.0, the command output is as follows:

```
{
Get system version(20.2.0) succeed, deviceId(0)
 {"device_id":0, "version":20.2.0}
Get system version(20.2.0) succeed, deviceId(1)
 {"device_id":1, "version":20.2.0}
...
}
```

 Copyright © Huawei Technologies Co., Ltd.

Docker Drive Partitions

Table 2-3 lists the recommended Docker drive partitions.

Table 2-3 Drive space plan

Partition	Description	Partition Format	Size	bootable flag
/boot	Boot partition	EFI	500M	on
/var	Docker and log partition **NOTE** Docker images and logs are stored in the **/var** partition. If the usage of the **/var** partition is greater than 85%, Kubernetes automatically deletes the images. Ensure that the usage of the **/var** partition is less than 85%.	EXT4	> 400G	off
/data	Data partition	EXT4	> 400G	off
/	Primary partition	EXT4	> 100G	off

Environment Check

For details, see **Environment Check**.

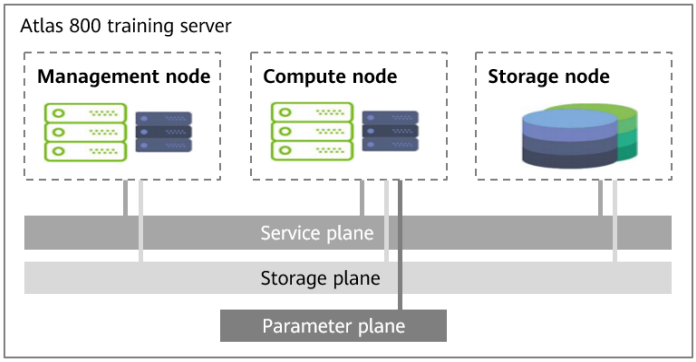
2.2.2 Networking Schemes

2.2.2.1 Logical Networking Scheme

Logical Network of Single-node Deployment

The management node, compute node, and storage node are deployed on the same Atlas 800 training server. **Figure 2-1** shows the logical networking.

Figure 2-1 Single-node deployment

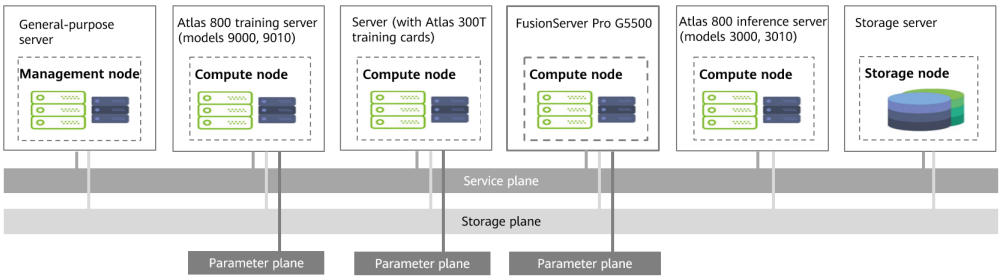


Copyright © Huawei Technologies Co., Ltd.

Logical Network of Cluster Deployment

The management node uses a general-purpose server. The compute nodes consist of multiple Atlas 800 training servers, Atlas 800 inference servers, servers (with Atlas 300T training cards), and GPU training servers. The storage node uses an external storage server. All node networks must be configured in the same network segment. **Figure 2-2** shows the logical networking.

Figure 2-2 Cluster deployment



2.2.2.2 Typical Physical Networking Scheme

2.2.3 Installation Scenarios

This section describes three installation scenarios of MindX DL. You can select a proper installation scenario based on the site requirements.

NOTE

You can download images used during the installation from **Ascend Hub** or by building the source code.

- Three installation scenarios are supported by **Ascend Hub** for downloading images.
- Images obtained by building the source code supports only offline installation and manual installation.

Table 2-4 Installation scenarios

Scenario	Description
MindX DL Online Installation	All nodes must be connected to the Internet. During online installation, Kubernetes, Docker, network file system (NFS), and MindX DL can be deployed using scripts without manual installation.

Copyright © Huawei Technologies Co., Ltd.

Scenario	Description
MindX DL Offline Installation	Offline installation is to prepare the software packages required for installation in advance and distribute the software packages to each node from the management node for installation and deployment. Offline installation improves the reusability and stability of installation and deployment, facilitates the installation of new nodes, and improves the installation efficiency. If a node loses internet access, you can use scripts to deploy Kubernetes, Docker, NFS, and MindX DL in offline mode. **NOTE** For offline installation, you need to prepare the Kubernetes basic image, MindX DL image, and necessary dependencies in advance. The node where the images and dependencies are prepared must be connected to the Internet. Other nodes do not need to be connected to the Internet.
MindX DL Manual Installation	In this scenario, you need to manually install Kubernetes, Docker, NFS, and MindX DL. This document describes how to manually install MindX DL. You need to install other components by referring to their installation guides. **NOTE** • If the MindX DL images required by all nodes are prepared on only one node, only this node needs to be connected to the Internet. • If each node needs to prepare its own MindX DL image, each node needs to be connected to the Internet. • For details about how to install Kubernetes and Docker, see official **Kubernetes** and **Docker** websites, respectively.

2.3 MindX DL Installation

2.3.1 MindX DL Online Installation

2.3.1.1 Preparations for Installation

Prerequisites

- An **OS** has been installed.
- The **NPU driver** has been installed.
- All nodes have access to the Internet.
- User permissions meet the requirements. For details, see **Modify the Permission of /etc/passwd**.

 Copyright © Huawei Technologies Co., Ltd.

Precautions

- The images used for online installation can be automatically downloaded from Ascend Hub. If you want to use manually built images, select the offline installation or manual installation mode.
- When MindX DL is installed, the system automatically creates the **hwMindX** user and the **HwHiAiUser** user group, and adds the user to the user group. The **hwMindX** user is used to run HCCL-Controller and Volcano. The **root** user is used to run other components.

Scripts

Obtain online installation scripts from the MindX DL deployment file repository, as listed in **Table 2-5**.

Link: **Gitee Code Repository**

Table 2-5 Script list

Script Name	Description	Script Path
entry.sh	Provides an entry for online deployment.	deploy/online/steps
set_global_env.yaml	Sets global variables.	deploy/online/steps
online_install_package.yaml	Installs software packages and dependencies.	deploy/online/steps
online_load_images.yaml	Obtains Docker images from Ascend Hub online.	deploy/online/steps
init_kubernetes.yaml	Creates a Kubernetes cluster.	deploy/online/steps
clean_services.yaml	Clears the MindX DL service.	deploy/online/steps
online_deploy_service.yaml	Deploys MindX DL components.	deploy/online/steps
calico.yaml	Kubernetes network plugin configuration file.	yamls

Copyright © Huawei Technologies Co., Ltd.

Script Name	Description	Script Path
ascendplugin-volcano-v20.2.0.yaml	Ascend Device Plugin configuration file for Ascend 910 AI Processor.	yamls
ascendplugin-310-v20.2.0.yaml	Ascend Device Plugin configuration file for Ascend 310 AI Processor.	yamls
cadvisor-v0.34.0-r40.yaml	Configuration file of the NPU monitoring component.	yamls
hccl-controller-v20.2.0.yaml	NPU training job component configuration file.	yamls
volcano-v1.0.1-r40.yaml	NPU training job scheduling component configuration file.	yamls

2.3.1.2 Online Installation

Procedure

Step 1 Log in to the management node as the **root** user.

Step 2 Check whether Python 3.7.5 and Ansible have been installed on the management node.

For details, see **Checking the Python and Ansible Versions**.

Step 3 Configure Ansible host information on the management node.

For details, see **Configuring Ansible Host Information**.

Step 4 Upload the files in the **deploy/online/steps** directory obtained in **Table 2-5** to any directory on the management node, for example, **/home/online_deploy**.

Step 5 On the management node, copy the files in the **yamls** directory in **Table 2-5** to the **dls_root_dir** directory defined in **/etc/ansible/hosts** on the management node. **/tmp** is used as an example of the **dls_root_dir** directory.

The reference directory structure is as follows:

 Copyright © Huawei Technologies Co., Ltd.

```
/tmp
├── ...
└── yamls
    ├── ascendplugin-volcano-v20.2.0.yaml
    ├── ascendplugin-310-v20.2.0.yaml
    ├── calico.yaml
    ├── hccl-controller-v20.2.0.yaml
    ├── cadvisor-v0.34.0-r40.yaml
    └── volcano-v1.0.1-r40.yaml
```

Step 6 Install MindX DL.

1. Go to the **/home/online_deploy** directory in **Step 4** and run the following command to modify the **entry.sh** file:

   chmod 600 entry.sh

   vim entry.sh

   - If you need to install basic dependencies such as Docker, Kubernetes, and NFS, change the value of the scope field in **entry.sh** to **full**, save the modification, and exit.

     ```
     set -e

     scope="full"
     ...
     ```

   - If Docker and NFS have been installed on the node and the Kubernetes cluster has been set up, you only need to install MindX DL. Change the value of the scope field in **entry.sh** to **basic**, save the modification, and exit.

     ```
     set -e

     scope="basic"
     ...
     ```

2. Run the following commands to perform automatic installation:

   dos2unix *

   chmod 500 entry.sh

   bash -x entry.sh

   NOTE

   The ufw firewall is disabled during the installation. If you need to enable the firewall after the installation, see **Hardening OS Security**.

Step 7 Verify the installation. For details, see **Environment Check**.

----End

2.3.2 MindX DL Offline Installation

2.3.2.1 Preparations for Installation

Precautions

- There are ARM and x86 software packages and images. You need to obtain the software packages and images based on the site requirements.
  - The packages whose names end with **arm64.***XXX* are applicable to the ARM architecture.

Copyright © Huawei Technologies Co., Ltd.

  - The packages whose names end with **amd64.***XXX* are applicable to the x86 architecture.
- When MindX DL is installed, the system automatically creates the **hwMindX** user and the **HwHiAiUser** user group, and adds the user to the user group. The **hwMindX** user is used to run HCCL-Controller and Volcano. The **root** user is used to run other components.

Prerequisites

- An **OS** has been installed.
- The **NPU driver** has been installed.
- You have obtained the MindX DL component images.
  - For details about how to obtain images from **Ascend Hub**, see **Obtaining MindX DL Images**.
  - For details about how to manually build images, see **Building MindX DL Images**.
- User permissions meet the requirements. For details, see **Modify the Permission of /etc/passwd**.

Software Packages

Prepare the software packages and dependencies required for the installation, compress them into a .zip package based on the format described in the following table, upload the .zip package to any directory on the management node, and decompress the package. Software packages are classified into Ubuntu and CentOS software packages. You need to obtain software packages as required. *Download paths* in the **How to Obtain** column indicate the paths for storing the downloaded software packages. Change them based on the actual situation.

NOTICE

- NFS, Docker, and Kubernetes offline installation packages are used to install Docker, Kubernetes, and NFS on all nodes. Obtain the software packages based on the actual architecture of each node. If all nodes in the cluster use the same architecture (ARM or x86), you only need to obtain the software packages of the corresponding architecture. If nodes of both architectures exist in the cluster, you need to obtain the software packages of both architectures.
- Ensure that you have the read and write permissions on the directory where the software packages are stored.
- Download the software packages using the methods provided in this document. The software package versions in the table are only examples and may be different from the actual versions, which does not affect the use of the software packages.

- Ubuntu 18.04

 Copyright © Huawei Technologies Co., Ltd.

Table 2-6 NFS, Docker, and Kubernetes offline installation packages

Package Name	Package Content	How to Obtain
offline-pkg-arm64.zip	• conntrack_1%3a1.4.4+snapshot20161117-6ubuntu2_arm64.deb • cri-tools_1.13.0-01_arm64.deb • haveged_1.9.1-6_arm64.deb • keyutils_1.5.9-9.2ubuntu2_arm64.deb • libhavege1_1.9.1-6_arm64.deb • libltdl7_2.4.6-2_arm64.deb • libnfsidmap2_0.25-5.1_arm64.deb • libtirpc-dev_0.2.5-1.2ubuntu0.1_arm64.deb • libtirpc1_0.2.5-1.2ubuntu0.1_arm64.deb • nfs-common_1%3a1.3.4-2.1ubuntu5.3_arm64.deb • nfs-kernel-server_1%3a1.3.4-2.1ubuntu5.3_arm64.deb • rpcbind_0.2.3-0.6ubuntu0.18.04.1_arm64.deb • socat_1.7.3.2-2ubuntu2_arm64.deb • sshpass_1.06-1_arm64.deb	• **cat <<EOF >/etc/apt/sources.list.d/kubernetes.list deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main** **EOF** • **curl -s https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -** • **apt-get update** • **apt-get download conntrack cri-tools haveged keyutils libhavege1 libltdl7 libnfsidmap2 libtirpc-dev libtirpc1 nfs-common nfs-kernel-server rpcbind socat sshpass**
	docker-ce_18.06.3~ce~3-0~ubuntu_arm64.deb	**wget --no-check-certificate https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/arm64/docker-ce_18.06.3~ce~3-0~ubuntu_arm64.deb**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• kubernetes-cni_0.8.6-00_arm64.deb • kubeadm_1.17.3-00_arm64.deb • kubectl_1.17.3-00_arm64.deb • kubelet_1.17.3-00_arm64.deb	**apt-get download kubelet=1.17.3-00 kubeadm=1.17.3-00 kubectl=1.17.3-00 kubernetes-cni=0.8.6-00**
offline-pkg-amd64.zip	• conntrack_1.4.4_amd64.deb • cri-tools_1.13.0-01_amd64.deb • haveged_1.9.1-6_amd64.deb • keyutils_1.5.9-9.2ubuntu2_amd64.deb • libhavege1_1.9.1-6_amd64.deb • libltdl7_2.4.6-2_amd64.deb • libnfsidmap2_0.25-5.1_amd64.deb • libtirpc-dev_0.2.5-1.2ubuntu0.1_amd64.deb • libtirpc1_0.2.5-1.2ubuntu0.1_amd64.deb • nfs-common_1.3.4-2.1ubuntu5.3_amd64.deb • nfs-kernel-server_1.3.4-2.1ubuntu5.3_amd64.deb • rpcbind_0.2.3-0.6ubuntu0.18.04.1_amd64.deb • socat_1.7.3.2-2ubuntu2_amd64.deb • sshpass_1.06-1_amd64.deb	• **cat <<EOF >/etc/apt/sources.list.d/kubernetes.list deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main** **EOF** • **curl -s https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -** • **apt-get update** • **apt-get download conntrack cri-tools haveged keyutils libhavege1 libltdl7 libnfsidmap2 libtirpc-dev libtirpc1 nfs-common nfs-kernel-server rpcbind socat sshpass**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	docker-ce_18.06.3~ce~3-0~ubuntu_amd64.deb	**wget --no-check-certificate https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/docker-ce_18.06.3~ce~3-0~ubuntu_amd64.deb**
	• kubernetes-cni_0.8.6-00_amd64.deb • kubeadm_1.17.3-00_amd64.deb • kubectl_1.17.3-00_amd64.deb • kubelet_1.17.3-00_amd64.deb	**apt-get download kubelet=1.17.3-00 kubeadm=1.17.3-00 kubectl=1.17.3-00 kubernetes-cni=0.8.6-00**

- CentOS 7.6

 Copyright © Huawei Technologies Co., Ltd.

Table 2-7 NFS, Docker, and Kubernetes offline installation packages

Package Name	Folder Name	Package Content	How to Obtain
offline-pkg-arm64.zip	nfs	• gssproxy-0.7.0-29.el7.aarch64.rpm • keyutils-1.5.8-3.el7.aarch64.rpm • libbasicobjects-0.1.1-32.el7.aarch64.rpm • libcollection-0.7.0-32.el7.aarch64.rpm • libevent-2.0.21-4.el7.aarch64.rpm • libini_config-1.3.1-32.el7.aarch64.rpm • libnfsidmap-0.25-19.el7.aarch64.rpm • libpath_utils-0.2.1-32.el7.aarch64.rpm • libref_array-0.1.5-32.el7.aarch64.rpm • libtirpc-0.2.4-0.16.el7.aarch64.rpm • libverto-libevent-0.2.5-4.el7.aarch64.rpm • nfs-utils-1.3.0-0.68.el7.aarch64.rpm • quota-4.01-19.el7.aarch64.rpm • quota-nls-4.01-19.el7.noarch.rpm • rpcbind-0.2.0-49.el7.aarch64.rpm • tcp_wrappers-7.6-77.el7.aarch64.rpm	**yum install --downloadonly --downloaddir=***Download path* **nfs-utils**
	versionlock	yum-plugin-versionlock-1.1.31-54.el7_8.noarch.rpm	**yum install --downloadonly --downloaddir=***Download path* ***yum-plugin-versionlock***

Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	yum-utils	• yum-utils-1.1.31-54.el7_8.noarch.rpm • libxml2-2.9.1-6.el7.5.aarch64.rpm • libxml2-python-2.9.1-6.el7.5.aarch64.rpm • python-chardet-2.2.1-3.el7.noarch.rpm • python-kitchen-1.1.1-5.el7.noarch.rpm	**yum install --downloadonly --downloaddir=***Download path* **yum-utils**
	lvm2	• lvm2-2.02.187-6.el7.aarch64.rpm • lvm2-libs-2.02.187-6.el7.aarch64.rpm • device-mapper-1.02.170-6.el7.aarch64.rpm • device-mapper-event-1.02.170-6.el7.aarch64.rpm • device-mapper-event-libs-1.02.170-6.el7.aarch64.rpm • device-mapper-libs-1.02.170-6.el7.aarch64.rpm	**yum install --downloadonly --downloaddir=***Download path* **lvm2**
	selinux	• distro-1.5.0-py2.py3-none-any.whl • selinux-0.2.1-py2.py3-none-any.whl	**pip3.7 download distro==1.5.0 selinux==0.2.1**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	libselinux	• libselinux-2.5-15.el7.aarch64.rpm • libselinux-python-2.5-15.el7.aarch64.rpm • libselinux-python3-2.5-15.el7.aarch64.rpm • libselinux-utils-2.5-15.el7.aarch64.rpm • libtirpc-0.2.4-0.16.el7.aarch64.rpm • python3-3.6.8-18.el7.aarch64.rpm • python3-libs-3.6.8-18.el7.aarch64.rpm • python3-pip-9.0.3-8.el7.noarch.rpm • python3-setuptools-39.2.0-10.el7.noarch.rpm	**yum install --downloadonly --downloaddir=***Download path* **libselinux-python3**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	docker	• audit-2.8.5-4.el7.aarch64.rpm • audit-libs-2.8.5-4.el7.aarch64.rpm • audit-libs-python-2.8.5-4.el7.aarch64.rpm • checkpolicy-2.5-8.el7.aarch64.rpm • container-selinux-2.119.2-1.911c772.el7_8.noarch.rpm • docker-ce-18.06.3.ce-3.el7.aarch64.rpm • libcgroup-0.41-21.el7.aarch64.rpm • libseccomp-2.3.1-4.el7.aarch64.rpm • libsemanage-python-2.5-14.el7.aarch64.rpm • libtool-ltdl-2.4.2-22.el7_3.aarch64.rpm • policycoreutils-2.5-34.el7.aarch64.rpm • policycoreutils-python-2.5-34.el7.aarch64.rpm • python-IPy-0.75-6.el7.noarch.rpm • setools-libs-3.3.8-4.el7.aarch64.rpm	**yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo** **yum install --downloadonly --downloaddir=***Download path* **docker-ce-18.06.3.ce**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	kubernetes	• *-kubectl-1.17.3-0.aarch64.rpm • *-cri-tools-1.13.0-0.aarch64.rpm • *-kubelet-1.17.3-0.aarch64.rpm • *-kubernetes-cni-0.8.7-0.aarch64.rpm • conntrack-tools-1.4.4-7.el7.aarch64.rpm • *-kubeadm-1.17.3-0.aarch64.rpm • libnetfilter_cthelper-1.0.0-11.el7.aarch64.rpm • libnetfilter_cttimeout-1.0.0-7.el7.aarch64.rpm • libnetfilter_queue-1.0.2-2.el7.aarch64.rpm • socat-1.7.3.2-2.el7.aarch64.rpm	• **cat << EOF > /etc +B2:E65/ yum.repos.d/ kubernetes.repo [kubernetes]** **name=Kubernetes** **baseurl=http:// mirrors.aliyun.com/ kubernetes/yum/repos/ kubernetes-el7-aarch64** **enabled=1** **gpgcheck=1** **repo_gpgcheck=1** **gpgkey=http:// mirrors.aliyun.com/ kubernetes/yum/doc/yum-key.gpg http:// mirrors.aliyun.com/ kubernetes/yum/doc/rpm-package-key.gpg** **EOF** • **yum install --downloadonly --downloaddir=***Download path* **kubelet-1.17.3 kubeadm-1.17.3 kubectl-1.17.**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
			3 --disableexcludes=kubernetes
offline-pkg-amd64.zip	nfs	• gssproxy-0.7.0-28.el7.x86_64.rpm • keyutils-1.5.8-3.el7.x86_64.rpm • libbasicobjects-0.1.1-32.el7.x86_64.rpm • libcollection-0.7.0-32.el7.x86_64.rpm • libevent-2.0.21-4.el7.x86_64.rpm • libini_config-1.3.1-32.el7.x86_64.rpm • libnfsidmap-0.25-19.el7.x86_64.rpm • libpath_utils-0.2.1-32.el7.x86_64.rpm • libref_array-0.1.5-32.el7.x86_64.rpm • libtirpc-0.2.4-0.16.el7.x86_64.rpm • libverto-libevent-0.2.5-4.el7.x86_64.rpm • nfs-utils-1.3.0-0.66.el7_8.x86_64.rpm • quota-4.01-19.el7.x86_64.rpm • quota-nls-4.01-19.el7.noarch.rpm • rpcbind-0.2.0-49.el7.x86_64.rpm • tcp_wrappers-7.6-77.el7.x86_64.rpm	**yum install --downloadonly --downloaddir=***Download path* **nfs-utils**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	versionlock	yum-plugin-versionlock-1.1.31-54.el7_8.noarch.rpm	**yum install --downloadonly --downloaddir=***Download path* **yum-plugin-versionlock**
	yum-utils	• yum-utils-1.1.31-54.el7_8.noarch.rpm • libxml2-2.9.1-6.el7.5.x86_64.rpm • libxml2-python-2.9.1-6.el7.5.x86_64.rpm • python-chardet-2.2.1-3.el7.noarch.rpm • python-kitchen-1.1.1-5.el7.noarch.rpm	**yum install --downloadonly --downloaddir=***Download path* **yum-utils**
	lvm2	• lvm2-2.02.187-6.el7.x86_64.rpm • lvm2-libs-2.02.187-6.el7.x86_64.rpm • device-mapper-1.02.170-6.el7.x86_64.rpm • device-mapper-event-1.02.170-6.el7.x86_64.rpm • device-mapper-event-libs-1.02.170-6.el7.x86_64.rpm • device-mapper-libs-1.02.170-6.el7.x86_64.rpm	**yum install --downloadonly --downloaddir=***Download path* **lvm2**
	selinux	• distro-1.5.0-py2.py3-none-any.whl • selinux-0.2.1-py2.py3-none-any.whl	**pip3.7 download distro==1.5.0 selinux==0.2.1**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	libselinux	• libselinux-2.5-15.el7.x86_64.rpm • libselinux-python-2.5-15.el7.x86_64.rpm • libselinux-python3-2.5-15.el7.x86_64.rpm • libselinux-utils-2.5-15.el7.x86_64.rpm • libtirpc-0.2.4-0.16.el7.x86_64.rpm • python3-3.6.8-13.el7.x86_64.rpm • python3-libs-3.6.8-13.el7.x86_64.rpm • python3-pip-9.0.3-7.el7_7.noarch.rpm • python3-setuptools-39.2.0-10.el7.noarch.rpm	**yum install --downloadonly --downloaddir=***Download path* **libselinux-python3**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	docker	• audit-2.8.5-4.el7.x86_64.rpm • audit-libs-2.8.5-4.el7.x86_64.rpm • audit-libs-python-2.8.5-4.el7.x86_64.rpm • checkpolicy-2.5-8.el7.x86_64.rpm • container-selinux-2.119.2-1.911c772.el7_8.noarch.rpm • docker-ce-18.06.3.ce-3.el7.x86_64.rpm • libcgroup-0.41-21.el7.x86_64.rpm • libseccomp-2.3.1-4.el7.x86_64.rpm • libsemanage-python-2.5-14.el7.x86_64.rpm • libtool-ltdl-2.4.2-22.el7_3.x86_64.rpm • policycoreutils-2.5-34.el7.x86_64.rpm • policycoreutils-python-2.5-34.el7.x86_64.rpm • python-IPy-0.75-6.el7.noarch.rpm • setools-libs-3.3.8-4.el7.x86_64.rpm	• **yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo** • **yum install --downloadonly --downloaddir=***Download path* **docker-ce-18.06.3.ce**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
	kubernetes	• *-kubectl-1.17.3-0.x86_64.rpm • *-cri-tools-1.13.0-0.x86_64.rpm • *-kubelet-1.17.3-0.x86_64.rpm • *-kubernetes-cni-0.8.7-0.x86_64.rpm • conntrack-tools-1.4.4-7.el7.x86_64.rpm • *-kubeadm-1.17.3-0.x86_64.rpm • libnetfilter_cthelper-1.0.0-11.el7.x86_64.rpm • libnetfilter_cttimeout-1.0.0-7.el7.x86_64.rpm • libnetfilter_queue-1.0.2-2.el7_2.x86_64.rpm • socat-1.7.3.2-2.el7.x86_64.rpm • ebtables-2.0.10-16.el7.x86_64.rpm	• **cat << EOF > /etc +B2:E65/ yum.repos.d/ kubernetes.repo [kubernetes]** **name=Kubernetes** **baseurl=http://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64** **enabled=1** **gpgcheck=1** **repo_gpgcheck=1** **gpgkey=http://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg http://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg** **EOF** • **yum install --downloadonly --downloaddir=***Download path* **kubelet-1.17.3 kubeadm-1.17.3 kubectl-1.17.**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Folder Name	Package Content	How to Obtain
			3 --disableexcludes=kubernetes

Scripts

Obtain offline installation scripts from the MindX DL deployment file repository, as listed in **Table 2-8**.

Link: **Gitee Code Repository**

Table 2-8 Script list

Script Name	Description	Script Path
entry.sh	Provides an entry for offline deployment.	deploy/offline/steps
set_global_env.yaml	Sets global variables.	deploy/offline/steps
offline_install_package.yaml	Installs software packages and dependencies.	deploy/offline/steps
offline_load_images.yaml	Imports the required Docker image.	deploy/offline/steps
init_kubernetes.yaml	Creates a Kubernetes cluster.	deploy/offline/steps
clean_services.yaml	Clears the MindX DL service.	deploy/offline/steps
offline_deploy_service.yaml	Deploys MindX DL components.	deploy/offline/steps
calico.yaml	Kubernetes network plugin configuration file.	yamls

Copyright © Huawei Technologies Co., Ltd.

Script Name	Description	Script Path
ascendplugin-volcano-v20.2.0.yaml	Ascend Device Plugin configuration file for Ascend 910 AI Processor.	yamls
ascendplugin-310-v20.2.0.yaml	Ascend Device Plugin configuration file for Ascend 910 AI Processor.	yamls
cadvisor-v0.34.0-r40.yaml	Configuration file of the NPU monitoring component.	yamls
hccl-controller-v20.2.0.yaml	NPU training job component configuration file.	yamls
volcano-v1.0.1-r40.yaml	NPU training job scheduling component configuration file.	yamls

Base Image Packages

Table 2-9 lists base image packages.

NOTICE

The Docker image in **Table 2-9** is used to create a Kubernetes cluster. The images need to be stored on the management node. Obtain the corresponding images based on the actual architecture of each node. If all nodes in the cluster use the same architecture (ARM or x86), you only need to obtain the images of the corresponding architecture. If nodes of both architectures exist in the cluster, you need to obtain the images of both architectures.

To obtain the packages, perform the following operations:

1. Run the following command to obtain and save the image packages:
   docker pull *XXX*
   docker save -o *Image package name Image name:tag*

 Copyright © Huawei Technologies Co., Ltd.

NOTE

docker pull *XXX*: For details about the command to be run, see the official obtaining method in **Table 2-9**.

For example, run the following commands to obtain and save the calico/node:v3.11.3 ARM image:

docker pull *calico/node:v3.11.3*

docker save -o *calico-cni_arm64.tar.gz calico/node:v3.11.3*

2. Save the image packages generated in **1** to the local PC.

Table 2-9 Base image packages

Description	Image Package	Official Obtaining Method
Kubernetes network plugin	• calico-cni_arm64.tar.gz • calico-kube-controllers_arm64.tar.gz • calico-node_arm64.tar.gz • calico-pod2daemon-flexvol_arm64.tar.gz	• **docker pull calico/node:v3.11.3** • **docker pull calico/pod2daemon-flexvol:v3.11.3** • **docker pull calico/cni:v3.11.3** • **docker pull calico/kube-controllers:v3.11.3**
	• calico-cni_amd64.tar.gz • calico-kube-controllers_amd64.tar.gz • calico-node_amd64.tar.gz • calico-pod2daemon-flexvol_amd64.tar.gz	
Kubernetes domain name service	coredns_arm64.tar.gz	**docker pull coredns/coredns:1.6.5**
	coredns_amd64.tar.gz	
Kubernetes cluster database	etcd_arm64.tar.gz	**docker pull cruse/etcd-arm64:3.4.3-0**
	etcd_amd64.tar.gz	**docker pull cruse/etcd-amd64:3.4.3-0**
Kubernetes cluster data center	kube-apiserver_arm64.tar.gz	**docker pull cruse/kube-apiserver-arm64:v1.17.3**
	kube-apiserver_amd64.tar.gz	**docker pull kubesphere/kube-apiserver:v1.17.3**
Kubernetes cluster management controller	kube-controller-manager_arm64.tar.gz	**docker pull cruse/kube-controller-manager-arm64:v1.17.3**

Copyright © Huawei Technologies Co., Ltd.

Description	Image Package	Official Obtaining Method
	kube-controller-manager_amd64.tar.gz	**docker pull kubesphere/kube-controller-manager:v1.17.3**
Kubernetes cluster HCCL and load balancing	kube-proxy_arm64.tar.gz	**docker pull cruse/kube-proxy-arm64:v1.17.3-beta.0**
	kube-proxy_amd64.tar.gz	**docker pull kubesphere/kube-proxy:v1.17.3**
Kubernetes cluster scheduler	kube-scheduler_arm64.tar.gz	**docker pull cruse/kube-scheduler-arm64:v1.17.3-beta.0**
	kube-scheduler_amd64.tar.gz	**docker pull kubesphere/kube-scheduler:v1.17.3**
Kubernetes basic container	pause_arm64.tar.gz	**docker pull cruse/pause-arm64:3.1**
	pause_amd64.tar.gz	**docker pull kubesphere/pause:3.1**

MindX DL Image Packages

Table 2-10 lists image packages.

To obtain the packages, perform the following operations:

1. Obtain the component image. Select one of the following methods based on the site requirements:
   - Obtain images from **Ascend Hub**. For details, see **Obtaining MindX DL Images**.
   - Manually build images. For details, see **Building MindX DL Images**.
2. Run the following command on each node where the MindX DL image exists to obtain the image packages:

   docker save -o *XXX*

   NOTE

   XXX: For details about the commands, see **Table 2-10**.
3. Save the images generated in **2** to the local PC.

Copyright © Huawei Technologies Co., Ltd.

Table 2-10 Image list

Item	Image Package	Obtaining Command	Obtaining Node
MindX DL Kubernetes device plugin	Ascend-K8sDevicePlugin-v20.2.0-arm64-Docker.tar.gz	**docker save -o Ascend-K8sDevicePlugin-v20.2.0-arm64-Docker.tar.gz ascend-k8sdeviceplugin:v20.2.0**	Node where the Ascend Device Plugin image has been obtained
	Ascend-K8sDevicePlugin-v20.2.0-amd64-Docker.tar.gz	**docker save -o Ascend-K8sDevicePlugin-v20.2.0-amd64-Docker.tar.gz ascend-k8sdeviceplugin:v20.2.0**	
MindX DL training job HCCL plugin	hccl-controller-v20.2.0-arm64.tar.gz	**docker save -o hccl-controller-v20.2.0-arm64.tar.gz hccl-controller:v20.2.0**	Node where the HCCL-Controller image has been obtained
	hccl-controller-v20.2.0-amd64.tar.gz	**docker save -o hccl-controller-v20.2.0-amd64.tar.gz hccl-controller:v20.2.0**	
MindX DL device monitoring plugin	huawei-cadvisor-v0.34.0-r40-arm64.tar.gz	**docker save -o huawei-cadvisor-v0.34.0-r40-arm64.tar.gz google/cadvisor:v0.34.0-r40**	Node where the cAdvisor image has been obtained
	huawei-cadvisor-v0.34.0-r40-amd64.tar.gz	**docker save -o huawei-cadvisor-v0.34.0-r40-amd64.tar.gz google/cadvisor:v0.34.0-r40**	

Copyright © Huawei Technologies Co., Ltd.

Item	Image Package	Obtaining Command	Obtaining Node
MindX DL job scheduling plugin	• vc-controller-manager-v1.0.1-r40-arm64.tar.gz • vc-scheduler-v1.0.1-r40-arm64.tar.gz • vc-webhook-manager-base-v1.0.1-r40-arm64.tar.gz • vc-webhook-manager-v1.0.1-r40-arm64.tar.gz	• **docker save -o vc-controller-manager-v1.0.1-r40-arm64.tar.gz volcanosh/vc-controller-manager:v1.0.1-r40** • **docker save -o vc-scheduler-v1.0.1-r40-arm64.tar.gz volcanosh/vc-scheduler:v1.0.1-r40** • **docker save -o vc-webhook-manager-base-v1.0.1-r40-arm64.tar.gz volcanosh/vc-webhook-manager-base:v1.0.1-r40** • **docker save -o vc-webhook-manager-v1.0.1-r40-arm64.tar.gz volcanosh/vc-webhook-manager:v1.0.1-r40**	Node where the Volcano image has been obtained

Copyright © Huawei Technologies Co., Ltd.

Item	Image Package	Obtaining Command	Obtaining Node
	• vc-controller-manager-v1.0.1-r40-amd64.tar.gz • vc-scheduler-v1.0.1-r40-amd64.tar.gz • vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz • vc-webhook-manager-v1.0.1-r40-amd64.tar.gz	• **docker save -o vc-controller-manager-v1.0.1-r40-amd64.tar.gz volcanosh/vc-controller-manager:v1.0.1-r40** • **docker save -o vc-scheduler-v1.0.1-r40-amd64.tar.gz volcanosh/vc-scheduler:v1.0.1-r40** • **docker save -o vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz volcanosh/vc-webhook-manager-base:v1.0.1-r40** • **docker save -o vc-webhook-manager-v1.0.1-r40-amd64.tar.gz volcanosh/vc-webhook-manager:v1.0.1-r40**	

2.3.2.2 Offline Installation

Procedure

Step 1 Log in to the management node as the **root** user.

Step 2 Check whether Python 3.7.5 and Ansible have been installed on the management node.

For details, see **Checking the Python and Ansible Versions**.

Step 3 Configure Ansible host information on the management node.

For details, see **Configuring Ansible Host Information**.

Step 4 Upload the files listed in **Software Packages**, **Base Image Packages**, **MindX DL Image Packages**, and **Table 2-8** (only the files in the **yamls** directory) to the

 Copyright © Huawei Technologies Co., Ltd.

dls_root_dir directory defined in **/etc/ansible/hosts** on the management node. **/tmp** is used as an example of the **dls_root_dir** directory. The folder names must be the same as the following. (A heterogeneous cluster running Ubuntu 18.04 is used as an example.)

The directory structure is as follows:

```
/tmp
├─ docker_images
│   ├── Ascend-K8sDevicePlugin-v20.2.0-amd64-Docker.tar.gz
│   ├── Ascend-K8sDevicePlugin-v20.2.0-arm64-Docker.tar.gz
│   ├── calico-cni_amd64.tar.gz
│   ├── calico-cni_arm64.tar.gz
│   ├── calico-kube-controllers_amd64.tar.gz
│   ├── calico-kube-controllers_arm64.tar.gz
│   ├── calico-node_amd64.tar.gz
│   ├── calico-node_arm64.tar.gz
│   ├── calico-pod2daemon-flexvol_amd64.tar.gz
│   ├── calico-pod2daemon-flexvol_arm64.tar.gz
│   ├── coredns_amd64.tar.gz
│   ├── coredns_arm64.tar.gz
│   ├── etcd_amd64.tar.gz
│   ├── etcd_arm64.tar.gz
│   ├── hccl-controller-v20.2.0-amd64.tar.gz
│   ├── hccl-controller-v20.2.0-arm64.tar.gz
│   ├── huawei-cadvisor-v0.34.0-r40-amd64.tar.gz
│   ├── huawei-cadvisor-v0.34.0-r40-arm64.tar.gz
│   ├── kube-apiserver_amd64.tar.gz
│   ├── kube-apiserver_arm64.tar.gz
│   ├── kube-controller-manager_amd64.tar.gz
│   ├── kube-controller-manager_arm64.tar.gz
│   ├── kube-proxy_amd64.tar.gz
│   ├── kube-proxy_arm64.tar.gz
│   ├── kube-scheduler_amd64.tar.gz
│   ├── kube-scheduler_arm64.tar.gz
│   ├── pause_amd64.tar.gz
│   ├── pause_arm64.tar.gz
│   ├── vc-controller-manager-v1.0.1-r40-amd64.tar.gz
│   ├── vc-controller-manager-v1.0.1-r40-arm64.tar.gz
│   ├── vc-scheduler-v1.0.1-r40-amd64.tar.gz
│   ├── vc-scheduler-v1.0.1-r40-arm64.tar.gz
│   ├── vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz
│   ├── vc-webhook-manager-base-v1.0.1-r40-arm64.tar.gz
│   ├── vc-webhook-manager-v1.0.1-r40-amd64.tar.gz
│   └── vc-webhook-manager-v1.0.1-r40-arm64.tar.gz
├─ offline-pkg-amd64.zip
├─ offline-pkg-arm64.zip
└─ yamls
    ├── ascendplugin-volcano-v20.2.0.yaml
    ├── ascendplugin-310-v20.2.0.yaml
    ├── calico.yaml
    ├── hccl-controller-v20.2.0.yaml
    ├── cadvisor-v0.34.0-r40.yaml
    └── volcano-v1.0.1-r40.yaml
```

Step 5 Upload the files (listed in **Table 2-8**) in the **deploy/offline/steps** directory to any directory on the management node, for example, **/home/offline_deploy**.

Step 6 Install MindX DL.

1. Go to the **/home/offline_deploy** directory in **Step 5** and run the following commands in the directory to modify the **entry.sh** file:

   chmod 600 entry.sh

   vim entry.sh

   – If you need to install basic dependencies such as Docker, Kubernetes, and NFS, change the value of the **scope** field in **entry.sh** to **full**, save the modification, and exit.

 Copyright © Huawei Technologies Co., Ltd.

```
set -e

scope="full"
...
```

– If Docker and NFS have been installed on the node and the Kubernetes cluster has been set up, you only need to install MindX DL. Change the value of the **scope** field in **entry.sh** to **basic**, save the modification, and exit.

```
set -e

scope="basic"
...
```

2. Run the following commands to perform automatic installation:

   dos2unix *

   chmod 500 entry.sh

   bash -x entry.sh

   NOTE

   The ufw firewall is disabled during the installation. If you need to enable the firewall after the installation, see **Hardening OS Security**.

Step 7 Verify the installation. For details, see **Environment Check**.

----End

2.3.3 MindX DL Manual Installation

2.3.3.1 Preparations for Installation

Prerequisites

A Kubernetes cluster has been set up.

Component Installation Positions

Table 2-11 shows the installation positions of the components.

Table 2-11 Component installation positions

Node	Component	Function
Management node HCCL-Controller	HCCL-Controller	Plugin developed based on the Kubernetes controller mechanism, which is used to generate the ranktable information of the cluster HCCL.
	Volcano	Open-source Volcano cluster scheduling, which enhances the affinity scheduling function of Ascend 910 AI Processors.

 Copyright © Huawei Technologies Co., Ltd.

Node	Component	Function
Compute node cAdvisor	Ascend Device Plugin	Provides the common device plug-in mechanism and standard device APIs for Kubernetes to use devices.
	cAdvisor	Enhances open-source cAdvisor to monitor Ascend AI Processors.

Creating a Node Label

Run the following commands on the management node to label the corresponding node so that MindX DL can schedule work nodes of different forms:

NOTICE

In a single-node system, the management node and compute node are the same node, which should be labeled.

Table 2-12 Labeling commands

Label Number	Command	Description
1	**kubectl label nodes** *Hostname* **node-role.kubernetes.io/worker=worker**	Identifies Kubernetes compute nodes. *Hostname* indicates the names of all compute nodes.
2	**kubectl label nodes** *Hostname* **accelerator=huawei-Ascend910**	Identifies Ascend 910 AI Processor nodes. *Hostname* indicates the names of all training nodes.
3	**kubectl label nodes** *Hostname* **accelerator=huawei-Ascend310**	Identifies Ascend 310 AI Processor node. *Hostname* indicates the names of all inference nodes.
4	**kubectl label nodes** *Hostname* **masterselector=dls-master-node**	Identifies the MindX DL management node. *Hostname* indicates the name of the management node.
5	**kubectl label nodes** *Hostname* **workerselector=dls-worker-node**	Identifies MindX DL compute nodes. *Hostname* indicates the names of all compute nodes.

Copyright © Huawei Technologies Co., Ltd.

Label Number	Command	Description
6	**kubectl label nodes** *Hostname* **host-arch=huawei-x86**	Identifies x86 nodes. *Hostname* indicates the names of all x86 compute nodes.
7	**kubectl label nodes** *Hostname* **host-arch=huawei-arm**	Identifies ARM nodes. *Hostname* indicates the names of all ARM compute nodes.
8	**kubectl label nodes** *Hostname* **accelerator-type=card**	Identifies servers (with Atlas 300T training cards). *Hostname* indicates the names of all compute nodes of the servers (with Atlas 300T training cards).

NOTE

The rules for creating a node label are as follows:

- The label numbers that may be used by an ARM management node are as follows:
  - 4: The node functions only as a management node in a cluster.
  - 1, 2, 4, 5, and 7: The node that is equipped with Ascend 910 AI Processors functions as both a management node and a compute node in a single-node system.
- The label numbers that are used by an x86 compute node with Ascend 310 AI Processors are 1, 3, 5, and 6.
- The label numbers that must be used by an x86 compute node with Atlas 300T training cards are 1, 2, 5, 6, and 8.

Creating a User

NOTICE

Run HCCL-Controller and Volcano as user **hwMindX**. Run the Ascend Device Plugin and cAdvisor as user **root**.

Run MindX DL as the **hwMindX** user and run the following commands to create a user:

useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX

usermod -a -G HwHiAiUser hwMindX

Creating MindX DL Log Paths

You need to manually create MindX DL log paths. **Table 2-13** lists the log paths.

Copyright © Huawei Technologies Co., Ltd.

Table 2-13 MindX DL log path list

MindX DL Component Name	Component Log Path	Permission	Owner Group	Node
Ascend Device Plugin	/var/log/devicePlugin	750	root:root	All compute nodes
cAdvisor	/var/log/cadvisor	750	root:root	
HCCL-Controller	/var/log/atlas_dls/hccl-controller	750	hwMindX:hwMindX	Management node
Volcano	/var/log/atlas_dls/volcano-admission	750	hwMindX:hwMindX	
	/var/log/atlas_dls/volcano-controller	750	hwMindX:hwMindX	
	/var/log/atlas_dls/volcano-scheduler	750	hwMindX:hwMindX	

Step 1 Run the following command to create a log path:

mkdir -p *Component log path*

NOTE

Component log path: indicates a log path in **Table 2-13**.

Example:

mkdir -p */var/log/devicePlugin*

Step 2 Run the following command to set the permission on the log path:

chmod -R *Permissions Component log path*

NOTE

Permissions: indicate the permissions on the log paths of the corresponding components in **Table 2-13**.

Example:

chmod -R *750 /var/log/devicePlugin*

Step 3 Run the following command to set the owner of the log path:

chown -R hwMindX:hwMindX *Component log path*

----End

Configuring Log Dumping

A large number of logs are generated after the components run for a long time. Therefore, you need to configure log dumping rules. The recommended log dumping configuration for MindX DL is as follows:

 Copyright © Huawei Technologies Co., Ltd.

- Management node
  a. In the **/etc/logrotate.d** directory, run the following command to create a log dump configuration file:

     vim /etc/logrotate.d/*File name*

     Example:

     vim /etc/logrotate.d/*mindx_dl_scheduler*

  b. Add the following content to the file and run the **:wq** command to save the file.

```
/var/log/atlas_dls/volcano-*/*.log
/var/log/atlas_dls/hccl-*/*.log{
    daily
    rotate 8
    size 20M
    compress
    dateext
    missingok
    notifempty
    copytruncate
    create 0640 hwMindX hwMindX
    sharedscripts
    postrotate
        chmod 640 /var/log/atlas_dls/volcano-*/*.log
        chmod 640 /var/log/atlas_dls/hccl-*/*.log
        chmod 440 /var/log/atlas_dls/volcano-*/*.log-*
        chmod 440 /var/log/atlas_dls/hccl-*/*.log-*
    endscript
}
```

  c. Run the following commands to set the file permission to 640 and owner to **root**:

     chmod 640 /etc/logrotate.d/*File name*

     chown root /etc/logrotate.d/*File name*

     Example:

     chmod 640 /etc/logrotate.d/*mindx_dl_scheduler*

     chown root /etc/logrotate.d/*mindx_dl_scheduler*

- All compute nodes
  a. In the **/etc/logrotate.d** directory, run the following command to create a log dump configuration file:

     vim /etc/logrotate.d/*File name*

     Example:

     vim /etc/logrotate.d/*mindx_dl_cadvisor*

  b. Add the following content to the file and run the **:wq** command to save the file.

```
/var/log/devicePlugin/*.log
/var/log/cadvisor/*.log{
    daily
    rotate 8
    size 20M
    compress
    dateext
    missingok
    notifempty
    copytruncate
    create 0640 root root
    sharedscripts
    postrotate
```

 Copyright © Huawei Technologies Co., Ltd.

```
    chmod 640 /var/log/devicePlugin/*.log
    chmod 640 /var/log/cadvisor/*.log
    chmod 440 /var/log/devicePlugin/*.log-*
    chmod 440 /var/log/cadvisor/*.log-*
  endscript
}
```

c. Run the following commands to set the file permission to 640 and owner to **root**:

chmod 640 /etc/logrotate.d/*File name*

chown root /etc/logrotate.d/*File name*

Example:

chmod 640 /etc/logrotate.d/*mindx_dl_cadvisor*

chown root /etc/logrotate.d/*mindx_dl_cadvisor*

2.3.3.2 Manual Installation (Using Images Downloaded from Ascend Hub)

This section describes how to manually install the Volcano, HCCL-Controller, Ascend Device Plugin, and cAdvisor components.

Precautions

The images for x86 servers and ARM servers are different. Use proper images during installation.

Prerequisites

1. The environment dependencies have been configured. For details, see **Environment Dependencies**.
2. The MindX DL images have been obtained. For details, see **Obtaining MindX DL Images**.
3. Installation preparations have been completed. For details, see **Preparations for Installation**.
4. All files, except **calico.yaml**, in the **yamls** directory have been obtained from **Gitee Code Repository** and uploaded to any directory on the management node, for example, **/home/yamls**.

   The directory structure of the obtained files is as follows:

```
/home
├── ...
└── yamls
    ├── ascendplugin-volcano-v20.2.0.yaml
    ├── ascendplugin-310-v20.2.0.yaml
    ├── hccl-controller-v20.2.0.yaml
    ├── cadvisor-v0.34.0-r40.yaml
    └── volcano-v1.0.1-r40.yaml
```

Installing Volcano

Step 1 Run the following command on the management node to view the image:

docker images

NOTE

If no image is available, obtain images. For details, see **Obtaining MindX DL Images**.

 Copyright © Huawei Technologies Co., Ltd.

```
[root@centos-19 ascend-device-plugin]# docker images|grep volcanosh
volcanosh/vc-webhook-manager         v1.0.1-r40          b234843da818        7 days ago          134MB
volcanosh/vc-scheduler               v1.0.1-r40          99bb8e9d020c        7 days ago          108MB
volcanosh/vc-controller-manager      v1.0.1-r40          b3b879a1024d        7 days ago          92.6MB
volcanosh/vc-webhook-manager-base    v1.0.1-r40          2d540c610363        2 weeks ago         47.6MB
```

Step 2 Run the following command to go to the directory (for example, **/home/yamls**) where the YAML file for starting Volcano is stored:

cd /home/yamls

Step 3 Run the following command to run the YAML file for starting Volcano:

kubectl apply -f volcano-*{version}*.yaml

Example:

kubectl apply -f volcano-v1.0.1-r40.yaml

```
root@ubuntu:/home/yamls# kubectl apply -f volcano-v1.0.1-r40.yaml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply
namespace/volcano-system configured
configmap/volcano-scheduler-configmap created
serviceaccount/volcano-scheduler created
clusterrole.rbac.authorization.k8s.io/volcano-scheduler created
clusterrolebinding.rbac.authorization.k8s.io/volcano-scheduler-role created
deployment.apps/volcano-scheduler created
serviceaccount/volcano-admission created
clusterrole.rbac.authorization.k8s.io/volcano-admission created
clusterrolebinding.rbac.authorization.k8s.io/volcano-admission-role created
deployment.apps/volcano-admission created
service/volcano-admission-service created
job.batch/volcano-admission-init created
serviceaccount/volcano-controllers created
clusterrole.rbac.authorization.k8s.io/volcano-controllers created
clusterrolebinding.rbac.authorization.k8s.io/volcano-controllers-role created
deployment.apps/volcano-controllers created
customresourcedefinition.apiextensions.k8s.io/jobs.batch.volcano.sh created
customresourcedefinition.apiextensions.k8s.io/commands.bus.volcano.sh created
customresourcedefinition.apiextensions.k8s.io/podgroups.scheduling.volcano.sh created
customresourcedefinition.apiextensions.k8s.io/queues.scheduling.volcano.sh created
```

----End

Installing HCCL-Controller

Step 1 Run the following command on the management node to view the image:

docker images

NOTE

If no image is available, obtain images. For details, see **Obtaining MindX DL Images**.

```
[root@centos-19 ascend-device-plugin]# docker images|grep hccl-controller
hccl-controller                      v20.2.0             914deb02403e        3 weeks ago         151MB
```

Step 2 Run the following command to go to the directory (for example, **/home/yamls**) where the YAML file for starting HCCL-Controller is stored:

cd /home/yamls

Step 3 Run the following command to start HCCL-Controller:

kubectl apply -f hccl-controller-*{version}*.yaml

 Copyright © Huawei Technologies Co., Ltd.

Example:

kubectl apply -f hccl-controller-*v20.2.0*.yaml

----End

Installing Ascend Device Plugin

Step 1 Run the following command on a compute node to view the image:

docker images|grep k8sdeviceplugin

NOTE

If no image is available, obtain images. For details, see **Obtaining MindX DL Images**.

```
root@ubuntu:~# docker images|grep k8sdeviceplugin
ascend-k8sdeviceplugin          v20.2.0             43a5c145ac8c        About an hour ago   768MB
root@ubuntu:~#
```

Step 2 On the management node, run the following command to go to the path (for example, **/home/yamls**) where the YAML file for starting Ascend Device Plugin is stored:

cd /home/yamls

Step 3 Run the following command on the management node to start the image:

- Start a training node.

  kubectl apply -f ascendplugin-volcano-*{version}*.yaml

  Example:

  kubectl apply -f ascendplugin-volcano-*v20.2.0*.yaml

- Start an inference node. If there is no inference node, skip this step.

  kubectl apply -f ascendplugin-310-*{version}*.yaml

  Example:

  kubectl apply -f ascendplugin-310-*v20.2.0*.yaml

NOTE

Ascend Device Plugin requires Docker to enable Ascend Docker Runtime. If Ascend Docker Runtime is not installed, install the toolbox **Ascend-cann-toolbox_{*version*}_linux-{*arch*}.run** by referring to "Installing the Operating Environment (Training)" > "Installing the Training Software" in the ***CANN Software Installation Guide***.

----End

Installing cAdvisor

Step 1 Run the following command on a compute node to view the image:

docker images | grep cadvisor

NOTE

If no image is available, obtain images. For details, see **Obtaining MindX DL Images**.

```
[root@centos-19 ascend-device-plugin]# docker images | grep cadvisor
google/cadvisor                 v0.34.0-r40         391393dea5f8        2 weeks ago         98.7MB
```

 Copyright © Huawei Technologies Co., Ltd.

Step 2 On the management node, run the following command to go to the path (for example, **/home/yamls**) where the YAML file for starting cAdvisor is stored:

cd /home/yamls

Step 3 Run the following command on the management node to install cAdvisor:

kubectl apply -f cadvisor-*{version}*.yaml

Example:

kubectl apply -f cadvisor-*v0.34.0-r40*.yaml

```
root@ubuntu:/home/yamls# kubectl apply -f cadvisor-v0.34.0-r40.yaml
namespace/cadvisor created
serviceaccount/cadvisor created
clusterrole.rbac.authorization.k8s.io/cadvisor created
clusterrolebinding.rbac.authorization.k8s.io/cadvisor created
daemonset.apps/cadvisor created
podsecuritypolicy.policy/cadvisor created
root@ubuntu:/home/yamls# kubectl get pod -n cadvisor
NAME            READY   STATUS    RESTARTS   AGE
cadvisor-p8qp8  1/1     Running   0          60s
```

----End

Verifying the Installation

Check that the components are installed properly. For details, see **Environment Check**.

2.3.3.3 Manual Installation (Using Manually Built Images)

This section describes how to manually install the Volcano, HCCL-Controller, Ascend Device Plugin, and cAdvisor components.

Precautions

The images for x86 servers and ARM servers are different. Use proper images during installation.

Prerequisites

1. Installation preparations have been completed. For details, see **Environment Dependencies**.
2. The components has been built. For details, see **Building MindX DL Images**.
3. Pre-installation operations have been performed. For details, see **Preparations for Installation**.

Installing Volcano

Step 1 Copy the files required for installing Volcano to any directory on the management node, for example, **/home/install**.

In the list of required files, *{REL_OSARCH}* indicates the system architecture. The value can be amd64 or arm64.

 Copyright © Huawei Technologies Co., Ltd.

Table 2-14 Files

File Name	Description	Path in the Compilation Environment
vc-controller-manager-v1.0.1-r40-*{REL_OSARCH}*.tar.gz	Controller-Manager installation image	${GOPATH}/src/volcano.sh/volcano/_output/DockFile/vc-controller-manager-v1.0.1-r40-*{REL_OSARCH}*.tar.gz
vc-scheduler-v1.0.1-r40-*{REL_OSARCH}*.tar.gz	vc-scheduler installation image	${GOPATH}/src/volcano.sh/volcano/_output/DockFile/vc-scheduler-v1.0.1-r40-*{REL_OSARCH}*.tar.gz
vc-webhook-manager-base-v1.0.1-r40-*{REL_OSARCH}*.tar.gz	webhook-manager-base installation image	${GOPATH}/src/volcano.sh/volcano/_output/DockFile/vc-webhook-manager-base-v1.0.1-r40-*{REL_OSARCH}*.tar.gz
vc-webhook-manager-v1.0.1-r40-*{REL_OSARCH}*.tar.gz	Webhook-manager installation image	${GOPATH}/src/volcano.sh/volcano/_output/DockFile/vc-webhook-manager-v1.0.1-r40-*{REL_OSARCH}*.tar.gz
volcano-v1.0.1-r40.yaml	YAML file for installing Volcano	${GOPATH}/src/volcano.sh/volcano/hack/../_output/release/volcano-v1.0.1-r40.yaml

The following uses amd64 as an example. After the operation is complete, the files in the directory are as follows:

```
root@ubuntu:/home/install# ll
total 433964
drwxr-xr-x 2 root root      4096 Nov 17 20:09 ./
drwxr-xr-x 5 root root      4096 Nov 17 20:09 ../
-rw------- 1 root root  96946176 Nov 17 20:09 vc-controller-manager-v1.0.1-r40-amd64.tar.gz
-rw------- 1 root root 113263616 Nov 17 20:09 vc-scheduler-v1.0.1-r40-amd64.tar.gz
-rw------- 1 root root  50782720 Nov 17 20:09 vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz
-rw------- 1 root root 141210624 Nov 17 20:09 vc-webhook-manager-v1.0.1-r40-amd64.tar.gz
-rw-r--r-- 1 root root     24413 Nov 17 20:09 volcano-v1.0.1-r40.yaml
```

Step 2 Import images.

1. Run the following commands to import the images:

   docker load --input vc-controller-manager-v1.0.1-r40-amd64.tar.gz

   docker load --input vc-scheduler-v1.0.1-r40-amd64.tar.gz

   docker load --input vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz

   docker load --input vc-webhook-manager-v1.0.1-r40-amd64.tar.gz

   ```
   root@ubuntu:/home/install# docker load --input vc-controller-manager-v1.0.1-r40-amd64.tar.gz
   5555a23bac37: Loading layer [==================================================>]
   ```

 Copyright © Huawei Technologies Co., Ltd.

```
42.42MB/42.42MB
9d71c9e305a9: Loading layer [==================================================>]
45.98MB/45.98MB
Loaded image: volcanosh/vc-controller-manager:v1.0.1-r40
root@ubuntu:/home/install# docker load --input vc-scheduler-v1.0.1-r40-amd64.tar.gz
3c5cfcf6e497: Loading layer [==================================================>]
49.68MB/49.68MB
6c299227fb43: Loading layer [==================================================>]
53.23MB/53.23MB
Loaded image: volcanosh/vc-scheduler:v1.0.1-r40
root@ubuntu:/home/install# docker load --input vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz
Loaded image: volcanosh/vc-webhook-manager-base:v1.0.1-r40
root@ubuntu:/home/install# docker load --input vc-webhook-manager-v1.0.1-r40-amd64.tar.gz
df6d1282497d: Loading layer [==================================================>]
39.9MB/39.9MB
d835292e5e19: Loading layer [==================================================>]
4.608kB/4.608kB
Loaded image: volcanosh/vc-webhook-manager:v1.0.1-r40
```

2. Run the following command to check whether the images are imported successfully.

   docker images|grep vc

```
root@ubuntu:/home/install# docker images|grep vc
volcanosh/vc-webhook-manager                v1.0.1-r40          3aca3aa7dd10        5 hours ago
85.9MB
volcanosh/vc-scheduler                      v1.0.1-r40          9958222963de        5 hours ago         161MB
volcanosh/vc-controller-manager             v1.0.1-r40          7e94f8150198        5 hours ago
146MB
volcanosh/vc-webhook-manager-base           v1.0.1-r40          bbadada24a40        15 months
ago       46MB
```

Step 3 Run the following command to run the YAML file for starting Volcano:

kubectl apply -f volcano-v1.0.1-r40.yaml

```
root@ubuntu:/home/install# kubectl apply -f volcano-v1.0.1-r40.yaml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl
apply
namespace/volcano-system configured
configmap/volcano-scheduler-configmap created
serviceaccount/volcano-scheduler created
clusterrole.rbac.authorization.k8s.io/volcano-scheduler created
clusterrolebinding.rbac.authorization.k8s.io/volcano-scheduler-role created
deployment.apps/volcano-scheduler created
serviceaccount/volcano-admission created
clusterrole.rbac.authorization.k8s.io/volcano-admission created
clusterrolebinding.rbac.authorization.k8s.io/volcano-admission-role created
deployment.apps/volcano-admission created
service/volcano-admission-service created
job.batch/volcano-admission-init created
serviceaccount/volcano-controllers created
clusterrole.rbac.authorization.k8s.io/volcano-controllers created
clusterrolebinding.rbac.authorization.k8s.io/volcano-controllers-role created
deployment.apps/volcano-controllers created
customresourcedefinition.apiextensions.k8s.io/jobs.batch.volcano.sh created
customresourcedefinition.apiextensions.k8s.io/commands.bus.volcano.sh created
customresourcedefinition.apiextensions.k8s.io/podgroups.scheduling.volcano.sh created
customresourcedefinition.apiextensions.k8s.io/queues.scheduling.volcano.sh created
```

----End

Installing HCCL-Controller

Step 1 Run the following command to check whether the HCCL-Controller image exists:

docker images

 Copyright © Huawei Technologies Co., Ltd.

NOTE

If no image is available, build the image by referring to **Building HCCL-Controller**.

Step 2 Run the following command to go to the HCCL-Controller source code building directory:

cd /home/ascend-hccl-controller/output

Step 3 Run the following command to start HCCL-Controller:

kubectl apply -f hccl-controller-*.yaml

----End

Installing Ascend Device Plugin

Step 1 Upload the images generated in **Building Ascend Device Plugin** to the ***/home/*ascend-device-plugin/output** directory on a compute node with Ascend AI Processors.

Step 2 Install the image.

- ARM

  docker load -i Ascend-K8sDevicePlugin-v20.2.0-arm64-Docker.tar.gz

- x86

  docker load -i Ascend-K8sDevicePlugin-v20.2.0-amd64-Docker.tar.gz

Step 3 Run the following command to query images:

docker images|grep k8sdeviceplugin

```
root@ubuntu:~# docker images|grep k8sdeviceplugin
ascend-k8sdeviceplugin          v20.2.0             43a5c145ac8c        About an hour ago   768MB
root@ubuntu:~#
```

Step 4 Copy the YAML files in the root directory (**/home/ascend-device-plugin** is used as an example) of the ascend-device-plugin source code on the compute node to the active Kubernetes node.

cd /*home*/ascend-device-plugin

scp ascendplugin-*.yaml root@masterIp:/home/ascend-device-plugin

Step 5 Run the following command to start the image:

kubectl apply -f ascendplugin-310.yaml

kubectl apply -f ascendplugin-volcano.yaml

----End

Installing cAdvisor

Step 1 Upload the **huawei-cadvisor-beta.tar.gz** image in the **$GOPATH/src/github.com/google/cadvisor** directory built in **Building cAdvisor** to a compute node with Ascend AI Processors and run the following command to load the image:

docker load < huawei-cadvisor-*.tar.gz

 Copyright © Huawei Technologies Co., Ltd.

Step 2 Run the following command to copy the **yaml** folder in the cAdvisor source code directory to the management node, for example, **/home**:

scp cadvisor-v0.34.0-*.yaml root@masterIp:*/home*

Step 3 Go to the directory in **Step 2**, for example, **/home**, and run the following command to install cAdvisor:

kubectl apply -f cadvisor-v0.34.0-*.yaml

```
root@ubuntu:/home# kubectl apply -f cadvisor-v0.34.0-*.yaml
namespace/cadvisor created
serviceaccount/cadvisor created
clusterrole.rbac.authorization.k8s.io/cadvisor created
clusterrolebinding.rbac.authorization.k8s.io/cadvisor created
daemonset.apps/cadvisor created
podsecuritypolicy.policy/cadvisor created
root@ubuntu:/home# kubectl get pod -n cadvisor
NAME            READY   STATUS    RESTARTS   AGE
cadvisor-p8qp8  1/1     Running   0          60s
```

----End

Verifying the Installation

Check that the components are installed properly. For details, see **Environment Check**.

2.4 Environment Check

2.4.1 Checking the Environment Manually

Procedure

Step 1 Run the following command on all nodes to check the Docker version and runtime:

docker info

Information similar to the following is displayed:

```
[root@centos-21 ~]# docker info
Client:
 Debug Mode: false

Server:
 Containers: 59
  Running: 30
  Paused: 0
  Stopped: 29
 Images: 113
 Server Version: 18.06.3
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: systemd
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
```

 Copyright © Huawei Technologies Co., Ltd.

```
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: ascend runc
Default Runtime: ascend
Init Binary: docker-init
...
```

NOTE

If the value of **Runtimes** is not **ascend runc**, install the toolbox **Ascend-cann-toolbox_{*version*}_linux-{*arch*}.run** by referring to "Installing the Operating Environment (Training)" > "Installing the Training Software" in the ***CANN Software Installation Guide***.

Step 2 On the management node, run the following command to check the pod status of ascend-device-plugin, cadvisor, hccl-controller, volcano-admission, volcano-controllers, volcano-scheduler, and volcano-admission-init:

kubectl get pod --all-namespaces

```
root@ubuntu:/home# kubectl get pod --all-namespaces
NAMESPACE        NAME                                       READY   STATUS             RESTARTS   AGE
cadvisor         cadvisor-kr59p                             1/1     Running            1          3d8h
default          hccl-controller-c8dc9ff76-wsxqg            1/1     Running            1          3d8h
kube-system      ascend-device-plugin-daemonset-k6nd7       1/1     Running            3          4h56m
...
volcano-system   volcano-admission-7c4cb5ff8-cf44s          1/1     Running            0          19h
volcano-system   volcano-admission-init-895wh               0/1     Completed          0          19h
volcano-system   volcano-controllers-6786db54f-j4pd2        1/1     Running            0          19h
volcano-system   volcano-scheduler-844f9b547b-6fctn         1/1     Running            0          19h
```

NOTE

- The volcano-admission-init component of Volcano is **Completed**, and all other components must be in the **Running** state, including each component on each compute node of cAdvisor and Ascend Device Plugin.
- The hccl-controller, volcano-admission, volcano-controllers, volcano-scheduler, and volcano-admission-init components run only on the management node.
- The ascend-device-plugin and cadvisor components run on nodes equipped with Ascend AI Processors.

Step 3 Run the following command on the management node to check the number of processors:

kubectl describe node *hostName*

NOTE

hostName: name of the node where Ascend Device Plugin is installed in Kubernetes.

- Number of Ascend 310 AI Processors

```
...
Hostname:    ubuntu
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend310:  4
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend310:  4
 hugepages-2Mi:      0
 memory:             792205068Ki
```

 Copyright © Huawei Technologies Co., Ltd.

```
 pods:                110
...
```

- Number of Ascend 910 AI Processors

```
...
 Hostname:    ubuntu
Capacity:
 cpu:                 192
 ephemeral-storage:     1537233808Ki
 huawei.com/Ascend910:  8    # The value is 2 if Atlas 300T training cards are installed.
 hugepages-2Mi:         0
 memory:              792307468Ki
 pods:                110
Allocatable:
 cpu:                 192
 ephemeral-storage:     1416714675108
 huawei.com/Ascend910:  8    # The value is 2 if Atlas 300T training cards are installed.
 hugepages-2Mi:         0
 memory:              792205068Ki
 pods:                110
...
```

----End

2.4.2 Checking the Environment Using a Script

MindX DL provides the component check function. You can use a check script to obtain the status and version information of the NPU driver, Docker, Kubernetes components (kubelet, Kubectl, and Kubeadm), and MindX DL. MindX DL supports single-node environment check and cluster environment check. To check the cluster environment, you need to run Ansible commands on the management node to distribute check scripts to check each node in the cluster. For details about how to view the number of processors on a node, see **Step 3** in **Checking the Environment Manually**.

NOTICE

- You need to temporarily enable the read and write permissions of the compute nodes for Kubernetes. The read and write permissions are valid only for the compute node resources and cannot process other node resources. After the check is complete, the temporary permissions of the compute nodes are cancelled.
- If you do not want to enable the read and write permissions for Kubernetes, perform a manual check. For details, see **Checking the Environment Manually**.

Prerequisites

To check a single-node environment, requirements **1** to **3** must be met. To check a cluster environment, requirements **1** to **4** must be met.

1. The compute nodes have the permission to access the Kubernetes cluster. If the compute nodes do not have the permission, the check result may be affected. In a single-node environment, you need to manually enable the permission. In a cluster environment, the Ansible script automatically enables the access permission on each compute node. After the check is complete, disable the access permission on each compute node.

 Copyright © Huawei Technologies Co., Ltd.

2. Obtain all files in the **check_evn** directory from **Gitee Code Repository** and upload them to the **/home/check_env** directory.
3. The dos2unix tool has been installed.

   NOTE

   - For Ubuntu, run the following command to install dos2unix:

     apt install -y dos2unix
   - For CentOS, run the following command to install dos2unix:

     yum install -y dos2unix
4. Python 3.7.5 and Ansible have been installed on the management node. For details about how to check the installation, see **Checking the Python and Ansible Versions**.

Checking a Single-Node Environment

Step 1 Run the following command to check whether the node has the permission to access the Kubernetes cluster:

kubectl get nodes

- If yes, go to **Step 3**.
- If no, go to **Step 2**.

If the following information is displayed, the node has the permission to access the Kubernetes cluster:

```
NAME       STATUS   ROLES          AGE     VERSION
centos-19  Ready    worker         102s    v1.17.3
centos-21  Ready    master,worker  3h17m   v1.17.3
centos39   Ready    worker         88m     v1.17.3
```

Step 2 Temporarily enable the node access permission.

- In the shell window that is displayed, run the following command to temporarily enable the access permission for the compute node:

  export KUBECONFIG=/etc/kubernetes/kubelet.conf
- Enable the access permission on the management node.

  By default, the management node has the permission to view the Kubernetes cluster information. If the management node does not have the permission, run the following command in the shell window to temporarily enable the permission:

  export KUBECONFIG=/etc/kubernetes/admin.conf

After the function is enabled, run the **kubectl get nodes** command.

If the following information is displayed, the node has the permission to access the Kubernetes cluster:

```
NAME       STATUS   ROLES          AGE     VERSION
centos-19  Ready    worker         102s    v1.17.3
centos-21  Ready    master,worker  3h17m   v1.17.3
centos39   Ready    worker         88m     v1.17.3
```

NOTE

If "Unable to connect to the server: x509: certificate signed by unknown authority" is displayed, a proxy may be configured. Run the **unset http_proxy https_proxy** command and enable the permission again.

 Copyright © Huawei Technologies Co., Ltd.

Step 3 Run the following command to switch to the **/*home*/check_env** directory:

cd /*home*/check_env

Step 4 Run the following commands to check the environment using the script:

dos2unix *

chmod 500 check_env.sh

bash check_env.sh *node_type ip*

NOTE

- *node_type*: The options are as follows. Check different services of MindX DL for different node types.
  - **master** indicates the management node. Only the Volcano and HCCL-Controller services of MindX DL are checked.
  - **worker** indicates a compute node. Only the Ascend Device Plugin and cAdvisor services of MindX DL are checked.
  - **master-worker** indicates that the node is both a management node and a compute node. The Volcano, HCCL-Controller, Ascend Device Plugin, and cAdvisor services are checked.
- *ip*: This parameter is optional. If this parameter is specified, the **IP** column in the command output is empty.

Example:

dos2unix *

chmod 500 check_env.sh

bash check_env.sh *master-worker 10.10.56.78*

```
[root@centos-21 check_env]# bash check_env.sh master-worker 10.10.56.78
-------------------------------------------------------------------------------------------------------------------------
---
| hostname  | ip          | service               | status                  | version                                    |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | npu-driver            | Normal                  | 20.1.0                                     |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | docker                | Normal[active (running)] | 19.03.13
|
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | kubelet               | Normal[active (running)] | v1.17.3                                    |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | kubeadm               | Normal                  | v1.17.3                                    |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | kubectl               | Normal                  | v1.17.3                                    |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | HCCL-Controller       | Not install             | hccl-controller:v20.2.0,hccl-
controller:v20.1.0 |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | volcano-admission     | Normal[Running]         | volcanosh/vc-webhook-
manager:v1.0.1-r40         |
-------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | volcano-admission-init | Normal[Completed]       | volcanosh/vc-webhook-
```

 Copyright © Huawei Technologies Co., Ltd.

```
manager:v1.0.1-r40         |
-------------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | volcano-controllers    | Normal[Running]          | volcanosh/vc-controller-
manager:v1.0.1-r40      |
-------------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | volcano-scheduler      | Normal[Running]          | volcanosh/vc-scheduler:v1.0.1-
r40               |
-------------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | Ascend-Device-Plugin   | Normal[Running]          | ascend-
k8sdeviceplugin:v20.2.0                  |
-------------------------------------------------------------------------------------------------------------------------------
---
| centos-21 | 10.10.56.78 | cAdvisor               | Normal[Running]          | google/cadvisor:v0.34.0-
r40                     |
-------------------------------------------------------------------------------------------------------------------------------
---

Finished! The check report is stored in the /home/check_env/env_check_report.txt
```

NOTE

- **status**
  - **Not install**: The service is not installed.
  - **Normal**: The service is normal. The specific status is displayed in [].
  - **Error**: The service is abnormal. The error status is displayed in [].
  - **Completed**: This state is valid only for the volcano-admission-init service.

  Use Kubernetes to check the MindX DL service. If the node does not have the permission to access the Kubernetes cluster, "Can't get service status, permission denied" is displayed.
- **version**: If the MindX DL service is not running, information about all available images of a service on the current node is displayed. Images are separated by commas (,). If the Docker service is not running, the message "Docker service not running" is displayed.
- The last line shows the location of the output report. In the example, the output report is in the **/home/check_env** directory.

Step 5 Disable the temporary access permission on each node.

After closing the current shell window, you can disable the temporary access permission of a node or run the following command to disable the temporary access permission:

unset KUBECONFIG

NOTE

- If the node itself has the access permission, running the preceding command can only disable the temporarily enabled access permission but cannot disable the permission of the node itself.
- The preceding command is valid only for the current shell window and can disable only the access permission that is temporarily enabled.

----End

Checking a Cluster Environment

Step 1 Configure Ansible host information.

For details, see **Configuring Ansible Host Information**.

 Copyright © Huawei Technologies Co., Ltd.

An example is as follows:

```
[all:vars]
# Master IP
master_ip=10.10.56.78

[master]
ubuntu-example ansible_host=10.10.56.78 ansible_ssh_user="root" ansible_ssh_pass="ad34#$"

[training_node]
ubuntu-example2 ansible_host=10.10.56.79 ansible_ssh_user="root" ansible_ssh_pass="ad34#$"

[inference_node]


[workers:children]
training_node
inference_node
```

NOTE

- The configuration file **/etc/ansible/hosts** of the cluster environment checked using Ansible must contain at least the preceding content.
- If Ansible is used to install and deploy a cluster, you can directly use the **/etc/ansible/hosts** file configured during installation and deployment to check the cluster environment. You do not need to modify the file.
- Some groups do not have servers and can be left empty, such as **[inference_node]** in the example.
- The content under **[workers:children]** cannot be modified.

Step 2 Run the following command to switch to the **/*home*/check_env** directory:

cd /*home*/check_env

The directory structure is as follows:

```
/home/check_env
├── check_env.sh
└── check_evn.yaml
```

Step 3 Run the following commands to check the environment:

dos2unix *

ansible-playbook -vv check_env.yaml

NOTE

You are advised to run the following commands to set the permission on the **check_env.yaml** file to **400** and the permission on the **check_env.sh** file to **500**:

chmod 400 check_env.yaml

chmod 500 check_env.sh

If the following message is displayed, the operation is successful.

```
TASK [Generate final report]
*********************************************************************************************************************************************************************
*********************************************************************************
task path: /home/check_env/check_env.yaml:158
changed: [centos39] => (item=/home/check_env/reports/env_check_report_master.txt) =>
{"ansible_loop_var": "item", "changed": true, "cmd": "cat /home/check_env/reports/
env_check_report_master.txt >> /home/check_env/env_check_report_all.txt; echo \"\" >> /home/check_env/
env_check_report_all.txt", "delta": "0:00:00.006126", "end": "2020-11-26 15:51:23.865208", "item": "/home/
check_env/reports/env_check_report_master.txt", "rc": 0, "start": "2020-11-26 15:51:23.859082", "stderr": "",
"stderr_lines": [], "stdout": "", "stdout_lines": []}
```

 Copyright © Huawei Technologies Co., Ltd.

```
changed: [centos39] => (item=/home/check_env/reports/env_check_report_10.10.56.78.txt) =>
{"ansible_loop_var": "item", "changed": true, "cmd": "cat /home/check_env/reports/
env_check_report_10.10.56.78.txt >> /home/check_env/env_check_report_all.txt; echo \"\" >> /home/
check_env/env_check_report_all.txt", "delta": "0:00:00.006343", "end": "2020-11-26 15:51:24.367935",
"item": "/home/check_env/reports/env_check_report_10.10.56.78.txt", "rc": 0, "start": "2020-11-26
15:51:24.361592", "stderr": "", "stderr_lines": [], "stdout": "", "stdout_lines": []}

TASK [Print report path]
*********************************************************************************************************************************************************
*********************************************************************************
task path: /home/check_env/check_env.yaml:166
changed: [centos39] => {"changed": true, "cmd": "echo \"Finished! The check report is stored in the /home/
check_env/env_check_report_all.txt on the master node.\"", "delta": "0:00:00.002918", "end": "2020-11-26
15:51:24.957293", "rc": 0, "start": "2020-11-26 15:51:24.954375", "stderr": "", "stderr_lines": [], "stdout":
"Finished! The check report is stored in the /home/check_env/env_check_report_all.txt on the master node.",
"stdout_lines": ["Finished! The check report is stored in the /home/check_env/env_check_report_all.txt on
the master node."]}
META: ran handlers
META: ran handlers

PLAY RECAP
*********************************************************************************************************************************************************
*********************************************************************************************
centos131                  : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
centos39                   : ok=14   changed=8    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0
[root@centos39 check_env]#
```

NOTE

- In the command output, "Finished! The check report is stored in the *XXX*" following the **TASK [Print report path]** field indicates the location of the report on the management node. In this example, the report is stored in **/home/check_env/env_check_report_all.txt**.
- The report contains the check result of each node.
- The independent report of each node is stored in the **reports** directory of the management node. In this example, the directory is **/home/check_env/reports**.

----End

2.5 MindX DL Uninstallation

2.5.1 Automatic Uninstallation

Prerequisites

- MindX DL has been installed.
- Python and Ansible have been installed on the management node. For details about how to check the installation, see **Installing Python and Ansible**.
- Ansible host information has been configured on the management node. For details, see **Configuring Ansible Host Information**.

Script Obtaining

Obtain the uninstallation scripts from the MindX DL deployment file repository, as listed in **Table 2-15**.

Link: **Gitee Code Repository**

 Copyright © Huawei Technologies Co., Ltd.

Table 2-15 Script list

Script Name	Description	Path in the Code Repository
entry.sh	Entry script for offline uninstallation.	uninstall
uninstall.yaml	Components and software uninstallation script.	uninstall
ascendplugin-volcano-v20.2.0.yaml	Ascend Device Plugin configuration file for Ascend 910 AI Processor.	yamls
ascendplugin-310-v20.2.0.yaml	Ascend Device Plugin configuration file for Ascend 310 AI Processor.	yamls
cadvisor-v0.34.0-r40.yaml	Configuration file of the NPU monitoring component.	yamls
hccl-controller-v20.2.0.yaml	NPU training job component configuration file.	yamls
volcano-v1.0.1-r40.yaml	NPU training job scheduling component configuration file.	yamls

Procedure

Step 1 Log in to the management node as the **root** user.

Step 2 Uninstall MindX DL.

1. Copy the files in the **yamls** directory obtained in **Table 2-15** to the **dls_root_dir** directory defined in the **/etc/ansible/hosts** file on the management node. **/tmp** is used as an example of **dls_root_dir**. The directory structure is as follows:

```
/tmp
└─ yamls
   ├── ascendplugin-volcano-v20.2.0.yaml
   ├── ascendplugin-310-v20.2.0.yaml
   ├── hccl-controller-v20.2.0.yaml
   ├── cadvisor-v0.34.0-r40.yaml
   └── volcano-v1.0.1-r40.yaml
```

2. Copy the files in the **uninstall** directory obtained in **Table 2-15** to any directory on the management node, go to the directory, and run the following scripts to automatically uninstall MindX DL:

   dos2unix *

   chmod 500 entry.sh

 Copyright © Huawei Technologies Co., Ltd.

bash -x entry.sh

NOTE

- Determine whether to delete MindX DL logs, uninstall Kubernetes and Docker on all nodes, and uninstall NFS as prompted.
- MindX DL images are deleted during uninstallation.
- The shared path **/data/atlas_dls** may contain files, such as datasets uploaded by users. This directory is not deleted when the NFS is uninstalled.

----End

2.5.2 Manual Uninstallation

2.5.2.1 Clearing Running Resources

You need to clear resources such as the pod, vcjob, namespace, and image. The vcjob is automatically cleared when the Volcano is uninstalled. To clear other resources, use the following procedure.

Procedure

Step 1 On the management node, run the following command to delete the pod:

kubectl delete -f *File name*

Example:

kubectl delete -f *hccl-controller-v20.2.0.yaml*

Use YAML to perform the uninstallation. For details about the path of the YAML files, see **MindX DL Installation**. Obtain the files in the **yamls** directory and upload the files to a directory on the server.

Step 2 (Optional) Run the following command to delete the namespace:

This step is required when the namespace created in YAML exists.

kubectl delete namespace *vcjob*

In this example, the namespace is **vcjob**.

```
root@ubuntu:/home/install# kubectl delete namespace vcjob
namespace "vcjob" deleted
```

Step 3 Delete the MindX DL image.

NOTE

- After the component images are deleted using YAML, there are still records. Therefore, you need to delete the records.
- Deleted images cannot be recovered. Exercise caution when performing this operation.
- Volcano and HCCL-Controller can be deleted only on the management node. Ascend Device Plugin and cAdvisor can be deleted only on compute nodes.

 Copyright © Huawei Technologies Co., Ltd.

Component	Image Name	Image Deletion Command
Volcano	• volcanosh/vc-webhook-manager:v1.0.1-r40 • volcanosh/vc-scheduler:v1.0.1-r40 • volcanosh/vc-controller-manager:v1.0.1-r40 • volcanosh/vc-webhook-manager-base: v1.0.1-r40	**docker image rm $(docker images |grep volcano|awk '{print $3}')**
cAdvisor	google/ cadvisor:v0.34.0-r40	**docker image rm $(docker images |grep cadvisor|awk '{print $3}')**
Ascend Device Plugin	ascend-k8sdeviceplugin:v20.2.0	**docker image rm $(docker images |grep ascend-device-plugin|awk '{print $3}')**
HCCL-Controller	hccl-controller:v20.2.0	**docker image rm $(docker images |grep hccl-controller| awk '{print $3}')**

----End

2.5.2.2 Deleting Component Logs

Procedure

Step 1 Run the following command to delete component logs. The log path **/var/log/atlas_dls** is used as an example.

rm -rf /var/log/atlas_dls/

The content in **/var/log/atlas_dls/** is as follows:

```
root@ubuntu:~# ll /var/log/atlas_dls/
total 24
drwxr-x--- 14 hwMindX hwMindX 4096 Sep 28 20:28 ./
drwxrwxr-x 18 root    syslog  4096 Sep 29 06:25 ../
drwxr-x---  2 hwMindX hwMindX 4096 Sep 28 20:30 hccl-controller/
drwxr-x---  2 hwMindX hwMindX 4096 Sep 28 20:28 volcano-admission/
drwxr-x---  2 hwMindX hwMindX 4096 Sep 28 20:28 volcano-controller/
drwxr-x---  2 hwMindX hwMindX 4096 Sep 29 06:25 volcano-scheduler
```

For details about other log paths, see **Table 2-13**.

----End

 Copyright © Huawei Technologies Co., Ltd.

2.5.2.3 Removing a Node from a Cluster

Prerequisites

- The running resources have been cleared. For details about how to clear running resources, see **Clearing Running Resources**.
- Component logs have been deleted. For details about how to delete logs, see **Deleting Component Logs**.

Procedure

Step 1 Run the following command on the management node to remove the node from the cluster:

kubectl drain *hostname* **--delete-local-data --force --ignore-daemonsets**

kubectl delete node *hostname*

NOTE

hostname: indicates the name of the node from which a cluster is to be removed.

Step 2 Run the following commands on the removed node to reset the node:

kubeadm reset -f

rm -rf ~/.kube

Step 3 Run the following command on the removed node to delete the **/etc/kubernetes** directory:

rm -rf /etc/kubernetes

```
root@ubuntu:/home# cd /etc/kubernetes/
root@ubuntu:/etc/kubernetes# ll
total 44
drwxr-xr-x  4 root root 4096 Sep 28 14:36 ./
drwxr-xr-x 99 root root 4096 Sep 30 13:05 ../
-rw-------  1 root root 5454 Sep 28 14:36 admin.conf
-rw-------  1 root root 5490 Sep 28 14:36 controller-manager.conf
-rw-------  1 root root 1862 Sep 28 14:36 kubelet.conf
drwxr-xr-x  2 root root 4096 Sep 28 14:36 manifests/
drwxr-xr-x  3 root root 4096 Sep 28 14:36 pki/
-rw-------  1 root root 5434 Sep 28 14:36 scheduler.conf
root@ubuntu:/etc/kubernetes# rm  -rf /etc/kubernetes/
```

----End

2.6 MindX DL Upgrade

2.6.1 Preparing for the Upgrade

Before the upgrade, you need to obtain the image files and configuration files required for the upgrade to improve the upgrade efficiency.

 Copyright © Huawei Technologies Co., Ltd.

Prerequisites

- MindX DL has been installed.
- Python and Ansible have been installed on the management node. For details about how to check the installation, see **Installing Python and Ansible**.
- Ansible host information has been configured on the management node. For details, see **Configuring Ansible Host Information**.

Obtaining Image Files

Table 2-16 lists the image files. In the table, *{version}* in a package name indicates the version number. Change it based on the actual situation. For details, see **Obtaining MindX DL Images**.

Table 2-16 Image list

Item	Image Package
Ascend Device Plugin image file	Ascend-K8sDevicePlugin-*{version}*-arm64-Docker.tar.gz
	Ascend-K8sDevicePlugin-*{version}*-amd64-Docker.tar.gz
HCCL-Controller image file	hccl-controller-*{version}*-arm64.tar.gz
	hccl-controller-*{version}*-amd64.tar.gz
cAdvisor image file	huawei-cadvisor-*{version}*-arm64.tar.gz
	huawei-cadvisor-*{version}*-amd64.tar.gz
Volcano image file	• vc-controller-manager-*{version}*-arm64.tar.gz • vc-scheduler-*{version}*-arm64.tar.gz • vc-webhook-manager-base-*{version}*-arm64.tar.gz • vc-webhook-manager-*{version}*-arm64.tar.gz
	• vc-controller-manager-*{version}*-amd64.tar.gz • vc-scheduler-*{version}*-amd64.tar.gz • vc-webhook-manager-base-*{version}*-amd64.tar.gz • vc-webhook-manager-*{version}*-amd64.tar.gz

Obtaining Configuration Files

Obtain offline upgrade configuration files from the MindX DL deployment file repository, as listed in **Table 2-17**. In the table, *{version}* in a file name indicates the version number. Change it based on the actual situation.

Link: **Gitee Code Repository**

 Copyright © Huawei Technologies Co., Ltd.

Table 2-17 Configuration file list

Script Name	Description	Script Path
ascendplugin-volcano-*{version}*.yaml	Ascend Device Plugin configuration file for Ascend 910 AI Processor.	yamls
ascendplugin-310-*{version}*.yaml	Ascend Device Plugin configuration file for Ascend 310 AI Processor.	yamls
cadvisor-*{version}*.yaml	cAdvisor configuration file.	yamls
hccl-controller-*{version}*.yaml	HCCL-Controller configuration file.	yamls
volcano-*{version}*.yaml	Volcano configuration file.	yamls

Obtaining Upgrade Scripts

Obtain the upgrade scripts from the MindX DL deployment file repository, as listed in **Table 2-18**.

Link: **Gitee Code Repository**

Table 2-18 Upgrade script list

Script Name	Description	Path in the Code Repository
entry.sh	Entry script for an offline upgrade.	upgrade
upgrade.yaml	Component upgrade script.	upgrade
volcano-v0.4.0-r03.yaml	First version configuration file of Volcano.	upgrade/volcano-difference
gen-admission-secret.sh	Script for generating a Volcano startup key.	upgrade/volcano-difference

2.6.2 Upgrading MindX DL

The upgrade mode provided in this document is offline upgrade. You need to prepare the MindX DL component images and configuration files of the latest version in advance and upgrade the existing MindX DL using scripts. After the

 Copyright © Huawei Technologies Co., Ltd.

upgrade, the component service status will be checked. If the status is abnormal, the components can be rolled back to the source version.

Prerequisites

- MindX DL has been installed.
- The MindX DL component images and configuration files of the latest version have been obtained. For details, see **Preparing for the Upgrade**.

Procedure

Step 1 Log in to the management node as the **root** user.

Step 2 Upgrade MindX DL.

1. Create the **upgrade_dependencies** directory in the **dls_root_dir** directory defined in the **/etc/ansible/hosts** file on the management node, and copy the image files of the latest version and the files in the **yamls** directory obtained in **Obtaining Configuration Files** to the **upgrade_dependencies** directory. **/tmp** directory is used as an example of **dls_root_dir**. The directory structure after the upload is as follows:

```
/tmp/upgrade_dependencies/
├── images
│   ├── Ascend-K8sDevicePlugin-v20.2.0-amd64-Docker.tar.gz
│   ├── Ascend-K8sDevicePlugin-v20.2.0-arm64-Docker.tar.gz
│   ├── hccl-controller-v20.2.0-amd64.tar.gz
│   ├── hccl-controller-v20.2.0-arm64.tar.gz
│   ├── huawei-cadvisor-v0.34.0-r40-amd64.tar.gz
│   ├── huawei-cadvisor-v0.34.0-r40-arm64.tar.gz
│   ├── vc-controller-manager-v1.0.1-r40-amd64.tar.gz
│   ├── vc-controller-manager-v1.0.1-r40-arm64.tar.gz
│   ├── vc-scheduler-v1.0.1-r40-amd64.tar.gz
│   ├── vc-scheduler-v1.0.1-r40-arm64.tar.gz
│   ├── vc-webhook-manager-base-v1.0.1-r40-amd64.tar.gz
│   ├── vc-webhook-manager-base-v1.0.1-r40-arm64.tar.gz
│   ├── vc-webhook-manager-v1.0.1-r40-amd64.tar.gz
│   └── vc-webhook-manager-v1.0.1-r40-arm64.tar.gz
└── yamls
    ├── ascendplugin-310.yaml
    ├── ascendplugin-volcano.yaml
    ├── cadvisor-v0.34.0-r40.yaml
    ├── hccl-controller-v20.2.0.yaml
    └── volcano-v1.0.1-r40.yaml
```

2. Copy the files in the **upgrade** directory obtained in **Table 2-18** to any directory on the management node, go to the directory, and run the following scripts to automatically upgrade MindX DL:

   dos2unix *

   chmod 500 entry.sh

   bash -x entry.sh

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- After the upgrade is successful, the message "Do you want to remove previous version images?(yes/no)" is displayed. You can determine whether to retain the images of the source version.
- If the upgrade fails, "Do you want to roll back to previous version?(yes/no)" is displayed. Enter **yes** to roll back to the source version.
- Before the upgrade, the version information of each component will be printed and exported to the **pre_check.txt** file in the same directory.
- After the upgrade, the version information of each component will be printed and exported to the **post_check.txt** file in the same directory.

----End

2.7 Security Hardening

2.7.1 Hardening OS Security

After an OS is installed, if a common user is configured, you can add the **ALWAYS_SET_PATH** field to the **/etc/login.defs** file and set it to **yes** to prevent unauthorized operations.

The ufw firewall is disabled during the installation and deployment. After the installation and deployment are complete, run the following command to enable the firewall:

ufw enable

ufw allow ssh

2.7.2 Hardening Container Security

Harden container security.

Host Configuration

- The host provides the function of auditing the Docker daemon process.

  NOTE

  The Auditd software has been installed.

  The Docker daemon process runs on the host as the **root** user. You can configure an audit mechanism on the host to audit the running and usage status of the Docker daemon. Once the Docker daemon process encounters unauthorized attacks, the root cause of the attack event can be traced.

  By default, the host does not enable the audit function for the Docker daemon. You can add an audit rule as follows:

  a. Add the following command to the **/etc/audit/audit.rules** file:

     -w /usr/bin/docker -k docker

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- **-w**: filters file paths.
- **-k**: filters strings based on specified keywords.

b. Restart the log daemon process.

service auditd restart

NOTICE

If the **/etc/audit/audit.rules** file contains "This file is automatically generated from /etc/audit/rules.d", the modification of the **/etc/audit/audit.rules** file is invalid. You must modify the **/etc/audit/rules.d/audit.rules** file for the modification to take effect. For example, in the Ubuntu system, you need to modify the **/etc/audit/rules.d/audit.rules** file.

- The host provides the audit function for key Docker files and directories.

  The directories and key files are as follows:

  – /var/lib/docker
  – /etc/docker
  – /etc/default/docker
  – /etc/docker/daemon.json
  – /usr/bin/docker-containerd
  – /usr/bin/docker-runc
  – docker.service
  – docker.socket

  NOTICE

  The preceding directories are the default Docker installation directories. If a separate partition is created for Docker, the paths may change.

  The host must provide the audit function for the directories because key Docker information is saved in the directories.

  By default, the host does not enable the audit function for these directories and files. You can add an audit rule in either of the following ways:

  a. Add the following command to the **/etc/audit/audit.rules** file:

     -w /etc/docker -k docker

  b. Restart the log daemon process.

     service auditd restart

 Copyright © Huawei Technologies Co., Ltd.

NOTICE

If the **/etc/audit/audit.rules** file contains "This file is automatically generated from /etc/audit/rules.d", the modification of the **/etc/audit/audit.rules** file is invalid. You must modify the **/etc/audit/rules.d/audit.rules** file for the modification to take effect. For example, in the Ubuntu system, you need to modify the **/etc/audit/rules.d/audit.rules** file.

Docker Daemon File Permission Configuration

- Set the owner and owner group of the TLS CA certificate file to **root:root**, and set the permission to **400**.

  The TLS CA certificate file (the path of the CA certificate file is specified by the **--tlscacert** parameter) is protected from being tampered with. The certificate file is used by the specified CA certificate to authenticate the Docker server. Therefore, the owner and owner group of the CA certificate must be root, and the permission must be 400 to ensure the integrity of the CA certificate.

  You can perform the following operations to set the file properties:

  a. Run the following command to set the owner and owner group of the file to **root**:

     chown root:root <*path to TLS CA certificate file*>

     NOTE

     Generally, the path to TLS CA certificate file is **/usr/local/share/ca-certificates**.

  b. Set the file permission to **400**.

     chmod 400 <*path to TLS CA certificate file*>

- The owner and owner group of the **daemon.json** file are set to **root:root**, and the file permission is set to **600**.

  The **daemon.json** file contains sensitive parameters for changing the Docker daemon process. It is an important global configuration file. The owner and owner group of the file must be **root**, and only the **root** user has the write permission on the file to ensure file integrity. This file does not exist by default.

  – If the **daemon.json** file does not exist by default, the product does not use this file for configuration. In this case, you can run the following command to set the configuration file to empty in the boot parameters so that the file is not used as the default configuration file to prevent attackers from maliciously creating and modifying configurations.

    docker --config-file=""

  – If the **daemon.json** file exists in the product environment, the file has been used for configuration. In this case, you need to set the corresponding permission to prevent malicious modification.

    i. Run the following command to set the owner and owner group of the file to **root**:

       chown root:root /etc/docker/daemon.json

 Copyright © Huawei Technologies Co., Ltd.

ii. Run the following command to set the file permission to **600**:
chmod 600 /etc/docker/daemon.json

Docker Permission Control

You are advised to use non-root users or non-privileged root users to use Docker except for special cases, such as cAdvisor and Ascend Device Plugin.

2.7.3 Security Hardening for Ownerless Files

The official Docker image is different from the OS on a physical machine. Therefore, the users in the system may not correspond to each other. As a result, the files generated during the running of the physical machine or container become ownerless files.

You can run the **find / -nouser -nogroup** command to search for ownerless files in a container or on a physical machine. Then create users and user groups based on the UIDs and GIDs of the files, or change the UIDs of existing users or GIDs of user groups to assign file owners, preventing security risks caused by ownerless files.

2.7.4 Hardening the cAdvisor Monitoring Port

When cAdvisor is deployed in the Kubernetes cluster, the monitoring service port is displayed only in the Kubernetes cluster by default. If network plugins that support network policies, such as Calico, are used during Kubernetes deployment, you can add corresponding network policies to display only network traffic. For example, to receive only traffic from Prometheus, you can configure the following network policies:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: cadvisor-network-policy
  namespace: cadvisor
spec:
  podSelector:
    matchLabels:
      name: cadvisor
  policyTypes:
    - Ingress
    - Egress
  ingress:
    - from:
        - namespaceSelector: {}
          podSelector:
            matchLabels:
              app: prometheus
  egress:
    - to:
        - namespaceSelector: {}
          podSelector:
            matchLabels:
              app: prometheus
```

2.8 Common Operations

 Copyright © Huawei Technologies Co., Ltd.

2.8.1 Checking the Python and Ansible Versions

Check whether the versions of the installed Python and Ansible meet the requirements.

Procedure

Step 1 Log in to the management node as the **root** user.

Step 2 Check whether the Python development environment is installed.

python3.7.5 --version

pip3.7.5 --version

If the following information is displayed, the tools have been installed:

```
Python 3.7.5
pip 19.2.3 from /usr/local/python3.7.5/lib/python3.7/site-packages/pip (python 3.7)
```

NOTE

If the Python of the required version has not been installed, install it by referring to **Installing Python and Ansible**.

Step 3 Check the Ansible version.

ansible --version

If the following information is displayed, the tools have been installed:

```
ansible 2.9.7
config file = /etc/ansible/ansible.cfg
configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
ansible python module location = /usr/local/python3.7.5/lib/python3.7/site-packages/ansible-2.9.7-
py3.7.egg/ansible
executable location = /usr/local/bin/ansible
python version = 3.7.5 (default, Nov  9 2020, 03:44:00) [GCC 7.5.0]
```

NOTE

If the Ansible of the required version has not been installed, install it by referring to **Installing Python and Ansible**.

Step 4 Check whether the **sshpass** library has been installed on all nodes.

Run the following command on each node to check **sshpass**:

sshpass -V

If the following information is displayed, the tools have been installed:

```
sshpass 1.06

(C) 2006-2011 Lingnu Open Source Consulting Ltd.
(C) 2015-2016 Shachar Shemesh
This program is free software, and can be distributed under the terms of the GPL
See the COPYING file for more information.

Using "assword" as the default password prompt indicator.
```

If it is not installed, perform the following steps to install it:

- If all nodes can communicate with the Internet:

  Install **sshpass** on each node.

 Copyright © Huawei Technologies Co., Ltd.

Table 2-19 Installation commands

OS	Name
Ubuntu	**apt install -y sshpass**
CentOS	**yum install -y sshpass**

- If a node in the cluster cannot communicate with the Internet, you need to download the offline installation package from the node that can be connected to the Internet and distribute the package to other nodes.

  Download the offline package.

  Table 2-20 Download commands

OS	Name
Ubuntu	**apt download -y sshpass**
CentOS	**yum install --downloadonly --downloaddir=***Download path* **sshpass**

  After the package is distributed to each node, go to the directory where the offline installation package is stored and install **sshpass**.

  Table 2-21 Installation commands

OS	Name
Ubuntu	**dpkg -i sshpass*.deb**
CentOS	**rpm -ivh sshpass*.rpm**

Step 5 Check whether the management node can access each node using the password by SSH.

Run the **ssh root@** *Node IP address* command on the management node and enter the login password of the node after password to log in to the node.

For example, check whether the CentOS management node can access one of the nodes. The check method for other nodes is similar.

ssh root@*10.10.11.12*

root@10.10.11.12's password:

If the following information is displayed, the login is successful:

```
Last failed login: Fri Dec 25 15:21:29 CST 2020 from 10.10.11.12 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Fri Dec 25 14:53:52 2020 from 10.10.11.10
```

----End

 Copyright © Huawei Technologies Co., Ltd.

2.8.2 Installing Python and Ansible

2.8.2.1 Installing Python and Ansible Online

Installing the Python Development Environment

Step 1 Log in to the management node as the **root** user.

Step 2 Install the software required for building Python.

NOTE

If only some software is not installed, install only the software that is not installed:

- Ubuntu ARM

  sudo apt-get install -y gcc g++ make cmake zlib1g zlib1g-dev libbz2-dev openssl libsqlite3-dev libssl-dev libxslt1-dev libffi-dev unzip pciutils net-tools libblas-dev gfortran libblas3 libopenblas-dev

- Ubuntu x86

  sudo apt-get install gcc g++ make cmake zlib1g zlib1g-dev libbz2-dev libsqlite3-dev libssl-dev libxslt1-dev libffi-dev unzip pciutils net-tools -y

- CentOS (ARM and x86)

  sudo yum install -y gcc gcc-c++ make cmake unzip zlib-devel libffi-devel openssl-devel pciutils net-tools sqlite-devel blas-devel lapack-devel openblas-devel gcc-gfortran

Step 3 Run the following command to download the Python 3.7.5 source code package:

wget https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz

Step 4 Go to the download directory and run the following command to decompress the source code package:

tar -zxvf Python-3.7.5.tgz

Step 5 Go to the decompressed folder and run the following commands to install Python:

cd Python-3.7.5

./configure --prefix=*/usr/local/python3.7.5* **--enable-shared**

make

sudo make install

NOTE

--prefix: specifies the Python installation path.

Step 6 Run the following commands to set the soft link:

```
sudo ln -s /usr/local/python3.7.5/bin/python3 /usr/local/python3.7.5/bin/python3.7.5
sudo ln -s /usr/local/python3.7.5/bin/pip3 /usr/local/python3.7.5/bin/pip3.7.5
```

Step 7 Set the Python 3.7.5 environment variables.

1. Run the **vi ~/.bashrc** command in any directory as the running user to open the **.bashrc** file and append the following content to the file:

 Copyright © Huawei Technologies Co., Ltd.

```
# Set the Python 3.7.5 library path.
export LD_LIBRARY_PATH=/usr/local/python3.7.5/lib:$LD_LIBRARY_PATH
# If multiple Python 3 versions exist in the user environment, specify Python 3.7.5.
export PATH=/usr/local/python3.7.5/bin:$PATH
```

2. Run the **:wq!** command to save the file and exit.
3. Run the **source ~/.bashrc** command for the modification to take effect immediately.

Step 8 After the installation is complete, run the following commands to check the version:

python3.7.5 --version

pip3.7.5 --version

If the following information is displayed, the installation is successful:

```
Python 3.7.5
pip 19.2.3 from /usr/local/python3.7.5/lib/python3.7/site-packages/pip (python 3.7)
```

----End

Installing Ansible

Step 1 Log in to the management node as the **root** user.

Step 2 Run the following command to download the Ansible source code package:

wget --no-check-certificate https://releases.ansible.com/ansible/ansible-2.9.7.tar.gz

Step 3 Go to the download directory and run the following command to decompress the source code package:

tar -zxvf ansible-2.9.7.tar.gz

Step 4 Go to the decompressed folder and run the following commands to install Ansible:

cd ansible-2.9.7

python3.7 setup.py install --record files.txt

mkdir -p /etc/ansible

cp -rf examples/ansible.cfg examples/hosts /etc/ansible/

ln -sf /usr/local/python3.7.5/bin/ansible* /usr/local/bin/

Step 5 After the installation is complete, run the following commands to check the version:

ansible --version

If the following information is displayed, the installation is successful:

```
ansible 2.9.7
config file = /etc/ansible/ansible.cfg
configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
ansible python module location = /usr/local/python3.7.5/lib/python3.7/site-packages/ansible-2.9.7-
py3.7.egg/ansible
executable location = /usr/local/bin/ansible
python version = 3.7.5 (default, Nov  9 2020, 03:44:00) [GCC 7.5.0]
```

----End

 Copyright © Huawei Technologies Co., Ltd.

2.8.2.2 Installing Python and Ansible Offline

Obtaining Software Packages

Prepare the Ansible installation package and dependencies required for the installation, and compress them into a .zip package based on the format described in the following table. The software packages are classified into Ubuntu and CentOS software packages. You need to obtain the software packages based on the actual OS.

NOTICE

- Python and Ansible offline installation packages: used to install Python 3.7.5 and Ansible 2.9.7 on the management node. This software is installed only on the management node. Obtain the corresponding software packages based on the actual OS of the management node.
- Download the software packages using the methods provided in this document. The software package versions in the table are only examples and may be different from the actual versions, which does not affect the use of the software packages.

- Ubuntu 18.04

 Copyright © Huawei Technologies Co., Ltd.

Table 2-22 Python and Ansible offline installation packages

Package Name	Package Content	How to Obtain
base-pkg-arm64.zip	• ansible-2.9.7.tar.gz • Python-3.7.5.tgz • setuptools-19.6.tar.gz	• **wget --no-check-certificate https://releases.ansible.com/ansible/ansible-2.9.7.tar.gz** • **wget --no-check-certificate https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz** • **wget --no-check-certificate https://pypi.python.org/packages/source/s/setuptools/setuptools-19.6.tar.gz**
	• cffi-1.14.3.tar.gz • pycparser-2.20-py2.py3-none-any.whl • cryptography-3.1.1.tar.gz • six-1.15.0-py2.py3-none-any.whl • glob3-0.0.1.tar.gz • Jinja2-2.11.2-py2.py3-none-any.whl • MarkupSafe-1.1.1.tar.gz • PyYAML-5.3.1.tar.gz	**pip3.7 download cffi==1.14.3 cryptography==3.1.1 glob3 Jinja2 PyYAML==5.3.1**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• dos2unix_7.3.4-3_arm64.deb • haveged_1.9.1-6_arm64.deb • libffi-dev_3.2.1-8_arm64.deb • libhavege1_1.9.1-6_arm64.deb • sshpass_1.06-1_arm64.deb • zlib1g-dev_1%3a1.2.11.dfsg-0ubuntu2_arm64.deb	**apt-get download dos2unix haveged libffi-dev libhavege1 sshpass zlib1g-dev**
base-pkg-amd64.zip	• ansible-2.9.7.tar.gz • Python-3.7.5.tgz • setuptools-19.6.tar.gz	• **wget --no-check-certificate https://releases.ansible.com/ansible/ansible-2.9.7.tar.gz** • **wget --no-check-certificate https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz** • **wget --no-check-certificate https://pypi.python.org/packages/source/s/setuptools/setuptools-19.6.tar.gz**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• cffi-1.14.3-cp37-cp37m-manylinux1_x86_64.whl • pycparser-2.20-py2.py3-none-any.whl • cryptography-3.1-cp35-abi3-manylinux2010_x86_64.whl • six-1.15.0-py2.py3-none-any.whl • glob3-0.0.1.tar.gz • Jinja2-2.11.2-py2.py3-none-any.whl • MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl • PyYAML-5.3.1.tar.gz	**pip3.7 download cffi==1.14.3 cryptography==3.1.1 glob3 Jinja2 PyYAML==5.3.1**
	• dos2unix_7.3.4-3_amd64.deb • haveged_1.9.1-6_amd64.deb • libffi-dev_3.2.1-8_amd64.deb • libhavege1_1.9.1-6_amd64.deb • sshpass_1.06-1_amd64.deb • zlib1g-dev_1%3a1.2.11.dfsg-0ubuntu2_amd64.deb	**apt-get download dos2unix haveged libffi-dev libhavege1 sshpass zlib1g-dev**

- CentOS 7.6

 Copyright © Huawei Technologies Co., Ltd.

Table 2-23 Python and Ansible offline installation packages

Package Name	Package Content	How to Obtain
base-pkg-arm64.zip	• ansible-2.9.7.tar.gz • Python-3.7.5.tgz • perl-5.28.0.tar.gz • openssl-1.1.1a.tar.gz	• **wget --no-check-certificate https://releases.ansible.com/ansible/ansible-2.9.7.tar.gz** • **wget --no-check-certificate https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz** • **wget --no-check-certificate https://www.cpan.org/src/5.0/perl-5.28.0.tar.gz** • **wget --no-check-certificate https://www.openssl.org/source/openssl-1.1.1a.tar.gz**
	• cffi-1.14.3.tar.gz • pycparser-2.20-py2.py3-none-any.whl • cryptography-3.2.1-cp35-abi3-manylinux2014_aarch64.whl • pip-20.2.4-py2.py3-none-any.whl • six-1.15.0-py2.py3-none-any.whl • Jinja2-2.11.2-py2.py3-none-any.whl • MarkupSafe-1.1.1.tar.gz • PyYAML-5.3.1.tar.gz • setuptools-50.3.2-py3-none-any.whl	**pip3.7 download pip==20.2.4 cffi==1.14.3 cryptography==3.2.1 Jinja2==2.11.2 PyYAML==5.3.1 setuptools==50.3.2**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• zlib-devel-1.2.7-18.el7.aarch64.rpm • bzip2-devel-1.0.6-13.el7.aarch64.rpm • epel-release-7-11.noarch.rpm • ncurses-devel-5.9-14.20130511.el7_4.aarch64.rpm • mpfr-3.1.1-4.el7.aarch64.rpm • libmpc-1.0.1-3.el7.aarch64.rpm • kernel-headers-4.18.0-193.28.1.el7.aarch64.rpm • glibc-2.17-317.el7.aarch64.rpm • glibc-common-2.17-317.el7.aarch64.rpm • glibc-headers-2.17-317.el7.aarch64.rpm • glibc-devel-2.17-317.el7.aarch64.rpm • cpp-4.8.5-44.el7.aarch64.rpm • libgcc-4.8.5-44.el7.aarch64.rpm • libgomp-4.8.5-44.el7.aarch64.rpm • gcc-4.8.5-44.el7.aarch64.rpm • libstdc++-4.8.5-44.el7.aarch64.rpm • libstdc++-devel-4.8.5-44.el7.aarch64.rpm	**yum install --downloadonly --downloaddir=***Download path* **gcc-c++ libffi-devel zlib-devel bzip2-devel epel-release nurses-devel unzip sshpass dos2unix haveged**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• gcc-c++-4.8.5-44.el7.aarch64.rpm • libffi-devel-3.0.13-19.el7.aarch64.rpm • libffi-3.0.13-19.el7.aarch64.rpm • unzip-6.0-21.el7.aarch64.rpm • sshpass-1.06-2.el7.aarch64.rpm • dos2unix-6.0.3-7.el7.aarch64.rpm • haveged-1.9.1-1.el7.aarch64.rpm	
base-pkg-amd64.zip	• ansible-2.9.7.tar.gz • Python-3.7.5.tgz • perl-5.28.0.tar.gz • openssl-1.1.1a.tar.gz	• **wget --no-check-certificate https://releases.ansible.com/ansible/ansible-2.9.7.tar.gz** • **wget --no-check-certificate https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz** • **wget --no-check-certificate https://www.cpan.org/src/5.0/perl-5.28.0.tar.gz** • **wget --no-check-certificate https://www.openssl.org/source/openssl-1.1.1a.tar.gz**

 Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• cffi-1.14.3-cp37-cp37m-manylinux1_x86_64.whl • pycparser-2.20-py2.py3-none-any.whl • cryptography-3.2.1-cp35-abi3-manylinux2010_x86_64.whl • six-1.15.0-py2.py3-none-any.whl • Jinja2-2.11.2-py2.py3-none-any.whl • MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl • PyYAML-5.3.1.tar.gz • setuptools-50.3.2-py3-none-any.whl	**pip3.7 download cffi==1.14.3 cryptography==3.2.1 Jinja2==2.11.2 PyYAML==5.3.1 setuptools==50.3.2**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• zlib-devel-1.2.7-18.el7.x86_64.rpm • bzip2-devel-1.0.6-13.el7.x86_64.rpm • epel-release-7-11.noarch.rpm • ncurses-devel-5.9-14.20130511.el7_4.x86_64.rpm • mpfr-3.1.1-4.el7.x86_64.rpm • libmpc-1.0.1-3.el7.x86_64.rpm • kernel-headers-3.10.0-1127.19.1.el7.x86_64.rpm • glibc-2.17-307.el7.1.x86_64.rpm • glibc-common-2.17-307.el7.1.x86_64.rpm • glibc-headers-2.17-307.el7.1.x86_64.rpm • glibc-devel-2.17-307.el7.1.x86_64.rpm • cpp-4.8.5-39.el7.x86_64.rpm • libgcc-4.8.5-39.el7.x86_64.rpm • libgomp-4.8.5-39.el7.x86_64.rpm • gcc-4.8.5-39.el7.x86_64.rpm • libstdc++-4.8.5-39.el7.x86_64.rpm • libstdc++-devel-4.8.5-39.el7.x86_64.rpm • gcc-c++-4.8.5-39.el7.x86_64.rpm	**yum install --downloadonly --downloaddir=***Download path* **gcc-c++ libffi-devel zlib-devel bzip2-devel epel-release nurses-devel unzip sshpass dos2unix haveged**

Copyright © Huawei Technologies Co., Ltd.

Package Name	Package Content	How to Obtain
	• libffi-devel-3.0.13-19.el7.x86_64.rpm • libffi-3.0.13-19.el7.x86_64.rpm • unzip-6.0-21.el7.x86_64.rpm • sshpass-1.06-2.el7.x86_64.rpm • dos2unix-6.0.3-7.el7.x86_64.rpm • haveged-1.9.13-1.el7.x86_64.rpm	

Installing Python and Ansible

Step 1 Log in to the management node as the **root** user.

Step 2 Copy the **obtained software packages** to any directory on the server and decompress them.

Step 3 Install the Python development environment.

Table 2-24 Python installation commands

OS	Command
Ubuntu	**dpkg -i dos2unix*.deb zlib1g-dev*.deb libffi-dev*.deb** **tar -zxvf Python-3.7.5.tgz** **cd Python-3.7.5** **./configure --prefix=/usr/local/python3.7.5 --enable-shared** **make** **sudo make install** **sudo cp /usr/local/python3.7.5/lib/libpython3.7m.so.1.0 /usr/lib** **sudo ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3.7** **sudo ln -s /usr/local/python3.7.5/bin/pip3 /usr/bin/pip3.7** **sudo ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3.7.5** **sudo ln -s /usr/local/python3.7.5/bin/pip3 /usr/bin/pip3.7.5**

Copyright © Huawei Technologies Co., Ltd.

OS	Command
CentOS	**yum install *.rpm** **tar -xzf perl-5.28.0.tar.gz** **cd perl-5.28.0** **./Configure -des -Dprefix=$HOME/localperl** **make && make install** **cd ..** **tar -zxvf openssl-1.1.1a.tar.gz** **cd openssl-1.1.1a** **./config --prefix=/usr/local/openssl no-zlib** **make && make install** **mv /usr/bin/openssl /usr/bin/openssl.bak** **ln -s /usr/local/openssl/include/openssl /usr/include/openssl** **ln -s /usr/local/openssl/lib/libssl.so.1.1 /usr/local/lib64/libssl.so** **ln -s /usr/local/openssl/bin/openssl /usr/bin/openssl** **echo "/usr/local/openssl/lib" >> /etc/ld.so.conf** **ldconfig** **cd ..** **tar -xzvf Python-3.7.5.tgz** **cd Python-3.7.5** **./configure --prefix=/usr/local/python3.7.5 --enable-shared --with-openssl=/usr/local/openssl** **make && make install** **ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3** **ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3.7** **ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3.7.5** **echo "/usr/local/python3.7.5/lib" > /etc/ld.so.conf.d/python3.7.conf** **ldconfig** **ln -s /usr/local/python3.7.5/bin/pip3.7 /usr/bin/pip3** **ln -s /usr/local/python3.7.5/bin/pip3.7 /usr/bin/pip3.7** **ln -s /usr/local/python3.7.5/bin/pip3.7 /usr/bin/pip3.7.5**

 Copyright © Huawei Technologies Co., Ltd.

Step 4 Install Ansible.

Table 2-25 Ansible installation commands

OS	Architecture	Command
Ubuntu	• ARM • x86	**pip3.7 install Jinja2-2.11.2* MarkupSafe-1.1.1* PyYAML-5.3.1* pycparser-2.20* cffi-1.14.3* six-1.15.0* cryptography-3.1*** **tar zxf setuptools-19.6.tar.gz** **cd setuptools-19.6** **python3.7 setup.py install** **cd ..** **dpkg -i libhavege1_1.9.1-6*.deb** **dpkg -i haveged_1.9.1-6*.deb** **tar -zxvf ansible-2.9.7.tar.gz** **cd ansible-2.9.7** **python3.7 setup.py install --record files.txt** **mkdir -p /etc/ansible** **cp -rf examples/ansible.cfg examples/hosts /etc/ansible/** **ln -sf /usr/local/python3.7.5/bin/ansible* /usr/local/bin/** **cd ..** **dpkg -i sshpass_1.06-1*.deb**

 Copyright © Huawei Technologies Co., Ltd.

OS	Architecture	Command
CentOS	ARM	**rpm -ivh haveged*.rpm** **tar zxf MarkupSafe-1.1.1.tar.gz** **cd MarkupSafe-1.1.1** **python3.7 setup.py install** **cd ..** **pip3.7 install pip-20.2.4* Jinja2-2.11.2* pycparser-2.20* six-1.15.0* setuptools-50.3.2* cryptography-3.2.1*** **tar zxf cffi-1.14.3.tar.gz** **cd cffi-1.14.3** **python3.7 setup.py install** **cd ..** **tar zxf PyYAML-5.3.1.tar.gz** **cd PyYAML-5.3.1** **python3.7 setup.py install** **cd ..** **tar zxf ansible-2.9.7.tar.gz** **cd ansible-2.9.7** **python3.7 setup.py install** **mkdir -p /etc/ansible** **cp -rf examples/ansible.cfg examples/hosts /etc/ansible/** **ln -sf /usr/local/python3.7.5/bin/ansible* /usr/local/bin/** **cd ..** **rpm -ivh unzip*.rpm** **rpm -ivh sshpass-1.06*.rpm** **rpm -ivh dos2unix-6.0.3*.rpm**

 Copyright © Huawei Technologies Co., Ltd.

OS	Architecture	Command
	x86	**pip3.7 install Jinja2-2.11.2* MarkupSafe-1.1.1* pycparser-2.20* cffi-1.14.3* six-1.15.0* cryptography-3.2.1* setuptools-50.3.2*** **tar zxf PyYAML-5.3.1.tar.gz** **cd PyYAML-5.3.1** **python3.7 setup.py install** **cd ..** **rpm -ivh haveged*.rpm** **tar -zxvf ansible-2.9.7.tar.gz** **cd ansible-2.9.7** **python3.7 setup.py install** **mkdir -p /etc/ansible** **cp -rf examples/ansible.cfg examples/hosts /etc/ansible/** **ln -sf /usr/local/python3.7.5/bin/ansible* /usr/local/bin/** **cd ..** **rpm -ivh unzip*.rpm** **rpm -ivh sshpass-1.06*.rpm** **rpm -ivh dos2unix-6.0.3*.rpm**

----End

2.8.3 Configuring Ansible Host Information

This section describes how to configure Ansible host information in the single-node system and cluster scenarios.

Prerequisites

Python 3.7.5 and Ansible have been installed on the management node. For details, see **Checking the Python and Ansible Versions**.

Precautions

- You are advised to run the **chmod 400 /etc/ansible/hosts** command to set the permission on the **hosts** file in the **/etc/ansible** directory to **400**.
- You are advised to back up the **hosts** file that has been used in the **/etc/ansible** directory for subsequent log collection or reinstallation. The **hosts** file contains the IP address of the server and the username and password for logging in to the server. After the backup is complete, delete the **hosts** file from the server.

 Copyright © Huawei Technologies Co., Ltd.

Single-Node Scenario

Step 1 Log in to the management node as the **root** user.

Step 2 Run the following command to edit the **hosts** file:

vim /etc/ansible/hosts

Modify the following content based on the actual situation. Do not modify the following template structure.

```
[all:vars]
# default shared directory, you can change it as yours
nfs_shared_dir=/data/atlas_dls

# NFS service IP
nfs_service_ip=nfs-host-ip

# Master IP
master_ip=master-host-ip

# dls install package dir
dls_root_dir=install_dir

# set proxy
proxy=proxy_address

# Command for logging in to the Ascend hub
ascendhub_login_command=login_command

# Generally, you do not need to change the value or delete it.
ascendhub_prefix="swr.cn-south-1.myhuaweicloud.com/public-ascendhub"

# versions
deviceplugin_version="v20.2.0"
cadvisor_version="v0.34.0-r40"
volcano_version="v1.0.1-r40"
hccl_version="v20.2.0"

[nfs_server]
single-node-host-name ansible_host=IP ansible_ssh_user="username" ansible_ssh_pass="passwd"

[localnode]
single-node-host-name ansible_host=IP ansible_ssh_user="username" ansible_ssh_pass="passwd"

[training_node]

[inference_node]

[A300T_node]

[arm]

[x86]

[workers:children]
training_node
inference_node
A300T_node
```

Set the following parameters based on the actual situation:

- ***nfs-host-ip***: IP address of the NFS server, that is, the IP address of the server. If the NFS is not installed, set **nfs-host-ip** to an empty string, for example, "".
- ***master-host-ip***: IP address of the management node server, that is, the server IP address.

 Copyright © Huawei Technologies Co., Ltd.

- ***install_dir***: directory to which the basic software package, image package, and **yamls** folder are uploaded.
- ***proxy_address***: proxy address. Set this parameter based on the site requirements. If no proxy is required, set this parameter to an empty string, for example, "".
- ***login_command***: login command used to obtain images from the **Ascend Hub**. This parameter is required only for online installation, for example, "**docker login -u** *xxxxxx@xxxxxx* **-p** *xxxxxxxx* **swr.cn-south-1.myhuaweicloud.com**". For details about how to obtain the login command, see **Step 1** to **Step 2** in **Obtaining MindX DL Images**. This parameter can be set to an empty string, for example, "", for offline installation.
- ***single-node-host-name***: hostname of a single node. You can run the **hostname** command to view the hostname.
- ***IP***: server IP address.
- ***username***: username for logging in to the server.
- ***passwd***: password for logging in to the server.

NOTE

- If the server is a training server, copy the host line under **[localnode]** to **[training_node]**.
- If the server is an inference server, copy the host line under the **[localnode]** label to **[inference_node]**.
- If Atlas 300T training card is configured on the server, copy the host line under **[localnode]** to **[A300T_node]**.
- If the server is an ARM server, copy the host line under **[localnode]** to **[arm]**.
- If the server is an x86 server, copy the host line under **[localnode]** to **[x86]**.

Step 3 Run the following command to edit **ansible.cfg**:

vim /etc/ansible/ansible.cfg

Uncomment the following content and change the value of **deprecation_warnings** to **False**:

```
log_path = /var/log/ansible.log
host_key_checking = False
deprecation_warnings = False
```

----End

Cluster Scenario

Step 1 Log in to the management node as the **root** user.

Step 2 Run the following command to edit the **hosts** file:

vim /etc/ansible/hosts

Modify the following content based on the actual situation. Do not modify the following template structure.

```
[all:vars]
# default shared directory, you can change it as yours
nfs_shared_dir=/data/atlas_dls
```

 Copyright © Huawei Technologies Co., Ltd.

```
# NFS service IP
nfs_service_ip=nfs-host-ip

# Master IP
master_ip=master-host-ip

# dls install package dir
dls_root_dir=install_dir

# set proxy
proxy=proxy_address

# Command for logging in to the Ascend hub
ascendhub_login_command=login_command

# Generally, you do not need to change the value or delete it.
ascendhub_prefix="swr.cn-south-1.myhuaweicloud.com/public-ascendhub"

# versions
deviceplugin_version="v20.2.0"
cadvisor_version="v0.34.0-r40"
volcano_version="v1.0.1-r40"
hccl_version="v20.2.0"

[nfs_server]
nfs-host-name ansible_host=nfs-host-ip ansible_ssh_user="username" ansible_ssh_pass="password"

[master]
master-host-name ansible_host=master-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"

[training_node]
training-node1-host-name ansible_host=training-node1-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
training-node2-host-name ansible_host=training-node2-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
...

[inference_node]
inference-node1-host-name ansible_host=inference-node1-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
inference-node2-host-name ansible_host=inference-node2-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
...

[A300T_node]
A300T-node1-host-name ansible_host=A300T-node1-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
A300T-node2-host-name ansible_host=A300T-node2-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
...

[arm]
arm-node1-host-name ansible_host=inference-node1-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
arm-node2-host-name ansible_host=inference-node2-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"

[x86]
x86-node1-host-name ansible_host=inference-node1-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"
x86-node2-host-name ansible_host=inference-node2-host-ip ansible_ssh_user="username"
ansible_ssh_pass="password"

[workers:children]
training_node
inference_node
A300T_node
```

 Copyright © Huawei Technologies Co., Ltd.

```
[cluster:children]
master
workers
```

Set the following parameters based on the actual situation:

- ***nfs-host-ip***: IP address of the NFS server. If the NFS is not installed, set **nfs-host-ip** to an empty string, for example, "".
- ***master-host-ip***: IP address of the management node server.
- ***install_dir***: directory to which the basic software package, image package, and **yamls** folder are uploaded.
- ***proxy_address***: proxy address. Set this parameter based on the site requirements. If no proxy is required, set this parameter to an empty string, for example, "".
- ***login_command***: login command used to obtain images from the **Ascend Hub**. This parameter is required only for online installation, for example, "**docker login -u** *xxxxxx@xxxxxx* **-p** *xxxxxxxx* **swr.cn-south-1.myhuaweicloud.com**". For details about how to obtain the login command, see **Step 1** to **Step 2** in **Obtaining MindX DL Images**. This parameter can be set to an empty string, for example, "", for offline installation.
- ***XXX-host-name***: hostname of a node. The value must be the same as the hostname of the OS and must be unique. You can run the **hostname** command on the node to view the hostname. The hostname of the Kubernetes cluster must be unique.
- ***XXX-host-ip***: IP address of the node.
- ***username***: username of the corresponding node.
- ***passwd***: user password of the node.

NOTE

- If the server is a training server, add the server node information to **[training_node]**.
- If the server is an inference server, add the server node information to **[inference_node]**.
- If Atlas 300T training cards are configured on the server, add the node information to **[A300T_node]**.
- If the server is an ARM compute node, add the server node information to **[arm]**.
- If the server is an x86 compute node, add the server node information to **[x86]**.

Step 3 Run the following command to edit **ansible.cfg**:

vim /etc/ansible/ansible.cfg

Uncomment the following content and change the value of **deprecation_warnings** to **False**:

```
log_path = /var/log/ansible.log
host_key_checking = False
deprecation_warnings = False
```

----End

 Copyright © Huawei Technologies Co., Ltd.

2.8.4 Obtaining MindX DL Images

Procedure

Step 1 Log in to **Ascend Hub**.

You can obtain the operation commands only after the account is activated. If you do not have an account, register one.

1. Register an account.

   Register an account as prompted on the **Ascend Hub** login page.

2. Activate the account.

   Log in to **Ascend Hub** and click **Activate Account**. Submit the activation application as prompted. The activation takes effect after being approved by the administrator.

Step 2 Obtain and copy the **docker login** command.

1. Click to open an image.

   The image page is displayed.

2. Click next to **docker login *XXX*** to copy the login command.

NOTICE

The validity period of the **docker login** command is one day. If the command has expired, obtain the command again.

Step 3 Log in to the server as the **root** user and run the **docker login** command obtained in **Step 2**.

- If the following information is displayed, the login is successful. Go to **Step 4**.

```
[root@centos39 ~]# docker login -u xxxxxx -p xxxxxxxx swr.cn-south-1.myhuaweicloud.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store


Login Succeeded
[root@centos39 ~]#
```

- If the following information is displayed, the login fails:

```
[root@centos39 ~]# docker login -u xxxxx -p xxxxxxx swr.cn-south-1.myhuaweicloud.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Error response from daemon: Get http://swr.cn-south-1.myhuaweicloud.com/v2/: dial tcp: lookup
swr.cn-south-1.myhuaweicloud.com on 127.0.0.53:53: server misbehaving
```

  The possible cause is that the proxy is not configured. To configure the proxy, perform the following steps:

  a. Run the following command to create proxy configuration file **proxy.conf** for Docker:

     mkdir -p /etc/systemd/system/docker.service.d

     vim /etc/systemd/system/docker.service.d/proxy.conf

  b. Add the following information to the **proxy.conf** file, set the proxy address based on the site requirements, save the file, and exit.

Copyright © Huawei Technologies Co., Ltd.

```
[Service]
Environment="HTTP_PROXY=http://xxxx.xxx.xxxx.xxxx"
Environment="HTTPS_PROXY=http://xxxx.xxx.xxxx.xxxx"
```

c. Run the following command to configure **insecure-registries** for Docker:

vim /etc/docker/daemon.json

- If the **insecure-registries** field exists in the file, add the following content to the field:

```
swr.cn-south-1.myhuaweicloud.com
```

Example:

```
{

    "insecure-registries": ["xxxxxxxxxxxxxxx",
                            "swr.cn-south-1.myhuaweicloud.com"],

}
```

- If the **insecure-registries** field does not exist in the file, add it to the file.

Example:

```
{

  "exec-opts": ["native.cgroupdriver=systemd"], # Assume that this line is the last line of the original /etc/docker/daemon.json file. After adding the insecure-registries field, you need to add a comma (,) after the square bracket (]) in this line.
  "insecure-registries": ["swr.cn-south-1.myhuaweicloud.com"]

}
```

d. Restart the Docker service.

```
systemctl daemon-reload;
systemctl restart docker;
```

Step 4 Obtain proper component images.

Table 2-26 lists the component images to be obtained for each server.

Table 2-26 Image list

Component Name	Target Server
Ascend Device Plugin	Compute node with NPUs
Volcano	Management node where Kubernetes is installed
HCCL-Controller	Management node where Kubernetes is installed
cAdvisor	Compute node with NPUs

NOTICE

Change the image version in each command with the actual one.

Copyright © Huawei Technologies Co., Ltd.

Table 2-27 Command

Architecture	Component	Command
ARM	Ascend Device Plugin	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/ascend-k8sdeviceplugin_arm64:v20.2.0**
	Volcano	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_arm64:v1.0.1-r40** **docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-scheduler_arm64:v1.0.1-r40** **docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager_arm64:v1.0.1-r40** **docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager-base_arm64:v1.0.1-r40**
	HCCL-Controller	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/hccl-controller_arm64:v20.2.0**
	cAdvisor	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/cadvisor_arm64:v0.34.0-r40**
x86	Ascend Device Plugin	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/ascend-k8sdeviceplugin_amd64:v20.2.0**
	Volcano	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_amd64:v1.0.1-r40** **docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-scheduler_amd64:v1.0.1-r40** **docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager_amd64:v1.0.1-r40** **docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager-base_amd64:v1.0.1-r40**
	HCCL-Controller	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/hccl-controller_amd64:v20.2.0**
	cAdvisor	**docker pull swr.cn-south-1.myhuaweicloud.com/public-ascendhub/cadvisor_amd64:v0.34.0-r40**

Step 5 Rename the image.

 Copyright © Huawei Technologies Co., Ltd.

Run the following command to rename the image obtained in **Step 4** and delete the original image:

docker tag *<Old image name>:<Old image version> <Newimage name>:<New image version>*

Example:

docker tag *swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_arm64:v1.0.1-r40 volcanosh/vc-controller-manager:v1.0.1-r40*

For details, see **Table 2-28**.

Copyright © Huawei Technologies Co., Ltd.

Table 2-28 Commands

Architecture	Command
ARM	**docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_arm64:v1.0.1-r40 volcanosh/vc-controller-manager:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-scheduler_arm64:v1.0.1-r40 volcanosh/vc-scheduler:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager_arm64:v1.0.1-r40 volcanosh/vc-webhook-manager:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager-base_arm64:v1.0.1-r40 volcanosh/vc-webhook-manager-base:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/hccl-controller_arm64:v20.2.0 hccl-controller:v20.2.0** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/ascend-k8sdeviceplugin_arm64:v20.2.0 ascend-k8sdeviceplugin:v20.2.0** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/cadvisor_arm64:v0.34.0-r40 google/cadvisor:v0.34.0-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_arm64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-scheduler_arm64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager_arm64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager-base_arm64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/hccl-controller_arm64:v20.2.0** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/ascend-k8sdeviceplugin_arm64:v20.2.0** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/cadvisor_arm64:v0.34.0-r40**

Copyright © Huawei Technologies Co., Ltd.

Architecture	Command
x86	**docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_amd64:v1.0.1-r40 volcanosh/vc-controller-manager:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-scheduler_amd64:v1.0.1-r40 volcanosh/vc-scheduler:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager_amd64:v1.0.1-r40 volcanosh/vc-webhook-manager:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager-base_amd64:v1.0.1-r40 volcanosh/vc-webhook-manager-base:v1.0.1-r40** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/hccl-controller_amd64:v20.2.0 hccl-controller:v20.2.0** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/ascend-k8sdeviceplugin_amd64:v20.2.0 ascend-k8sdeviceplugin:v20.2.0** **docker tag swr.cn-south-1.myhuaweicloud.com/public-ascendhub/cadvisor_amd64:v0.34.0-r40 google/cadvisor:v0.34.0-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-controller-manager_amd64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-scheduler_amd64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager_amd64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/vc-webhook-manager-base_amd64:v1.0.1-r40** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/hccl-controller_amd64:v20.2.0** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/ascend-k8sdeviceplugin_amd64:v20.2.0** **docker rmi swr.cn-south-1.myhuaweicloud.com/public-ascendhub/cadvisor_amd64:v0.34.0-r40**

----End

2.8.5 Building MindX DL Images

Precautions

The images built for x86 servers and ARM servers are different. Build the image based on your server CPU architecture.

Copyright © Huawei Technologies Co., Ltd.

Environment Dependencies

Table 2-29 Environment dependencies

Software	Version	How to Obtain
Go	1.14 or later	Official Go website Download the required version from **https://golang.org/dl/**.
Docker	For details, see the version mapping of the Kubernetes software.	Official Docker website Download the required version from **https://docs.docker.com/engine/install/**.
Git	-	Official Git website Download the required version from **https://git-scm.com/downloads**.

Uploading Building Files

Step 1 Log in as the **root** user.

- Log in to the management node for Volcano and HCCL-Controller.
- Log in to each compute node for Ascend Device Plugin and cAdvisor.

Step 2 Run the following command to create a directory:

mkdir -p ${GOPATH}/{src/github.com/google,src/k8s.io,src/volcano.sh}

Step 3 Upload the **ascend-for-cadvisor** folder obtained from **https://gitee.com/ascend/ascend-for-cadvisor** to the **${GOPATH}/src/github.com/google/** directory and rename the folder **cadvisor**.

NOTE

If you do not have the permission, contact Huawei technical support.

Step 4 Upload the **ascend-for-volcano** folder obtained from **https://gitee.com/ascend/ascend-for-volcano** to the **${GOPATH}/src/volcano.sh/** directory and rename the folder **volcano**.

Step 5 Upload the **ascend-device-plugin** folder obtained from **https://gitee.com/ascend/ascend-device-plugin** to the **/home/** directory.

Step 6 Upload the **ascend-hccl-controller** folder obtained from **https://gitee.com/ascend/ascend-hccl-controller** to the **/home/** directory.

----End

Building Volcano

Step 1 Log in to the management node as the **root** user.

Copyright © Huawei Technologies Co., Ltd.

Step 2 Run the following command to modify the environment variable:

export GO111MODULE=""

Step 3 Run the following commands to create the **build** folder:

cd ${GOPATH}/src/volcano.sh/volcano/

mkdir -p build

Step 4 Run the following commands to create and edit the **build.sh** file:

cd ${GOPATH}/src/volcano.sh/volcano/build

vim build.sh

Add the following content to the file:

```
#!/bin/sh
cd ${GOPATH}/src/volcano.sh/volcano/
make clean
export PATH=$GOPATH/bin:$PATH
export GO111MODULE=off
export GOMOD=""
export GIT_SSL_NO_VERIFY=1
make webhook-manager-base-image
make image_bins
make images
make generate-yaml

REL_VERSION=v1.0.1-r40
REL_OSARCH="amd64"
OUT_PATH=_output/DockFile/

machine_arch=`uname -m`
if [ $machine_arch = "aarch64" ]; then
 REL_OSARCH="arm64"
fi

mkdir -p $OUT_PATH
docker save -o ${OUT_PATH}/vc-webhook-manager-base-${REL_VERSION}-${REL_OSARCH}.tar.gz
volcanosh/vc-webhook-manager-base:${REL_VERSION}
docker save -o ${OUT_PATH}/vc-webhook-manager-${REL_VERSION}-${REL_OSARCH}.tar.gz volcanosh/vc-
webhook-manager:${REL_VERSION}
docker save -o ${OUT_PATH}/vc-controller-manager-${REL_VERSION}-${REL_OSARCH}.tar.gz volcanosh/vc-
controller-manager:${REL_VERSION}
docker save -o ${OUT_PATH}/vc-scheduler-${REL_VERSION}-${REL_OSARCH}.tar.gz volcanosh/vc-scheduler:$
{REL_VERSION}
```

Step 5 Run the following commands to build the image:

chmod +x build.sh

./build.sh

NOTE

For other information, see the **Volcano compilation guide**.

Step 6 After the images are built, run the **docker images** command to check whether the **volcanosh/vc-webhook-manager:v1.0.1-r40 volcanosh/vc-webhook-manager-base:v1.0.1-r40**, **volcanosh/vc-scheduler:v1.0.1-r40**, and **volcanosh/vc-controller-manager:v1.0.1-r40** images exist.

----End

 Copyright © Huawei Technologies Co., Ltd.

Building HCCL-Controller

Step 1 Log in to the management node as the **root** user.

Step 2 Run the following command to go to the source code building directory:

cd /home/ascend-hccl-controller/build

Step 3 Run the following commands to build the image:

dos2unix *.sh

chmod +x build.sh

./build.sh

Step 4 Run the following command to go to the **/home/ascend-hccl-controller/output** directory to obtain the built binary file and install the YAML file:

cd /home/ascend-hccl-controller/output

Step 5 After the image is generated, run the **docker images** command to check whether the **hccl-controller:v20.2.0** image exists.

----End

Building Ascend Device Plugin

Step 1 Log in to the training node as the **root** user.

Step 2 Go to the device plugin building directory.

cd /home/ascend-device-plugin/build/

Step 3 Run the following commands to set environment variables:

export GO111MODULE=on

export GOPROXY=*Proxy address*

export GONOSUMDB=*

NOTE

- Use the actual GOPROXY proxy address. You can run the **go mod download** command in the **ascend-device-plugin** directory to check the address.
- If no error information is displayed, the proxy is set successfully.

Step 4 Run the following commands to change the file permission and run the .sh files:

chmod +x build.sh

dos2unix build.sh

./build.sh dockerimages

Step 5 After the image is generated, run the **docker images** command to check whether the **ascend-k8sdeviceplugin:v20.2.0** image exists.

----End

 Copyright © Huawei Technologies Co., Ltd.

Building cAdvisor

Step 1 Log in to the training node as the **root** user.

Step 2 Run the following command to modify the environment variable:

export GO111MODULE=""

Step 3 Run the following command to generate an executable cAdvisor binary file:

cd $GOPATH/src/github.com/google/cadvisor

Step 4 Run the following command to create a Docker image:

chmod +x build/*.sh

chmod +x deploy/*.sh

./deploy/build.sh

NOTE

For other information, see the **cAdvisor compilation guide**.

Step 5 After the image is generated, run the **docker images** command to check whether the **google/cadvisor:v0.34.0-r40** image exists.

----End

2.8.6 Modify the Permission of /etc/passwd

During online and offline installation, a user named **hwMindX** is automatically created on the management node and NFS node. Run the **lsattr** command to check the **/etc/passwd** file and ensure that the file does not contain the **-i** attribute, that is, the file can be modified.

- If the file has the **-i** attribute, run the following command to delete the attribute:

  chattr -i /etc/passwd
- After the installation is complete, run the following command to add the **-i** attribute:

  chattr +i /etc/passwd

2.8.7 Installing the NFS

2.8.7.1 Ubuntu

Installing the NFS server

Step 1 Log in to the storage node as an administrator and run the following command to install the NFS server:

apt install -y nfs-kernel-server

Step 2 Run the following command to disable the firewall:

ufw disable

 Copyright © Huawei Technologies Co., Ltd.

Step 3 Run the following commands to create a shared directory (for example, **/data/atlas_dls**) and change the directory permission:

mkdir -p */data/atlas_dls*

chmod 755 */data/atlas_dls/*

Step 4 Run the following command to add the following content to the end of the file:

vi /etc/exports

```
/data/atlas_dls *(rw,sync,no_root_squash)
```

Step 5 Run the following commands to start rpcbind:

systemctl restart rpcbind.service

systemctl enable rpcbind

Step 6 Run the following command to check whether rpcbind is started:

systemctl status rpcbind

If the following information is displayed, the service is running properly:

```
root@ubuntu-211:/data/kfa# service rpcbind status
● rpcbind.service - RPC bind portmap service
   Loaded: loaded (/lib/systemd/system/rpcbind.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-01-08 16:39:03 CST; 6 days ago
     Docs: man:rpcbind(8)
 Main PID: 2952 (rpcbind)
    Tasks: 1 (limit: 29491)
   CGroup: /system.slice/rpcbind.service
           └─2952 /sbin/rpcbind -f -w


Jan 08 16:39:03 ubuntu-211 systemd[1]: Starting RPC bind portmap service...
Jan 08 16:39:03 ubuntu-211 systemd[1]: Started RPC bind portmap service.
```

Step 7 After rpcbind is started, run the following commands to start the NFS service:

systemctl restart nfs-server.service

systemctl enable nfs-server

Step 8 Run the following command to check whether the NFS service is started:

systemctl status nfs-server.service

If the following information is displayed, the service is running properly:

```
root@ubuntu-211:/data/kfa# service nfs-kernel-server status
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)
   Active: active (exited) since Fri 2021-01-08 16:39:03 CST; 6 days ago
 Main PID: 3220 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 29491)
   CGroup: /system.slice/nfs-server.service


Jan 08 16:39:03 ubuntu-211 systemd[1]: Starting NFS server and services...
Jan 08 16:39:03 ubuntu-211 exportfs[3181]: exportfs: /etc/exports [1]: Neither 'subtree_check' or
'no_subtree_check' specified for export "*:/data/atlas_dls".
Jan 08 16:39:03 ubuntu-211 exportfs[3181]:   Assuming default behaviour ('no_subtree_check').
Jan 08 16:39:03 ubuntu-211 exportfs[3181]:   NOTE: this default has changed since nfs-utils version 1.0.x
Jan 08 16:39:03 ubuntu-211 systemd[1]: Started NFS server and services.
```

Step 9 Run the following command to check the mounting permission of the shared directory (for example, **/data/atlas_dls**):

 Copyright © Huawei Technologies Co., Ltd.

cat /var/lib/nfs/etab

If the following information is displayed, the service is running properly:

```
root@ubuntu203:~# cat /var/lib/nfs/etab
/data/atlas_dls
*(rw,sync,wdelay,hide,nocrossmnt,secure,no_root_squash,no_all_squash,no_subtree_check,secure_locks,acl,no
_pnfs,anonuid=65534,anongid=65534,sec=sys,rw,secure,no_root_squash,no_all_squash)
```

----End

Installing the NFS Client

Step 1 Log in to another server as an administrator and run the following command to install the NFS client:

apt install -y nfs-common

Step 2 Run the following commands to start rpcbind:

systemctl restart rpcbind.service

systemctl enable rpcbind

Step 3 After rpcbind is started, run the following commands to start the NFS service:

systemctl restart nfs-server.service

systemctl enable nfs-server

----End

2.8.7.2 CentOS

Installing the NFS server

Step 1 Log in to the storage node as an administrator and run the following command to install the NFS server:

yum install nfs-utils -y

Step 2 Run the following command to disable the firewall:

systemctl stop firewalld.service

systemctl disable firewalld.service

Step 3 Run the following commands to create a shared directory (for example, **/data/atlas_dls**) and change the directory permission:

mkdir -p */data/atlas_dls*

chmod 755 */data/atlas_dls/*

Step 4 Run the following command to add the following content to the end of the file:

vi /etc/exports

```
/data/atlas_dls *(rw,sync,no_root_squash)
```

Step 5 Run the following commands to start rpcbind:

systemctl restart rpcbind.service

 Copyright © Huawei Technologies Co., Ltd.

systemctl enable rpcbind

Step 6 Run the following command to check whether rpcbind is started:

systemctl status rpcbind

If the following information is displayed, the service is running properly:

```
[root@centos39 ~]# systemctl status rpcbind
● rpcbind.service - RPC bind service
   Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-01-15 15:54:44 CST; 28s ago
 Main PID: 63008 (rpcbind)
   CGroup: /system.slice/rpcbind.service
           └─63008 /sbin/rpcbind -w


Jan 15 15:54:44 centos39 systemd[1]: Starting RPC bind service...
Jan 15 15:54:44 centos39 systemd[1]: Started RPC bind service.
```

Step 7 After rpcbind is started, run the following commands to start the NFS service:

systemctl restart nfs

systemctl enable nfs

Step 8 Run the following command to check whether the NFS service is started:

systemctl status nfs

If the following information is displayed, the service is running properly:

```
[root@centos39 ~]# systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor preset: disabled)
  Drop-In: /run/systemd/generator/nfs-server.service.d
           └─order-with-mounts.conf
   Active: active (exited) since Fri 2021-01-15 15:56:15 CST; 8s ago
 Main PID: 67145 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nfs-server.service


Jan 15 15:56:15 centos39 systemd[1]: Starting NFS server and services...
Jan 15 15:56:15 centos39 systemd[1]: Started NFS server and services.
```

Step 9 Run the following command to check the mounting permission of the shared directory (for example, **/data/atlas_dls**):

cat /var/lib/nfs/etab

If the following information is displayed, the service is running properly:

```
[root@centos39 ~]# cat /var/lib/nfs/etab
/data/atlas_dls
*(rw,sync,wdelay,hide,nocrossmnt,secure,no_root_squash,no_all_squash,no_subtree_check,secure_locks,acl,no
_pnfs,anonuid=65534,anongid=65534,sec=sys,rw,secure,no_root_squash,no_all_squash)
```

----End

Installing the NFS Client

Step 1 Log in to another server as an administrator and run the following command to install the NFS client:

yum install nfs-utils -y

 Copyright © Huawei Technologies Co., Ltd.

Step 2 Run the following commands to start rpcbind:

systemctl restart rpcbind.service

systemctl enable rpcbind

Step 3 After rpcbind is started, run the following commands to start the NFS service:

systemctl restart nfs

systemctl enable nfs

----End

2.9 User Information

System Users

User	Description	Initial Password	Password Changing Method
HwHiAiUser	Running user of the .run driver package	Custom	Run the **passwd** command to change the password.
hwMindX	Default user for starting a container	Randomly generated	Run the **passwd** command to change the password.

Container Users

User	Description	Initial Password	Password Changing Method
HwHiAiUser	Running user of the .run driver package	Custom	Run the **passwd** command to change the password.
hwMindX	Default user for starting a container	Randomly generated	Run the **passwd** command to change the password.

 Copyright © Huawei Technologies Co., Ltd.

3 Usage Guidelines

3.1 Instructions

MindX DL is applicable to certain scenarios. You are advised to use MindX DL in the following scenarios:

- A data center performs training and inference.
- A device contains Huawei NPUs.
- Deployment is based on containerization technologies.
- The Kubernetes functions as the basic platform for job scheduling.

3.2 Interconnection Programming Guide

MindX DL is a reference design that uses the deep learning components of Huawei NPUs. It is positioned to enable partners to quickly build a deep learning system. MindX DL is operated by using commands and resource configuration files and does not provide a GUI. You need to convert the resource files required in this section into the required format using **Kubernetes Clients** (supporting multiple languages) and send the converted files to kube-apiserver for programmatic job scheduling so that the files can be integrated into upper-layer systems.

Go

Demo for starting a training job in Go (For reference only. For details, see **https://github.com/kubernetes-client/go**. To simplify the code, error handling is omitted in the example.)

```
package main

import (
 "k8s.io/api/core/v1"
 "k8s.io/client-go/kubernetes"
 "k8s.io/client-go/tools/clientcmd"
 "volcano.sh/volcano/pkg/apis/batch/v1alpha1"
 vcClientset "volcano.sh/volcano/pkg/client/clientset/versioned"
)

func main() {
 config, _ := clientcmd.BuildConfigFromFlags("", "")
```

 Copyright © Huawei Technologies Co., Ltd.

```
 kubeClient, _ := kubernetes.NewForConfig(config)      //Create a Kubernetes native client.
 vcjobClient, _ := vcClientset.NewForConfig(config)    //Create a Volcano client.
 label := make(map[string]string)
 label["ring-controller.atlas"] = "ascend-910"
 cmdata := make(map[string]string)
 cmdata["hccl.json"] = "{\n   \"status\":\"initializing\"\n   }\n"
 cm := &v1.ConfigMap{...}
 kubeClient.CoreV1().ConfigMaps(v1.NamespaceDefault).Create(cm) //Create a ConfigMap using the
Kubernetes source client.
 vcjob := &v1alpha1.Job{...}
 vcjobClient.BatchV1alpha1().Jobs(v1.NamespaceDefault).Create(vcjob) //Create a vcjob using the Volcano
client.
}
```

Java

Demo for starting a training job in Java (For reference only. For details, see **https://github.com/kubernetes-client/java**.)

```
import io.kubernetes.client.openapi.ApiClient;
import io.kubernetes.client.openapi.ApiException;
import io.kubernetes.client.openapi.Configuration;
import io.kubernetes.client.openapi.apis.CoreV1Api;
import io.kubernetes.client.openapi.apis.CustomObjectsApi;
import io.kubernetes.client.openapi.models.V1ConfigMap;
import io.kubernetes.client.util.ClientBuilder;
import io.kubernetes.client.util.KubeConfig;
import io.kubernetes.client.util.Yaml;

import java.io.File;
import java.io.FileReader;
import java.io.IOException;

public class CustomObjExample {
    public static void main(String[] args) throws IOException, ApiException {
        String kubeConfigPath = "~/.kube/config";    //KubeConfig file path
        ApiClient   client = ClientBuilder.kubeconfig(KubeConfig.loadKubeConfig(new
FileReader(kubeConfigPath))).build();
        // ApiClient client = ClientBuilder.cluster().build();  //Used in the K8scluster
        Configuration.setDefaultApiClient(client);
        File cmFile = new File("configMap.yaml");
        V1ConfigMap cm = (V1ConfigMap) Yaml.load(cmFile);
        CoreV1Api api = new CoreV1Api();
        api.createNamespacedConfigMap("default", cm, null, null, null);
        File file = new File("vcjob.yaml");
        VcJob job = (VcJob) Yaml.load(file); //Convert the YAML file into a user-defined VCJob object.
        CustomObjectsApi customObjectsApi = new CustomObjectsApi();
        //Create a Vcjob object.
        customObjectsApi.createNamespacedCustomObject("batch.volcano.sh", "v1alpha1",
                "default", "jobs", job, null, null, null);
    }
}
```

Python

Demo for starting a training job in Python (For reference only. For details, see **https://github.com/kubernetes-client/python**.)

```
from os import path
import yaml
from kubernetes import client, config

def main():
    config.load_kube_config()

    with open(path.join(path.dirname(__file__), "config_map.yaml")) as f:   # Load ConfigMaps from YAML.
        cm = yaml.safe_load(f)
        k8s_core_v1 = client.CoreV1Api
        k8s_core_v1.create_namespaced_config_map(body=cm, namespace="default")
```

 Copyright © Huawei Technologies Co., Ltd.

```
    api = client.CustomObjectsApi()
    vcjob = {...}
    api.create_namespaced_custom_object(          #Create a vcjob using a customized API.
        group="batch.volcano.sh",
        version="v1alpha1",
        namespace="default",
        plural="jobs",
        body=vcjob,
    )

if __name__ == "__main__":
    main()
```

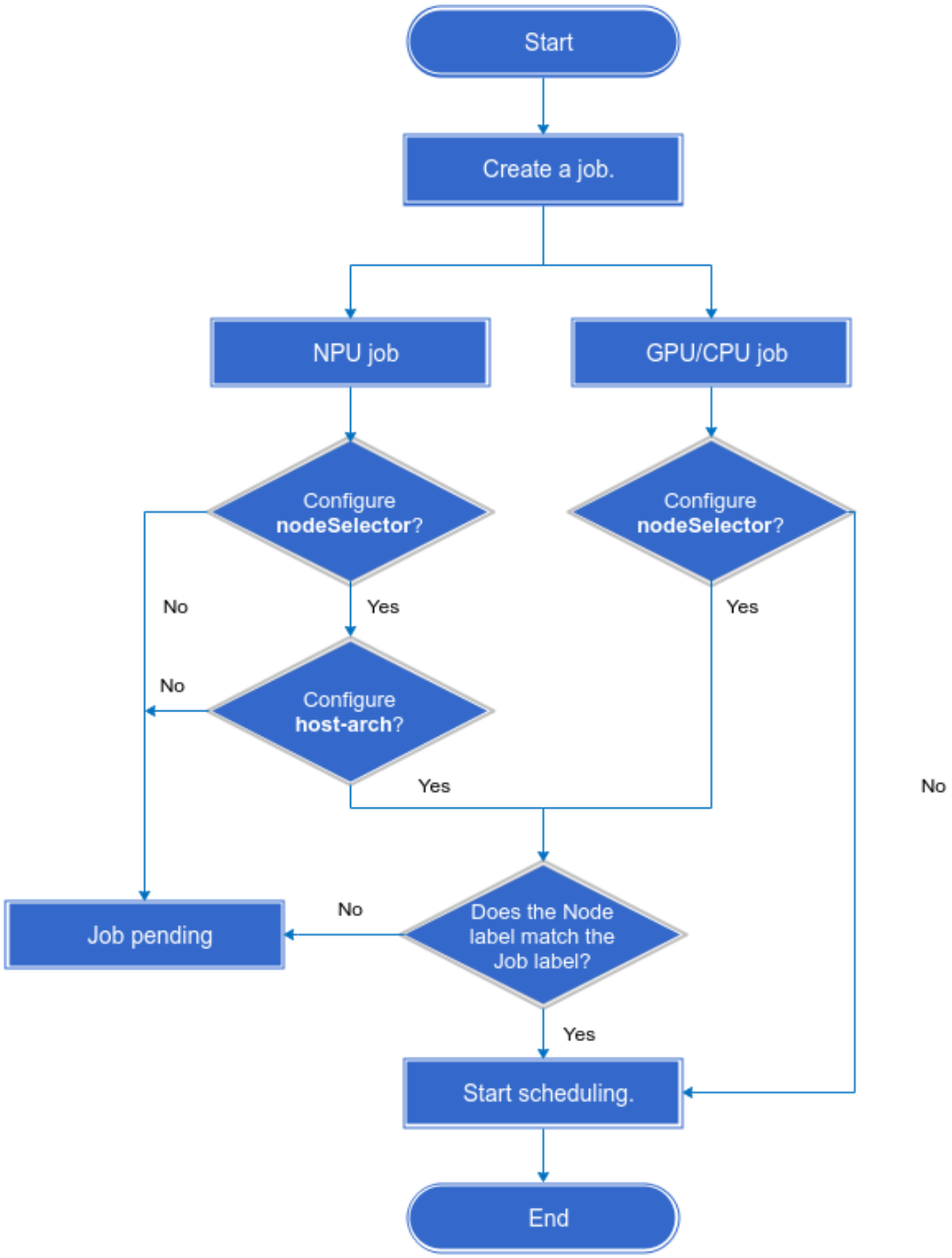
3.3 Scheduling Configuration

To support hybrid deployment of NPUs and GPUs, x86 and ARM platforms, and standard cards and modules, you need to configure labels for each worker node so that MindX DL can schedule worker nodes of different forms.

You can configure a label to specify the node where a job is to be run. Label configuration involves **Job**, **volcano-scheduler**, and **Node**. The three labels must be matched (that is, the labels configured for **Job** can be found in **volcano-scheduler** and **Node**). **Figure 3-1** shows the relationship between them.

 Copyright © Huawei Technologies Co., Ltd.

Figure 3-1 Process of customizing a label



- Jobs are classified into NPU, GPU, and CPU by resource type.
- The **nodeSelector** label of **host-arch** must be configured for NPU jobs. The default content is **huawei-arm** or **huawei-x86**. Modification is not supported.
- If no label is configured for GPU and CPU jobs, the jobs are scheduled based on other rules.
- If a label is configured for a job, the label must match the label configured by **volcano-scheduler**. If they do not match, the job is in the **Pending** state and the reason is provided. If they match, the process goes to the next step.

 Copyright © Huawei Technologies Co., Ltd.

- The job label is in the label list of **volcano-scheduler**. The **volcano-scheduler** needs to select the node with the same label. If there is no such node, the job is in the **Pending** state and the reason is provided. If there is a marched node, the scheduling is performed according to other rules.

Customizing a volcano-scheduler Label

In the Volcano deployment file **volcano-v*.yaml**, set **configurations** in bold as follows.

```
...
data:
  volcano-scheduler.conf: |-
    actions: "enqueue, allocate, backfill"
    tiers:
    - plugins:
      - name: topology910
    - plugins:
      - name: priority
      - name: gang
      - name: conformance
    - plugins:
      - name: drf
      - name: predicates
      - name: proportion
      - name: nodeorder
      - name: binpack
    configurations:
      - arguments: {"host-arch":"huawei-arm|huawei-x86",
      "accelerator":"huawei-Ascend910|nvidia-tesla-v100|nvidia-tesla-p40",
      "accelerator-type":"card|module"
...
```

NOTE

- The configuration mode adopts the map format. Currently, only English input is supported. If a label has multiple values, use vertical bars (|) to separate them.
- If the value of **ascend device plugin** is **Ascend910**, **"host-arch":"huawei-arm|huawei-x86"** in **arguments** is the default configuration and cannot be modified. If you need to use other labels, add them.
- If **host-arch** is set to **huawei-arm|huawei-x86**, it cannot be configured or modified and takes effect only for NPU jobs.

Customizing a Job Label

You can add customized labels to the YAML file of a training job as required. For details about a complete YAML file, see the section "Creating a YAML File." NPU jobs must contain the **nodeSelector** label of **host-arch:huawei-arm** or **host-arch:huawei-x86**. There is no restriction on jobs of other types.

The related configuration of the YAML file is as follows:

```
...
 spec:
        containers:
        ...
        nodeSelector:
          accelerator: nvidia-tesla-v100
        volumes:
...
```

 Copyright © Huawei Technologies Co., Ltd.

Customizing a Node Label

The node label must be operated on the management node where Kubernetes is installed.

- Creating a Node Label

  kubectl label nodes *{HostName} {label_key}={label_value}*

  NOTE

  Parameter description:

  - *{HostName}*: indicates the name of the host to be added.
  - *label_key* and *label_value* must match the configurations in **Job** and **volcano-scheduler**.

  Example:

  kubectl label nodes *ubuntu-11 accelerator=nvidia-tesla-p40*
- Modifying a Node Label

  kubectl label nodes *{HostName} {label_key}={label_value}* **--overwrite=true**

  Example:

  kubectl label nodes *ubuntu-11accelerator=vidia-tesla-p40* **--overwrite=true**
- Deleting a Node Label

  kubectl label nodes *{HostName} {label_key}* **-**

  Example:

  kubectl label nodes *ubuntu-11accelerator* **-**

3.4 ResNet-50 Model Use Examples

3.4.1 TensorFlow

3.4.1.1 Atlas 800 Training Server

3.4.1.1.1 Preparing the NPU Training Environment

After MindX DL is installed, you can use YAML to deliver a vcjob (short for Volcano job, which is a job resource type customized by Volcano) to check whether the system can run properly.

This section uses the environment described in **Table 3-1** as an example.

Table 3-1 Test environment requirements

Resource Item	Name	Version
OS	• Ubuntu 18.04 • CentOS 7.6 • EulerOS 2.8	-

 Copyright © Huawei Technologies Co., Ltd.

Resource Item	Name	Version
Training script	ModelZoo_Resnet50_HC	-
OS architecture	ARM	-

Creating a Training Image

For details, see **Creating a Container Image Using a Dockerfile (TensorFlow)**. You can rename the training image, for example, **tf_arm64:b030**.

Preparing a Dataset

The **imagenet_TF** dataset is used only as an example.

Step 1 You need to prepare the dataset by yourself. The imagenet_TF dataset is recommended.

Step 2 Upload the dataset to the storage node as an administrator.

1. Go to the **/data/atlas_dls/public** directory and upload the **imagenet_TF** dataset to any directory, for example, **/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF**.

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF# pwd
/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF
```

2. Run the following command to check the dataset size:

   du -sh

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF# du -sh
144G
```

Step 3 Run the following command to change the owner of the dataset:

chown -R hwMindX:HwHiAiUser /data/atlas_dls/public

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF# chown -R
hwMindX:HwHiAiUser /data/atlas_dls/public
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF#
```

Step 4 Run the following command to change the dataset permission:

chmod -R 750 /data/atlas_dls/public

Step 5 Run the following command to check the file status:

ll /data/atlas_dls/public/*Dataset location*

Example:

ll /data/atlas_dls/public/*dataset/resnet50/resnet50/imagenet_TF*

```
root@ubuntu:~# ll /data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF
total 150649408
drwxr-x--- 2 hwMindX HwHiAiUser     53248 Sep 12 16:00 ./
drwxr-x--- 4 hwMindX HwHiAiUser      4096 Oct  7 16:52 ../
-rwxr-x--- 1 hwMindX HwHiAiUser 139619127 Sep 12 15:58 train-00000-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 141465049 Sep 12 16:00 train-00001-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 138414827 Sep 12 16:00 train-00002-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 135107647 Sep 12 15:58 train-00003-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 139356668 Sep 12 15:58 train-00004-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 140990868 Sep 12 15:58 train-00005-of-01024*
```


Copyright © Huawei Technologies Co., Ltd.

```
-rwxr-x--- 1 hwMindX HwHiAiUser 150652029 Sep 12 15:56 train-00006-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 136866315 Sep 12 16:00 train-00007-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 149972019 Sep 12 15:58 train-00008-of-01024*
...
```

----End

Obtaining and Modifying the Training Script

Step 1 Obtain the training script.

1. Log in to **ModelZoo**, download the ResNet-50 training code package of the TensorFlow framework, and decompress the package to the local host.
2. Find the following folders in the directory generated after the decompression and save them to the **resnet50_train** directory:

```
├── configs
├── data_loader
├── hyper_param
├── layers
├── losses
├── mains
├── models
├── optimizers
├── trainers
└── utils
```

3. Download **train_start.sh** and **main.sh** from **mindxdl-sample** (**download address**) and build the following directory structure on the host by referring to **Step 1.2**:

```
/data/atlas_dls/code/ModelZoo_Resnet50_HC/
├── code
│   └── resnet50_train
│       ├── configs
│       ├── data_loader
│       ├── hyper_param
│       ├── layers
│       ├── losses
│       ├── mains
│       ├── models
│       ├── optimizers
│       ├── trainers
│       └── utils
├── config(folder)
├── main.sh
└── train_start.sh
```

Step 2 Change the script permission and owner.

1. Upload the training script to the **/data/atlas_dls/code** directory on the storage node and decompress it.
2. Run the following command to assign the execute permission recursively:

   chmod -R 770 /data/atlas_dls/code

```
root@ubuntu:/data/atlas_dls/code# chmod -R 770 /data/atlas_dls/code/
root@ubuntu:/data/atlas_dls/code#
```

3. Run the following command to change the owner:

   chown -R hwMindX:HwHiAiUser /data/atlas_dls/code

```
root@ubuntu:/data/atlas_dls/code# chown -R hwMindX:HwHiAiUser /data/atlas_dls/code
root@ubuntu:/data/atlas_dls/code#
```

4. Run the following command to view the output result:

   ll /data/atlas_dls/code

```
root@ubuntu-infer:/data/atlas_dls/code# ll /data/atlas_dls/code
total 64
```

 Copyright © Huawei Technologies Co., Ltd.

```
drwxrwx--- 3 hwMindX HwHiAiUser  4096 Nov  3 15:50 ./
drwxrwx--- 5 hwMindX HwHiAiUser  4096 Nov  2 16:05 ../
drwxrwx--- 3 hwMindX HwHiAiUser  4096 Sep 24 18:55 ModelZoo_Resnet50_HC/
```

Step 3 Modify the permission on the script output directory.

1. Create the **/data/atlas_dls/output/model** directory for storing PB models on the storage node and run the following command to assign permissions:

   mkdir -p /data/atlas_dls/output/model

   chmod -R 770 /data/atlas_dls/output

2. Run the following command to change the owner:

   chown -R hwMindX:HwHiAiUser /data/atlas_dls/output

   ll /data/atlas_dls/output

   ```
   root@ubuntu-infer:/data/atlas_dls/output/# ll /data/atlas_dls/output
   total 12
   drwxrwx--- 3 hwMindX HwHiAiUser 4096 Nov  2 16:05 ./
   drwxrwx--- 3 hwMindX HwHiAiUser 4096 Nov  2 16:05 ../
   drwxrwx--- 2 hwMindX HwHiAiUser 4096 Nov  2 16:05 model/
   ```

Step 4 Modify the **main.sh** file in **ModelZoo_Resnet50_HC**.

1. Obtain the location of the **freeze_graph.py** file.

   a. Run the following command to access the container where the training image is located:

      docker run -ti *Image name_System architecture:Image tag* **/bin/bash**

      Example:

      docker run -ti *tf_arm64:b030* **/bin/bash**

   b. Run the following command to obtain the **naming.log** file:

      find /usr/local/ -name "freeze_graph.py"

      ```
      root@ubuntu-216-210:~# find /usr/local/ -name "freeze_graph.py"
      /usr/local/python3.7.5/lib/python3.7/site-packages/tensorflow_core/python/tools/freeze_graph.py
      ```

   c. Run the **exit** command to exit.

2. Run the following command in the directory of the **main.sh** file to modify the file:

   vim *{main.sh filepath}*

   Example:

   vim /data/atlas_dls/code/ModelZoo_Resnet50_HC/main.sh

   ```
   python3.7 /job/code/ModelZoo_Resnet50_HC/code/resnet50_train/mains/res50.py \
     --config_file=res50_256bs_1p \
     --max_train_steps=1000 \
     --iterations_per_loop=100 \
     --debug=True \  # Display precision.
     --eval=False \
     --model_dir=${model_dir} \
     | tee -a ${currentDir}/result/${log_id}/train_${device_id}.log 2>&1

   ...
   cd ${model_dir}
   python3.7 /usr/local/lib/python3.7/dist-packages/tensorflow_core/python/tools/freeze_graph.py \
       --input_checkpoint=${ckpt_name} \
       --output_graph=/job/output/model/resnet50_final.pb \
       --output_node_names=fp32_vars/final_dense \
       --input_graph=graph.pbtxt
   ...
   ```

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- In the example, **--config_file** indicates the configuration file of the training parameters.

  res50_256bs_1p indicates that the configuration file **res50_256bs_1p.py** is used.
- In the example, *${ckpt_name}* needs to be replaced with the value of **max_train_steps**.
  - If **max_train_steps=1000**, this parameter is **./model.ckpt-1000**.
  - If **max_train_steps=100**, this parameter is **./model.ckpt-100**.

3. Move the following content:

```
currentDir=$(cd "$(dirname "$0")"; pwd)
cd ${currentDir}
```

   To the position under **umask 007**. That is:

```
umask 007
currentDir=$(cd "$(dirname "$0")"; pwd)
cd ${currentDir}
```

4. Change **export RANK_TABLE_FILE=/user/serverid/devindex/config/hccl.json** to **export RANK_TABLE_FILE=/hccl_config/hccl.json**.
5. Change **DEVICE_INDEX=$((DEVICE_ID + RANK_INDEX * 8))** to **DEVICE_INDEX=${RANK_ID}**.
6. Modify the following content:

```
model_dir="/job/output/logs/ckpt${device_id}"
if [ "$first_card" = "true" ]; then
    model_dir="/job/output/logs/ckpt_first"
fi
```

   To:

```
pod_id=$6
model_dir="/job/output/logs/${pod_id}/ckpt${device_id}"
if [ "$first_card" = "true" ]; then
    model_dir="/job/output/logs/${pod_id}/ckpt_first"
fi
```

Step 5 Modify the following content in the **train_start.sh** file in the **ModelZoo_Resnet50_HC** directory.

1. Add **export NEW_RANK_INFO_FILE=/hccl_config/rank_id_info.txt** under **export RANK_TABLE_FILE=/user/serverid/devindex/config/hccl.json line**.
2. Change **rm -rf ${currentDir}/config/hccl.json** to **rm -rf ${currentDir}/config/* /hccl_config**.
3. Add the following information before **cp ${RANK_TABLE_FILE} ${currentDir}/config/hccl.json**:

```
mkdir -p /hccl_config
python3.7 ${currentDir}/trans_hccl_json_file.py
if [ ! $? -eq 0 ]
then
    exit 1
fi
chown -R HwHiAiUser:HwHiAiUser /hccl_config
cp ${NEW_RANK_INFO_FILE} ${currentDir}/config/rank_id_info.txt
```

4. Replace the following content:

```
mkdir -p /var/log/npu/slog/slogd
/usr/local/Ascend/driver/tools/docker/slogd &
/usr/local/Ascend/driver/tools/sklogd &
/usr/local/Ascend/driver/tools/log-daemon &
```

   With:

 Copyright © Huawei Technologies Co., Ltd.

```
# mkdir -p /var/log/npu/slog/slogd
# /usr/local/Ascend/driver/tools/docker/slogd &
# /usr/local/Ascend/driver/tools/sklogd &
# /usr/local/Ascend/driver/tools/log-daemon &
```

5. Locate the following content:

```
# Single-node training scenario
if [[ "$instance_count" == "1" ]]; then
    pod_name=$(get_json_value ${RANK_TABLE_FILE} pod_name)
    mkdir -p ${currentDir}/result/${train_start_time}
    chmod 770 -R ${currentDir}/result
    chgrp HwHiAiUser -R ${currentDir}/result

    for (( i=1;i<=$device_count;i++ ));do
    {
        dev_id=$(get_json_value ${RANK_TABLE_FILE} device_id ${i})
        device_count=$(get_json_value ${RANK_TABLE_FILE} device_count)

        first_card=false
        if [[ "$i" == "1" ]]; then
            first_card=true
        fi
        su - HwHiAiUser -c "bash ${currentDir}/main.sh ${dev_id} ${device_count} ${pod_name} $
{train_start_time} ${first_card}" &
    }
    done
# Distributed training scenario
else
    rank_index=`echo $HOSTNAME | awk -F"-" '{print $NF}'`
    device_count=8
    log_id=${train_start_time}${pod_name}
    mkdir -p ${currentDir}/result/${log_id}
    chmod 770 -R ${currentDir}/result
    chgrp HwHiAiUser -R ${currentDir}/result

    for (( i=1;i<=$device_count;i++ ));do
    {
        dev_id=$(get_json_value ${RANK_TABLE_FILE} device_id ${i})
        su - HwHiAiUser -c "bash ${currentDir}/main.sh ${dev_id} ${device_count} ${rank_index} $
{log_id}" &
    }
    done
fi
```

Replace it with the following content:

```
device_count=$(( $(cat ${NEW_RANK_INFO_FILE} | grep "single_pod_npu_count" | awk -F ':' '{print
$2}') ))
rank_size=$(cat ${NEW_RANK_INFO_FILE} | grep "rank_size" | awk -F ':' '{print $2}')
# IP address of the pod
rank_index=`echo $HOSTNAME | awk -F"-" '{print $NF}'`
pod_name="pod_${rank_index}"
#Information about all ranks in the current pod
pod_rank_info=$(cat ${NEW_RANK_INFO_FILE} | grep "${pod_name}")
pod_rank_info=${pod_rank_info#*:}

log_id=${train_start_time}${rank_index}

mkdir -p ${currentDir}/result/${log_id}
chmod 770 -R ${currentDir}/result
chgrp HwHiAiUser -R ${currentDir}/result

for (( i=0;i<$device_count;i++ ));do
{
    first_card=false
    if [[ "$i" == "0" ]]; then
        first_card=true
    fi
    rank_info=$(echo "${pod_rank_info}" | awk -F ':' '{print $1}')
    dev_id=$(echo ${rank_info} | awk -F ' ' '{print $1}')
    rank_id=$(echo ${rank_info} | awk -F ' ' '{print $2}')
```

 Copyright © Huawei Technologies Co., Ltd.

```
    su - HwHiAiUser -c "bash ${currentDir}/main.sh ${dev_id} ${rank_size} ${rank_id} ${log_id} $
{first_card} ${pod_name}" &
    pod_rank_info=${pod_rank_info#*:}
  }
  done
```

Step 6 Add the **trans_hccl_json_file.py** file to the same directory as the **main.sh** file and add the following content to the file:

```
import sys
import json


HCCL_JSON_FILE_PATH = "/user/serverid/devindex/config/hccl.json"
NEW_HCCL_JSON_FILE_PATH = "/hccl_config/hccl.json"
RANK_ID_INFO_FILE_PATH = "/hccl_config/rank_id_info.txt"

def read_old_hccl_json_content():
    hccl_json_str = ""
    try:
        with open(HCCL_JSON_FILE_PATH, "r") as f:
            hccl_json_str = f.read()
    except FileNotFoundError as e:
        print("File {} not exists !".format(HCCL_JSON_FILE_PATH))
        sys.exit(1)

    if not hccl_json_str:
        print("File {} is empty !".format(HCCL_JSON_FILE_PATH))
        sys.exit(1)

    try:
        hccl_json = json.loads(hccl_json_str)
    except TypeError as e:
        print("File {} content format is incorrect.".format(HCCL_JSON_FILE_PATH))
        sys.exit(1)

    return hccl_json


def create_new_hccl_content():
    hccl_json = read_old_hccl_json_content()
    group_list = hccl_json.get("group_list")[0]
    device_count = group_list.get("device_count")
    node_count = group_list.get("instance_count")
    instance_lists = group_list.get("instance_list")
    status = hccl_json.get("status")
    single_pod_npu_count = 0
    new_hccl_json_dict = {}
    rank_id_info_list = []

    new_hccl_json = {
        "status": status,
        "server_list": [],
        "server_count": node_count,
        "version": "1.0"
    }


    for instance_list in instance_lists:
        pod_id = int(instance_list.get("pod_name"))
        device_info_list = instance_list.get("devices")
        server_id = instance_list.get("server_id")

        server_info = {
            "server_id": server_id,
            "device": []
        }

        single_pod_npu_count = len(device_info_list)
        rankid_info_list = ["pod_{}".format(pod_id)]
```

 Copyright © Huawei Technologies Co., Ltd.

```
        device_info_list = sorted(device_info_list, key=lambda x: int(x.get("device_id")))
        index = 0

        for device_info in device_info_list:
            device_id = device_info.get("device_id")
            device_ip = device_info.get("device_ip")

            rank_id = single_pod_npu_count * pod_id + index

            new_device_info = {
                "device_id": device_id,
                "device_ip": device_ip,
                "rank_id": str(rank_id)
            }

            rank_info = device_id + " " + str(rank_id)
            rankid_info_list.append(rank_info)
            server_info.get("device").append(new_device_info)
            index += 1


        rankid_info_str = ":".join(rankid_info_list)
        rank_id_info_list.append(rankid_info_str)
        new_hccl_json_dict[pod_id] = server_info

    for pod_id in range(int(node_count)):
        server_info = new_hccl_json_dict.get(pod_id)
        new_hccl_json.get("server_list").append(server_info)

    rank_id_info_list.append("rank_size:{}".format(device_count))
    rank_id_info_list.append("single_pod_npu_count:{}".format(single_pod_npu_count))

    return new_hccl_json, rank_id_info_list


def write_new_hccl_to_file():
    new_hccl_json, rank_id_info_list = create_new_hccl_content()

    with open(NEW_HCCL_JSON_FILE_PATH, "w") as hccl_f:
        hccl_f.write(json.dumps(new_hccl_json))

    with open(RANK_ID_INFO_FILE_PATH, "w") as node_info_f:
        for rank_id_info in rank_id_info_list:
            node_info_f.write(rank_id_info)
            node_info_f.write("\n")


if __name__ == "__main__":
    write_new_hccl_to_file()
```

----End

3.4.1.1.2 Creating a YAML File

This section describes the YAML files in the single-node system and cluster scenarios. You can select proper YAML files based on the actual situation.

The YAML example is used in the NFS scenario. The NFS needs to be installed on the storage node. For details about how to install NFS, see **Installing the NFS**.

NOTE

If MindX DL is fully installed in online or offline mode, the NFS can be automatically installed.

 Copyright © Huawei Technologies Co., Ltd.

Single-Node Scenario

The following uses a single-server single-device training job as an example. Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *File name*.**yaml**

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

NOTICE

When using the following code, you need to delete the number signs (#) and comments, and modify the YAML file configurations based on the site requirements, such as used images, code paths, output paths, and output log paths.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob         # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   # The value cannot be modified. Service operations will be performed based on this label.
data:
  hccl.json: |
    {
        "status":"initializing"
    }
---
apiVersion: batch.volcano.sh/v1alpha1   # The value cannot be changed. The volcano API must be used.
kind: Job                          # Only the job type is supported at present.
metadata:
  name: mindx-dls-test               # The value must be consistent with the name of ConfigMap.
  namespace: vcjob     # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   # The value must be the same as the label in ConfigMap and cannot be changed.
spec:
  minAvailable: 1
  schedulerName: volcano             # Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 1                # For a single-node system, the value is 1 and the number of NPUs in the requests field is 1.
    template:
      metadata:
```

 Copyright © Huawei Technologies Co., Ltd.

```
      labels:
        app: tf
        ring-controller.atlas: ascend-910  # The value must be the same as the label in ConfigMap and cannot be changed.
    spec:
      containers:
      - image: tf_arm64:b030              # Training framework image, which can be modified.
        imagePullPolicy: IfNotPresent
        name: tf
        env:
        - name: RANK_TABLE_FILE
          value: "/user/serverid/devindex/config/hccl.json"  # Data mounting path in ConfigMap. If you need to change the value, ensure that it is consistent with the mounting path of ConfigMap.
        command:
        - "/bin/bash"
        - "-c"
        # Commands for running the training script. Ensure that the involved commands and paths exist on Docker.
        - "cd /job/code/ModelZoo_Resnet50_HC; bash train_start.sh"
        #args: [ "while true; do sleep 30000; done;" ]     # Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
        resources:
          requests:
            huawei.com/Ascend910: 1          # Number of required NPUs. The maximum value is 8. You can add lines below to configure resources such as memory and CPU.
          limits:
            huawei.com/Ascend910: 1          # The value must be consistent with that in requests..
        volumeMounts:
        - name: ascend-910-config
          mountPath: /user/serverid/devindex/config
        - name: code
          mountPath: /job/code/           # Path of the training script in the container.
        - name: data
          mountPath: /job/data            # Path of the training dataset in the container.
        - name: output
          mountPath: /job/output          # Training output path in the container.
        - name: slog
          mountPath: /var/log/npu
        - name: localtime
          mountPath: /etc/localtime
      nodeSelector:
        host-arch: huawei-arm     # Configure the label based on the actual job.
      volumes:
      - name: ascend-910-config
        configMap:
          name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
      - name: code
        nfs:
          server: 127.0.0.1                # IP address of the NFS server. In this example, the shared path is /data/atlas_dls/.
          path: "/data/atlas_dls/code/"     # Configure the training script path.
      - name: data
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/public/dataset/resnet50"    # Configure the path of the training set.
      - name: output
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/output/"    # Configure the path for saving the configuration model, which is related to the script.
      - name: slog
        hostPath:
          path: /var/log/npu        # Configure the NPU log path and mount it to the local host.
      - name: localtime
        hostPath:
          path: /etc/localtime   # Configure the Docker time.
      env:
      - name: mindx-dls-test  # The value must be consistent with the value of JobName.
        valueFrom:
```

 Copyright © Huawei Technologies Co., Ltd.

```
        fieldRef:
          fieldPath: metadata.name
    restartPolicy: OnFailure
```

Cluster Scenario

The following uses two training nodes running 2 x 8P distributed training jobs as an example. Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *File name*.**yaml**

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

NOTICE

When using the following code, you need to delete the number signs (#) and comments, and modify the YAML file configurations based on the site requirements, such as used images, code paths, output paths, and output log paths.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob         # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   # The value cannot be modified. Service operations will be performed based on this label.
data:
  hccl.json: |
    {
        "status":"initializing"
    }
---
apiVersion: batch.volcano.sh/v1alpha1   # The value cannot be changed. The volcano API must be used.
kind: Job                       # Only the job type is supported at present.
metadata:
  name: mindx-dls-test              # The value must be consistent with the name of ConfigMap.
  namespace: vcjob     # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The value must be the same as the label in ConfigMap and cannot be changed.
spec:
  minAvailable: 1
  schedulerName: volcano            #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
```

 Copyright © Huawei Technologies Co., Ltd.

```
 - name: "default-test"
   replicas: 2               # The value of replicas is N in an N-node scenario. The number of NPUs in the  field is 8 in an N-node scenario.
   template:
     metadata:
       labels:
         app: tf
         ring-controller.atlas: ascend-910  #The value must be the same as the label in ConfigMap and cannot be changed.
     spec:
       containers:
       - image: tf_arm64:b030              # Training framework image, which can be modified.
         imagePullPolicy: IfNotPresent
         name: tf
         env:
         - name: RANK_TABLE_FILE
           value: "/user/serverid/devindex/config/hccl.json"  # Data mounting path in ConfigMap. If you need to change the value, ensure that it is consistent with the mounting path of ConfigMap.
         command:
         - "/bin/bash"
         - "-c"
         #Commands for running the training script. Ensure that the involved commands and paths exist on Docker.
         - "cd /job/code/ModelZoo_Resnet50_HC; bash train_start.sh"
         #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
         resources:
           requests:
             huawei.com/Ascend910: 8          # Number of required NPUs. The maximum value is 8. You can add lines below to configure resources such as memory and CPU.
           limits:
             huawei.com/Ascend910: 8          # The value must be consistent with that in requests.
         volumeMounts:
         - name: ascend-910-config
           mountPath: /user/serverid/devindex/config
         - name: code
           mountPath: /job/code/           # Path of the training script in the container.
         - name: data
           mountPath: /job/data            # Path of the training dataset in the container.
         - name: output
           mountPath: /job/output          # Training output path in the container.
         - name: slog
           mountPath: /var/log/npu
         - name: localtime
           mountPath: /etc/localtime
       nodeSelector:
         host-arch: huawei-arm    # Configure the label based on the actual job.
       volumes:
       - name: ascend-910-config
         configMap:
          name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
       - name: code
         nfs:
           server: xxx.xxx.xxx.xxx                # IP address of the NFS server.
           path: "/data/atlas_dls/code/"     # Configure the training script path.
       - name: data
         nfs:
           server: xxx.xxx.xxx.xxx                # IP address of the NFS server.
           path: "/data/atlas_dls/public/dataset/resnet50"    # Configure the path of the training set.
       - name: output
         nfs:
           server: xxx.xxx.xxx.xxx                # IP address of the NFS server.
           path: "/data/atlas_dls/output/"    # Configure the path for saving the configuration model, which is related to the script.
       - name: slog
         hostPath:
           path: /var/log/npu        #Configure the NPU log path and mount it to the local host.
       - name: localtime
         hostPath:
```

 Copyright © Huawei Technologies Co., Ltd.

```
    path: /etc/localtime   # Configure the Docker time.
env:
- name: mindx-dls-test  # The value must be consistent with the value of JobName.
  valueFrom:
    fieldRef:
      fieldPath: metadata.name
restartPolicy: OnFailure
```

3.4.1.1.3 Preparing for Running a Training Job

Procedure

Step 1 Run the following command to modify the resources in the YAML file.

vim *XXX*.**yaml**

NOTE

XXX: YAML name generated in **Creating a YAML File**.

Example:

- Single-server single-device training job

  vim *Mindx-dl-test*.**yaml**

  Modify the items based on the resource requirements. For details about how to modify other items, see **Creating a YAML File**.

```
...
- name: "default-test"
   replicas: 1               # The value is 1 for a single node.
   template:
     metadata:
...
          resources:
            requests:
              huawei.com/Ascend910: 1        # Number of required NPUs. The maximum value is 8.You can add lines below to configure resources such as memory and CPU.
            limits:
              huawei.com/Ascend910: 1        # The value must be consistent with that in requests.
...
```

  NOTE

  - For a single-server single-device scenario, the value of **huawei.com/Ascend910** is **1**.
  - For a single-server multi-device scenario, the value of **huawei.com/Ascend910** is **2**, **4**, or **8**.

- Two training nodes running 2 x 8P distributed training jobs

  vim *Mindx-dl-test*.**yaml**

  Modify the items based on the resource requirements. For details about how to modify other items, see **Creating a YAML File**.

```
...
- name: "default-test"
   replicas: 2               # The value of replicas is N in an N-node scenario. The number of NPUs in the  field is 8 in an N-node scenario.replicas1requests
   template:
     metadata:
...
          resources:
            requests:
              huawei.com/Ascend910: 8        # Number of required NPUs. The maximum value is 8.
You can add lines below to configure resources such as memory and CPU.
```

 Copyright © Huawei Technologies Co., Ltd.

```
          limits:
            huawei.com/Ascend910: 8          # The value must be consistent with that in requests.
...
```

- If CPU and memory resources need to be configured, configure them as follows and set the values based on the site requirements:

```
...
- name: "default-test"
   replicas: 1
   template:
     metadata:
...
         resources:
           requests:
             huawei.com/Ascend910: 1
             cpu: 100m               # means 100 milliCPU.For example 100m CPU, 100 milliCPU, and 0.1 CPU are all the same
             memory: 100Gi           # means 100*2^30 bytes of memory
           limits:
             huawei.com/Ascend910: 1
             cpu: 100m
             memory: 100Gi
...
```

Step 2 Modify the training script.

1. Go to the **/data/atlas_dls/code/ModelZoo_Resnet50_HC/code/resnet50_train/configs** directory and modify the configuration file.

   In this example, the configuration file **res50_256bs_1p** is used. Therefore, modify the **res50_256bs_1p.py** file as follows (only one epoch is run):

   – Example of a single-server single-device training job

   ```
   ...
         'rank_size': 1,                      # total number of npus
         'shard': False,                      # set to False
   ...
         'mode':'train',                      # "train","evaluate","train_and_evaluate"
         'epochs_between_evals': 4,                  #used if mode is "train_and_evaluate"
         'stop_threshold': 80.0,                     #used if mode is "train_and_evaluate"
         'data_dir':'/opt/npu/resnet_data_new',
         'data_url': '/job/data/resnet50/imagenet_TF',    # dataset path
         'data_type': 'TFRECORD',
         'model_name': 'resnet50',
         'num_classes': 1001,
         'num_epochs': 1,
         'height':224,
         'width':224,
         'dtype': tf.float32,
         'data_format': 'channels_last',
         'use_nesterov': True,
         'eval_interval': 1,
         'loss_scale': 1024,
   ...
   #=======  logger config =======
         'display_every': 1,
         'log_name': 'resnet50.log',
         'log_dir': '/job/output/logs',    # Location of the resnet50.log file. The file content indicates the training precision.
   ...
   ```

   – Two training nodes running 2 x 8P distributed training jobs

   ```
   ...
         'rank_size': 16,                     # total number of npus
         'shard': True,                       # set to True
   ...
         'mode':'train',                      # "train","evaluate","train_and_evaluate"
         'epochs_between_evals': 4,                  #used if mode is "train_and_evaluate"
         'stop_threshold': 80.0,                     #used if mode is "train_and_evaluate"
         'data_dir':'/opt/npu/resnet_data_new',
   ```

 Copyright © Huawei Technologies Co., Ltd.

```
    'data_url': '/job/data/resnet50/imagenet_TF',    # dataset path
    'data_type': 'TFRECORD',
    'model_name': 'resnet50',
    'num_classes': 1001,
    'num_epochs': 1,
    'height':224,
    'width':224,
    'dtype': tf.float32,
    'data_format': 'channels_last',
    'use_nesterov': True,
    'eval_interval': 1,
    'loss_scale': 1024,
...
#=======  logger config =======
    'display_every': 1,
    'log_name': 'resnet50.log',
    'log_dir': '/job/output/logs',    # Location of the resnet50.log file. The file content indicates the training precision.
...
```

2. Modify the **main.sh** file in **ModelZoo_Resnet50_HC**. **--config_file** parameter is the **res50_256bs_1p.py** configuration file modified in the previous step.

```
python3.7 /job/code/ModelZoo_Resnet50_HC/code/resnet50_train/mains/res50.py \
  --config_file=res50_256bs_1p \
  --max_train_steps=1000 \
  --iterations_per_loop=100 \
  --debug=False \
  --eval=False \
  --model_dir=${model_dir} \
  | tee -a ${currentDir}/result/${log_id}/train_${device_id}.log 2>&1
```

3. Go to the **/data/atlas_dls/code/ModelZoo_Resnet50_HC/code/resnet50_train/mains** directory, find **resnet50.py**, and add **sys.path.append("/job/code/ModelZoo_Resnet50_HC/code/resnet50_train")** to the file.

```
......
print (path_2)
path_3 = base_path + "/../../"
print (path_3)


sys.path.append("/job/code/ModelZoo_Resnet50_HC/code/resnet50_train")
sys.path.append(base_path + "/../models")
sys.path.append(base_path + "/../../")
sys.path.append(base_path + "/../../models")

from utils import create_session as cs
from utils import logger as lg
......
```

----End

3.4.1.1.4 Delivering a Training Job

Procedure

Step 1 Run the following command to create a namespace for the training job:

kubectl create namespace vcjob

Step 2 Run the following command on the management node to deliver training jobs using YAML:

kubectl apply -f *XXX*.yaml

Example:

 Copyright © Huawei Technologies Co., Ltd.

kubectl apply -f *Mindx-dl-test*.yaml

```
root@ubuntu:/home/test/yaml# kubectl apply -f Mindx-dl-test.yaml
configmap/rings-config-mindx-dls-test created
job.batch.volcano.sh/mindx-dls-test created
```

----End

3.4.1.1.5 Checking the Running Status

Procedure

Step 1 Run the following command to check the pod running status:

kubectl get pod --all-namespaces -o wide

- Example of a single-server single-device training job

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                      1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                      1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                      1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n    1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v  1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk  1/1     Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                   1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                   1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      calico-node-fh58s                   1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      coredns-6955765f44-4pdhg            1/1     Running            0          8d
192.168.243.249  ubuntu        <none>           <none>
kube-system      coredns-6955765f44-n9pg4            1/1     Running            2          8d
192.168.243.237  ubuntu        <none>           <none>
kube-system      etcd-ubuntu                         1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-controller-manager-ubuntu      1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-b5flw                    1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      kube-proxy-ttsjp                    1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-zp9xw                    1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      kube-scheduler-ubuntu               1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-0       1/1     Running            0          4m
192.168.243.198  ubuntu        <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r   1/1     Running            2          8d
192.168.243.215  ubuntu        <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l    1/1     Running            1          8d
192.168.243.238  ubuntu        <none>           <none>
volcano-system   volcano-admission-init-bbx5z        0/1     Completed          0          39s
10.174.217.96    ubuntu-96     <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q  1/1     Running            2          8d
192.168.243.211  ubuntu        <none>           <none>
```

 Copyright © Huawei Technologies Co., Ltd.

- Two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                  READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                        1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                        1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                        1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n      1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx  1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9  1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v  1/1  Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk  1/1  Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                     1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                     1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      calico-node-fh58s                     1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      coredns-6955765f44-4pdhg              1/1     Running            0          8d
192.168.243.249  ubuntu        <none>           <none>
kube-system      coredns-6955765f44-n9pg4              1/1     Running            2          8d
192.168.243.237  ubuntu        <none>           <none>
kube-system      etcd-ubuntu                           1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-controller-manager-ubuntu        1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-b5flw                      1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      kube-proxy-ttsjp                      1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-zp9xw                      1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      kube-scheduler-ubuntu                 1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-0         1/1     Running            0          3m
192.168.243.198  ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-1         1/1     Running            0          3m
192.168.243.199  ubuntu        <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r     1/1     Running            2          8d
192.168.243.215  ubuntu        <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l  1/1     Running            1          8d
192.168.243.238  ubuntu        <none>           <none>
volcano-system   volcano-admission-init-bbx5z          0/1     Completed          0          39s
10.174.217.96    ubuntu-96     <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q    1/1     Running            2          8d
192.168.243.211  ubuntu        <none>           <none>
```

Step 2 View the NPU allocation of compute nodes.

Run the following command on the management node:

kubectl describe nodes

NOTE

- The **huawei.com/Ascend910** field of **Annotations** indicates the available NPU of the compute node.
- The **huawei.com/Ascend910** field in **Allocated resources** indicates the number of used NPUs.

 Copyright © Huawei Technologies Co., Ltd.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910:
Ascend910-0,Ascend910-1,Ascend910-2,Ascend910-4,Ascend910-5,Ascend910-6,Ascend910-7
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests        Limits
 --------            --------        ------
 cpu                 37250m (19%)    37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)          0 (0%)
 huawei.com/Ascend910  1               1
Events:             <none>
```

NOTE

The **Annotations** field does not contain the Ascend 910-3 AI Processor, and the value of the **Allocated resources** field **huawei.com/Ascend910** is **1**, indicating that one processor is used for training.

- One of the two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
```

 Copyright © Huawei Technologies Co., Ltd.

```
                workerselector=dls-worker-node
Annotations:    huawei.com/Ascend910:
                kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                node.alpha.kubernetes.io/ttl: 0
                projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests       Limits
 --------            --------       ------
 cpu                 37250m (19%)   37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)         0 (0%)
 huawei.com/Ascend910  8            8
Events:             <none>
```

NOTE

No NPU is available in the **Annotations** field, and the value of the **huawei.com/Ascend910** field in **Allocated resources** is **8**, all NPUs are used for distributed training.

Step 3 View the NPU usage of a pod.

In this example, run the **kubectl describe pod mindx-dls-test-default-test-0 -n vcjob** command to check the running status of the pod.

NOTE

Annotations displays the NPU information.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 15:38:22 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"3","device_ip":"192.168.20.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-3
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
```

 Copyright © Huawei Technologies Co., Ltd.

```
            volcano.sh/task-spec: default-test
Status:     Running
```

- Two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 15:38:22 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"0","device_ip":"192.168.20.100"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910:
Ascend910-0,Ascend910-1,Ascend910-2,Ascend910-3,Ascend910-4,Ascend910-5,Ascend910-6,Asce
nd910-7
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

----End

3.4.1.1.6 Viewing the Running Result

Step 1 Log in to the storage server.

The following uses the local NFS whose hostname is **Ubuntu** as an example.

Step 2 Run the following command to go to the output directory of the job running YAML file:

The **resnet50.log** file in the **/data/atlas_dls/output/logs/** directory records the training FPS value. In this example, the directory structure of a single-node training job is the same as that of a distributed training job.

```
root@ubuntu:/home# ll /data/atlas_dls/output/logs/
total 16896
drwxr-x--- 2 HwHiAiUser HwHiAiUser    4096 Oct  7 16:06 ./
drwxr-x--- 4 hwMindX    HwHiAiUser    4096 Oct  7 15:26 ../
...
-rwxr-x--- 1 HwHiAiUser HwHiAiUser     682 Oct  7 16:06 resnet50.log*
```

Step 3 View the information in **resnet50.log**.

cat /data/atlas_dls/output/logs/resnet50.log

If the FPS value is displayed in the command output, the training is successful.

```
PY3.7.5 (default, Dec 10 2020, 04:31:28)
[GCC 7.5.0]TF1.15.0
Step   Epoch   Speed   Loss   FinLoss   LR
step:   100  epoch:  0.0  FPS:   82.6  loss: 6.902  total_loss: 8.242  lr:0.00002
step:   200  epoch:  0.0  FPS: 1771.5  loss: 6.988  total_loss: 8.328  lr:0.00005
step:   300  epoch:  0.0  FPS: 1771.8  loss: 6.969  total_loss: 8.305  lr:0.00007
step:   400  epoch:  0.0  FPS: 1769.3  loss: 6.988  total_loss: 8.328  lr:0.00010
step:   500  epoch:  0.0  FPS:  691.6  loss: 6.895  total_loss: 8.234  lr:0.00012
step:   600  epoch:  0.0  FPS:  741.0  loss: 6.895  total_loss: 8.234  lr:0.00015
step:   700  epoch:  0.0  FPS:  563.2  loss: 6.922  total_loss: 8.258  lr:0.00017
step:   800  epoch:  0.0  FPS:  659.6  loss: 6.934  total_loss: 8.273  lr:0.00020
```

 Copyright © Huawei Technologies Co., Ltd.

```
step:  900  epoch:  0.0  FPS:  851.5  loss: 6.898  total_loss: 8.234  lr:0.00022
step: 1000  epoch:  0.0  FPS: 1524.7  loss: 6.949  total_loss: 8.289  lr:0.00025
```

Step 4 Go to the directory for storing the PB model and view the generated PB file.

ls -l /data/atlas_dls/output/model

```
root@ubuntu:/home# ls -l /data/atlas_dls/output/model/
total 99960
-rw-rw---- 1 HwHiAiUser HwHiAiUser 102356214 Mar  3 14:30 resnet50_final.pb
```

----End

3.4.1.1.7 Deleting a Training Job

Run the following command in the directory where YAML is running to delete a training job:

kubectl delete -f *XXX.yaml*

Example:

kubectl delete -f *Mindx-dl-test.yaml*

```
root@ubuntu:/home/test/yaml# kubectl delete -f Mindx-dl-test.yaml
configmap "rings-config-mindx-dls-test" deleted
job.batch.volcano.sh "mindx-dls-test" deleted
```

3.4.1.2 Server (with Atlas 300T Training Cards)

3.4.1.2.1 Preparing the NPU Training Environment

After MindX DL is installed, you can use YAML to deliver a vcjob to check whether the system can run properly.

This section uses the environment described in **Table 3-2** as an example.

Table 3-2 Test environment requirements

Item	Name	Version
OS	Ubuntu 18.04	-
Training script	ModelZoo_Resnet50_HC	-
OS architecture	x86	-

NOTE

Only the x86 environment of Ubuntu 18.04 is supported at present.

Creating a Training Image

Create a training image. For details, see **Creating a Container Image Using a Dockerfile (TensorFlow)**. You can rename the training image, for example, **tf_x86:b030**.

 Copyright © Huawei Technologies Co., Ltd.

Preparing a Dataset

The **imagenet_TF** dataset is used only as an example.

Step 1 You need to prepare the dataset by yourself. The imagenet_TF dataset is recommended.

Step 2 Upload the dataset to the storage node as an administrator.

1. Go to the **/data/atlas_dls/public** directory and upload the **imagenet_TF** dataset to any directory, for example, **/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF**.
```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF# pwd
/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF
```
2. Run the following command to check the dataset size:

   du -sh
```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF# du -sh
144G
```

Step 3 Run the following command to change the owner of the dataset:

chown -R hwMindX:HwHiAiUser /data/atlas_dls/public

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF# chown -R
hwMindX:HwHiAiUser /data/atlas_dls/public
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF#
```

Step 4 Run the following command to change the dataset permission:

chmod -R 750 /data/atlas_dls/public

Step 5 Run the following command to check the file status:

ll /data/atlas_dls/public/*Dataset location*

Example:

ll /data/atlas_dls/public/*dataset/resnet50/resnet50/imagenet_TF*

```
root@ubuntu:~# ll /data/atlas_dls/public/dataset/resnet50/resnet50/imagenet_TF
total 150649408
drwxr-x--- 2 hwMindX HwHiAiUser     53248 Sep 12 16:00 ./
drwxr-x--- 4 hwMindX HwHiAiUser      4096 Oct  7 16:52 ../
-rwxr-x--- 1 hwMindX HwHiAiUser 139619127 Sep 12 15:58 train-00000-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 141465049 Sep 12 16:00 train-00001-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 138414827 Sep 12 16:00 train-00002-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 135107647 Sep 12 15:58 train-00003-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 139356668 Sep 12 15:58 train-00004-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 140990868 Sep 12 15:58 train-00005-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 150652029 Sep 12 15:56 train-00006-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 136866315 Sep 12 16:00 train-00007-of-01024*
-rwxr-x--- 1 hwMindX HwHiAiUser 149972019 Sep 12 15:58 train-00008-of-01024*
...
```

----End

Obtaining and Modifying the Training Script

Step 1 Obtain the training script.

1. Log in to **ModelZoo**, download the ResNet-50 training code package of the TensorFlow framework, and decompress the package to the local host.

 Copyright © Huawei Technologies Co., Ltd.

2. Find the following folders in the directory generated after the decompression and save them to the **resnet50_train** directory:

```
├── configs
├── data_loader
├── hyper_param
├── layers
├── losses
├── mains
├── models
├── optimizers
├── trainers
└── utils
```

3. Download **train_start.sh** and **main.sh** from **mindxdl-sample** (**download address**) and build the following directory structure on the host by referring to **Step 1.2**:

```
/data/atlas_dls/code/ModelZoo_Resnet50_HC/
├── code
│   └── resnet50_train
│       ├── configs
│       ├── data_loader
│       ├── hyper_param
│       ├── layers
│       ├── losses
│       ├── mains
│       ├── models
│       ├── optimizers
│       ├── trainers
│       └── utils
├── config(folder)
├── main.sh
└── train_start.sh
```

Step 2 Change the script permission and owner.

1. Upload the training script to the **/data/atlas_dls/code** directory on the storage node and decompress it.
2. Run the following command to assign the execute permission recursively:

   chmod -R 770 /data/atlas_dls/code

```
root@ubuntu:/data/atlas_dls/code# chmod -R 770 /data/atlas_dls/code/
root@ubuntu:/data/atlas_dls/code#
```

3. Run the following command to change the owner:

   chown -R hwMindX:HwHiAiUser /data/atlas_dls/code

```
root@ubuntu:/data/atlas_dls/code# chown -R hwMindX:HwHiAiUser /data/atlas_dls/code
root@ubuntu:/data/atlas_dls/code#
```

4. Run the following command to view the output result:

   ll /data/atlas_dls/code

```
root@ubuntu-infer:/data/atlas_dls/code# ll /data/atlas_dls/code
total 64
drwxrwx--- 3 hwMindX HwHiAiUser  4096 Nov  3 15:50 ./
drwxrwx--- 5 hwMindX HwHiAiUser  4096 Nov  2 16:05 ../
drwxrwx--- 3 hwMindX HwHiAiUser  4096 Sep 24 18:55 ModelZoo_Resnet50_HC/
```

Step 3 Modify the permission on the script output directory.

1. Create the **/data/atlas_dls/output/model** directory for storing PB models on the storage node and run the following command to assign permissions:

   mkdir -p /data/atlas_dls/output/model

   chmod -R 770 /data/atlas_dls/output

2. Run the following command to change the owner:

 Copyright © Huawei Technologies Co., Ltd.

chown -R hwMindX:HwHiAiUser /data/atlas_dls/output

ll /data/atlas_dls/output

```
root@ubuntu-infer:/data/atlas_dls/output/# ll /data/atlas_dls/output
total 12
drwxrwx--- 3 hwMindX HwHiAiUser 4096 Nov  2 16:05 ./
drwxrwx--- 3 hwMindX HwHiAiUser 4096 Nov  2 16:05 ../
drwxrwx--- 2 hwMindX HwHiAiUser 4096 Nov  2 16:05 model/
```

Step 4 Modify the **main.sh** file in **ModelZoo_Resnet50_HC**.

1. Obtain the location of the **freeze_graph.py** file.
   a. Run the following command to access the container where the training image is located:

      docker run -ti *Image name_System architecture:Image tag* **/bin/bash**

      Example:

      docker run -ti *tf_x86:b030* **/bin/bash**

   b. Run the following command to obtain the **naming.log** file:

      find /usr/local/ -name "freeze_graph.py"

      ```
      root@ubuntu-216-210:~# find /usr/local/ -name "freeze_graph.py"
      /usr/local/python3.7.5/lib/python3.7/site-packages/tensorflow_core/python/tools/freeze_graph.py
      ```

   c. Run the **exit** command to exit.
2. Run the following command in the directory of the **main.sh** file to modify the file:

   vim *{main.sh filepath}*

   Example:

   vim /data/atlas_dls/code/ModelZoo_Resnet50_HC/main.sh

   ```
   python3.7 /job/code/ModelZoo_Resnet50_HC/code/resnet50_train/mains/res50.py \
     --config_file=res50_256bs_1p \
     --max_train_steps=1000 \
     --iterations_per_loop=100 \
     --debug=True \  # Display precision.
     --eval=False \
     --model_dir=${model_dir} \
     | tee -a ${currentDir}/result/${log_id}/train_${device_id}.log 2>&1


   ...
   cd ${model_dir}
   python3.7 /usr/local/lib/python3.7/dist-packages/tensorflow_core/python/tools/freeze_graph.py \
      --input_checkpoint=${ckpt_name} \
      --output_graph=/job/output/model/resnet50_final.pb \
      --output_node_names=fp32_vars/final_dense \
      --input_graph=graph.pbtxt
   ...
   ```

   NOTE

   - In the example, **--config_file** indicates the configuration file of the training parameters.

     res50_256bs_1p indicates that the configuration file **res50_256bs_1p.py** is used.
   - In the example, *${ckpt_name}* needs to be replaced with the value of **max_train_steps**.
     - If **max_train_steps=1000**, this parameter is **./model.ckpt-1000**.
     - If **max_train_steps=100**, this parameter is **./model.ckpt-100**.

 Copyright © Huawei Technologies Co., Ltd.

3. Move the following content:
```
currentDir=$(cd "$(dirname "$0")"; pwd)
cd ${currentDir}
```
To the position under **umask 007**. That is:
```
umask 007
currentDir=$(cd "$(dirname "$0")"; pwd)
cd ${currentDir}
```
4. Change **export RANK_TABLE_FILE=/user/serverid/devindex/config/hccl.json** to **export RANK_TABLE_FILE=/hccl_config/hccl.json**.
5. Change **DEVICE_INDEX=$((DEVICE_ID + RANK_INDEX * 8))** to **DEVICE_INDEX=${RANK_ID}**.
6. Modify the following content:
```
model_dir="/job/output/logs/ckpt${device_id}"
if [ "$first_card" = "true" ]; then
    model_dir="/job/output/logs/ckpt_first"
fi
```
To:
```
pod_id=$6
model_dir="/job/output/logs/${pod_id}/ckpt${device_id}"
if [ "$first_card" = "true" ]; then
    model_dir="/job/output/logs/${pod_id}/ckpt_first"
fi
```

Step 5 Modify the following content in the **train_start.sh** file in the **ModelZoo_Resnet50_HC** directory.

1. Add **export NEW_RANK_INFO_FILE=/hccl_config/rank_id_info.txt** under **export RANK_TABLE_FILE=/user/serverid/devindex/config/hccl.json line**.
2. Change **rm -rf ${currentDir}/config/hccl.json** to **rm -rf ${currentDir}/config/* /hccl_config**.
3. Add the following information before **cp ${RANK_TABLE_FILE} ${currentDir}/config/hccl.json**:
```
mkdir -p /hccl_config
python3.7 ${currentDir}/trans_hccl_json_file.py
if [ ! $? -eq 0 ]
then
    exit 1
fi
chown -R HwHiAiUser:HwHiAiUser /hccl_config
cp ${NEW_RANK_INFO_FILE} ${currentDir}/config/rank_id_info.txt
```
4. Replace the following content:
```
mkdir -p /var/log/npu/slog/slogd
/usr/local/Ascend/driver/tools/docker/slogd &
/usr/local/Ascend/driver/tools/sklogd &
/usr/local/Ascend/driver/tools/log-daemon &
```
With:
```
# mkdir -p /var/log/npu/slog/slogd
# /usr/local/Ascend/driver/tools/docker/slogd &
# /usr/local/Ascend/driver/tools/sklogd &
# /usr/local/Ascend/driver/tools/log-daemon &
```
5. Locate the following content:
```
# Single-node training scenario
if [[ "$instance_count" == "1" ]]; then
    pod_name=$(get_json_value ${RANK_TABLE_FILE} pod_name)
    mkdir -p ${currentDir}/result/${train_start_time}
    chmod 770 -R ${currentDir}/result
    chgrp HwHiAiUser -R ${currentDir}/result

    for (( i=1;i<=$device_count;i++ ));do
```

 Copyright © Huawei Technologies Co., Ltd.

```
    {
        dev_id=$(get_json_value ${RANK_TABLE_FILE} device_id ${i})
        device_count=$(get_json_value ${RANK_TABLE_FILE} device_count)

        first_card=false
        if [[ "$i" == "1" ]]; then
            first_card=true
        fi
        su - HwHiAiUser -c "bash ${currentDir}/main.sh ${dev_id} ${device_count} ${pod_name} $
{train_start_time} ${first_card}" &
    }
    done
# Distributed training scenario
else
    rank_index=`echo $HOSTNAME | awk -F"-" '{print $NF}'`
    device_count=8
    log_id=${train_start_time}${pod_name}
    mkdir -p ${currentDir}/result/${log_id}
    chmod 770 -R ${currentDir}/result
    chgrp HwHiAiUser -R ${currentDir}/result

    for (( i=1;i<=$device_count;i++ ));do
    {
        dev_id=$(get_json_value ${RANK_TABLE_FILE} device_id ${i})
        su - HwHiAiUser -c "bash ${currentDir}/main.sh ${dev_id} ${device_count} ${rank_index} $
{log_id}" &
    }
    done
fi
```

Replace it with the following content:

```
device_count=$(( $(cat ${NEW_RANK_INFO_FILE} | grep "single_pod_npu_count" | awk -F ':' '{print
$2}') ))
rank_size=$(cat ${NEW_RANK_INFO_FILE} | grep "rank_size" | awk -F ':' '{print $2}')
# IP address of the pod
rank_index=`echo $HOSTNAME | awk -F"-" '{print $NF}'`
pod_name="pod_${rank_index}"
#Information about all ranks in the current pod
pod_rank_info=$(cat ${NEW_RANK_INFO_FILE} | grep "${pod_name}")
pod_rank_info=${pod_rank_info#*:}

log_id=${train_start_time}${rank_index}

mkdir -p ${currentDir}/result/${log_id}
chmod 770 -R ${currentDir}/result
chgrp HwHiAiUser -R ${currentDir}/result

for (( i=0;i<$device_count;i++ ));do
{
    first_card=false
    if [[ "$i" == "0" ]]; then
        first_card=true
    fi
    rank_info=$(echo "${pod_rank_info}" | awk -F ':' '{print $1}')
    dev_id=$(echo ${rank_info} | awk -F ' ' '{print $1}')
    rank_id=$(echo ${rank_info} | awk -F ' ' '{print $2}')
    su - HwHiAiUser -c "bash ${currentDir}/main.sh ${dev_id} ${rank_size} ${rank_id} ${log_id} $
{first_card} ${pod_name}" &
    pod_rank_info=${pod_rank_info#*:}
}
done
```

Step 6 Add the **trans_hccl_json_file.py** file to the same directory as the **main.sh** file and add the following content to the file:

```
import sys
import json


HCCL_JSON_FILE_PATH = "/user/serverid/devindex/config/hccl.json"
```

 Copyright © Huawei Technologies Co., Ltd.

```
NEW_HCCL_JSON_FILE_PATH = "/hccl_config/hccl.json"
RANK_ID_INFO_FILE_PATH = "/hccl_config/rank_id_info.txt"

def read_old_hccl_json_content():
    hccl_json_str = ""
    try:
        with open(HCCL_JSON_FILE_PATH, "r") as f:
            hccl_json_str = f.read()
    except FileNotFoundError as e:
        print("File {} not exists !".format(HCCL_JSON_FILE_PATH))
        sys.exit(1)

    if not hccl_json_str:
        print("File {} is empty !".format(HCCL_JSON_FILE_PATH))
        sys.exit(1)

    try:
        hccl_json = json.loads(hccl_json_str)
    except TypeError as e:
        print("File {} content format is incorrect.".format(HCCL_JSON_FILE_PATH))
        sys.exit(1)

    return hccl_json


def create_new_hccl_content():
    hccl_json = read_old_hccl_json_content()
    group_list = hccl_json.get("group_list")[0]
    device_count = group_list.get("device_count")
    node_count = group_list.get("instance_count")
    instance_lists = group_list.get("instance_list")
    status = hccl_json.get("status")
    single_pod_npu_count = 0
    new_hccl_json_dict = {}
    rank_id_info_list = []

    new_hccl_json = {
        "status": status,
        "server_list": [],
        "server_count": node_count,
        "version": "1.0"
    }


    for instance_list in instance_lists:
        pod_id = int(instance_list.get("pod_name"))
        device_info_list = instance_list.get("devices")
        server_id = instance_list.get("server_id")

        server_info = {
            "server_id": server_id,
            "device": []
        }

        single_pod_npu_count = len(device_info_list)
        rankid_info_list = ["pod_{}".format(pod_id)]
        device_info_list = sorted(device_info_list, key=lambda x: int(x.get("device_id")))
        index = 0

        for device_info in device_info_list:
            device_id = device_info.get("device_id")
            device_ip = device_info.get("device_ip")

            rank_id = single_pod_npu_count * pod_id + index

            new_device_info = {
                "device_id": device_id,
                "device_ip": device_ip,
                "rank_id": str(rank_id)
```

 Copyright © Huawei Technologies Co., Ltd.

```
            }

            rank_info = device_id + " " + str(rank_id)
            rankid_info_list.append(rank_info)
            server_info.get("device").append(new_device_info)
            index += 1


        rankid_info_str = ":".join(rankid_info_list)
        rank_id_info_list.append(rankid_info_str)
        new_hccl_json_dict[pod_id] = server_info

    for pod_id in range(int(node_count)):
        server_info = new_hccl_json_dict.get(pod_id)
        new_hccl_json.get("server_list").append(server_info)

    rank_id_info_list.append("rank_size:{}".format(device_count))
    rank_id_info_list.append("single_pod_npu_count:{}".format(single_pod_npu_count))

    return new_hccl_json, rank_id_info_list


def write_new_hccl_to_file():
    new_hccl_json, rank_id_info_list = create_new_hccl_content()

    with open(NEW_HCCL_JSON_FILE_PATH, "w") as hccl_f:
        hccl_f.write(json.dumps(new_hccl_json))

    with open(RANK_ID_INFO_FILE_PATH, "w") as node_info_f:
        for rank_id_info in rank_id_info_list:
            node_info_f.write(rank_id_info)
            node_info_f.write("\n")


if __name__ == "__main__":
    write_new_hccl_to_file()
```

----End

3.4.1.2.2 Creating a YAML File

This section describes the YAML files in the single-node system and cluster scenarios. You can select proper YAML files based on the actual situation.

The YAML example is used in the NFS scenario. The NFS needs to be installed on the storage node. For details about how to install NFS, see **Installing the NFS**.

NOTE

If MindX DL is fully installed in online or offline mode, the NFS can be automatically installed.

Single-Node Scenario

The following uses a single-server single-device training job as an example. Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *XXX*.**yaml**

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

 Copyright © Huawei Technologies Co., Ltd.

NOTICE

When using the following code, you need to delete the number signs (#) and comments, and modify the YAML file configurations based on the site requirements, such as used images, code paths, output paths, and output log paths.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob       # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   # The value cannot be modified. Service operations will be performed based on this label.
data:
  hccl.json: |
    {
        "status":"initializing"
    }
---
apiVersion: batch.volcano.sh/v1alpha1   # The value cannot be changed. The volcano API must be used.
kind: Job                          #Only the job type is supported at present.
metadata:
  name: mindx-dls-test               # The value must be consistent with the name of ConfigMap.
  namespace: vcjob     # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The value must be the same as the label in ConfigMap and cannot be changed.
spec:
  minAvailable: 1
  schedulerName: volcano             #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 1                # For a single-node system, the value is 1 and the maximum number of NPUs in the requests field is 2.
    template:
      metadata:
        labels:
          app: tf
          ring-controller.atlas: ascend-910  #The value must be the same as the label in ConfigMap and cannot be changed.
      spec:
        containers:
        - image: tf_x86:b030             # Training framework image, which can be modified.
          imagePullPolicy: IfNotPresent
          name: tf
          env:
          - name: RANK_TABLE_FILE
            value: "/user/serverid/devindex/config/hccl.json"  # Data mounting path in ConfigMap. If you need to change the value, ensure that it is consistent with the mounting path of ConfigMap.
          command:
          - "/bin/bash"
```

 Copyright © Huawei Technologies Co., Ltd.

```
        - "-c"
        #Commands for running the training script. Ensure that the involved commands and paths exist on
Docker.
        - "cd /job/code/ModelZoo_Resnet50_HC; bash train_start.sh"
        #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this
line. You can manually run the training script in the container to facilitate debugging.
        resources:
          requests:
            huawei.com/Ascend910: 1          # Number of required NPUs. The maximum value is 2. You
can add lines below to configure resources such as memory and CPU.
          limits:
            huawei.com/Ascend910: 1          #The value must be consistent with that in requests.
        volumeMounts:
        - name: ascend-910-config
          mountPath: /user/serverid/devindex/config
        - name: code
          mountPath: /job/code/          # Path of the training script in the container.
        - name: data
          mountPath: /job/data/resnet50/imagenet_TF          # Path of the training dataset in the container.
        - name: output
          mountPath: /job/output          # Training output path in the container.
        - name: slog
          mountPath: /var/log/npu
        - name: localtime
          mountPath: /etc/localtime
      nodeSelector:
        host-arch: huawei-x86    # Configure the label based on the actual job.
        accelerator-type: card          #servers (with Atlas 300T training cards)
      volumes:
      - name: ascend-910-config
        configMap:
          name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
      - name: code
        nfs:
          server: 127.0.0.1                #IP address of the NFS server. In this example, the shared path is /data/
atlas_dls/.
          path: "/data/atlas_dls/code/"     # Configure the training script path.
      - name: data
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/public/dataset/resnet50/imagenet_TF"    # Configure the path of the
training set.
      - name: output
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/output/"    # Configure the path for saving the configuration model,
which is related to the script.
      - name: slog
        hostPath:
          path: /var/log/npu        #Configure the NPU log path and mount it to the local host.
      - name: localtime
        hostPath:
          path: /etc/localtime   # Configure the Docker time.
      env:
      - name: mindx-dls-test  # The value must be consistent with the value of JobName.
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      restartPolicy: OnFailure
```

Cluster Scenario

The following uses two training nodes running 2 x 2P distributed training jobs as an example. Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *XXX*.yaml

 Copyright © Huawei Technologies Co., Ltd.

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

NOTICE

When using the following code, you need to delete the number signs (#) and comments, and modify the YAML file configurations based on the site requirements, such as used images, code paths, output paths, and output log paths.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob       # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   # The value cannot be modified. Service operations will be performed based on this label.
data:
  hccl.json: |
    {
        "status":"initializing"
    }
---
apiVersion: batch.volcano.sh/v1alpha1   # The value cannot be changed. The volcano API must be used.
kind: Job                          #Only the job type is supported at present.
metadata:
  name: mindx-dls-test               # The value must be consistent with the name of ConfigMap.
  namespace: vcjob     # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The value must be the same as the label in ConfigMap and cannot be changed.
spec:
  minAvailable: 1
  schedulerName: volcano             #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 2                # In distributed mode, the value is greater than 1 and the maximum number of NPUs in the requests field is 2.
    template:
      metadata:
        labels:
          app: tf
          ring-controller.atlas: ascend-910  #The value of replicas is N in an N-node scenario. The number of NPUs in the requests field is 2 in an N-node scenario.
      spec:
        containers:
        - image: tf_x86:b030             # Training framework image, which can be modified.
          imagePullPolicy: IfNotPresent
          name: tf
```

 Copyright © Huawei Technologies Co., Ltd.

```
        env:
        - name: RANK_TABLE_FILE
          value: "/user/serverid/devindex/config/hccl.json"  # Data mounting path in ConfigMap. If you need to change the value, ensure that it is consistent with the mounting path of ConfigMap.
        command:
        - "/bin/bash"
        - "-c"
        #Commands for running the training script. Ensure that the involved commands and paths exist on Docker.
        - "cd /job/code/ModelZoo_Resnet50_HC; bash train_start.sh"
        #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
        resources:
          requests:
            huawei.com/Ascend910: 2          # Number of required NPUs. The maximum value is 2. You can add lines below to configure resources such as memory and CPU.
          limits:
            huawei.com/Ascend910: 2          # The value must be consistent with that in requests.
        volumeMounts:
        - name: ascend-910-config
          mountPath: /user/serverid/devindex/config
        - name: code
          mountPath: /job/code/           # Path of the training script in the container.
        - name: data
          mountPath: /job/data/resnet50/imagenet_TF            # Path of the training dataset in the container.
        - name: output
          mountPath: /job/output          # Training output path in the container.
        - name: slog
          mountPath: /var/log/npu
        - name: localtime
          mountPath: /etc/localtime
      nodeSelector:
        host-arch: huawei-x86    # Configure the label based on the actual job.
        accelerator-type: card            #servers (with Atlas 300T training cards)
      volumes:
      - name: ascend-910-config
        configMap:
          name: rings-config-mindx-dls-test   #Corresponds to the name of ConfigMap in the preceding information.
      - name: code
        nfs:
          server: xxx.xxx.xxx.xxx                # IP address of the NFS server.
          path: "/data/atlas_dls/code/"     #Configure the training script path.
      - name: data
        nfs:
          server: xxx.xxx.xxx.xxx                # IP address of the NFS server.
          path: "/data/atlas_dls/public/dataset/resnet50/imagenet_TF"    # Configure the training set path.
      - name: output
        nfs:
          server: xxx.xxx.xxx.xxx                # IP address of the NFS server.
          path: "/data/atlas_dls/output/"    # Configure the path for saving the configuration model, which is related to the script.
      - name: slog
        hostPath:
          path: /var/log/npu        #Configure the NPU log path and mount it to the local host.
      - name: localtime
        hostPath:
          path: /etc/localtime   # Configure the Docker time.
      env:
      - name: mindx-dls-test  # The value must be consistent with the value of JobName.
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      restartPolicy: OnFailure
```

 Copyright © Huawei Technologies Co., Ltd.

3.4.1.2.3 Preparing for Running a Training Job

Procedure

Step 1 Run the following command to modify the resources in the YAML file.

vim *XXX*.**yaml**

NOTE

XXX: YAML name generated in **Creating a YAML File**.

Example:

- Single-server single-device training job

  vim *Mindx-dl-test*.**yaml**

  Modify the items based on the resource requirements. For details about how to modify other items, see **Creating a YAML File**.

```
...
 - name: "default-test"
   replicas: 1               # The value of replicas is 1 in a single-node scenario and N in an N-node scenario. The number of NPUs in the requests field is 2 in an N-node scenario.
   template:
     metadata:
...
         resources:
           requests:
             huawei.com/Ascend910: 1        # Number of required NPUs. The maximum value is 2. You can add lines below to configure resources such as memory and CPU.
           limits:
             huawei.com/Ascend910: 1        # The value must be consistent with that in requests.
...
```

  NOTE

  - For a single-server single-device scenario, the value of **huawei.com/Ascend910** is **1**.
  - For a single-server multi-device scenario, the value of **huawei.com/Ascend910** is **2**.

- Two training nodes running 2 x 2P distributed training jobs

  vim *Mindx-dl-test*.**yaml**

  Modify the items based on the resource requirements. For details about how to modify other items, see **Creating a YAML File**.

```
...
 - name: "default-test"
   replicas: 2               # In distributed mode, the value is greater than 1 and the maximum number of NPUs in the requests field is 2.
   template:
     metadata:
...
         resources:
           requests:
             huawei.com/Ascend910: 2        # Number of required NPUs. The maximum value is 2. You can add lines below to configure resources such as memory and CPU.
           limits:
             huawei.com/Ascend910: 2        # The value must be consistent with that in requests.
...
```

- If CPU and memory resources need to be configured, configure them as follows and set the values based on the site requirements:

 Copyright © Huawei Technologies Co., Ltd.

```
...
- name: "default-test"
   replicas: 1
   template:
     metadata:
...
         resources:
           requests:
             huawei.com/Ascend910: 1
             cpu: 100m              # means 100 milliCPU.For example 100m CPU, 100 milliCPU, and 0.1
CPU are all the same
             memory: 100Gi          # means 100*2^30 bytes of memory
           limits:
             huawei.com/Ascend910: 1
             cpu: 100m
             memory: 100Gi
...
```

Step 2 Modify the training script.

1. Go to the **/data/atlas_dls/code/ModelZoo_Resnet50_HC/code/resnet50_train/configs** directory and modify the processor configuration file.

   In this example, the **res50_256bs_1p.py** file is modified as follows: (Only one epochs is run.)

   - Example of a single-server single-device training job

     ```
     ...
           'rank_size': 1,                    # total number of npus
           'shard': False,                    # set to False
     ...
           'mode':'train',                    # "train","evaluate","train_and_evaluate"
           'epochs_between_evals': 4,                 #used if mode is "train_and_evaluate"
           'stop_threshold': 80.0,                  #used if mode is "train_and_evaluate"
           'data_dir':'/opt/npu/resnet_data_new',
           'data_url': '/job/data/resnet50/imagenet_TF',    # dataset path
           'data_type': 'TFRECORD',
           'model_name': 'resnet50',
           'num_classes': 1001,
           'num_epochs': 1,
           'height':224,
           'width':224,
           'dtype': tf.float32,
           'data_format': 'channels_last',
           'use_nesterov': True,
           'eval_interval': 1,
           'loss_scale': 1024,
     ...
     #=======  logger config =======
           'display_every': 1,
           'log_name': 'resnet50.log',
           'log_dir': '/job/output/logs',    # Location of the resnet50.log file. The file content indicates
     the training precision.
     ...
     ```

   - Two training nodes running 2 x 2P distributed training jobs

     ```
     ...
           'rank_size': 4,                    # total number of npus
           'shard': True,                     # set to True
     ...
           'mode':'train',                    # "train","evaluate","train_and_evaluate"
           'epochs_between_evals': 4,                 #used if mode is "train_and_evaluate"
           'stop_threshold': 80.0,                  #used if mode is "train_and_evaluate"
           'data_dir':'/opt/npu/resnet_data_new',
           'data_url': '/job/data/resnet50/imagenet_TF',    # dataset path
           'data_type': 'TFRECORD',
           'model_name': 'resnet50',
           'num_classes': 1001,
           'num_epochs': 1,
           'height':224,
     ```

 Copyright © Huawei Technologies Co., Ltd.

```
        'width':224,
        'dtype': tf.float32,
        'data_format': 'channels_last',
        'use_nesterov': True,
        'eval_interval': 1,
        'loss_scale': 1024,
...
#=======  logger config =======
        'display_every': 1,
        'log_name': 'resnet50.log',
        'log_dir': '/job/output/logs',    # Location of the resnet50.log file. The file content indicates the training precision.
...
```

2. Modify the **main.sh** file in **ModelZoo_Resnet50_HC**. **--config_file** parameter is the **res50_256bs_1p.py** configuration file modified in the previous step.

```
python3.7 /job/code/ModelZoo_Resnet50_HC/code/resnet50_train/mains/res50.py \
  --config_file=res50_256bs_1p \
  --max_train_steps=1000 \
  --iterations_per_loop=100 \
  --debug=True \  # Display precision.
  --eval=False \
  --model_dir=${model_dir} \
  | tee -a ${currentDir}/result/${log_id}/train_${device_id}.log 2>&1
```

3. Go to the **/data/atlas_dls/code/ModelZoo_Resnet50_HC/code/resnet50_train/mains** directory, find **resnet50.py**, and add **sys.path.append("/job/code/ModelZoo_Resnet50_HC/code/resnet50_train")** to the file.

```
......
print (path_2)
path_3 = base_path + "/../../"
print (path_3)



sys.path.append("/job/code/ModelZoo_Resnet50_HC/code/resnet50_train")
sys.path.append(base_path + "/../models")
sys.path.append(base_path + "/../../")
sys.path.append(base_path + "/../../models")

from utils import create_session as cs
from utils import logger as lg
......
```

----End

3.4.1.2.4 Delivering a Training Job

Procedure

Step 1 Run the following command to create a namespace for the training job:

kubectl create namespace vcjob

Step 2 Run the following command on the management node to deliver training jobs using YAML:

kubectl apply -f *XXX*.**yaml**

Example:

kubectl apply -f *Mindx-dl-test*.**yaml**

 Copyright © Huawei Technologies Co., Ltd.

```
root@ubuntu:/home/test/yaml# kubectl apply -f Mindx-dl-test.yaml
configmap/rings-config-mindx-dls-test created
job.batch.volcano.sh/mindx-dls-test created
```

----End

3.4.1.2.5 Checking the Running Status

Procedure

Step 1 Run the following command to check the pod running status:

kubectl get pod --all-namespaces -o wide

- Example of a single-server single-device training job

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                  READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                        1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                        1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                        1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n      1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx  1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9  1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v  1/1  Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk  1/1  Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                     1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                     1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      calico-node-fh58s                     1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      coredns-6955765f44-4pdhg              1/1     Running            0          8d
192.168.243.249  ubuntu        <none>           <none>
kube-system      coredns-6955765f44-n9pg4              1/1     Running            2          8d
192.168.243.237  ubuntu        <none>           <none>
kube-system      etcd-ubuntu                           1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-controller-manager-ubuntu        1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-b5flw                      1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      kube-proxy-ttsjp                      1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-zp9xw                      1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      kube-scheduler-ubuntu                 1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-0         1/1     Running            0          4m
192.168.243.198  ubuntu        <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r     1/1     Running            2          8d
192.168.243.215  ubuntu        <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l  1/1     Running            1          8d
192.168.243.238  ubuntu        <none>           <none>
volcano-system   volcano-admission-init-bbx5z          0/1     Completed          0          39s
10.174.217.96    ubuntu-96     <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q    1/1     Running            2          8d
192.168.243.211  ubuntu        <none>           <none>
```

 Copyright © Huawei Technologies Co., Ltd.

- Two training nodes running 2 x 2P distributed training jobs

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                    READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                          1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                          1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                          1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n        1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v 1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk 1/1    Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                       1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                       1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      calico-node-fh58s                       1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      coredns-6955765f44-4pdhg                1/1     Running            0          8d
192.168.243.249  ubuntu        <none>           <none>
kube-system      coredns-6955765f44-n9pg4                1/1     Running            2          8d
192.168.243.237  ubuntu        <none>           <none>
kube-system      etcd-ubuntu                             1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-controller-manager-ubuntu          1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-b5flw                        1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      kube-proxy-ttsjp                        1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-zp9xw                        1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      kube-scheduler-ubuntu                   1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-0           1/1     Running            0          3m
192.168.243.198  ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-1           1/1     Running            0          3m
192.168.243.199  ubuntu        <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r       1/1     Running            2          8d
192.168.243.215  ubuntu        <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l    1/1     Running            1          8d
192.168.243.238  ubuntu        <none>           <none>
volcano-system   volcano-admission-init-bbx5z            0/1     Completed          0          39s
10.174.217.96    ubuntu-96     <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q      1/1     Running            2          8d
192.168.243.211  ubuntu        <none>           <none>
```

Step 2 View the NPU allocation of compute nodes.

Run the following command on the management node:

kubectl describe nodes

NOTE

- The **huawei.com/Ascend910** field of **Annotations** indicates the available NPU of the compute node.
- The **huawei.com/Ascend910** field in **Allocated resources** indicates the number of used NPUs.

 Copyright © Huawei Technologies Co., Ltd.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    accelerator-type=card
                    beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-x86
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910: Ascend910-0
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests        Limits
 --------            --------        ------
 cpu                 37250m (19%)    37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)          0 (0%)
 huawei.com/Ascend910  1               1
Events:             <none>
```

NOTE

The **Annotations** field does not contain the Ascend 910-1 AI Processor, and the value of the **Allocated resources** field **huawei.com/Ascend910** is **1**, indicating that one processor is used for training.

- One of the two training nodes running 2 x 2P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    accelerator-type=card
                    beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-x86
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
```

 Copyright © Huawei Technologies Co., Ltd.

```
                node-role.kubernetes.io/worker=worker
                workerselector=dls-worker-node
Annotations:    huawei.com/Ascend910:
                kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                node.alpha.kubernetes.io/ttl: 0
                projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests       Limits
 --------            --------       ------
 cpu                 37250m (19%)   37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)         0 (0%)
 huawei.com/Ascend910  2            2
Events:             <none>
```

NOTE

No NPU is available in the **Annotations** field, and the value of the **huawei.com/Ascend910** field in **Allocated resources** is **2**, all NPUs are used for distributed training.

Step 3 View the NPU usage of a pod.

In this example, run the **kubectl describe pod mindx-dls-test-default-test-0 -n vcjob** command to check the running status of the pod.

NOTE

Annotations displays the NPU information.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 15:38:22 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"4","device_ip":"192.168.21.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-1
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
```

 Copyright © Huawei Technologies Co., Ltd.

```
            volcano.sh/job-version: 0
            volcano.sh/task-spec: default-test
Status:     Running
```

- Two training nodes running 2 x 2P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 15:38:22 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"1","device_ip":"192.168.20.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-0,Ascend910-1
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

----End

3.4.1.2.6 Viewing the Running Result

Step 1 Log in to the storage server.

The following uses the local NFS whose hostname is **Ubuntu** as an example.

Step 2 Run the following command to go to the output directory of the job running YAML file:

The **resnet50.log** file in the **/data/atlas_dls/output/logs/** directory records the training FPS value. In this example, the directory structure of a single-node training job is the same as that of a distributed training job.

```
root@ubuntu:/home# ll /data/atlas_dls/output/logs/
total 16896
drwxr-x--- 2 HwHiAiUser HwHiAiUser    4096 Oct  7 16:06 ./
drwxr-x--- 4 hwMindX    HwHiAiUser    4096 Oct  7 15:26 ../
...
-rwxr-x--- 1 HwHiAiUser HwHiAiUser     682 Oct  7 16:06 resnet50.log*
```

Step 3 View the information in **resnet50.log**.

cat /data/atlas_dls/output/logs/resnet50.log

If the FPS value is displayed in the command output, the training is successful.

```
PY3.7.5 (default, Dec 10 2020, 04:31:28)
[GCC 7.5.0]TF1.15.0
Step   Epoch   Speed   Loss   FinLoss   LR
step:   100  epoch:  0.0  FPS:   82.6  loss: 6.902  total_loss: 8.242  lr:0.00002
step:   200  epoch:  0.0  FPS: 1771.5  loss: 6.988  total_loss: 8.328  lr:0.00005
step:   300  epoch:  0.0  FPS: 1771.8  loss: 6.969  total_loss: 8.305  lr:0.00007
step:   400  epoch:  0.0  FPS: 1769.3  loss: 6.988  total_loss: 8.328  lr:0.00010
step:   500  epoch:  0.0  FPS:  691.6  loss: 6.895  total_loss: 8.234  lr:0.00012
step:   600  epoch:  0.0  FPS:  741.0  loss: 6.895  total_loss: 8.234  lr:0.00015
step:   700  epoch:  0.0  FPS:  563.2  loss: 6.922  total_loss: 8.258  lr:0.00017
step:   800  epoch:  0.0  FPS:  659.6  loss: 6.934  total_loss: 8.273  lr:0.00020
step:   900  epoch:  0.0  FPS:  851.5  loss: 6.898  total_loss: 8.234  lr:0.00022
step:  1000  epoch:  0.0  FPS: 1524.7  loss: 6.949  total_loss: 8.289  lr:0.00025
```

 Copyright © Huawei Technologies Co., Ltd.

Step 4 Go to the directory for storing the PB model and view the generated PB file.

ls -l /data/atlas_dls/output/model

```
root@ubuntu:/home# ls -l /data/atlas_dls/output/model/
total 99960
-rw-rw---- 1 HwHiAiUser HwHiAiUser 102356214 Mar  3 14:30 resnet50_final.pb
```

----End

3.4.1.2.7 Deleting a Training Job

Run the following command in the directory where YAML is running to delete a training job:

kubectl delete -f *XXX.yaml*

Example:

kubectl delete -f *Mindx-dl-test.yaml*

```
root@ubuntu:/home/test/yaml# kubectl delete -f Mindx-dl-test.yaml
configmap "rings-config-mindx-dls-test" deleted
job.batch.volcano.sh "mindx-dls-test" deleted
```

3.4.2 PyTorch

3.4.2.1 Atlas 800 Training Server

3.4.2.1.1 Preparing the NPU Training Environment

After MindX DL is installed, you can use YAML to deliver a vcjob to check whether the system can run properly.

This section uses the environment described in **Table 3-3** as an example.

Table 3-3 Test environment requirements

Resource Item	Name	Version
OS	• Ubuntu 18.04 • CentOS 7.6 • EulerOS 2.8	-
Training script	Benchmark **NOTE** This document uses the ResNet50 model.	-
OS architecture	• ARM • x86	-

Copyright © Huawei Technologies Co., Ltd.

Creating a Training Image

Create a training image. For details, see **Creating a Container Image Using a Dockerfile (PyTorch)**. You can rename the training image, for example, **torch:b035**.

Preparing a Dataset

The **imagenet** dataset is used only as an example.

Step 1 You need to prepare the dataset by yourself. The **imagenet** dataset is recommended.

Step 2 Upload the dataset to the storage node as an administrator.

1. Go to the **/data/atlas_dls/public** directory and upload the **imagenet** dataset to any directory, for example, **/data/atlas_dls/public/dataset/resnet50/imagenet**.

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50imagenet# pwd
/data/atlas_dls/public/dataset/resnet50/imagenet
```

2. Run the following command to check the dataset size:

   du -sh

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/imagenet# du -sh
146G
```

Step 3 Run the following command to change the owner of the dataset:

chown -R hwMindX:hwMindX /data/atlas_dls/

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/imagenet# chown -R hwMindX:hwMindX /data/
atlas_dls/
```

Step 4 Run the following command to change the dataset permission:

chmod -R 750 /data/atlas_dls/

Step 5 Run the following command to check the file status:

ll /data/atlas_dls/public/*Dataset location*

Example:

ll /data/atlas_dls/public/*dataset/resnet50/imagenet*

```
root@ubuntu:~# ll /data/atlas_dls/public/dataset/resnet50/imagenet
total 84
drwxr-x---    4 hwMindX hwMindX  4096 Oct 20 17:29 ./
drwxr-x---    3 hwMindX hwMindX  4096 Oct 16 11:35 ../
drwxr-x--- 1002 hwMindX hwMindX 36864 Sep 12 16:01 train/
drwxr-x--- 1002 hwMindX hwMindX 36864 Sep 12 16:15 val/
```

----End

Obtaining and Modifying the Training Script

Step 1 Obtain the **training script**.

NOTE

Currently, this function is available only to ISV partners. For other users, contact Huawei engineers.

 Copyright © Huawei Technologies Co., Ltd.

Step 2 Change the script permission and owner.

1. Upload the training script to the **/data/atlas_dls/code** directory on the storage node and decompress it.
2. Run the following command to assign the execute permission recursively:

   chmod -R 750 /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# chmod -R 750 /data/atlas_dls/code/
   ```

3. Run the following command to change the owner:

   chown -R hwMindX:hwMindX /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# chown -R hwMindX:hwMindX /data/atlas_dls/code
   ```

4. Run the following command to view the output result:

   ll /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# ll
   total 12
   drwxrwxrwx 3 hwMindX hwMindX 4096 Oct 20 20:01 ./
   drwxr-x--- 6 hwMindX hwMindX 4096 Oct 22 17:03 ../
   drwxrwx--- 5 hwMindX hwMindX 4096 Oct 20 20:01 benchmark_20200924-benchmark_Alpha/
   ```

----End

3.4.2.1.2 Creating a YAML File

The YAML example is used in the NFS scenario. The NFS needs to be installed on the storage node. For details about how to install NFS, see **Installing the NFS**.

NOTE

If MindX DL is fully installed in online or offline mode, the NFS can be automatically installed.

Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *XXX*.**yaml**

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

NOTICE

Delete # when using the file.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob     # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910
data:
  hccl.json: |
    {
        "status":"initializing"
    } #This line is automatically generated by HCCL-Controller. Keep it unchanged.
```

 Copyright © Huawei Technologies Co., Ltd.

```
---
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: mindx-dls-test               # The value must be consistent with the name of ConfigMap.
  namespace: vcjob                   # Select a proper namespace based on the site requirements. (The
namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add
exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The HCCL-Controller distinguishes Ascend 910 and other processors
based on this label.
spec:
  minAvailable: 1                  #The value of minAvailable is 1 in a single-node scenario and 2 in a
distributed scenario.
  schedulerName: volcano             #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 1                #The value of replicas is 1 in a single-node scenario and N in an N-node scenario.
The number of NPUs in the requests field is 8 in an N-node scenario.
    template:
      metadata:
        labels:
          app: tf
          ring-controller.atlas: ascend-910
      spec:
        hostNetwork: true
        containers:
        - image: torch:b035              # Training framework image, which can be modified.
          imagePullPolicy: IfNotPresent
          name: torch
# ==========Distributed scenario ============
          env:
          - name: NODE_IP
            valueFrom:
              fieldRef:
                fieldPath: status.hostIP
          - name: MY_POD_IP
            valueFrom:
              fieldRef:
                fieldPath: status.podIP
# ============================================
          command:
          - "/bin/bash"
          - "-c"
          - "cd /job/code/train;./benchmark.sh -e Resnet50 -hw 1p -f pytorch"   #Command for running
the training script. The command varies according to scenarios, Xp in a single-node scenario and ct in a
distributed scenario. X indicates the number of NPUs. Ensure that the involved commands and paths exist
on Docker.
          #args: [ "while true; do sleep 30000; done;"  ]     #Comment out the preceding line and enable this
line. You can manually run the training script in the container to facilitate debugging.
          resources:
            requests:
              huawei.com/Ascend910: 1          # Number of required NPUs. The maximum value is 8. You can
add lines below to configure resources such as memory and CPU.
            limits:
              huawei.com/Ascend910: 1          # The value must be consistent with that in requests.
          volumeMounts:
          - name: ascend-910-config
            mountPath: /user/serverid/devindex/config
          - name: code
```

 Copyright © Huawei Technologies Co., Ltd.

```
          mountPath: /job/code/
        - name: data
          mountPath: /job/data
        - name: output
          mountPath: /job/output
        - name: ascend-driver
          mountPath: /usr/local/Ascend/driver
        - name: ascend-add-ons
          mountPath: /usr/local/Ascend/add-ons
        - name: dshm
          mountPath: /dev/shm
        - name: localtime
          mountPath: /etc/localtime
      nodeSelector:
        host-arch: huawei-arm    # Configure the label based on the actual job.
      volumes:
      - name: ascend-910-config
        configMap:
         name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
      - name: code
        nfs:
          server: 127.0.0.1              #IP address of the NFS server. In this example, the shared path is /data/
atlas_dls/.
          path: "/data/atlas_dls/code/benchmark"     #Configure the path of the training script. Modify the
path based on the actual benchmark name.
      - name: data
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/public/dataset/resnet50/imagenet"    # Configure the path of the training
set.
      - name: output
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/output/"    # Configure the path for saving the configuration model, which is
related to the script.
      - name: ascend-driver
        hostPath:
          path: /usr/local/Ascend/driver     #Configure the NPU driver and mount it to Docker.
      - name: ascend-add-ons
        hostPath:
          path: /usr/local/Ascend/add-ons   #Configure the add-ons driver of the NPU and mount it to
Docker.
      - name: dshm
        emptyDir:
          medium: Memory
      - name: localtime
        hostPath:
          path: /etc/localtime   # Configure the Docker time.
      env:
      - name: mindx-dls-test  # The value must be consistent with the value of JobName.
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      restartPolicy: OnFailure
```

3.4.2.1.3 Preparing for Running a Training Job

Single-Node Scenario

Step 1 Run the following command to modify the resources in the YAML file.

vim *XXX*.yaml

NOTE

XXX: YAML name generated in **Creating a YAML File**.

Example:

 Copyright © Huawei Technologies Co., Ltd.

vim *Mindx-dl-test*.**yaml**

1. Change the number of NPUs based on resource requirements.

```
...
    resources:
        requests:
          huawei.com/Ascend910: X        #Number of required NPUs. The maximum value is 8. You can add lines below to configure resources such as memory and CPU.
        limits:
          huawei.com/Ascend910: X        #Number of required NPU chips. The maximum value is 8.
...
```

   NOTE

   X: indicates the number of NPUs.

   - For a single-server single-device scenario, the value of ***X*** is **1**.
   - For a single-server multi-device scenario, the value of ***X*** is **2**, **4**, or **8**.

2. Modify the boot command based on the resource requirements.

```
...
        command:
        - "/bin/bash"
        - "-c"
        - "cd /job/code/train;./benchmark.sh -e Resnet50 -hw Xp -f pytorch"   #Command for running the training script. Ensure that the path in the command exist on Docker.
        #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
...
```

   NOTE

   X: indicates the number of NPUs.

   - For a single-server single-device scenario, the value of ***X*** is **1**.
   - For a single-server multi-device scenario, the value of ***X*** is **2**, **4**, or **8**.

3. For details about how to modify other items, see **Creating a YAML File**.

Step 2 Modify the training script.

In the **/data/atlas_dls/code** directory, go to the YAML file configured for the training job, for example, **/data/atlas_dls/code/Benchmark/train/yaml**. Modify **ResNet50.yaml** based on the number of running processors.

NOTE

In the single-node scenario, you need to modify the following parameters:

- **data_url**: Set this parameter to the path of the dataset mounted to the container.
- **device_group_1p**: Set this parameter to **0** in the single-node scenario.

```
pytorch_config:
  # The input data dir
  data_url: /job/data

  # The mapping of the number of devices and batch_size(the number of devices : batch_size) is 1p:512, 2p:1024, 4p:2048
  # 8p:4096
  batch_size: 512

  # The number of training epochs. Set epochs 90 when wanting to get ACCURACY.
  epoches: 90

  # Training mode, train_and_eval or eval
  mode: train_and_eval

  # Only when the value of mode is eval, config this parameter.Input the ckpt path from training result.
  ckpt_path: /home/train/result/pt_resnet50/training_job_20200916042624/7/
```

 Copyright © Huawei Technologies Co., Ltd.

```
checkpoint_npu7model_best.pth.tar

    # Set this parameter only for docker situation. Docker image name:version number.
    docker_image: c73:b02

    # Learning rate, the value of 1p is 0.2, 2p/4p/8p is 1.6
    lr: 0.2

    # Set device id when training with one piece of npu device.
    device_group_1p: 0

    # Set device id if the number of devices when training is greater than 1. If the number of devices is 2, the
value of device_group_multi
    # can be '0, 1' or '2, 3'. If the number of devices is 4, the value of device_group_multi can be '0, 1, 2, 3' or
'4, 5, 6, 7'.
    # If the number of devices is 8, the value of device_group_multi can be '0, 1, 2, 3, 4, 5, 6, 7'
    device_group_multi: 0,1,2,3,4,5,6,7

    # Set this parameter only for multi-node deployment. Cluster master server ip.
    addr: 172.16.176.54

    # Set this parameter only for multi-node deployment. Node rank for distributed training,default value is
0.
    rank: 0

    # Set this parameter only for multi-node deployment. Start cluster server with mpirun. Tool mpirun
config,server1_ip:the number of
    # training shell process,server_ip2:the number of training shell process...
    # Training shell process default is 1, please do not modify this.
    mpirun_ip: 172.16.176.152:1,172.16.176.154:1

    # Set this parameter only for multi-node deployment. The first device id and the index of the first device
in every server in cluster.
    # The first device id: the index of the first device id.
    cluster_device_ip: 192.168.10.101:0 192.168.10.103:0

    # Set this parameter only for multi-node deployment. The number of devices training using in every
server.
    # The default value is 8, device count per server in cluster.
    cdc: 8

    # Set this parameter only for multi-node deployment. The number of servers in cluster.
    world_size: 2
```

----End

Distributed Scenario

Step 1 Run the following command to modify the resources in the YAML file.

vim *XXX*.**yaml**

NOTE

XXX: YAML name generated in **Creating a YAML File**.

Example:

vim *Mindx-dl-test*.**yaml**

1. Change the number of NPUs based on resource requirements.

```
...
    resources:
        requests:                # You can add lines below to configure resources such as memory
and CPU.
          huawei.com/Ascend910: X
        limits:
```

 Copyright © Huawei Technologies Co., Ltd.

```
            huawei.com/Ascend910: X
...
```

NOTE

X: number of NPUs. The value is **8**.

2. Modify the boot command based on the resource requirements.

```
...
        command:
        - "/bin/bash"
        - "-c"
        - "cd /job/code/train;./benchmark.sh -e Resnet50 -hw ct -f pytorch"   #Command for running the training script. Ensure that the path in the command exist on Docker.
        #args: [ "while true; do sleep 30000; done;" ]    #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
...
```

3. For details about how to modify other items, see **Creating a YAML File**.

Step 2 Modify the training script.

1. In the **/data/atlas_dls/code** directory, go to the YAML file configured for the training job, for example, **/data/atlas_dls/code/Benchmark/train/yaml**. Modify **ResNet50.yaml** based on the number of running processors.

NOTE

In the distributed scenario, you need to modify the following parameters:

- **data_url**: Set this parameter to the path of the dataset mounted to the container.
- **device_group_multi**: In the distributed scenario, each server uses eight NPUs. Set this parameter to **0**, **1**, **2**, **3**, **4**, **5**, **6**, or **7**.
- **addr**: Set this parameter to the IP address of the management node in a distributed cluster.
- **mpirun_ip**: Set this parameter to the IP addresses of the nodes in the distributed cluster. Use commas (,) to separate multiple IP addresses. The format is as follows: *Node IP address*:1,*Node IP address*:1.
- **cluster_device_ip**: Set this parameter to **device_ip:0** of the first NPU on each node in the distributed cluster. Use commas (,) to separate multiple IP addresses. You can run the **hccn_tool -i 0 -ip -g** command to query the value of **device_ip** corresponding to **device_id=0** on each node.
- **cdc**: Ensure that this parameter is set to **8**.
- **world_size**: Set this parameter to the number of nodes in the distributed cluster. For example, if a 2-node cluster runs 2 x 8P training jobs, set this parameter to **2**.

```
pytorch_config:
    # Change the value based on the actual location of the dataset.
    data_url: /job/data

    # The mapping of the number of devices and batch_size(the number of devices : batch_size) is 1p:512,2p:1024,4p:2048
    # 8p:4096
    batch_size: 512

    # The number of training epochs. Set epochs 90 when wanting to get ACCURACY.
    epoches: 90

    # Training mode, train_and_eval or eval
    mode: train_and_eval

    # Only when the value of mode is eval, config this parameter. Input the ckpt path from training result.
    ckpt_path: /home/train/result/pt_resnet50/training_job_20200916042624/7/checkpoint_npu7model_best.pth.tar
```

 Copyright © Huawei Technologies Co., Ltd.

```
    # Set this parameter only for docker situation. Docker image name:version number.
    docker_image: c73:b02

    # Learning rate, the value of 1p is 0.2, 2p/4p/8p is 1.6
    lr: 0.2

    # Set device id when training with one piece of npu device.
    device_group_1p: 0

    # Set device id if the number of devices when training is greater than 1. If the number of devices is
2, the value of device_group_multi
    # can be '0, 1' or '2, 3'. If the number of devices is 4, the value of device_group_multi can be '0, 1,
2, 3' or '4, 5, 6, 7'.
    # If the number of devices is 8, the value of device_group_multi can be '0, 1, 2, 3, 4, 5, 6, 7'
    device_group_multi: 0,1,2,3,4,5,6,7

   # Set this parameter only for multi-node deployment. Cluster master server ip.
    addr: 172.16.176.54

    # Set this parameter only for multi-node deployment. Node rank for distributed training, default
value is 0.
    rank: 0

    # Set this parameter only for multi-node deployment. Start cluster server with mpirun. Tool mpirun
config,server1_ip:the number of
    # training shell process,server_ip2:the number of training shell process...
    # Training shell process default is 1, please do not modify this.
    mpirun_ip: 172.16.176.152:1,172.16.176.154:1

    # Set this parameter only for multi-node deployment. The first device id and the index of the first
device in every server in cluster.
    # The first device id: the index of the first device id.
    cluster_device_ip: 192.168.10.101:0 192.168.10.103:0

    # Set this parameter only for multi-node deployment. The number of devices training using in
every server.
    # The default value is 8, device count per server in cluster.
    cdc: 8

    # Set this parameter only for multi-node deployment. The number of servers in cluster.
    world_size: 2
```

2. Modify the **run.sh** file.

   In the **/data/atlas_dls/code** directory, go to the directory where the model startup script is stored and modify the **run.sh** file to comment out the following content:

   For example, go to the **/data/atlas_dls/code/Benchmark\image_classification\ResNet50\pytorch\scripts** directory and comment out the following content:

```
...
rank_size=$1

yamlPath=$2
toolsPath=$3
#if [ -f /.dockerenv ];then
#     CLUSTER=$4
#     MPIRUN_ALL_IP="$5"
#     export CLUSTER=${CLUSTER}
#fi
...
# ==========Replace the original code. ==============================
if [ x"${CLUSTER}" == x"True" ];then
    echo "whether if will run into cluster"
    ln -snf ${currentDir%train*}/train/result/pt_resnet50/training_job_${currtime}/0/hw_resnet50.log $
{train_job_dir}
    this_ip=$NODE_IP
    if [ x"${addr}" == x"${this_ip}" ]; then
```

 Copyright © Huawei Technologies Co., Ltd.

```
        rm -rf ${currentDir}/config/hccl_bridge_device_file
        if [ ! -d "${currentDir}/config" ]; then
            mkdir ${currentDir}/config
        fi
        hccl_bridge_device_path=${currentDir}/config/hccl_bridge_device_file
        for i in ${cluster_device_ip[@]};do
            echo $i >> ${hccl_bridge_device_path}
        done
        chmod 755 ${hccl_bridge_device_path}
        export HCCL_BRIDGE_DEVICE_FILE=${hccl_bridge_device_path}
        ranks=0
        for ip in ${MPIRUN_ALL_IP[@]};do
            if [ x"$ip" != x"$this_ip" ];then
              bak_yaml=$(dirname "${yamlPath}")/ResNet50_$ip.yaml
              rm -rf ${bak_yaml}
              cp $yamlPath $bak_yaml
              ranks=$[ranks+1]
              sed -i "s/rank.*$/rank\: ${ranks}/g" ${bak_yaml}
            fi
        done
    fi
    if [ x"${addr}" != x"${this_ip}" ]; then
      yamlPath=$(dirname "${yamlPath}")/ResNet50_$this_ip.yaml
      while [ ! -f ${yamlPath} ];do
          echo "Wait for the generation of yaml files of worker nodes."
          sleep 2
      done
      echo "start run train.sh"
      echo "look at yamPath ${yamlPath}"
      bash ${currentDir}/scripts/train.sh 0 $rank_size $yamlPath $currtime ${toolsPath} ${CLUSTER}
    else
      echo "start run train.sh"
      echo "look at yamPath ${yamlPath}"
      bash ${currentDir}/scripts/train.sh 0 $rank_size $yamlPath $currtime ${toolsPath} ${CLUSTER}
    fi
else
# ===========================================
...
```

----End

3.4.2.1.4 Delivering a Training Job

Procedure

Step 1 Run the following command to create a namespace for the training job:

kubectl create namespace vcjob

Step 2 Run the following command on the management node to deliver training jobs using YAML:

kubectl apply -f *XXX*.yaml

Example:

kubectl apply -f *Mindx-dl-test*.yaml

```
root@ubuntu:/home/test/yaml# kubectl apply -f Mindx-dl-test.yaml
configmap/rings-config-mindx-dls-test created
job.batch.volcano.sh/mindx-dls-test created
```

----End

 Copyright © Huawei Technologies Co., Ltd.

3.4.2.1.5 Checking the Running Status

Procedure

Step 1 Run the following command to check the pod running status:

kubectl get pod --all-namespaces -o wide

- Example of a single-server single-device training job

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE       NAME                                READY   STATUS             RESTARTS   AGE
IP              NODE          NOMINATED NODE   READINESS GATES
cadvisor        cadvisor-8x86g                      1/1     Running            33         8d
192.168.243.252   ubuntu        <none>           <none>
cadvisor        cadvisor-hgbw8                      1/1     Running            0          26h
192.168.207.48    ubuntu-96     <none>           <none>
cadvisor        cadvisor-shwb4                      1/1     Running            0          6m46s
192.168.240.65    ubuntu-infer  <none>           <none>
default         hccl-controller-688c7cb8c6-4b88n    1/1     Running            0          8d
192.168.243.199   ubuntu        <none>           <none>
kube-system     ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218   ubuntu        <none>           <none>
kube-system     ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49    ubuntu-96     <none>           <none>
kube-system     ascend310-device-plugin-daemonset-fls4v 1/1     Running            0          4m15s
192.168.240.66    ubuntu-infer  <none>           <none>
kube-system     calico-kube-controllers-8464785d6b-bj4pk 1/1     Running            1          8d
192.168.243.198   ubuntu        <none>           <none>
kube-system     calico-node-bkbvl                   1/1     Running            0          8m16s
10.174.216.214    ubuntu-infer  <none>           <none>
kube-system     calico-node-bzd7q                   1/1     Running            3          8d
10.174.217.94     ubuntu        <none>           <none>
kube-system     calico-node-fh58s                   1/1     Running            1          8d
10.174.217.96     ubuntu-96     <none>           <none>
kube-system     coredns-6955765f44-4pdhg            1/1     Running            0          8d
192.168.243.249   ubuntu        <none>           <none>
kube-system     coredns-6955765f44-n9pg4            1/1     Running            2          8d
192.168.243.237   ubuntu        <none>           <none>
kube-system     etcd-ubuntu                         1/1     Running            3          8d
10.174.217.94     ubuntu        <none>           <none>
kube-system     kube-controller-manager-ubuntu      1/1     Running            4          8d
10.174.217.94     ubuntu        <none>           <none>
kube-system     kube-proxy-b5flw                    1/1     Running            1          8d
10.174.217.96     ubuntu-96     <none>           <none>
kube-system     kube-proxy-ttsjp                    1/1     Running            3          8d
10.174.217.94     ubuntu        <none>           <none>
kube-system     kube-proxy-zp9xw                    1/1     Running            0          8m16s
10.174.216.214    ubuntu-infer  <none>           <none>
kube-system     kube-scheduler-ubuntu               1/1     Running            4          8d
10.174.217.94     ubuntu        <none>           <none>
vcjob           mindx-dls-test-default-test-0       1/1     Running            0          4m
192.168.243.198   ubuntu        <none>           <none>
volcano-system  volcano-admission-5bcb6d799-rgk5r   1/1     Running            2          8d
192.168.243.215   ubuntu        <none>           <none>
volcano-system  volcano-controllers-7d6d465877-nnf7l    1/1     Running            1          8d
192.168.243.238   ubuntu        <none>           <none>
volcano-system  volcano-admission-init-bbx5z        0/1     Completed          0          39s
10.174.217.96     ubuntu-96     <none>           <none>
volcano-system  volcano-scheduler-67f89949b4-ncs8q  1/1     Running            2          8d
192.168.243.211   ubuntu        <none>           <none>
```

- Two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE       NAME                                READY   STATUS             RESTARTS   AGE
IP              NODE          NOMINATED NODE   READINESS GATES
cadvisor        cadvisor-8x86g                      1/1     Running            33         8d
192.168.243.252   ubuntu        <none>           <none>
cadvisor        cadvisor-hgbw8                      1/1     Running            0          26h
```

 Copyright © Huawei Technologies Co., Ltd.

```
192.168.207.48   ubuntu-96      <none>          <none>
cadvisor         cadvisor-shwb4                 1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer   <none>          <none>
default          hccl-controller-688c7cb8c6-4b88n          1/1     Running            0          8d
192.168.243.199  ubuntu         <none>          <none>
kube-system      ascend-device-plugin-daemonset-8f2dx      1/1     Running            2          8d
192.168.243.218  ubuntu         <none>          <none>
kube-system      ascend-device-plugin-daemonset-f2jk9      1/1     Running            1          8d
192.168.207.49   ubuntu-96      <none>          <none>
kube-system      ascend310-device-plugin-daemonset-fls4v   1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer   <none>          <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk  1/1     Running            1          8d
192.168.243.198  ubuntu         <none>          <none>
kube-system      calico-node-bkbvl                         1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer   <none>          <none>
kube-system      calico-node-bzd7q                         1/1     Running            3          8d
10.174.217.94    ubuntu         <none>          <none>
kube-system      calico-node-fh58s                         1/1     Running            1          8d
10.174.217.96    ubuntu-96      <none>          <none>
kube-system      coredns-6955765f44-4pdhg                  1/1     Running            0          8d
192.168.243.249  ubuntu         <none>          <none>
kube-system      coredns-6955765f44-n9pg4                  1/1     Running            2          8d
192.168.243.237  ubuntu         <none>          <none>
kube-system      etcd-ubuntu                               1/1     Running            3          8d
10.174.217.94    ubuntu         <none>          <none>
kube-system      kube-controller-manager-ubuntu            1/1     Running            4          8d
10.174.217.94    ubuntu         <none>          <none>
kube-system      kube-proxy-b5flw                          1/1     Running            1          8d
10.174.217.96    ubuntu-96      <none>          <none>
kube-system      kube-proxy-ttsjp                          1/1     Running            3          8d
10.174.217.94    ubuntu         <none>          <none>
kube-system      kube-proxy-zp9xw                          1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer   <none>          <none>
kube-system      kube-scheduler-ubuntu                     1/1     Running            4          8d
10.174.217.94    ubuntu         <none>          <none>
vcjob            mindx-dls-test-default-test-0             1/1     Running            0          3m
192.168.243.198  ubuntu         <none>          <none>
vcjob            mindx-dls-test-default-test-1             1/1     Running            0          3m
192.168.243.199  ubuntu         <none>          <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r         1/1     Running            2          8d
192.168.243.215  ubuntu         <none>          <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l      1/1     Running            1          8d
192.168.243.238  ubuntu         <none>          <none>
volcano-system   volcano-admission-init-bbx5z              0/1     Completed          0          39s
10.174.217.96    ubuntu-96      <none>          <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q        1/1     Running            2          8d
192.168.243.211  ubuntu         <none>          <none>
```

Step 2 (Optional) Run the following commands to check the run logs:

kubectl logs -n *[Pod running namespace] [Pod name]*

Example:

kubectl logs -n vcjob *mindx-dls-test-default-test-0*

Step 3 View the NPU allocation of compute nodes.

Run the following command on the management node:

kubectl describe nodes

NOTE

- The **huawei.com/Ascend910** field of **Annotations** indicates the available NPU of the compute node.
- The **huawei.com/Ascend910** field in **Allocated resources** indicates the number of used NPUs.

 Copyright © Huawei Technologies Co., Ltd.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910:
Ascend910-0,Ascend910-1,Ascend910-3,Ascend910-4,Ascend910-5,Ascend910-6,Ascend910-7
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests        Limits
 --------            --------        ------
 cpu                 37250m (19%)    37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)          0 (0%)
 huawei.com/Ascend910  1               1
Events:             <none>
```

NOTE

The **Annotations** field does not contain the Ascend 910-2 AI Processor, and the value of the **Allocated resources** field **huawei.com/Ascend910** is **1**, indicating that one processor is used for training.

- One of the two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
```

```
                workerselector=dls-worker-node
Annotations:    huawei.com/Ascend910:
                kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                node.alpha.kubernetes.io/ttl: 0
                projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:     1537233808Ki
 huawei.com/Ascend910:  8
 hugepages-2Mi:         0
 memory:                792307468Ki
 pods:                  110
Allocatable:
 cpu:                192
 ephemeral-storage:     1416714675108
 huawei.com/Ascend910:  8
 hugepages-2Mi:         0
 memory:                792205068Ki
 pods:                  110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests        Limits
 --------            --------        ------
 cpu                 37250m (19%)    37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)          0 (0%)
 huawei.com/Ascend910  8               8
Events:              <none>
```

NOTE

No NPU is available in the **Annotations** field, and the value of the **huawei.com/Ascend910** field in **Allocated resources** is **8**, all NPUs are used for distributed training.

Step 4 View the NPU usage of a pod.

In this example, run the **kubectl describe pod mindx-dls-test-default-test-0 -n vcjob** command to check the running status of the pod.

NOTE

Annotations displays the NPU information.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 17:42:32 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"3","device_ip":"192.168.20.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-2
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
```

 Copyright © Huawei Technologies Co., Ltd.

```
            volcano.sh/task-spec: default-test
Status:     Running
```

- Two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 18:12:07 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"0","device_ip":"192.168.20.100"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910:
Ascend910-0,Ascend910-1,Ascend910-2,Ascend910-3,Ascend910-4,Ascend910-5,Ascend910-6,Asce
nd910-7
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

----End

3.4.2.1.6 Viewing the Running Result

Step 1 Log in to the storage server.

The following uses the local NFS whose hostname is **Ubuntu** as an example.

Step 2 Run the following command to view the output directory of the job running YAML file:

ll /data/atlas_dls/code/Benchmark/train/result/pt_resnet50

```
root@ubuntu:/home# ll /data/atlas_dls/code/Benchmark/train/result/pt_resnet50
total 28
drwxr-xr-x 7 root root 4096 Oct 22 17:16 ./
drwxr-xr-x 3 root root 4096 Oct 22 17:10 ../
drwxr-xr-x 4 root root 4096 Oct 22 17:10 training_job_20201022074421/
drwxr-xr-x 6 root root 4096 Oct 22 17:10 training_job_20201022080648/
drwxr-xr-x 6 root root 4096 Oct 22 17:10 training_job_20201022082409/
drwxr-xr-x 6 root root 4096 Oct 22 17:12 training_job_20201022091259/
drwxr-xr-x 6 root root 4096 Oct 22 17:16 training_job_20201022091619/
```

Step 3 Run the following command to access the corresponding training job:

cd /data/atlas_dls/code/Benchmark/train/result/pt_resnet50

cd training_job_20201022091619/

```
root@ubuntu:cd /data/atlas_dls/code/Benchmark/train/result/pt_resnet50/training_job_20201022091619# ll
total 324
drwxr-xr-x 6 root root   4096 Oct 22 17:16 ./
drwxr-xr-x 7 root root   4096 Oct 22 17:16 ../
drwxr-xr-x 3 root root   4096 Oct 22 17:46 0/
drwxr-xr-x 3 root root   4096 Oct 22 17:17 1/
drwxr-xr-x 3 root root   4096 Oct 22 17:17 2/
drwxr-xr-x 3 root root   4096 Oct 22 17:17 3/
lrwxrwxrwx 1 root root     82 Oct 22 17:16 hw_resnet50.log -> /job/code//train/result/pt_resnet50/
training_job_20201022091619//0/hw_resnet50.log
-rw-r--r-- 1 root root 300160 Oct 22 17:46 train_4p.log
```

 Copyright © Huawei Technologies Co., Ltd.

The **train_4p.log** file in this directory records the training precision.

```
root@ubuntu:/data/atlas_dls/public/pytorch/train/result/pt_resnet50/training_job_20201022091619# tail -f
train_4p.log
[gpu id: 0 ] Test: [2495/2502] Time  0.282 ( 0.553) Loss 7.0266 (7.0780) Acc@1   0.59 (  0.15) Acc@5   1.37
(  0.68)
[gpu id: 0 ] Test: [2496/2502] Time  0.256 ( 0.552) Loss 7.0124 (7.0780) Acc@1   0.39 (  0.15) Acc@5   0.98
(  0.68)
[gpu id: 0 ] Test: [2497/2502] Time  0.257 ( 0.552) Loss 7.0578 (7.0779) Acc@1   0.00 (  0.15) Acc@5   0.39
(  0.68)
[gpu id: 0 ] Test: [2498/2502] Time  0.265 ( 0.552) Loss 7.1090 (7.0780) Acc@1   0.00 (  0.15) Acc@5   0.98
(  0.68)
[gpu id: 0 ] Test: [2499/2502] Time  0.318 ( 0.552) Loss 7.0254 (7.0779) Acc@1   0.00 (  0.15) Acc@5   0.20
(  0.68)
[gpu id: 0 ] Test: [2500/2502] Time  0.359 ( 0.552) Loss 7.0409 (7.0779) Acc@1   0.00 (  0.15) Acc@5   0.78
(  0.68)
[gpu id: 0 ] Test: [2501/2502] Time  0.445 ( 0.552) Loss 7.0593 (7.0779) Acc@1   0.39 (  0.15) Acc@5   1.37
(  0.68)
[gpu id: 0 ] [AVG-ACC] * Acc@1 0.151 Acc@5 0.683
THPModule_npu_shutdown success.
:::ABK 1.0.0 resnet50 train success
```

----End

3.4.2.1.7 Deleting a Training Job

Run the following command in the directory where YAML is running to delete a training job:

kubectl delete -f *XXX.yaml*

Example:

kubectl delete -f *Mindx-dl-test.yaml*

```
root@ubuntu:/home/test/yaml# kubectl delete -f Mindx-dl-test.yaml
configmap "rings-config-mindx-dls-test" deleted
job.batch.volcano.sh "mindx-dls-test" deleted
```

3.4.2.2 Server (with Atlas 300T Training Cards)

3.4.2.2.1 Preparing the NPU Training Environment

After MindX DL is installed, you can use YAML to deliver a vcjob to check whether the system can run properly.

This section uses the environment described in **Table 3-4** as an example.

Table 3-4 Test environment requirements

Item	Name	Version
OS	Ubuntu 18.04	-
Training script	Benchmark **NOTE** This document uses the ResNet50 model.	-
OS architecture	x86	-

 Copyright © Huawei Technologies Co., Ltd.

Creating a Training Image

Create a training image. For details, see **Creating a Container Image Using a Dockerfile (PyTorch)**. You can rename the training image, for example, **torch:b035**.

Preparing a Dataset

The **imagenet** dataset is used only as an example.

Step 1 You need to prepare the dataset by yourself. The **imagenet** dataset is recommended.

Step 2 Upload the dataset to the storage node as an administrator.

1. Go to the **/data/atlas_dls/public** directory and upload the **imagenet** dataset to any directory, for example, **/data/atlas_dls/public/dataset/resnet50/imagenet**.

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/imagenet# pwd
/data/atlas_dls/public/dataset/resnet50/imagenet
```

2. Run the following command to check the dataset size:

   du -sh

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/imagenet# du -sh
146G
```

Step 3 Run the following command to change the owner of the dataset:

chown -R hwMindX:hwMindX /data/atlas_dls/

```
root@ubuntu:/data/atlas_dls/public/dataset/resnet50/imagenet# chown -R hwMindX:hwMindX /data/
atlas_dls/
```

Step 4 Run the following command to change the dataset permission:

chmod -R 750 /data/atlas_dls/

Step 5 Run the following command to check the file status:

ll /data/atlas_dls/public/*Dataset location*

Example:

ll /data/atlas_dls/public/*dataset/resnet50/imagenet*

```
root@ubuntu:~# ll /data/atlas_dls/public/dataset/resnet50/imagenet
total 84
drwxr-x---    4 hwMindX hwMindX  4096 Oct 20 17:29 ./
drwxr-x---    3 hwMindX hwMindX  4096 Oct 16 11:35 ../
drwxr-x--- 1002 hwMindX hwMindX 36864 Sep 12 16:01 train/
drwxr-x--- 1002 hwMindX hwMindX 36864 Sep 12 16:15 val/
```

----End

Obtaining and Modifying the Training Script

Step 1 Obtain the **training script**.

NOTE

Currently, this function is available only to ISV partners. For other users, contact Huawei engineers.

 Copyright © Huawei Technologies Co., Ltd.

Step 2 Change the script permission and owner.

1. Upload the training script to the **/data/atlas_dls/code** directory on the storage node and decompress it.
2. Run the following command to assign the execute permission recursively:

   chmod -R 750 /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# chmod -R 750 /data/atlas_dls/code/
   ```

3. Run the following command to change the owner:

   chown -R hwMindX:hwMindX /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# chown -R hwMindX:hwMindX /data/atlas_dls/code
   ```

4. Run the following command to view the output result:

   ll /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# ll
   total 12
   drwxrwxrwx 3 hwMindX hwMindX 4096 Oct 20 20:01 ./
   drwxr-x--- 6 hwMindX hwMindX 4096 Oct 22 17:03 ../
   drwxrwx--- 5 hwMindX hwMindX 4096 Oct 20 20:01 benchmark_20200924-benchmark_Alpha/
   ```

----End

3.4.2.2.2 Creating a YAML File

The YAML example is used in the NFS scenario. The NFS needs to be installed on the storage node. For details about how to install NFS, see **Installing the NFS**.

NOTE

If MindX DL is fully installed in online or offline mode, the NFS can be automatically installed.

Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *XXX*.yaml

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.yaml

NOTICE

Delete # when using the file.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob     # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910
data:
  hccl.json: |
    {
        "status":"initializing"
    } #This line is automatically generated by HCCL-Controller. Keep it unchanged.
```

 Copyright © Huawei Technologies Co., Ltd.

```
---
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: mindx-dls-test              # The value must be consistent with the name of ConfigMap.
  namespace: vcjob                  # Select a proper namespace based on the site requirements. (The
namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add
exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The HCCL-Controller distinguishes Ascend 910 and other processors
based on this label.
spec:
   minAvailable: 1                  #The value of minAvailable is 1 in a single-node scenario and 2 in a
distributed scenario.
  schedulerName: volcano            #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 1                     #The value of replicas is 1 in a single-node scenario and 2 in a distributed
scenario, and the maximum number of NPUs in the requests field is 2 in a distributed scenario.
    template:
      metadata:
        labels:
          app: tf
          ring-controller.atlas: ascend-910
      spec:
        hostNetwork: true
        containers:
        - image: torch:b035             # Training framework image, which can be modified.
          imagePullPolicy: IfNotPresent
          name: torch
# ==========Distributed scenario ============
          env:
          - name: NODE_IP
            valueFrom:
              fieldRef:
                fieldPath: status.hostIP
          - name: MY_POD_IP
            valueFrom:
              fieldRef:
                fieldPath: status.podIP
# ============================================
          command:
          - "/bin/bash"
          - "-c"
          - "cd /job/code/train;./benchmark.sh -e Resnet50 -hw 1p -f pytorch"   #Command for running
the training script. The command varies according to scenarios, Xp in a single-node scenario and ct in a
distributed scenario. X indicates the number of NPUs. Ensure that the involved commands and paths exist
on Docker.
          #args: [ "while true; do sleep 30000; done;"  ]     #Comment out the preceding line and enable this
line. You can manually run the training script in the container to facilitate debugging.
          resources:
            requests:
              huawei.com/Ascend910: 1         # Number of required NPUs. The maximum value is 2. You can
add lines below to configure resources such as memory and CPU.
            limits:
              huawei.com/Ascend910: 1         # The value must be consistent with that in requests.
          volumeMounts:
          - name: ascend-910-config
            mountPath: /user/serverid/devindex/config
          - name: code
```

 Copyright © Huawei Technologies Co., Ltd.

```
          mountPath: /job/code/
        - name: data
          mountPath: /job/data
        - name: output
          mountPath: /job/output
        - name: ascend-driver
          mountPath: /usr/local/Ascend/driver
        - name: ascend-add-ons
          mountPath: /usr/local/Ascend/add-ons
        - name: dshm
          mountPath: /dev/shm
        - name: localtime
          mountPath: /etc/localtime
      nodeSelector:
        host-arch: huawei-arm    # Configure the label based on the actual job.
# ========servers (with Atlas 300T training cards)=========================
        accelerator-type: card
      volumes:
      - name: ascend-910-config
        configMap:
         name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
      - name: code
        nfs:
          server: 127.0.0.1                #IP address of the NFS server. In this example, the shared path is /data/
atlas_dls/.
          path: "/data/atlas_dls/code/benchmark"     #Configure the path of the training script. Modify the
path based on the actual benchmark name.
      - name: data
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/public/dataset/resnet50/imagenet"    # Configure the path of the training
set.
      - name: output
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/output/"    # Configure the path for saving the configuration model, which is
related to the script.
      - name: ascend-driver
        hostPath:
          path: /usr/local/Ascend/driver     #Configure the NPU driver and mount it to Docker.
      - name: ascend-add-ons
        hostPath:
          path: /usr/local/Ascend/add-ons   #Configure the add-ons driver of the NPU and mount it to
Docker.
      - name: dshm
        emptyDir:
          medium: Memory
      - name: localtime
        hostPath:
          path: /etc/localtime   # Configure the Docker time.
      env:
      - name: mindx-dls-test  # The value must be consistent with the value of JobName.
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      restartPolicy: OnFailure
```

3.4.2.2.3 Preparing for Running a Training Job

Single-Node Scenario

Step 1 Run the following command to modify the resources in the YAML file.

vim *XXX*.yaml

NOTE

XXX: YAML name generated in **Creating a YAML File**.

 Copyright © Huawei Technologies Co., Ltd.

Example:

vim *Mindx-dl-test*.**yaml**

1. Change the number of NPUs based on resource requirements.

```
...
    resources:
        requests:                       #You can also configure resources such as the men and cpu.
          huawei.com/Ascend910: X       #Number of NPUs required.
        limits:
          huawei.com/Ascend910: X       #Number of NPUs required.
...
```

   NOTE

   X: indicates the number of NPUs.

   - For a single-server single-device scenario, the value of ***X*** is **1**.
   - For a single-server multi-device scenario, the value of ***X*** is **2**.

2. Modify the boot command based on the resource requirements.

```
...
        command:
        - "/bin/bash"
        - "-c"
        - "cd /job/code/train;./benchmark.sh -e Resnet50 -hw Xp -f pytorch"    #Command for running the training script. Ensure that the path in the command exist on Docker.
        #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
...
```

   NOTE

   X: indicates the number of NPUs.

   - For a single-server single-device scenario, the value of ***X*** is **1**.
   - For a single-server multi-device scenario, the value of ***X*** is **2**.

3. For details about how to modify other items, see **Creating a YAML File**.

Step 2 Modify the training script.

In the **/data/atlas_dls/code** directory, go to the YAML file configured for the training job, for example, **/data/atlas_dls/code/Benchmark/train/yaml**. Modify **ResNet50.yaml** based on the NPU count and running parameters.

NOTE

In the single-node scenario, you need to modify the following parameters:

- **data_url**: Set this parameter to the path of the dataset mounted to the container.
- **device_group_1p**: Set this parameter to **0** in the single-server single-device scenario.
- **device_group_multi**: Set this parameter to **0** or **1** in the single-server multi-device scenario.

```
pytorch_config:
  # The input data dir
  data_url: /job/data
  # The mapping of the number of devices and batch_size(the number of devices : batch_size) is 1p:512, 2p:1024, 4p:2048
  # 8p:4096
  batch_size: 512

  # The number of training epochs. Set epochs 90 when wanting to get ACCURACY.
  epoches: 90

  # Training mode, train_and_eval or eval
  mode: train_and_eval
```

 Copyright © Huawei Technologies Co., Ltd.

```
    # Only when the value of mode is eval, config this parameter.Input the ckpt path from training result.
    ckpt_path: /home/train/result/pt_resnet50/training_job_20200916042624/7/
checkpoint_npu7model_best.pth.tar

    # Set this parameter only for docker situation. Docker image name:version number.
    docker_image: c73:b02

    # Learning rate, the value of 1p is 0.2, 2p/4p/8p is 1.6
    lr: 0.2

    # Set device id when training with one piece of npu device.
    device_group_1p: 0

    # Set device id if the number of devices when training is greater than 1. If the number of devices is 2, the
value of device_group_multi
    # can be '0, 1' or '2, 3'. If the number of devices is 4, the value of device_group_multi can be '0, 1, 2, 3' or
'4, 5, 6, 7'.
    # If the number of devices is 8, the value of device_group_multi can be '0, 1, 2, 3, 4, 5, 6, 7'
    device_group_multi: 0,1
    # Set this parameter only for multi-node deployment. Cluster master server ip.
    addr: 172.16.176.54

    # Set this parameter only for multi-node deployment. Node rank for distributed training, default value is
0.
    rank: 0

    # Set this parameter only for multi-node deployment. Start cluster server with mpirun. Tool mpirun
config,server1_ip:the number of
    # training shell process, server_ip2:the number of training shell process...
    # Training shell process default is 1, please do not modify this.
    mpirun_ip: 172.16.176.152:1,172.16.176.154:1

    # Set this parameter only for multi-node deployment. The first device id and the index of the first device
in every server in cluster.
    # The first device id: the index of the first device id.
    cluster_device_ip: 192.168.10.101:0 192.168.10.103:0

    # Set this parameter only for multi-node deployment. The number of devices training using in every
server.
    # The default value is 8, device count per server in cluster.
    cdc: 2

    # Set this parameter only for multi-node deployment. The number of servers in cluster.
    world_size: 2
```

----End

Distributed Scenario

Step 1 Run the following command to modify the YAML file based on the resource requirements.

vim *XXX*.**yaml**

NOTE

XXX: YAML name generated in **Creating a YAML File**.

Example:

vim *Mindx-dl-test*.**yaml**

1. Change the number of NPUs based on resource requirements.

```
...
    resources:
        requests:                    #You can also configure resources such as the men and cpu.
```

 Copyright © Huawei Technologies Co., Ltd.

```
            huawei.com/Ascend910: X
          limits:
            huawei.com/Ascend910: X
...
```

NOTE

X: number of NPUs. The value is **1** or **2**.

2. Modify the boot command based on the resource requirements.

```
...
        command:
        - "/bin/bash"
        - "-c"
        - "cd /job/code/train;./benchmark.sh -e Resnet50 -hw ct -f pytorch"   #Command for running the training script. Ensure that the path in the command exist on Docker.
        #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
...
```

3. For details about how to modify other items, see **Creating a YAML File**.

Step 2 Modify the training script.

1. In the **/data/atlas_dls/code** directory, go to the YAML file configured for the training job, for example, **/data/atlas_dls/code/Benchmark/train/yaml**. Modify **ResNet50.yaml** based on the number of running processors.

NOTE

In the distributed scenario, you need to modify the following parameters:

- **data_url**: Set this parameter to the path of the dataset mounted to the container.
- **device_group_1p**: Set this parameter to **0** in the distributed single-device scenario.
- **device_group_multi**: In the distributed multi-device scenario, each server uses two NPUs. Set this parameter to **0** or **1**.
- **addr**: Set this parameter to the IP address of the management node in a distributed cluster.
- **mpirun_ip**: Set this parameter to the IP addresses of the nodes in the distributed cluster. Use commas (,) to separate multiple IP addresses. The format is as follows: *Node IP address*:1,*Node IP address*:1.
- **cluster_device_ip**: Set this parameter to **device_ip:0** of the first NPU on each node in the distributed cluster. Use commas (,) to separate multiple IP addresses. You can run the **hccn_tool -i 0 -ip -g** command to query the value of **device_ip** corresponding to **device_id=0** on each node.
- **cdc**: Ensure that this parameter is set to **2**.
- **world_size**: Set this parameter to the number of nodes in the distributed cluster. For example, if a 2-node cluster runs 2 x 2P training jobs, set this parameter to **2**.

```
pytorch_config:
    # Change the value based on the actual location of the dataset.
    data_url: /job/data

    # The mapping of the number of devices and batch_size(the number of devices : batch_size) is 1p:512, 2p:1024, 4p:2048
    # 8p:4096
    batch_size: 512

    # The number of training epochs. Set epochs 90 when wanting to get ACCURACY.
    epoches: 90

    # Training mode, train_and_eval or eval
    mode: train_and_eval

    # Only when the value of mode is eval, config this parameter. Input the ckpt path from training result.
```

 Copyright © Huawei Technologies Co., Ltd.

```
    ckpt_path: /home/train/result/pt_resnet50/training_job_20200916042624/7/
checkpoint_npu7model_best.pth.tar

    # Set this parameter only for docker situation. Docker image name:version number.
    docker_image: c73:b02

    # Learning rate, the value of 1p is 0.2, 2p/4p/8p is 1.6
    lr: 0.2

    # Set device id when training with one piece of npu device.
    device_group_1p: 0

    # Set device id if the number of devices when training is greater than 1. If the number of devices is
2, the value of device_group_multi
    # can be '0, 1' or '2, 3'.If the number of devices is 4, the value of device_group_multi can be '0, 1, 2,
3' or '4, 5, 6, 7'.
    # If the number of devices is 8, the value of device_group_multi can be '0, 1, 2, 3, 4, 5, 6, 7'
   # For the distributed training scenario on servers (with Atlas 300T training cards), if there is only
one card, device_group_multi: 0; If there are two cards, device_group_multi: 0,1
    device_group_multi: 0,1

   # Set this parameter only for multi-node deployment. Cluster master server ip. Run the
kubectl get pods -o wide -n {Job namespace} command to query POD_IP of the active node of the
training jobs.
    addr: 172.16.176.54

    # Set this parameter only for multi-node deployment. Node rank for distributed training, default
value is 0.
    rank: 0

    # Set this parameter only for multi-node deployment.Start cluster server with mpirun. Tool mpirun
config, server1_ip:the number of
    # training shell process, server_ip2:the number of training shell process...
    # Training shell process default is 1, please do not modify this. Run the kubectl get pods -o wide -
n {Job namespace} command to query POD_IP of the two training jobs.
    mpirun_ip: 172.16.176.152:1,172.16.176.154:1

    # Set this parameter only for multi-node deployment. The first device id and the index of the first
device in every server in cluster.
    # The first device id: the index of the first device id. Run the hccn_tool -i 0 -ip -g command to
check device_ip corresponding to device_id=0 on the two servers.
    # For the distributed training scenario on servers (with Atlas 300T training cards), run the
hccn_tool -i 1 -ip -g command to check the value of device_ip corresponding to device_id=1 on the
two servers. The value is device_ip: device_id.
    cluster_device_ip: 192.168.10.101:0 192.168.10.103:0

    # Set this parameter only for multi-node deployment. The number of devices training using in
every server.
    # The default value is 8, device count per server in cluster.
    cdc: 2

    # Set this parameter only for multi-node deployment. The number of servers in cluster.
    world_size: 2
```

2. Modify the **run.sh** file.

   In the **/data/atlas_dls/code** directory, go to the directory where the model startup script is stored and modify the **run.sh** file to comment out the following content:

   For example, go to the **/data/atlas_dls/code/Benchmark\image_classification\ResNet50\pytorch\scripts** directory and comment out the following content:

```
...
rank_size=$1

yamlPath=$2
toolsPath=$3
#if [ -f /.dockerenv ];then
```

 Copyright © Huawei Technologies Co., Ltd.

```
#      CLUSTER=$4
#      MPIRUN_ALL_IP="$5"
#      export CLUSTER=${CLUSTER}
#fi
...
# ==========Replace the original code. ================================
if [ x"${CLUSTER}" == x"True" ];then
    echo "whether if will run into cluster"
    ln -snf ${currentDir%train*}/train/result/pt_resnet50/training_job_${currtime}/0/hw_resnet50.log ${train_job_dir}
    this_ip=$NODE_IP
    if [ x"${addr}" == x"${this_ip}" ]; then
        rm -rf ${currentDir}/config/hccl_bridge_device_file
        if [ ! -d "${currentDir}/config" ]; then
            mkdir ${currentDir}/config
        fi
        hccl_bridge_device_path=${currentDir}/config/hccl_bridge_device_file
        for i in ${cluster_device_ip[@]};do
            echo $i >> ${hccl_bridge_device_path}
        done
        chmod 755 ${hccl_bridge_device_path}
        export HCCL_BRIDGE_DEVICE_FILE=${hccl_bridge_device_path}
        ranks=0
        for ip in ${MPIRUN_ALL_IP[@]};do
            if [ x"$ip" != x"$this_ip" ];then
              bak_yaml=$(dirname "${yamlPath}")/ResNet50_$ip.yaml
              rm -rf ${bak_yaml}
              cp $yamlPath $bak_yaml
              ranks=$[ranks+1]
              sed -i "s/rank.*$/rank\: ${ranks}/g" ${bak_yaml}
            fi
        done
    fi
    if [ x"${addr}" != x"${this_ip}" ]; then
      yamlPath=$(dirname "${yamlPath}")/ResNet50_$this_ip.yaml
      while [ ! -f ${yamlPath} ];do
          echo "Wait for the generation of yaml files of worker nodes."
          sleep 2
      done
      echo "start run train.sh"
      echo "look at yamPath ${yamlPath}"
      bash ${currentDir}/scripts/train.sh 0 $rank_size $yamlPath $currtime ${toolsPath} ${CLUSTER}
    else
      echo "start run train.sh"
      echo "look at yamPath ${yamlPath}"
      bash ${currentDir}/scripts/train.sh 0 $rank_size $yamlPath $currtime ${toolsPath} ${CLUSTER}
    fi
else
...
# ========================================
```

----End

3.4.2.2.4 Delivering a Training Job

Procedure

Step 1 Run the following command to create a namespace for the training job:

kubectl create namespace vcjob

Step 2 Run the following command on the management node to deliver training jobs using YAML:

kubectl apply -f *XXX*.**yaml**

Example:

 Copyright © Huawei Technologies Co., Ltd.

kubectl apply -f *Mindx-dl-test*.yaml

```
root@ubuntu:/home/test/yaml# kubectl apply -f Mindx-dl-test.yaml
configmap/rings-config-mindx-dls-test created
job.batch.volcano.sh/mindx-dls-test created
```

----End

3.4.2.2.5 Checking the Running Status

Procedure

Step 1 Run the following command to check the pod running status:

kubectl get pod --all-namespaces -o wide

- Example of a single-server single-device training job

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                      1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                      1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                      1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n    1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v  1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk  1/1     Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                   1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                   1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      calico-node-fh58s                   1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      coredns-6955765f44-4pdhg            1/1     Running            0          8d
192.168.243.249  ubuntu        <none>           <none>
kube-system      coredns-6955765f44-n9pg4            1/1     Running            2          8d
192.168.243.237  ubuntu        <none>           <none>
kube-system      etcd-ubuntu                         1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-controller-manager-ubuntu      1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-b5flw                    1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      kube-proxy-ttsjp                    1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-zp9xw                    1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      kube-scheduler-ubuntu               1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-0       1/1     Running            0          4m
192.168.243.198  ubuntu        <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r   1/1     Running            2          8d
192.168.243.215  ubuntu        <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l    1/1     Running            1          8d
192.168.243.238  ubuntu        <none>           <none>
volcano-system   volcano-admission-init-bbx5z        0/1     Completed          0          39s
10.174.217.96    ubuntu-96     <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q  1/1     Running            2          8d
192.168.243.211  ubuntu        <none>           <none>
```

 Copyright © Huawei Technologies Co., Ltd.

- Two training nodes running 2 x 2P distributed training jobs

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                      1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                      1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                      1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n    1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v  1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk  1/1     Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                   1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                   1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      calico-node-fh58s                   1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      coredns-6955765f44-4pdhg            1/1     Running            0          8d
192.168.243.249  ubuntu        <none>           <none>
kube-system      coredns-6955765f44-n9pg4            1/1     Running            2          8d
192.168.243.237  ubuntu        <none>           <none>
kube-system      etcd-ubuntu                         1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-controller-manager-ubuntu      1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-b5flw                    1/1     Running            1          8d
10.174.217.96    ubuntu-96     <none>           <none>
kube-system      kube-proxy-ttsjp                    1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
kube-system      kube-proxy-zp9xw                    1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      kube-scheduler-ubuntu               1/1     Running            4          8d
10.174.217.94    ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-0       1/1     Running            0          3m
192.168.243.198  ubuntu        <none>           <none>
vcjob            mindx-dls-test-default-test-1       1/1     Running            0          3m
192.168.243.199  ubuntu        <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r    1/1     Running            2          8d
192.168.243.215  ubuntu        <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l  1/1     Running            1          8d
192.168.243.238  ubuntu        <none>           <none>
volcano-system   volcano-admission-init-bbx5z        0/1     Completed          0          39s
10.174.217.96    ubuntu-96     <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q   1/1     Running            2          8d
192.168.243.211  ubuntu        <none>           <none>
```

Step 2 (Optional) Run the following commands to check the run logs:

kubectl logs -n *[Pod running namespace] [Pod name]*

Example:

kubectl logs -n vcjob *mindx-dls-test-default-test-0*

Step 3 View the NPU allocation of compute nodes.

Run the following command on the management node:

kubectl describe nodes

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- The **huawei.com/Ascend910** field of **Annotations** indicates the available NPU of the compute node.
- The **huawei.com/Ascend910** field in **Allocated resources** indicates the number of used NPUs.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    accelerator-type=card
                    beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-x86
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910: Ascend910-0
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests        Limits
 --------            --------        ------
 cpu                 37250m (19%)    37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)          0 (0%)
 huawei.com/Ascend910  1               1
Events:              <none>
```

NOTE

The **Annotations** field does not contain the Ascend 910-1 AI Processor, and the value of the **Allocated resources** field **huawei.com/Ascend910** is **1**, indicating that one processor is used for training.

- One of the two training nodes running 2 x 2P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    accelerator-type=card
```

 Copyright © Huawei Technologies Co., Ltd.

```
                    beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-x86
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910:
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  2
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests        Limits
 --------            --------        ------
 cpu                 37250m (19%)    37500m (19%)
 memory              117536Mi (15%)  119236Mi (15%)
 ephemeral-storage   0 (0%)          0 (0%)
 huawei.com/Ascend910  2             2
Events:              <none>
```

NOTE

No NPU is available in the **Annotations** field, and the value of the **huawei.com/Ascend910** field in **Allocated resources** is **2**, all NPUs are used for distributed training.

Step 4 View the NPU usage of a pod.

In this example, run the **kubectl describe pod mindx-dls-test-default-test-0 -n vcjob** command to check the running status of the pod.

NOTE

Annotations displays the NPU information.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 19:15:12 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
```

 Copyright © Huawei Technologies Co., Ltd.

```
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"4","device_ip":"192.168.21.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-1
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

- Two training nodes running 2 x 2P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 20:39:50 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"1","device_ip":"192.168.20.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-0,Ascend910-1
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

----End

3.4.2.2.6 Viewing the Running Result

Step 1 Log in to the storage server.

The following uses the local NFS whose hostname is **Ubuntu** as an example.

Step 2 Run the following command to view the output directory of the job running YAML file:

ll /data/atlas_dls/code/Benchmark/train/result/pt_resnet50

```
root@ubuntu:/home# ll /data/atlas_dls/code/Benchmark/train/result/pt_resnet50
total 28
drwxr-xr-x 7 root root 4096 Oct 22 17:16 ./
drwxr-xr-x 3 root root 4096 Oct 22 17:10 ../
drwxr-xr-x 4 root root 4096 Oct 22 17:10 training_job_20201022074421/
drwxr-xr-x 6 root root 4096 Oct 22 17:10 training_job_20201022080648/
drwxr-xr-x 6 root root 4096 Oct 22 17:10 training_job_20201022082409/
drwxr-xr-x 6 root root 4096 Oct 22 17:12 training_job_20201022091259/
drwxr-xr-x 6 root root 4096 Oct 22 17:16 training_job_20201022091619/
```

Step 3 Run the following command to access the corresponding training job:

cd /data/atlas_dls/code/Benchmark/train/result/pt_resnet50

cd training_job_20201022091619/

```
root@ubuntu:cd /data/atlas_dls/code/Benchmark/train/result/pt_resnet50/training_job_20201022091619# ll
total 324
drwxr-xr-x 6 root root   4096 Oct 22 17:16 ./
```

 Copyright © Huawei Technologies Co., Ltd.

```
drwxr-xr-x 7 root root   4096 Oct 22 17:16 ../
drwxr-xr-x 3 root root   4096 Oct 22 17:46 0/
drwxr-xr-x 3 root root   4096 Oct 22 17:17 1/
drwxr-xr-x 3 root root   4096 Oct 22 17:17 2/
drwxr-xr-x 3 root root   4096 Oct 22 17:17 3/
lrwxrwxrwx 1 root root     82 Oct 22 17:16 hw_resnet50.log -> /job/code//train/result/pt_resnet50/
training_job_20201022091619//0/hw_resnet50.log
-rw-r--r-- 1 root root 300160 Oct 22 17:46 train_4p.log
```

The **train_4p.log** file in this directory records the training precision.

```
root@ubuntu:/data/atlas_dls/public/pytorch/train/result/pt_resnet50/training_job_20201022091619# tail -f
train_4p.log
[gpu id: 0 ] Test: [2495/2502] Time  0.282 ( 0.553) Loss 7.0266 (7.0780) Acc@1   0.59 (  0.15) Acc@5   1.37
(  0.68)
[gpu id: 0 ] Test: [2496/2502] Time  0.256 ( 0.552) Loss 7.0124 (7.0780) Acc@1   0.39 (  0.15) Acc@5   0.98
(  0.68)
[gpu id: 0 ] Test: [2497/2502] Time  0.257 ( 0.552) Loss 7.0578 (7.0779) Acc@1   0.00 (  0.15) Acc@5   0.39
(  0.68)
[gpu id: 0 ] Test: [2498/2502] Time  0.265 ( 0.552) Loss 7.1090 (7.0780) Acc@1   0.00 (  0.15) Acc@5   0.98
(  0.68)
[gpu id: 0 ] Test: [2499/2502] Time  0.318 ( 0.552) Loss 7.0254 (7.0779) Acc@1   0.00 (  0.15) Acc@5   0.20
(  0.68)
[gpu id: 0 ] Test: [2500/2502] Time  0.359 ( 0.552) Loss 7.0409 (7.0779) Acc@1   0.00 (  0.15) Acc@5   0.78
(  0.68)
[gpu id: 0 ] Test: [2501/2502] Time  0.445 ( 0.552) Loss 7.0593 (7.0779) Acc@1   0.39 (  0.15) Acc@5   1.37
(  0.68)
[gpu id: 0 ] [AVG-ACC] * Acc@1 0.151 Acc@5 0.683
THPModule_npu_shutdown success.
:::ABK 1.0.0 resnet50 train success
```

----End

3.4.2.2.7 Deleting a Training Job

Run the following command in the directory where YAML is running to delete a training job:

kubectl delete -f *XXX.yaml*

Example:

kubectl delete -f *Mindx-dl-test.yaml*

```
root@ubuntu:/home/test/yaml# kubectl delete -f Mindx-dl-test.yaml
configmap "rings-config-mindx-dls-test" deleted
job.batch.volcano.sh "mindx-dls-test" deleted
```

3.4.3 MindSpore

3.4.3.1 Preparing the NPU Training Environment

After MindX DL is installed, you can use YAML to deliver a vcjob to check whether the system can run properly.

This section uses the environment described in **Table 3-5** as an example.

 Copyright © Huawei Technologies Co., Ltd.

Table 3-5 Test environment requirements

Item	Name	Version
OS	• Ubuntu 18.04 • CentOS 7.6 • EulerOS 2.8	-
Training script	ResNet	-
OS architecture	• ARM • x86	-

Creating a Training Image

Create a training image. For details, see **Creating a Container Image Using a Dockerfile (MindSpore)**. You can rename the training image, for example, **mindspore:b035**.

Preparing a Dataset

The **CIFAR-10** dataset is used only as an example.

Step 1 You need to prepare the dataset by yourself. The **CIFAR-10** dataset is recommended.

Step 2 Upload the dataset to the storage node as an administrator.

1. Go to the **/data/atlas_dls/public** directory and upload the **CIFAR-10** dataset to any directory, for example, **/data/atlas_dls/public/dataset/cifar-10**.
```
root@ubuntu:/data/atlas_dls/public/dataset/cifar-10# pwd
/data/atlas_dls/public/dataset/cifar-10
```
2. Run the following command to check the dataset size:

   du -sh
```
root@ubuntu:/data/atlas_dls/public/dataset/cifar-10# du -sh
176M
```

Step 3 Run the following command to change the owner of the dataset:

chown -R hwMindX:hwMindX /data/atlas_dls/

```
root@ubuntu:/data/atlas_dls/public/dataset/cifar-10# chown -R hwMindX:hwMindX /data/atlas_dls/
root@ubuntu:/data/atlas_dls/public/dataset/cifar-10#
```

Step 4 Run the following command to change the dataset permission:

chmod -R 750 /data/atlas_dls/

Step 5 Run the following command to check the file status:

ll /data/atlas_dls/public/*Dataset location*

Example:

ll /data/atlas_dls/public/*dataset/cifar-10*/cifar-10-batches-bin

```
root@ubuntu:~# ll /data/atlas_dls/public/dataset/cifar-10/cifar-10-batches-bin
total 180088
```

 Copyright © Huawei Technologies Co., Ltd.

```
drwxr-x--- 2 hwMindX HwHiAiUser     4096 Dec 15 16:00 ./
drwxr-x--- 4 hwMindX HwHiAiUser     4096 Dec 15 16:52 ../
-rwxr-x--- 1 hwMindX HwHiAiUser       61 Jul 11 10:23 batches.meta.txt
-rwxr-x--- 1 hwMindX HwHiAiUser 30730000 Jul 11 10:23 data_batch_1.bin
-rwxr-x--- 1 hwMindX HwHiAiUser 30730000 Jul 11 10:24 data_batch_2.bin
-rwxr-x--- 1 hwMindX HwHiAiUser 30730000 Jul 11 10:23 data_batch_3.bin
-rwxr-x--- 1 hwMindX HwHiAiUser 30730000 Jul 11 10:24 data_batch_4.bin
-rwxr-x--- 1 hwMindX HwHiAiUser 30730000 Jul 11 10:24 data_batch_5.bin
-rwxr-x--- 1 hwMindX HwHiAiUser       88 Jul 11 10:23 readme.html
-rwxr-x--- 1 hwMindX HwHiAiUser 30730000 Jul 11 10:24 test_batch.bin
```

----End

Obtaining and Modifying the Training Script

Step 1 Obtain the training script.

For details, see **Creating an NPU Training Script (MindSpore)**.

NOTE

The modified training script supports both the single-node and cluster scenarios.

Step 2 Change the script permission and owner.

1. Upload the training script to the **/data/atlas_dls/code** directory on the storage node and decompress it.
2. Run the following command to assign the execute permission recursively:

   chmod -R 750 /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# chmod -R 750 /data/atlas_dls/code/
   root@ubuntu:/data/atlas_dls/code#
   ```

3. Run the following command to change the owner:

   chown -R hwMindX:hwMindX /data/atlas_dls/code

   ```
   root@ubuntu:/data/atlas_dls/code# chown -R hwMindX:hwMindX /data/atlas_dls/code
   root@ubuntu:/data/atlas_dls/code#
   ```

4. Run the following command to view the output result:

   ll /data/atlas_dls/code

   ```
   root@ubuntu-infer:/data/atlas_dls/code# ll /data/atlas_dls/code
   total 64
   drwxr-x--- 3 hwMindX hwMindX  4096 Dec 15 15:50 ./
   drwxr-x--- 5 hwMindX hwMindX  4096 Dec 15 16:05 ../
   drwxr-x--- 3 hwMindX hwMindX  4096 Dec 15 18:55 resnet/
   ```

----End

3.4.3.2 Creating a YAML File

This section describes the YAML files in the single-node system and cluster scenarios. You can select proper YAML files based on the actual situation.

The YAML example is used in the NFS scenario. The NFS needs to be installed on the storage node. For details about how to install NFS, see **Installing the NFS**.

NOTE

If MindX DL is fully installed in online or offline mode, the NFS can be automatically installed.

 Copyright © Huawei Technologies Co., Ltd.

Single-Node Scenario

Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *XXX*.**yaml**

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

NOTICE

Delete # when using the file.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob      # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910
data:
  hccl.json: |
    {
        "status":"initializing"
    } #This line is automatically generated by HCCL-Controller. Keep it unchanged.
---
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: mindx-dls-test               # The value must be consistent with the name of ConfigMap.
  namespace: vcjob                    # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The HCCL-Controller distinguishes Ascend 910 and other processors based on this label.
spec:
  minAvailable: 1
  schedulerName: volcano             #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 1                 #The value of replicas is 1 in a single-node scenario. The value is N and the number of NPUs in the requests field is 8 in an N-node distributed scenario.
    template:
      metadata:
        labels:
          app: mindspore
          ring-controller.atlas: ascend-910
      spec:
        containers:
        - image: mindspore:b035             # Training framework image, which can be modified.
```

 Copyright © Huawei Technologies Co., Ltd.

```
        imagePullPolicy: IfNotPresent
        name: mindspore
        command:
        - "/bin/bash"
        - "-c"
        - "cd /job/code/resnet/scripts; chmod +x train_start.sh; ./train_start.sh" #Commands for running the training script. Ensure that the involved commands and paths exist on Docker.
        #args: [ "while true; do sleep 30000; done;" ]     #Comment out the preceding line and enable this line. You can manually run the training script in the container to facilitate debugging.
        resources:
          requests:
            huawei.com/Ascend910: 1            # Number of required NPUs. The maximum value is 8. You can add lines below to configure resources such as memory and CPU.
          limits:
            huawei.com/Ascend910: 1            # The value must be consistent with that in requests.
        volumeMounts:
        - name: ascend-910-config
          mountPath: /user/serverid/devindex/config
        - name: code
          mountPath: /job/code/
        - name: data
          mountPath: /job/data
        - name: ascend-driver
          mountPath: /usr/local/Ascend/driver
        - name: ascend-add-ons
          mountPath: /usr/local/Ascend/add-ons
        - name: localtime
          mountPath: /etc/localtime
      nodeSelector:
        host-arch: huawei-arm    # Configure the label based on the actual job.
      volumes:
      - name: ascend-910-config
        configMap:
          name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
      - name: code
        nfs:
          server: 127.0.0.1                # IP address of the NFS server. In this example, the shared path is /data/atlas_dls/.
          path: "/data/atlas_dls/code"     # Configure the path of the training script.
      - name: data
        nfs:
          server: 127.0.0.1
          path: "/data/atlas_dls/public/dataset/cifar-10"    # Configure the path of the training set.
      - name: ascend-driver
        hostPath:
          path: /usr/local/Ascend/driver     #Configure the NPU driver and mount it to Docker.
      - name: ascend-add-ons
        hostPath:
          path: /usr/local/Ascend/add-ons   # Configure the add-ons driver of the NPU and mount it to Docker.
      - name: localtime
        hostPath:
          path: /etc/localtime   # Configure the Docker time.
      env:
      - name: mindx-dls-test  # The value must be consistent with the value of JobName.
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
      restartPolicy: OnFailure
```

Cluster Scenario

The following uses two training nodes running 2 x 8P distributed training jobs as an example. Run the following command on the management node to create the YAML file for training jobs and add the content in this section to the YAML file.

vim *File name*.**yaml**

 Copyright © Huawei Technologies Co., Ltd.

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

NOTICE

Delete # when using the file.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rings-config-mindx-dls-test     # The value of JobName must be the same as the name attribute of the following job. The prefix rings-config- cannot be modified.
  namespace: vcjob      # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910
data:
  hccl.json: |
    {
        "status":"initializing"
    } #This line is automatically generated by HCCL-Controller. Keep it unchanged.
---
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: mindx-dls-test              # The value must be consistent with the name of ConfigMap.
  namespace: vcjob                   # Select a proper namespace based on the site requirements. (The namespaces of ConfigMap and Job must be the same. In addition, if the tjm component of MindX-add exists, the vcjob namespace cannot be used.)
  labels:
    ring-controller.atlas: ascend-910   #The HCCL-Controller distinguishes Ascend 910 and other processors based on this label.
spec:
  minAvailable: 1
  schedulerName: volcano            #Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    ssh: []
    env: []
    svc: []
  maxRetry: 3
  queue: default
  tasks:
  - name: "default-test"
    replicas: 2                #The value of replicas is 1 in a single-node scenario. The value is N and the number of NPUs in the requests field is 8 in an N-node distributed scenario.
    template:
      metadata:
        labels:
          app: mindspore
          ring-controller.atlas: ascend-910
      spec:
        containers:
        - image: mindspore:b035             # Training framework image, which can be modified.
          imagePullPolicy: IfNotPresent
          name: mindspore
          command:
          - "/bin/bash"
          - "-c"
          - "cd /job/code/resnet/scripts; chmod +x train_start.sh; ./train_start.sh" #Commands for running the training script. Ensure that the involved commands and paths exist on Docker.
          #args: [ "while true; do sleep 30000; done;"  ]     #Comment out the preceding line and enable this
```

 Copyright © Huawei Technologies Co., Ltd.

```
line. You can manually run the training script in the container to facilitate debugging.
          resources:
            requests:
              huawei.com/Ascend910: 8            # Number of required NPUs. The maximum value is 8. You can add lines below to configure resources such as memory and CPU.
            limits:
              huawei.com/Ascend910: 8          # The value must be consistent with that in requests.
          volumeMounts:
          - name: ascend-910-config
            mountPath: /user/serverid/devindex/config
          - name: code
            mountPath: /job/code/
          - name: data
            mountPath: /job/data
          - name: ascend-driver
            mountPath: /usr/local/Ascend/driver
          - name: ascend-add-ons
            mountPath: /usr/local/Ascend/add-ons
          - name: localtime
            mountPath: /etc/localtime
        nodeSelector:
          host-arch: huawei-arm    # Configure the label based on the actual job.
        volumes:
        - name: ascend-910-config
          configMap:
           name: rings-config-mindx-dls-test   # Correspond to the ConfigMap name above.
        - name: code
          nfs:
            server: 127.0.0.1                 # IP address of the NFS server. In this example, the shared path is /data/atlas_dls/.
            path: "/data/atlas_dls/code"     # Configure the path of the training script.
        - name: data
          nfs:
            server: 127.0.0.1
            path: "/data/atlas_dls/public/dataset/cifar-10"    # Configure the path of the training set.
          hostPath:
            path: /usr/local/Ascend/driver     #Configure the NPU driver and mount it to Docker.
        - name: ascend-add-ons
          hostPath:
            path: /usr/local/Ascend/add-ons   #Configure the add-ons driver of the NPU and mount it to Docker.
        - name: localtime
          hostPath:
            path: /etc/localtime   # Configure the Docker time.
        env:
        - name: mindx-dls-test  # The value must be consistent with the value of JobName.
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        restartPolicy: OnFailure
```

3.4.3.3 Preparing for Running a Training Job

Procedure

Step 1 Run the following command to modify the resources in the YAML file.

vim *XXX*.yaml

NOTE

XXX: YAML name generated in **Creating a YAML File**.

Example:

- Single-server single-device training job

 Copyright © Huawei Technologies Co., Ltd.

vim *Mindx-dl-test*.**yaml**

Modify the items based on the resource requirements. For details about how to modify other items, see **Creating a YAML File**.

```
...
    resources:
        requests:
          huawei.com/Ascend910: 1         #Number of required NPUs. The maximum value is 8. Resources such as the MEN and CPU can be configured.
        limits:
          huawei.com/Ascend910: 1         #Number of required NPUs. The maximum value is 8. Resources such as the MEN and CPU can be configured.
...
```

NOTE

- For a single-server single-device scenario, the value of **huawei.com/Ascend910** is **1**.
- For a single-server multi-device scenario, the value of **huawei.com/Ascend910** is **2**, **4**, or **8**.

- Two training nodes running 2 x 8P distributed training jobs

  vim *Mindx-dl-test*.**yaml**

  Modify the items based on the resource requirements. For details about how to modify other items, see **Creating a YAML File**.

```
...
- name: "default-test"
   replicas: 2               # The value of replicas is 1 in a single-node scenario and N in an N-node scenario. The number of NPUs in the requests field is 8 in an N-node scenario.
   template:
     metadata:
...
        resources:
          requests:
            huawei.com/Ascend910: 8         # Number of required NPUs. The maximum value is 8. Resources such as men and cpu can be configured.
          limits:
            huawei.com/Ascend910: 8         # The value must be consistent with that in requests.
...
```

Step 2 Modify the training script.

Go to the **/data/atlas_dls/code/resnet/src** directory and modify the code based on the network structure and dataset.

In this example, the network structure and data set are **resnet50** and **cifar-10**, respectively. Therefore, the following content is modified: (Only 10 epochs are run.)

```
...
# config for resent50, cifar10
config1 = ed({
    "class_num": 10,
    "batch_size": 32,
    "loss_scale": 1024,
    "momentum": 0.9,
    "weight_decay": 1e-4,
    "epoch_size": 10,
    "pretrain_epoch_size": 0,
    "save_checkpoint": True,
    "save_checkpoint_epochs": 5,
    "keep_checkpoint_max": 10,
    "save_checkpoint_path": "./",
    "warmup_epochs": 5,
    "lr_decay_mode": "poly",
```

 Copyright © Huawei Technologies Co., Ltd.

```
    "lr_init": 0.01,
    "lr_end": 0.00001,
    "lr_max": 0.1
})
...
```

----End

3.4.3.4 Delivering a Training Job

Procedure

Step 1 Run the following command to create a namespace for the training job:

kubectl create namespace vcjob

Step 2 Run the following command on the management node to deliver training jobs using YAML:

kubectl apply -f *XXX*.yaml

Example:

kubectl apply -f *Mindx-dl-test*.yaml

```
root@ubuntu:/home/test/yaml# kubectl apply -f Mindx-dl-test.yaml
configmap/rings-config-mindx-dls-test created
job.batch.volcano.sh/mindx-dls-test created
```

----End

3.4.3.5 Checking the Running Status

Procedure

Step 1 Run the following command to check the pod running status:

kubectl get pod --all-namespaces -o wide

- Example of a single-server single-device training job

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                    READY   STATUS             RESTARTS   AGE
IP               NODE          NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                          1/1     Running            33         8d
192.168.243.252  ubuntu        <none>           <none>
cadvisor         cadvisor-hgbw8                          1/1     Running            0          26h
192.168.207.48   ubuntu-96     <none>           <none>
cadvisor         cadvisor-shwb4                          1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer  <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n        1/1     Running            0          8d
192.168.243.199  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218  ubuntu        <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49   ubuntu-96     <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v 1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer  <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk 1/1    Running            1          8d
192.168.243.198  ubuntu        <none>           <none>
kube-system      calico-node-bkbvl                       1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer  <none>           <none>
kube-system      calico-node-bzd7q                       1/1     Running            3          8d
10.174.217.94    ubuntu        <none>           <none>
```

 Copyright © Huawei Technologies Co., Ltd.

```
kube-system      calico-node-fh58s                       1/1     Running            1          8d
10.174.217.96    ubuntu-96      <none>           <none>
kube-system      coredns-6955765f44-4pdhg                1/1     Running            0          8d
192.168.243.249  ubuntu         <none>           <none>
kube-system      coredns-6955765f44-n9pg4                1/1     Running            2          8d
192.168.243.237  ubuntu         <none>           <none>
kube-system      etcd-ubuntu                             1/1     Running            3          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      kube-controller-manager-ubuntu          1/1     Running            4          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      kube-proxy-b5flw                        1/1     Running            1          8d
10.174.217.96    ubuntu-96      <none>           <none>
kube-system      kube-proxy-ttsjp                        1/1     Running            3          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      kube-proxy-zp9xw                        1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer   <none>           <none>
kube-system      kube-scheduler-ubuntu                   1/1     Running            4          8d
10.174.217.94    ubuntu         <none>           <none>
vcjob            mindx-dls-test-default-test-0           1/1     Running            0          5m
192.168.243.198  ubuntu         <none>           <none>
volcano-system   volcano-admission-5bcb6d799-rgk5r       1/1     Running            2          8d
192.168.243.215  ubuntu         <none>           <none>
volcano-system   volcano-controllers-7d6d465877-nnf7l    1/1     Running            1          8d
192.168.243.238  ubuntu         <none>           <none>
volcano-system   volcano-admission-init-bbx5z            0/1     Completed          0          39s
10.174.217.96    ubuntu-96      <none>           <none>
volcano-system   volcano-scheduler-67f89949b4-ncs8q      1/1     Running            2          8d
192.168.243.211  ubuntu         <none>           <none>
```

- Example of executing a 2 x 8P distributed training task on two training nodes

```
root@ubuntu-96:~# kubectl get pod --all-namespaces -o wide
NAMESPACE        NAME                                    READY   STATUS             RESTARTS   AGE
IP               NODE           NOMINATED NODE   READINESS GATES
cadvisor         cadvisor-8x86g                          1/1     Running            33         8d
192.168.243.252  ubuntu         <none>           <none>
cadvisor         cadvisor-hgbw8                          1/1     Running            0          26h
192.168.207.48   ubuntu-96      <none>           <none>
cadvisor         cadvisor-shwb4                          1/1     Running            0          6m46s
192.168.240.65   ubuntu-infer   <none>           <none>
default          hccl-controller-688c7cb8c6-4b88n        1/1     Running            0          8d
192.168.243.199  ubuntu         <none>           <none>
kube-system      ascend-device-plugin-daemonset-8f2dx    1/1     Running            2          8d
192.168.243.218  ubuntu         <none>           <none>
kube-system      ascend-device-plugin-daemonset-f2jk9    1/1     Running            1          8d
192.168.207.49   ubuntu-96      <none>           <none>
kube-system      ascend310-device-plugin-daemonset-fls4v 1/1     Running            0          4m15s
192.168.240.66   ubuntu-infer   <none>           <none>
kube-system      calico-kube-controllers-8464785d6b-bj4pk 1/1    Running            1          8d
192.168.243.198  ubuntu         <none>           <none>
kube-system      calico-node-bkbvl                       1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer   <none>           <none>
kube-system      calico-node-bzd7q                       1/1     Running            3          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      calico-node-fh58s                       1/1     Running            1          8d
10.174.217.96    ubuntu-96      <none>           <none>
kube-system      coredns-6955765f44-4pdhg                1/1     Running            0          8d
192.168.243.249  ubuntu         <none>           <none>
kube-system      coredns-6955765f44-n9pg4                1/1     Running            2          8d
192.168.243.237  ubuntu         <none>           <none>
kube-system      etcd-ubuntu                             1/1     Running            3          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      kube-controller-manager-ubuntu          1/1     Running            4          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      kube-proxy-b5flw                        1/1     Running            1          8d
10.174.217.96    ubuntu-96      <none>           <none>
kube-system      kube-proxy-ttsjp                        1/1     Running            3          8d
10.174.217.94    ubuntu         <none>           <none>
kube-system      kube-proxy-zp9xw                        1/1     Running            0          8m16s
10.174.216.214   ubuntu-infer   <none>           <none>
kube-system      kube-scheduler-ubuntu                   1/1     Running            4          8d
```

 Copyright © Huawei Technologies Co., Ltd.

```
10.174.217.94     ubuntu        <none>          <none>
vcjob             mindx-dls-test-default-test-0        1/1     Running           0          10m
192.168.243.134   ubuntu        <none>          <none>
vcjob             mindx-dls-test-default-test-1        1/1     Running           0          10m
192.168.243.135   ubuntu        <none>          <none>
volcano-system    volcano-admission-5bcb6d799-rgk5r    1/1     Running           2          8d
192.168.243.215   ubuntu        <none>          <none>
volcano-system    volcano-controllers-7d6d465877-nnf7l 1/1     Running           1          8d
192.168.243.238   ubuntu        <none>          <none>
volcano-system    volcano-admission-init-bbx5z         0/1     Completed         0          39s
10.174.217.96     ubuntu-96     <none>          <none>
volcano-system    volcano-scheduler-67f89949b4-ncs8q   1/1     Running           2          8d
192.168.243.211   ubuntu        <none>          <none>
```

Step 2 View the NPU allocation of compute nodes.

Run the following command on the management node:

kubectl describe nodes

NOTE

- The **huawei.com/Ascend910** field of **Annotations** indicates the available NPU of the compute node.
- The **huawei.com/Ascend910** field in **Allocated resources** indicates the number of used NPUs.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910:
Ascend910-0,Ascend910-1,Ascend910-2,Ascend910-3,Ascend910-5,Ascend910-6,Ascend910-7
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
  cpu:                192
  ephemeral-storage:  1537233808Ki
  huawei.com/Ascend910:  8
  hugepages-2Mi:      0
  memory:             792307468Ki
  pods:               110
Allocatable:
  cpu:                192
  ephemeral-storage:  1416714675108
  huawei.com/Ascend910:  8
  hugepages-2Mi:      0
  memory:             792205068Ki
  pods:               110
...
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
```

 Copyright © Huawei Technologies Co., Ltd.

```
 Resource            Requests       Limits
 --------            --------       ------
 cpu                 37250m (19%)   37500m (19%)
 memory              117536Mi (15%) 119236Mi (15%)
 ephemeral-storage   0 (0%)         0 (0%)
 huawei.com/Ascend910  1            1
Events:              <none>
```

NOTE

The **Annotations** field does not contain the Ascend 910-4 AI Processor, and the value of the **Allocated resources** field **huawei.com/Ascend910** is **1**, indicating that one processor is used for training.

- One of the two training nodes running 2 x 8P distributed training jobs

```
root@ubuntu:/home/test/yaml# kubectl describe nodes
Name:               ubuntu
Roles:              master,worker
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    node-role.kubernetes.io/worker=worker
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910:
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: XXX.XXX.XXX.XXX/23
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.243.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Mon, 28 Sep 2020 14:36:54 +0800
...
Capacity:
 cpu:                192
 ephemeral-storage:  1537233808Ki
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792307468Ki
 pods:               110
Allocatable:
 cpu:                192
 ephemeral-storage:  1416714675108
 huawei.com/Ascend910:  8
 hugepages-2Mi:      0
 memory:             792205068Ki
 pods:               110
...
Allocated resources:
 (Total limits may be over 100 percent, i.e., overcommitted.)
 Resource            Requests       Limits
 --------            --------       ------
 cpu                 37250m (19%)   37500m (19%)
 memory              117536Mi (15%) 119236Mi (15%)
 ephemeral-storage   0 (0%)         0 (0%)
 huawei.com/Ascend910  8            8
Events:              <none>
```

NOTE

No NPU is available in the **Annotations** field, and the value of the **huawei.com/Ascend910** field in **Allocated resources** is **8**, all NPUs are used for distributed training.

Step 3 View the NPU usage of a pod.

 Copyright © Huawei Technologies Co., Ltd.

In this example, run the **kubectl describe pod mindx-dls-test-default-test-0 -n vcjob** command to check the running status of the pod.

NOTE

Annotations displays the NPU information.

- Example of a single-server single-device training job

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 15:38:22 +0800
Labels:       app=mindspore
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"4","device_ip":"192.168.20.102"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910: Ascend910-4
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

- Example of executing a 2 x 8P distributed training task on two training nodes

```
root@ubuntu:/home/test/yaml# kubectl describe pod mindx-dls-test-default-test-0 -n vcjob
Name:         mindx-dls-test-default-test-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu/XXX.XXX.XXX.XXX
Start Time:   Wed, 30 Sep 2020 15:38:22 +0800
Labels:       app=tf
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mindx-dls-test
              volcano.sh/job-namespace=vcjob
Annotations:  atlas.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"XXX.XXX.XXX.XXX","devices":
[{"device_id":"0","device_ip":"192.168.20.100"}...
              cni.projectcalico.org/podIP: 192.168.243.195/32
              cni.projectcalico.org/podIPs: 192.168.243.195/32
              huawei.com/Ascend910:
Ascend910-0,Ascend910-1,Ascend910-2,Ascend910-3,Ascend910-4,Ascend910-5,Ascend910-6,Asce
nd910-7
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mindx-dls-test
              volcano.sh/job-name: mindx-dls-test
              volcano.sh/job-version: 0
              volcano.sh/task-spec: default-test
Status:       Running
```

----End

3.4.3.6 Viewing the Running Result

Step 1 Log in to the storage server.

The following uses the local NFS whose hostname is **Ubuntu** as an example.

Step 2 Run the following command to go to the training output directory specified during job running:

 Copyright © Huawei Technologies Co., Ltd.

ll /data/atlas_dls/code/scripts/train

```
root@ubuntu:/home# ll /data/atlas_dls/code/scripts/train
total 16896
drwxr-x--- 2 HwHiAiUser HwHiAiUser      4096 Dec 15 16:06 ./
drwxr-x--- 4 hwMindX    HwHiAiUser      4096 Dec 15 15:26 ../
-rwxr-x--- 1 HwHiAiUser HwHiAiUser 188537157 Dec 15 19:59 resnet-10_1875.ckpt
-rwxr-x--- 1 HwHiAiUser HwHiAiUser 188537157 Dec 15 19:56 resnet-5_1875.ckpt
-rwxr-x--- 1 HwHiAiUser HwHiAiUser    938486 Dec 15 19:54 resnet-graph.meta
```

----End

3.4.3.7 Deleting a Training Job

Run the following command in the directory where YAML is running to delete a training job:

kubectl delete -f *XXX.yaml*

Example:

kubectl delete -f *Mindx-dl-test.yaml*

```
root@ubuntu:/home/test/yaml# kubectl delete -f Mindx-dl-test.yaml
configmap "rings-config-mindx-dls-test" deleted
job.batch.volcano.sh "mindx-dls-test" deleted
```

3.4.4 Inference Job

3.4.4.1 Preparing the NPU Inference Environment

Environment Dependencies

This section uses the environment described in **Table 3-6** as an example.

Table 3-6 Test environment requirements

Item	Name	Version
OS	• Ubuntu 18.04 • CentOS 7.6	-
RUN NPU package	RUN inference package	For details, see the **version mapping**.
Inference service code	*XXX*.tar **NOTE** *XXX*: name of the inference service code package. You need to prepare the package based on the inference service. This document uses **dvpp_resnet.tar** as an example.	-

 Copyright © Huawei Technologies Co., Ltd.

Item	Name	Version
OS architecture	• ARM • x86	-

Creating an Inference Image

Create an inference image. For details, see **Creating an Inference Image Using a Dockerfile**. Rename the inference image, for example, **ubuntu-infer:v1**.

3.4.4.2 Creating a YAML File

Run the following command on the management node to create the YAML file for inference jobs and add the content in this section to the ***XXX*.yaml** file.

vim *XXX*.**yaml**

The following uses **Mindx-dl-test.yaml** as an example:

vim *Mindx-dl-test*.**yaml**

```
apiVersion: batch/v1
kind: Job
metadata:
  name: resnetinfer1-1
  #namespace: kube-system
spec:
    template:
      spec:
        nodeSelector:
          accelerator: huawei-Ascend310          #Select an inference processor node.
        containers:
        - image: ubuntu-infer:v1                 #Inference image name
          imagePullPolicy: IfNotPresent
          name: resnet50infer
          resources:
            requests:
              huawei.com/Ascend310: 1            #Number of Ascend 310 AI Processors for inference.
            limits:
              huawei.com/Ascend310: 1            #The value must be the same as the number in requests.
          volumeMounts:
          - name: ascend-dirver
            mountPath: /usr/local/Ascend/driver   #Driver path.
          - name: slog
            mountPath: /var/log/npu/conf/slog/    #Log path.
          - name: localtime                       #The container time must be the same as the host time.
            mountPath: /etc/localtime
        volumes:
        - name: ascend-dirver
          hostPath:
            path: /usr/local/Ascend/driver
        - name: slog
          hostPath:
            path: /var/log/npu/conf/slog/
        - name: localtime
          hostPath:
            path: /etc/localtime
        restartPolicy: OnFailure
```

 Copyright © Huawei Technologies Co., Ltd.

3.4.4.3 Delivering Inference Jobs

Run the following command on the management node to deliver inference jobs using YAML:

kubectl apply -f *XXX*.yaml

Example:

kubectl apply -f *Mindx-dl-test*.yaml

```
root@ubuntu:/home/test/yaml# kubectl apply -f Mindx-dl-test.yaml
job.batch/resnetinfer1-2 created
```

3.4.4.4 Checking the Running Status

Procedure

Step 1 Run the following command to check the pod running status:

kubectl get pod --all-namespaces

```
NAMESPACE        NAME                                READY   STATUS    RESTARTS   AGE
cadvisor         cadvisor-r6qkq                      1/1     Running   0          3d21h
default          dls-cec-deploy-6544466488-fpgsj     1/1     Running   1          3d21h
default          dls-mms-deploy-6456b45b5f-z7nsd     1/1     Running   2          3d21h
default          hccl-controller-688c7cb8c6-r4zkm    1/1     Running   0          3d21h
default          resnetinfer1-2-scpr5                1/1     Running   0          8s
default          tjm-68dcc744fc-vm4zg                1/1     Running   2          3d21h
kube-system      ascend-device-plugin2-daemonset-8g2hb   1/1     Running   1          4d16h
```

Step 2 Run the following command on the management node to view the inference result:

kubectl logs -f resnetinfer1-2-scpr5

----End

3.4.4.5 Deleting an Inference Job

Run the following command in the directory where YAML is run to delete an inference job:

kubectl delete -f *XXX*.yaml

Example:

kubectl delete -f *Mindx-dl-test*.yaml

```
root@ubuntu:/home/test/yaml# kubectl delete -f Mindx-dl-test.yaml
job "resnetinfer1-1" deleted
```

3.5 Log Collection

MindX DL provides the log collection function. You can use scripts to obtain log files of the NPU driver, Docker, Kubernetes components (Kubelet, Kubectl, and Kubeadm), and MindX DL. Logs of a single node or a cluster can be collected. To collect cluster logs, you need to run Ansible commands on the management node to distribute collection scripts and collect logs of each node in the cluster.

 Copyright © Huawei Technologies Co., Ltd.

NOTE

- The log collection paths can be configured in the script.
- To use this script to collect logs, you need to install Python 2.7 or Python 3.

NOTICE

- Ensure that the server has sufficient space for collecting logs. Otherwise, the system may be abnormal.
- The collected information is not checked during log collection, and insecure information (such as sensitive information and confidential information) may be collected. You are advised to delete the collected information in a timely manner after using it.

Prerequisites

To run the log collection tool, the following conditions must be met:

1. Obtain all files in the **collect_log** directory from **Gitee Code Repository** and upload them to the **/home/collect_log** directory.
2. The dos2unix tool has been installed.

   NOTE

   - For Ubuntu, run the following command to install dos2unix:
     apt install -y dos2unix
   - For CentOS, run the following command to install dos2unix:
     yum install -y dos2unix
3. Python 3.7.5 and Ansible have been installed on the management node. For details about how to check the installation, see **Checking the Python and Ansible Versions**.

Collecting Logs of a Single Node

Step 1 Run the following command to switch to the ***/home*/collect_log** directory:

cd /*home*/collect_log

Step 2 Run the following commands to change the log collection paths:

chmod 600 collect_log.py

vim collect_log.py

NOTE

The format of the log collection path is **(os.path.join(base, "*Archive path in the compressed package*"),"*Path of logs to be collected in the system*")**. You can add or delete log collection paths based on the site requirements.

```
...
def get_log_path_src_and_dst(base):
    # compress all files from source folders into destination folders
    dst_src_paths = \
        [(os.path.join(base, "volcano-scheduler"),
```

```
        "/var/log/atlas_dls/volcano-scheduler"),
       (os.path.join(base, "volcano-admission"),
        "/var/log/atlas_dls/volcano-admission"),
       (os.path.join(base, "volcano-controller"),
        "/var/log/atlas_dls/volcano-controller"),
       (os.path.join(base, "hccl-controller"),
        "/var/log/atlas_dls/hccl-controller"),
       (os.path.join(base, "devicePlugin"), "/var/log/devicePlugin"),
       (os.path.join(base, "cadvisor"), "/var/log/cadvisor"),
       (os.path.join(base, "npuSlog"), "/var/log/npu/slog/host-0/"),
       (os.path.join(base, "apigw"), "/var/log/atlas_dls/apigw"),
       (os.path.join(base, "cec"), "/var/log/atlas_dls/cec"),
       (os.path.join(base, "dms"), "/var/log/atlas_dls/dms"),
       (os.path.join(base, "mms"), "/var/log/atlas_dls/mms"),
       (os.path.join(base, "mysql"), "/var/log/atlas_dls/mysql"),
       (os.path.join(base, "nginx"), "/var/log/atlas_dls/nginx"),
       (os.path.join(base, "tjm"), "/var/log/atlas_dls/tjm")]
   return dst_src_paths
...
```

Step 3 Run the following commands to collect logs:

dos2unix *

chmod 500 collectLog.py

python collectLog.py

Information similar to the following is displayed:

```
root@ubuntu560:/home/collect_log# python collect_log.py
begin to collect log files
creating dst folder:MindX_Report_2020_12_07_16_10_55/LogCollect
compress files:/var/log/atlas_dls/volcano-scheduler/volcano-scheduler.log
...
compress files:/var/log/npu/slog/host-0/host-0_20201206190134713.log
warning: /var/log/atlas_dls/apigw not exists
warning: /var/log/atlas_dls/cec not exists
warning: /var/log/atlas_dls/dms not exists
warning: /var/log/atlas_dls/mms not exists
warning: /var/log/atlas_dls/mysql not exists
warning: /var/log/atlas_dls/nginx not exists
warning: /var/log/atlas_dls/tjm not exists
compress files:/var/log/messages-20201122
compress files:/var/log/messages-20201129
compress files:/var/log/messages-20201206
compress files:/var/log/messages
create tar file:MindX_Report_2020_12_07_16_10_55-centos-127-LogCollect.tar.gz, from all compressed files
adding to tar: MindX_Report_2020_12_07_16_10_55/LogCollect

delete temp folderMindX_Report_2020_12_07_16_10_55
collect log files finish
```

NOTE

- In this example, the **collect_log.py** file has been copied to the **/home/collect_log** directory.
- In the command output, the log collection path is **/home/collect_log/MindX_Report_2020_12_02_19_58_10-ubuntu560-LogCollect.gz**, which is in the path where the **collect_log.py** file is stored.

----End

Collecting Logs of a Cluster

Step 1 Configure Ansible host information.

 Copyright © Huawei Technologies Co., Ltd.

For details, see **Configuring Ansible Host Information**.

An example is as follows:

```
[all:vars]
# Master IP
master_ip=10.10.56.78

[master]
ubuntu-example ansible_host=10.10.56.78 ansible_ssh_user="root" ansible_ssh_pass="ad34#$"

[training_node]
ubuntu-example2 ansible_host=10.10.56.79 ansible_ssh_user="root" ansible_ssh_pass="ad34#$"

[inference_node]


[workers:children]
training_node
inference_node
```

NOTE

- The configuration file **/etc/ansible/hosts** of the cluster logs collected using Ansible must contain at least the preceding content.
- If Ansible is used to install and deploy a cluster, you can directly use the **/etc/ansible/hosts** file configured during installation and deployment to collect cluster logs. You do not need to modify the file.
- Some groups do not have servers and can be left empty, such as **[inference_node]** in the example.
- The content under **[workers:children]** cannot be modified.

Step 2 Run the following command to switch to the **/*home*/collect_log** directory:

cd /*home*/collect_log

The directory structure is as follows:

```
/home/collect_log
├── collect_log.py
└── collect_log.yaml
```

Step 3 Run the following commands to collect logs:

dos2unix *

ansible-playbook -vv collect_log.yaml

NOTE

You are advised to run the following commands to set the permission on the **collect_log.yaml** file to **400** and the permission on the **collect_log.sh** file to **500**:

chmod 400 collect_log.yaml

chmod 500 collect_log.py

If the following message is displayed, the operation is successful.

```
root@ubuntu560:/home/collect_log# ansible-playbook -vv collect_log.yaml
ansible-playbook 2.9.7
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/python3.7.5/lib/python3.7/site-packages/ansible-2.9.7-
py3.7.egg/ansible
  executable location = /usr/local/bin/ansible-playbook
  python version = 3.7.5 (default, Sep 22 2020, 17:38:26) [GCC 7.5.0]
```

 Copyright © Huawei Technologies Co., Ltd.

```
Using /etc/ansible/ansible.cfg as config file

PLAYBOOK: collect_log.yaml
*************************************************************************************************************************************************************
***********************************
2 plays in collect_log.yaml

PLAY [master]
*************************************************************************************************************************************************************
***********************************************

...
*************************************************************************************************************************************************************
***********************************
task path: /home/collect_log/collect_log.yaml:102
changed: [ubuntu-11] => {"changed": true, "cmd": "echo \"Finished! The check report is stored in /home/
collect_log/MindXReport/ on the master node.\"", "delta": "0:00:00.003850", "end": "2020-12-02
12:26:36.867520", "rc": 0, "start": "2020-12-02 12:26:36.863670", "stderr": "", "stderr_lines": [], "stdout":
"Finished! The check report is stored in /home/collect_log/MindXReport/ on the master node.",
"stdout_lines": ["Finished! The check report is stored in /home/collect_log/MindXReport/ on the master
node."]}
META: ran handlers
META: ran handlers

PLAY RECAP
*************************************************************************************************************************************************************
**************************************************
ubuntu-11                  : ok=7    changed=4    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0
ubuntu560                  : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

NOTE

- In this example, the **collect_log.yaml** and **collect_log.py** file has been copied to the **/home/collect_log** directory.
- The independent report of each node is stored in the **MindXReport** directory where the **collect_log.yaml** file of the management node is stored. In this example, the directory is **/home/collect_log/MindXReport/**.

----End

3.6 Common Operations

3.6.1 Creating an NPU Training Script (MindSpore)

Procedure

Step 1 Log in to **ModelZoo**, download the ResNet-50 training code package of the MindSpore framework, and decompress the package to the local host.

Step 2 Create the **hccl2ranktable.py**, **train_start.sh**, and **main.sh** files in the **resnet/scripts** directory. The following is an example of the directory structure:

```
root@ubuntu:/data/atlas_dls/code/resnet/scripts/#
scripts/
├── hccl2ranktable.py
├── main.sh
├── run_distribute_train_gpu.sh
├── run_distribute_train.sh
├── run_eval_gpu.sh
├── run_eval.sh
├── run_gpu_resnet_benchmark.sh
├── run_parameter_server_train_gpu.sh
```

 Copyright © Huawei Technologies Co., Ltd.

```
├── run_parameter_server_train.sh
├── run_standalone_train_gpu.sh
├── run_standalone_train.sh
└── train_start.sh
```

Step 3 Perform the following steps to prepare the files:

- **hccl2ranktable.py** contains the conversion code of the old and new HCCL configuration files.
- **train_start.sh** and **main.sh** are training scripts.

1. Run the following command to create the **hccl2ranktable.py** file:

   vim hccl2ranktable.py

   Add the following content to the file and run the **:wq** command to save the file:

```
# Copyright 2020 Huawei Technologies Co., Ltd
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# ============================================================================
"""convert new format hccl config to old config script"""

import json
import sys


def creat_rank_table_file(hccl_path, rank_table_file_path):
    hccn_table = {'version': '1.0',
                  'server_count': '1'}

    server_list = []
    with open(hccl_path, 'r', encoding='UTF-8') as f:
        data = json.load(f)

    status = data.get('status')
    instance_list = data.get('group_list')[0].get('instance_list')
    server_count = data.get('group_list')[0].get('instance_count')
    rank_id = 0
    for instance in instance_list:
        del instance['pod_name']
        devices_list = instance.get('devices')
        for dev in devices_list:
            dev['rank_id'] = str(rank_id)
            rank_id += 1
        instance['device'] = devices_list
        del instance['devices']
        instance['host_nic_ip'] = 'reserve'
        server_list.append(instance)
    hccn_table['server_count'] = server_count
    hccn_table['server_list'] = server_list
    hccn_table['status'] = status

    with open(rank_table_file_path, 'w') as convert_fp:
        json.dump(hccn_table, convert_fp, indent=4)
    sys.stdout.flush()


if __name__ == "__main__":
```

```
    if len(sys.argv) != 3:
        print("Parameter is invalid, exit!!!")
        exit(1)
    hccl_path = sys.argv[1]
    rank_table_file_path = sys.argv[2]
    creat_rank_table_file(hccl_path, rank_table_file_path)
```

2. Run the following command to create the **train_start.sh** file:

   vim train_start.sh

   Add the following content to the file and run the **:wq** command to save the file:

```
#!/bin/bash
# set -x

# rank_table_file generated by HCCL-Controller
export RANK_TABLE_FILE=/user/serverid/devindex/config/hccl.json

# Parse rank_table_file.
function get_json_value()
{
  local json=$1
  local key=$2

  if [[ -z "$3" ]]; then
    local num=1
  else
    local num=$3
  fi

  local value=$(cat "${json}" | awk -F"[,:}]" '{for(i=1;i<=NF;i++){if($i~/'${key}'\042/){print $(i+1)}}}' |
                tr -d '"' | sed -n ${num}p)
  echo ${value}
}

# Check the status of rank_table_file.
function check_hccl_status()
{
    local retry_times=60
    local retry_interval=5
    for (( n=1;n<=$retry_times;n++ ));do
    {
        local status=$(get_json_value ${RANK_TABLE_FILE} status)
        if [[ "$status" != "completed" ]]; then
            echo "hccl status is not completed, wait 5s and retry." | tee -a hccl.log
            sleep $retry_interval
            continue
        else
            echo 0
            return
        fi
    }
    done
    echo 1
}

ret=$(check_hccl_status)
if [[ "${ret}" == "1" ]]; then
    echo "wait hccl status timeout, train job failed." | tee -a hccl.log
    exit 1
fi

sleep 1


# Obtain the value of the device_count field in the hcl.json file.
device_count=$(get_json_value ${RANK_TABLE_FILE} device_count)
if [[ "$device_count" == "" ]]; then
    echo "device count is 0, train job failed." | tee -a hccl.log
```

 Copyright © Huawei Technologies Co., Ltd.

```
    exit 1
fi

# Obtain the value of the instance_count field in the hcl.json file.
instance_count=$(get_json_value ${RANK_TABLE_FILE} instance_count)
if [[ "$instance_count" == "" ]]; then
    echo "instance count is 0, train job failed." | tee -a hccl.log
    exit 1
fi
# Single-node training scenario
if [[ "$instance_count" == "1" ]]; then
    device_count=$(get_json_value ${RANK_TABLE_FILE} device_count)
    server_id=0
    if [ ${device_count} -eq 1 ]; then
        bash main.sh /job/data/cifar-10-batches-bin/
    fi
    if [ ${device_count} -gt 1 ]; then
        python hccl2ranktable.py ${RANK_TABLE_FILE} /job/code/resnet/rank_table_${instance_count}x$
{device_count}pcs.json
        bash main.sh ${device_count} /job/code/resnet/rank_table_${instance_count}x$
{device_count}pcs.json ${server_id} /job/data/cifar-10-batches-bin/
    fi

# Distributed training scenario
else
    rank_index=`echo $HOSTNAME | awk -F"-" '{print $NF}'`
    device_count=8

    python hccl2ranktable.py ${RANK_TABLE_FILE} /job/code/resnet/rank_table_${instance_count}x$
{device_count}pcs.json
    bash main.sh ${device_count} /job/code/resnet/rank_table_${instance_count}x$
{device_count}pcs.json ${rank_index} /job/data/cifar-10-batches-bin/
fi

wait
```

3. Run the following command to create the **main.sh** file:

   vim main.sh

   Add the following content to the file and run the **:wq** command to save the file:

```
#!/bin/bash

ulimit -u unlimited

# Single-device single-card
if [ $# == 1 ]; then
    export DEVICE_NUM=1
    export DEVICE_ID=0
    export RANK_ID=0
    export RANK_SIZE=1

    if [ -d "train" ];
    then
        rm -rf ./train
    fi
    mkdir ./train
    cp ../*.py ./train
    cp *.sh ./train
    cp -r ../src ./train
    cd ./train || exit
    echo "start training for device $DEVICE_ID"
    env > env.log

# Keep the foreground output.
    python train.py --net=resnet50 --dataset=cifar10 --dataset_path=$1 | tee log
fi

# Single-device multi-card and distributed deployment
```

 Copyright © Huawei Technologies Co., Ltd.

```
if [ $# == 4 ]; then
    export DEVICE_NUM=$1
    export RANK_SIZE=$1
    export RANK_TABLE_FILE=$2

    export SERVER_ID=$3
    rank_start=$((DEVICE_NUM * SERVER_ID))

# Start the background job and check the log output of a foreground job.
    for((i=1; i<${DEVICE_NUM}; i++))
    do
        rankid=$((rank_start + i))
        export DEVICE_ID=${i}
        export RANK_ID=${rankid}
        rm -rf ./train_parallel${rankid}
        mkdir ./train_parallel${rankid}
        cp ../*.py ./train_parallel${rankid}
        cp *.sh ./train_parallel${rankid}
        cp -r ../src ./train_parallel${rankid}
        cd ./train_parallel${rankid} || exit
        echo "start training for rank $RANK_ID, device $DEVICE_ID"
        env > env.log

        python train.py --net=resnet50 --dataset=cifar10 --run_distribute=True --device_num=
$DEVICE_NUM --dataset_path=$4 &> log &

        cd ..
    done

    rankid=$((rank_start))
    export DEVICE_ID=0
    export RANK_ID=${rankid}
    rm -rf ./train_parallel${rankid}
    mkdir ./train_parallel${rankid}
    cp ../*.py ./train_parallel${rankid}
    cp *.sh ./train_parallel${rankid}
    cp -r ../src ./train_parallel${rankid}
    cd ./train_parallel${rankid} || exit
    echo "start training for rank $RANK_ID, device $DEVICE_ID"
    env > env.log

    python train.py --net=resnet50 --dataset=cifar10 --run_distribute=True --device_num=
$DEVICE_NUM --dataset_path=$4 | tee log

    cd ..
fi
```

----End

3.6.2 Creating a Container Image Using a Dockerfile (TensorFlow)

Prerequisites

Obtain the software packages of the corresponding operating system and the Dockerfile and script files required for packaging images by referring to **Table 3-7**.

In the names of the deep learning acceleration engine package and framework plugin package, ***{version}*** indicates the package version, ***{arch}*** indicates the OS.

 Copyright © Huawei Technologies Co., Ltd.

Table 3-7 Required software

Software Package	Description	How to Obtain
Ascend-cann-nnae_*{version}*_linux-*{arch}*.run	Deep learning engine software package	**Link**
Ascend-cann-tfplugin_*{version}*_linux-*{arch}*.run	Framework plugin package	**Link**
• ARM: tensorflow-1.15.0-cp37-cp37m-linux_aarch64.whl • x86: tensorflow_cpu-1.15.0-cp37-cp37m-manylinux2010_x86_64.whl	WHL package of the TensorFlow framework	• For the ARM architecture, see **Creating the WHL Package of the TensorFlow Framework**. • For the x86 architecture, download the **x86 TensorFlow framework**.
Dockerfile	Required for creating an image	Prepared by users
ascend_install.info	Software package installation log file	Copy the **/etc/ascend_install.info** file from the host.

 Copyright © Huawei Technologies Co., Ltd.

Software Package	Description	How to Obtain
version.info	Driver package version information file	Copy the **/usr/local/Ascend/driver/version.info** file from the host.
prebuild.sh	Script used to prepare for the installation of the training operating environment, for example, configuring the agent.	Prepared by users
install_ascend_pkgs.sh	Script for installing the Ascend software package.	
postbuild.sh	Delete the installation packages, scripts, and proxy configurations that do not need to be retained in the container.	

NOTE

This section uses Ubuntu ARM as an example.

Procedure

Step 1 Upload the software package, deep learning framework, host Ascend software package installation information file, and driver package version information file to the same directory (for example, **/home/test**) on the server.

 Copyright © Huawei Technologies Co., Ltd.

- Ascend-cann-nnae_*{version}*_linux-*{arch}*.run
- Ascend-cann-tfplugin_*{version}*_linux-*{arch}*.run
- tensorflow-1.15.0-cp37-cp37m-linux_aarch64.whl
- ascend_install.info
- version.info

Step 2 Log in to the server as the **root** user.

Step 3 Perform the following steps to prepare the **prebuild.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **prebuild.sh** file:

   vim prebuild.sh

2. For details about the content to be written, see the **prebuild.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 4 Perform the following steps to prepare the **install_ascend_pkgs.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **install_ascend_pkgs.sh** file:

   vim install_ascend_pkgs.sh

2. For details about the content to be written, see the **install_ascend_pkgs.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 5 Perform the following steps to prepare the **postbuild.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **postbuild.sh** file:

   vim postbuild.sh

2. For details about the content to be written, see the **postbuild.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 6 Perform the following steps to prepare the **Dockerfile** file:

1. Go to the directory where the software package is stored and run the following command to create the **Dockerfile** file:

   vim Dockerfile

2. For details about the content to be written, see the **Dockerfile in the Ubuntu ARM system**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

   NOTE

   To obtain the image **ubuntu:18.04**, you can also run the **docker pull ubuntu:18.04** command to obtain the image from Docker Hub.

Step 7 Go to the directory where the software package is stored and run the following command to create a container image:

docker build -t *Image name_System architecture:Image tag* **.**

Example:

 Copyright © Huawei Technologies Co., Ltd.

docker build -t *test_train_arm64:v1.0* **.**

Table 3-8 describes the commands.

Table 3-8 Command parameter description

Parameter	Description
-t	Image name.
Image name_System architecture:Image tag	Image name and tag. Change them based on the actual situation.

If "Successfully built xxx" is displayed, the image is successfully created. Do not omit **.** at the end of the command.

Step 8 After the image is created, run the following command to view the image information:

docker images

Example:

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
test_train_arm64    v1.0                d82746acd7f0        27 minutes ago      749MB
```

Step 9 Run the following command to access the container:

docker run -it *Image name_System architecture:Image tag* **bash**

Example:

docker run -it *test_train_arm64:v1.0* **bash**

Step 10 Run the following command to obtain the **naming.log** file:

find /usr/local/ -name "freeze_graph.py"

```
root@032953231d61:/tmp# find /usr/local/ -name "freeze_graph.py"
/usr/local/lib/python3.7/dist-packages/tensorflow_core/python/tools/freeze_graph.py
```

Step 11 Run the following command to modify the file in the image:

vim /usr/local/lib/python3.7/dist-packages/tensorflow_core/python/tools/freeze_graph.py

Add the following content.

```
from npu_bridge.estimator import npu_ops
from npu_bridge.estimator.npu.npu_config import NPURunConfig
from npu_bridge.estimator.npu.npu_estimator import NPUEstimator
from npu_bridge.estimator.npu.npu_optimizer import allreduce
from npu_bridge.estimator.npu.npu_optimizer import NPUDistributedOptimizer
from npu_bridge.hccl import hccl_ops
```

Run the **:wq** command to save the configuration and exit.

Step 12 Run the **exit** command to exit the container.

Step 13 Run the following command to save the current image:

docker commit *containerid Image name_System architecture:Image tag*

 Copyright © Huawei Technologies Co., Ltd.

Example:

```
root@032953231d61:/tmp# exit
exit
root@ubuntu-185:/data/kfa/train# docker commit 032953231d61 test_train_arm64:v2.0
```

NOTE

In the preceding example, the value of **containerid** is **032953231d61**.

----End

Compilation Examples

NOTE

Modify the software package version and architecture based on the actual situation.

1. Compilation example of **prebuild.sh** (Ubuntu)

```
#!/bin/bash
#----------------------------------------------------------------------------
# Use the bash syntax to compile script code and prepare for the installation, for example, configuring the proxy.
# This script will be run before the formal creation process is started.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#----------------------------------------------------------------------------
# DNS settings. If the DNS settings are not required, delete them.
tee /etc/resolv.conf <<- EOF
nameserver xxx.xxx.xxx.xxx #IP address of the DNS server. You can enter multiple IP addresses based on the site requirements.
nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx
EOF
# APT proxy settings
tee /etc/apt/apt.conf.d/80proxy <<- EOF
Acquire::http::Proxy "http://xxx.xxx.xxx.xxx:xxx";  #IP address and port number of the HTTP proxy server.
Acquire::https::Proxy "http://xxx.xxx.xxx.xxx:xxx";  #IP address and port number of the HTTPS proxy server.
EOF
chmod 777 -R /tmp
rm /var/lib/apt/lists/*
#APT source settings (The following uses Ubuntu 18.04 ARM as an example. Set the information based on the site requirements.)
tee /etc/apt/sources.list <<- EOF
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-backports main restricted universe multiverse
EOF
```

Example of compiling the **prebuild.sh** script for the Ubuntu x86 OS

```
#!/bin/bash
#----------------------------------------------------------------------------

# Use the bash syntax to compile script code and prepare for the installation, for example, configuring the proxy.
# This script will be run before the formal creation process is started.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained
```

```
in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
# APT proxy settings
tee /etc/apt/apt.conf.d/80proxy <<- EOF
Acquire::http::Proxy "http://xxx.xxx.xxx.xxx:xxx";    #IP address and port number of the HTTP proxy server.
Acquire::https::Proxy "http://xxx.xxx.xxx.xxx:xxx";   #IP address and port number of the HTTPS proxy server.
EOF

#APT source settings (The following uses Ubuntu 18.04 x86 as an example. Set the information based on the site requirements.)
tee /etc/apt/sources.list <<- EOF
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-security main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-security main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main multiverse restricted universe
EOF
```

2. Compilation example of **install_ascend_pkgs.sh**

```
#!/bin/bash
#------------------------------------------------------------------------------
# Use the bash syntax to compile script code and install the Ascend software package.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
# Copy the /etc/ascend_install.info file on the host to the current directory before creating the container image.
cp ascend_install.info /etc/
# Copy the /usr/local/Ascend/driver/version.info file on the host to the current directory before creating the container image.
mkdir -p /usr/local/Ascend/driver/
cp version.info /usr/local/Ascend/driver/

# Ascend-cann-nnae_{version}_linux-{arch}.run
chmod +x Ascend-cann-nnae_{version}_linux-{arch}.run
./Ascend-cann-nnae_{version}_linux-{arch}.run --install-path=/usr/local/Ascend/ --install --quiet

# Ascend-cann-tfplugin_{version}_linux-{arch}.run
chmod +x Ascend-cann-tfplugin_{version}_linux-{arch}.run
./Ascend-cann-tfplugin_{version}_linux-{arch}.run --install-path=/usr/local/Ascend/ --install --quiet

#Only for the installation of the nnae package. Therefore, the nnae package needs to be cleared. When the container is started, the nnae package is mounted by the Ascend Docker.
rm -f version.info
rm -rf /usr/local/Ascend/driver/
```

3. Compilation example of **postbuild.sh** (Ubuntu)

```
#!/bin/bash
#------------------------------------------------------------------------------
# Use the bash syntax to compile the script code and delete the installation packages, scripts, and proxy configurations that do not need to be retained in the container.
# This script will be run after the formal creation process ends.
#
# Note: After this script is run, it is automatically cleared and will not be left in the image. The scripts and Working Dir are stored in /root.
#------------------------------------------------------------------------------

rm -f ascend_install.info
rm -f prebuild.sh
rm -f install_ascend_pkgs.sh
rm -f Dockerfile
rm -f Ascend-cann-nnae_{version}_linux-{arch}.run
```

 Copyright © Huawei Technologies Co., Ltd.

```
rm -f Ascend-cann-tfplugin_{version}_linux-{arch}.run
rm -f tensorflow-1.15.0-cp37-cp37m-linux_{arch}.whl
rm -f /etc/apt/apt.conf.d/80proxy

# Delete if not required
tee /etc/resolv.conf <<- EOF
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "systemd-resolve --status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

options edns0

nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx
EOF
```

4. Dockerfile compilation sample
   - The following is an example of the Dockerfile in the Ubuntu ARM system.

```
FROM ubuntu:18.04

ARG TF_PKG=tensorflow-1.15.0-cp37-cp37m-linux_aarch64.whl
ARG HOST_ASCEND_BASE=/usr/local/Ascend
ARG NNAE_PATH=/usr/local/Ascend/nnae/latest
ARG TF_PLUGIN_PATH=/usr/local/Ascend/tfplugin/latest
ARG INSTALL_ASCEND_PKGS_SH=install_ascend_pkgs.sh
ARG PREBUILD_SH=prebuild.sh
ARG POSTBUILD_SH=postbuild.sh
WORKDIR /tmp
COPY . ./

# Trigger prebuild.sh.
RUN bash -c "test -f $PREBUILD_SH && bash $PREBUILD_SH || true"

ENV http_proxy http://xxx.xxx.xxx.xxx:xxx
ENV https_proxy http://xxx.xxx.xxx.xxx:xxx

# System package
RUN apt update && \
    apt install --no-install-recommends \
        python3.7 python3.7-dev \
        curl g++ pkg-config unzip \
        libblas3 liblapack3 liblapack-dev \
        libblas-dev gfortran libhdf5-dev \
        libffi-dev libicu60 libxml2 -y

# Configure the Python PIP source.
RUN mkdir -p ~/.pip \
&& echo '[global] \n\
index-url=https://pypi.doubanio.com/simple/\n\
trusted-host=pypi.doubanio.com' >> ~/.pip/pip.conf

# pip3.7
RUN curl -k https://bootstrap.pypa.io/get-pip.py -o get-pip.py && \
    cd /tmp && \
    apt-get download python3-distutils && \
    dpkg-deb -x python3-distutils_*.deb / && \
    rm python3-distutils_*.deb && \
```

 Copyright © Huawei Technologies Co., Ltd.

```
    cd - && \
    python3.7 get-pip.py && \
    rm get-pip.py

# HwHiAiUser, hwMindX
RUN useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX && \
    useradd -d /home/HwHiAiUser -u 1000 -m -s /bin/bash HwHiAiUser && \
    usermod -a -G HwHiAiUser hwMindX

# Python package
RUN pip3.7 install numpy && \
    pip3.7 install decorator && \
    pip3.7 install sympy==1.4 && \
    pip3.7 install cffi==1.12.3 && \
    pip3.7 install pyyaml && \
    pip3.7 install pathlib2 && \
    pip3.7 install grpcio && \
    pip3.7 install grpcio-tools && \
    pip3.7 install protobuf && \
    pip3.7 install scipy && \
    pip3.7 install requests

# Ascend package
RUN bash $INSTALL_ASCEND_PKGS_SH

# TensorFlow installation
ENV LD_LIBRARY_PATH=\
/usr/lib/aarch64-linux-gnu/hdf5/serial:\
$HOST_ASCEND_BASE/add-ons:\
$NNAE_PATH/fwkacllib/lib64:\
$HOST_ASCEND_BASE/driver/lib64/common:\
$HOST_ASCEND_BASE/driver/lib64/driver:$LD_LIBRARY_PATH

RUN pip3.7 install $TF_PKG

# Environment variables
ENV GLOG_v=2
ENV TBE_IMPL_PATH=$NNAE_PATH/opp/op_impl/built-in/ai_core/tbe
ENV TF_PLUGIN_PKG=$TF_PLUGIN_PATH/tfplugin/python/site-packages
ENV FWK_PYTHON_PATH=$NNAE_PATH/fwkacllib/python/site-packages
ENV PATH=$NNAE_PATH/fwkacllib/ccec_compiler/bin:$PATH
ENV ASCEND_OPP_PATH=$NNAE_PATH/opp
ENV PYTHONPATH=\
$FWK_PYTHON_PATH:\
$FWK_PYTHON_PATH/auto_tune.egg:\
$FWK_PYTHON_PATH/schedule_search.egg:\
$TF_PLUGIN_PKG:\
$TBE_IMPL_PATH:\
$PYTHONPATH

# Create /lib64/ld-linux-aarch64.so.1.
RUN umask 0022 && \
    if [ ! -d "/lib64" ]; \
    then \
        mkdir /lib64 && ln -sf /lib/ld-linux-aarch64.so.1 /lib64/ld-linux-aarch64.so.1; \
    fi

ENV http_proxy ""
ENV https_proxy ""

# Trigger postbuild.sh.
RUN bash -c "test -f $POSTBUILD_SH && bash $POSTBUILD_SH || true" && \
    rm $POSTBUILD_SH
```

- Dockerfile example for Ubuntu x86

```
FROM ubuntu:18.04
# The following lines are used for online download and installation during image compilation. It is mutually exclusive with the WHL configuration.
ARG TF_PKG=tensorflow-cpu==1.15.0
# Use the offline x86 TensorFlow package, comment out the upper line, and delete the
```

 Copyright © Huawei Technologies Co., Ltd.

```
comment tag (#) from the lower line.
#ARG TF_PKG=tensorflow_cpu-1.15.0-cp37-cp37m-manylinux2010_x86_64.whl
ARG HOST_ASCEND_BASE=/usr/local/Ascend
ARG NNAE_PATH=/usr/local/Ascend/nnae/latest
ARG TF_PLUGIN_PATH=/usr/local/Ascend/tfplugin/latest
ARG INSTALL_ASCEND_PKGS_SH=install_ascend_pkgs.sh
ARG PREBUILD_SH=prebuild.sh
ARG POSTBUILD_SH=postbuild.sh
WORKDIR /tmp
COPY . ./

# Triggering prebuild.sh
RUN bash -c "test -f $PREBUILD_SH && bash $PREBUILD_SH || true"

ENV http_proxy http://xxx.xxx.xxx.xxx:xxx
ENV https_proxy http://xxx.xxx.xxx.xxx:xxx

# System package
RUN apt update && \
    apt install --no-install-recommends \
        python3.7 python3.7-dev \
        curl g++ pkg-config unzip \
        libblas3 liblapack3 liblapack-dev \
        libblas-dev gfortran libhdf5-dev \
        libffi-dev libicu60 libxml2 -y

# Configuring the Python PIP source
RUN mkdir -p ~/.pip \
&& echo '[global] \n\
index-url=https://pypi.doubanio.com/simple/\n\
trusted-host=pypi.doubanio.com' >> ~/.pip/pip.conf

# pip3.7
RUN curl -k https://bootstrap.pypa.io/get-pip.py -o get-pip.py && \
    cd /tmp && \
    apt-get download python3-distutils && \
    dpkg-deb -x python3-distutils_*.deb / && \
    rm python3-distutils_*.deb && \
    cd - && \
    python3.7 get-pip.py && \
    rm get-pip.py

# HwHiAiUser, hwMindX
RUN useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX && \
    useradd -d /home/HwHiAiUser -u 1000 -m -s /bin/bash HwHiAiUser && \
    usermod -a -G HwHiAiUser hwMindX

# Python package
RUN pip3.7 install numpy && \
    pip3.7 install decorator && \
    pip3.7 install sympy==1.4 && \
    pip3.7 install cffi==1.12.3 && \
    pip3.7 install pyyaml && \
    pip3.7 install pathlib2 && \
    pip3.7 install grpcio && \
    pip3.7 install grpcio-tools && \
    pip3.7 install protobuf && \
    pip3.7 install scipy && \
    pip3.7 install requests

# Ascend package
RUN bash $INSTALL_ASCEND_PKGS_SH

# TensorFlow installation
ENV LD_LIBRARY_PATH=\
/usr/lib/x86_64-linux-gnu/hdf5/serial:\
$HOST_ASCEND_BASE/add-ons:\
$NNAE_PATH/fwkacllib/lib64:\
$HOST_ASCEND_BASE/driver/lib64/common:\
```

 Copyright © Huawei Technologies Co., Ltd.

```
$HOST_ASCEND_BASE/driver/lib64/driver:$LD_LIBRARY_PATH

RUN pip3.7 install $TF_PKG

# Environment variables
ENV GLOG_v=2
ENV TBE_IMPL_PATH=$NNAE_PATH/opp/op_impl/built-in/ai_core/tbe
ENV TF_PLUGIN_PKG=$TF_PLUGIN_PATH/tfplugin/python/site-packages
ENV FWK_PYTHON_PATH=$NNAE_PATH/fwkacllib/python/site-packages
ENV PATH=$NNAE_PATH/fwkacllib/ccec_compiler/bin:$PATH
ENV ASCEND_OPP_PATH=$NNAE_PATH/opp
ENV PYTHONPATH=\
$FWK_PYTHON_PATH:\
$FWK_PYTHON_PATH/auto_tune.egg:\
$FWK_PYTHON_PATH/schedule_search.egg:\
$TF_PLUGIN_PKG:\
$TBE_IMPL_PATH:\
$PYTHONPATH

ENV http_proxy ""
ENV https_proxy ""

# Triggering postbuild.sh
RUN bash -c "test -f $POSTBUILD_SH && bash $POSTBUILD_SH || true" && \
    rm $POSTBUILD_SH
```

3.6.3 Creating a Container Image Using a Dockerfile (PyTorch)

Prerequisites

Obtain the software packages of the corresponding operating system and the Dockerfile and script files required for packaging images by referring to **Table 3-9**.

In the software package names, ***{version}*** indicates the version and ***{arch}*** indicates the OS architecture.

Table 3-9 Required software

Software Package	Description	How to Obtain
Ascend-cann-nnae_*{version}*_linux-*{arch}*.run	Deep learning engine software package	**Link**
apex-0.1+ascend-cp37-cp37m-linux_*{arch}*.whl	Mixed precision module	**Link**
torch-1.5.0+ascend.post2-cp37-cp37m-linux_*{arch}*.whl	PyTorch Adapter plugin	**Link**
Dockerfile	Required for creating an image	Prepared by users

Copyright © Huawei Technologies Co., Ltd.

Software Package	Description	How to Obtain
dllogger-master	PyTorch log tool	Link
ascend_install.info	Software package installation log file	Copy the **/etc/ascend_install.info** file from the host.
version.info	Driver package version information file	Copy the **/usr/local/Ascend/driver/version.info** file from the host.
prebuild.sh	Script used to prepare for the installation of the training operating environment, for example, configuring the agent.	Prepared by users
install_ascend_pkgs.sh	Script for installing the Ascend software package.	
postbuild.sh	Delete the installation packages, scripts, and proxy configurations that do not need to be retained in the container.	

 Copyright © Huawei Technologies Co., Ltd.

NOTE

This section uses Ubuntu ARM as an example.

Procedure

Step 1 Upload the software package, deep learning framework, host Ascend software package installation information file, and driver package version information file to the same directory (for example, **/home/test**) on the server.

- Ascend-cann-nnae_*{version}*_linux-*{arch}*.run
- apex-0.1+ascend-cp37-cp37m-linux_*{arch}*.whl
- torch-1.5.0+ascend.post2-cp37-cp37m-linux_*{arch}*.whl
- dllogger-master
- ascend_install.info
- version.info

Step 2 Log in to the server as the **root** user.

Step 3 Perform the following steps to prepare the **prebuild.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **prebuild.sh** file:

   vim prebuild.sh

2. For details about the content to be written, see the **prebuild.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 4 Perform the following steps to prepare the **install_ascend_pkgs.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **install_ascend_pkgs.sh** file:

   vim install_ascend_pkgs.sh

2. For details about the content to be written, see the **install_ascend_pkgs.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 5 Perform the following steps to prepare the **postbuild.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **postbuild.sh** file:

   vim postbuild.sh

2. For details about the content to be written, see the **postbuild.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 6 Perform the following steps to prepare the **Dockerfile** file:

1. Go to the directory where the software package is stored and run the following command to create the **Dockerfile** file:

   vim Dockerfile

2. For details about the content to be written, see the **Dockerfile compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

 Copyright © Huawei Technologies Co., Ltd.

> **NOTE**
>
> To obtain the image **ubuntu:18.04**, you can also run the **docker pull ubuntu:18.04** command to obtain the image from Docker Hub.

Step 7 Go to the directory where the software package is stored and run the following command to create a container image:

docker build -t *Image name_System architecture:Image tag* **.**

Example:

docker build -t *test_train_arm64:v1.0* **.**

Table 3-10 describes the commands.

Table 3-10 Command parameter description

Parameter	Description
-t	Image name.
Image name_System architecture:Image tag	Image name and tag. Change them based on the actual situation.

If "Successfully built xxx" is displayed, the image is successfully created. Do not omit **.** at the end of the command.

Step 8 After the image is created, run the following command to view the image information:

docker images

Example:

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
test_train_arm64    v1.0                d82746acd7f0        27 minutes ago      749MB
```

----End

Compilation Examples

> **NOTE**
>
> Modify the software package version and architecture based on the actual situation.

1. Compilation example of **prebuild.sh**

   Example of compiling the **prebuild.sh** script for the Ubuntu ARM OS

```
#!/bin/bash
#------------------------------------------------------------------------------
# Use the bash syntax to compile script code and prepare for the installation, for example, configuring the proxy.
# This script will be run before the formal creation process is started.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
# DNS settings
tee /etc/resolv.conf <<- EOF
```

 Copyright © Huawei Technologies Co., Ltd.

```
nameserver xxx.xxx.xxx.xxx #IP address of the DNS server. You can enter multiple IP addresses based on the site requirements.
nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx
EOF
# APT proxy settings
tee /etc/apt/apt.conf.d/80proxy <<- EOF
Acquire::http::Proxy "http://xxx.xxx.xxx.xxx:xxx";  #IP address and port number of the HTTP proxy server.
Acquire::https::Proxy "http://xxx.xxx.xxx.xxx:xxx";  #IP address and port number of the HTTPS proxy server.
EOF
chmod 777 -R /tmp
rm /var/lib/apt/lists/*
#APT source settings (The following uses Ubuntu 18.04 ARM as an example. Set the information based on the site requirements.)
tee /etc/apt/sources.list <<- EOF
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-backports main restricted universe multiverse
EOF
```

Example of compiling the **prebuild.sh** script for the Ubuntu x86 OS

```
#!/bin/bash
#------------------------------------------------------------------------------

# Use the bash syntax to compile script code and prepare for the installation, for example, configuring the proxy.
# This script will be run before the formal creation process is started.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
# APT proxy settings
tee /etc/apt/apt.conf.d/80proxy <<- EOF
Acquire::http::Proxy "http://xxx.xxx.xxx.xxx:xxx";    #IP address and port number of the HTTP proxy server.
Acquire::https::Proxy "http://xxx.xxx.xxx.xxx:xxx";   #IP address and port number of the HTTPS proxy server.
EOF

#APT source settings (The following uses Ubuntu 18.04 x86 as an example. Set the information based on the site requirements.)
tee /etc/apt/sources.list <<- EOF
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-security main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-security main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main multiverse restricted universe
EOF
```

2. Compilation example of **install_ascend_pkgs.sh**

```
#------------------------------------------------------------------------------
# Use the bash syntax to compile script code and install the Ascend software package.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
```

 Copyright © Huawei Technologies Co., Ltd.

```
cp ascend_install.info /etc/
# Copy the /usr/local/Ascend/driver/version.info file on the host to the current directory before creating the container image.
mkdir -p /usr/local/Ascend/driver/
cp version.info /usr/local/Ascend/driver/
# Ascend-cann-nnae_{version}_linux-{arch}.run
chmod +x Ascend-cann-nnae_{version}_linux-{arch}.run
./Ascend-cann-nnae_{version}_linux-{arch}.run --install-path=/usr/local/Ascend/ --install --quiet
# Only for the installation of the nnae package. Therefore, the nnae package needs to be cleared. When the container is started, the nnae package is mounted by the Ascend Docker.
rm -f version.info
rm -rf /usr/local/Ascend/driver/
```

3. Compilation example of **postbuild.sh**

```
#-------------------------------------------------------------------------------
# Use the bash syntax to compile the script code and delete the installation packages, scripts, and proxy configurations that do not need to be retained in the container.
# This script will be run after the formal creation process ends.
#
# Note: After this script is run, it is automatically cleared and will not be left in the image. The scripts and Working Dir are stored in /tmp.
#-------------------------------------------------------------------------------
rm -f ascend_install.info
rm -f prebuild.sh
rm -f install_ascend_pkgs.sh
rm -f Dockerfile
rm -f Ascend-cann-nnae_{version}_linux-{arch}.run
rm -f apex-0.1+ascend-cp37-cp37m-linux_{arch}.whl
rm -f torch-1.5.0+ascend.post2-cp37-cp37m-linux_{arch}.whl
rm -f /etc/apt/apt.conf.d/80proxy
```

4. Dockerfile compilation sample

   The following is an example of the Dockerfile in the Ubuntu ARM system.

```
FROM ubuntu:18.04

ARG PYTORCH_PKG=torch-1.5.0+ascend-cp37-cp37m-linux_aarch64.whl
ARG APEX_PKG=apex-0.1+ascend-cp37-cp37m-linux_aarch64.whl
ARG HOST_ASCEND_BASE=/usr/local/Ascend
ARG NNAE_PATH=/usr/local/Ascend/nnae/latest
# ARG TF_PLUGIN_PATH=/usr/local/Ascend/tfplugin/latest
ARG INSTALL_ASCEND_PKGS_SH=install_ascend_pkgs.sh
ARG PREBUILD_SH=prebuild.sh
ARG POSTBUILD_SH=postbuild.sh

WORKDIR /tmp
COPY . ./

# Triggering prebuild.sh
RUN bash -c "test -f $PREBUILD_SH && bash $PREBUILD_SH || true"

ENV http_proxy http://xxx.xxx.xxx.xxx:xxx
ENV https_proxy http://xxx.xxx.xxx.xxx:xxx

# System package
RUN apt update && \
    apt install --no-install-recommends \
        python3.7 python3.7-dev \
        curl g++ pkg-config unzip \
        libblas3 liblapack3 liblapack-dev \
        libblas-dev gfortran libhdf5-dev \
        libffi-dev libicu60 libxml2 -y

# Configuring the Python PIP source
RUN mkdir -p ~/.pip \
&& echo '[global] \n\
index-url=https://pypi.doubanio.com/simple/\n\
trusted-host=pypi.doubanio.com' >> ~/.pip/pip.conf

# pip3.7
RUN curl -k https://bootstrap.pypa.io/get-pip.py -o get-pip.py && \
```

 Copyright © Huawei Technologies Co., Ltd.

```
    cd /tmp && \
    apt-get download python3-distutils && \
    dpkg-deb -x python3-distutils_*.deb / && \
    rm python3-distutils_*.deb && \
    cd - && \
    python3.7 get-pip.py && \
    rm get-pip.py

# HwHiAiUser, hwMindX
RUN useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX && \
    useradd -d /home/HwHiAiUser -u 1000 -m -s /bin/bash HwHiAiUser && \
    usermod -a -G HwHiAiUser hwMindX

# Python package
RUN pip3.7 install numpy && \
    pip3.7 install decorator && \
    pip3.7 install sympy==1.4 && \
    pip3.7 install cffi==1.12.3 && \
    pip3.7 install pyyaml && \
    pip3.7 install pathlib2 && \
    pip3.7 install grpcio && \
    pip3.7 install grpcio-tools && \
    pip3.7 install protobuf && \
    pip3.7 install scipy && \
    pip3.7 install requests && \
    pip3.7 install attrs

# Ascend package
RUN bash $INSTALL_ASCEND_PKGS_SH

# PyTorch installation
ENV LD_LIBRARY_PATH=\
/usr/lib/aarch64-linux-gnu/hdf5/serial:\
$HOST_ASCEND_BASE/add-ons:\
$NNAE_PATH/fwkacllib/lib64:\
$HOST_ASCEND_BASE/driver/lib64/common:\
$HOST_ASCEND_BASE/driver/lib64/driver:$LD_LIBRARY_PATH

RUN pip3.7 install $APEX_PKG

RUN pip3.7 install $PYTORCH_PKG

RUN pip3.7 install torchvision

RUN cd /tmp/dllogger-master/ && \
    python3.7 setup.py build && \
    python3.7 setup.py install

# Environment variables
ENV GLOG_v=2
ENV TBE_IMPL_PATH=$NNAE_PATH/opp/op_impl/built-in/ai_core/tbe
ENV FWK_PYTHON_PATH=$NNAE_PATH/fwkacllib/python/site-packages
ENV PATH=$NNAE_PATH/fwkacllib/ccec_compiler/bin:$PATH
ENV ASCEND_OPP_PATH=$NNAE_PATH/opp
ENV PYTHONPATH=\
$FWK_PYTHON_PATH:\
$FWK_PYTHON_PATH/auto_tune.egg:\
$FWK_PYTHON_PATH/schedule_search.egg:\
$TBE_IMPL_PATH:\
$PYTHONPATH

# Creating /lib64/ld-linux-aarch64.so.1
RUN umask 0022 && \
    if [ ! -d "/lib64" ]; \
    then \
        mkdir /lib64 && ln -sf /lib/ld-linux-aarch64.so.1 /lib64/ld-linux-aarch64.so.1; \
    fi

ENV http_proxy ""
```

 Copyright © Huawei Technologies Co., Ltd.

```
ENV https_proxy ""

# Triggering postbuild.sh
RUN bash -c "test -f $POSTBUILD_SH && bash $POSTBUILD_SH || true" && \
    rm $POSTBUILD_SH
```

Dockerfile example for Ubuntu x86

```
FROM ubuntu:18.04

ARG PYTORCH_PKG=torch-1.5.0+ascend-cp37-cp37m-linux_x86_64.whl
ARG APEX_PKG=apex-0.1+ascend-cp37-cp37m-linux_x86_64.whl
ARG HOST_ASCEND_BASE=/usr/local/Ascend
ARG NNAE_PATH=/usr/local/Ascend/nnae/latest
ARG INSTALL_ASCEND_PKGS_SH=install_ascend_pkgs.sh
ARG PREBUILD_SH=prebuild.sh
ARG POSTBUILD_SH=postbuild.sh
WORKDIR /tmp
COPY . ./

# Triggering prebuild.sh
RUN bash -c "test -f $PREBUILD_SH && bash $PREBUILD_SH || true"

#System package
RUN apt update
RUN apt install --no-install-recommends python3.7 python3.7-dev -y
RUN apt install --no-install-recommends curl g++ pkg-config unzip -y
RUN apt install --no-install-recommends libblas3 liblapack3 liblapack-dev libblas-dev gfortran libhdf5-dev libffi-dev \
                                        libicu60 libxml2 -y

ENV http_proxy http://xxx.xxx.xxx.xxx:xxx
ENV https_proxy http://xxx.xxx.xxx.xxx:xxx

# Configure the Python PIP source.
RUN mkdir -p ~/.pip \
&& echo '[global] \n\
index-url=https://pypi.doubanio.com/simple/\n\
trusted-host=pypi.doubanio.com' >> ~/.pip/pip.conf

# pip3.7
RUN curl -k https://bootstrap.pypa.io/get-pip.py -o get-pip.py && \
    cd /tmp && \
    apt-get download python3-distutils && \
    dpkg-deb -x python3-distutils_*.deb / && \
    rm python3-distutils_*.deb && \
    cd - && \
    python3.7 get-pip.py && \
    rm get-pip.py

# HwHiAiUser, hwMindX
RUN umask 0022 && \
useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX && \
useradd -d /home/HwHiAiUser -u 1000 -m -s /bin/bash HwHiAiUser && \
usermod -a -G HwHiAiUser hwMindX

# Python package
RUN pip3.7 install numpy && \
    pip3.7 install decorator && \
    pip3.7 install sympy==1.4 && \
    pip3.7 install cffi==1.12.3 && \
    pip3.7 install pyyaml && \
    pip3.7 install pathlib2 && \
    pip3.7 install grpcio && \
    pip3.7 install grpcio-tools && \
    pip3.7 install protobuf && \
    pip3.7 install scipy && \
    pip3.7 install requests && \
    pip3.7 install attrs && \
    pip3.7 install Pillow && \
    pip3.7 install torchvision==0.6.0
```

 Copyright © Huawei Technologies Co., Ltd.

```
# Ascend package
RUN bash $INSTALL_ASCEND_PKGS_SH

# PyTorch installation
ENV LD_LIBRARY_PATH=\
/usr/lib/x86_64-linux-gnu/hdf5/serial:\
$HOST_ASCEND_BASE/add-ons:\
$NNAE_PATH/fwkacllib/lib64:\
$HOST_ASCEND_BASE/driver/lib64/common:\
$HOST_ASCEND_BASE/driver/lib64/driver:$LD_LIBRARY_PATH

RUN pip3.7 install $APEX_PKG

RUN pip3.7 install $PYTORCH_PKG

RUN cd /tmp/dllogger-master/ && \    # Find the directory where the setup.py file is located based on the downloaded file and modify the file.
    python3.7 setup.py build && \
    python3.7 setup.py install

# Environment variables
ENV GLOG_v=2
ENV TBE_IMPL_PATH=$NNAE_PATH/opp/op_impl/built-in/ai_core/tbe
ENV FWK_PYTHON_PATH=$NNAE_PATH/fwkacllib/python/site-packages
ENV PATH=$NNAE_PATH/fwkacllib/ccec_compiler/bin:$PATH
ENV ASCEND_OPP_PATH=$NNAE_PATH/opp
ENV PYTHONPATH=\
$FWK_PYTHON_PATH:\
$FWK_PYTHON_PATH/auto_tune.egg:\
$FWK_PYTHON_PATH/schedule_search.egg:\
$TBE_IMPL_PATH:\
$PYTHONPATH

ENV http_proxy ""
ENV https_proxy ""

# Triggering postbuild.sh
RUN bash -c "test -f $POSTBUILD_SH && bash $POSTBUILD_SH || true" && \
    rm $POSTBUILD_SH
```

3.6.4 Creating a Container Image Using a Dockerfile (MindSpore)

Prerequisites

Obtain the software packages of the corresponding OS and the Dockerfile and script files required for packaging images by referring to **Table 3-11**.

In the names of the deep learning engine software package and MindSpore framework software package, ***{version}*** indicates the package version, ***{arch}*** indicates the OS.

NOTE

The MindSpore software package must match the Ascend 910 AI Processor software package. For details, see the ***MindSpore Installation Guide***.

Copyright © Huawei Technologies Co., Ltd.

Table 3-11 Required software

Software Package	Description	How to Obtain
Ascend-cann-nnae_*{version}*_linux-*{arch}*.run	Deep learning engine software package	**Link**
mindspore_ascend-*{version}*-cp37-cp37m-linux_*{arch}*.whl	WHL package of the MindSpore framework	**Link**
Dockerfile	Package required for creating an image.	Prepared by users.
ascend_install.info	Software package installation log file.	Copy the **/etc/ascend_install.info** file from the host.
version.info	Driver package version information file.	Copy the **/usr/local/Ascend/driver/version.info** file from the host.
prebuild.sh	Script used to prepare for the installation of the training operating environment, for example, configuring the agent.	Prepared by users.
install_ascend_pkgs.sh	Script for installing the Ascend software package.	

 Copyright © Huawei Technologies Co., Ltd.

Software Package	Description	How to Obtain
postbuild.sh	Delete the installation packages, scripts, and proxy configurations that do not need to be retained in a container.	

NOTE

This section uses Ubuntu ARM as an example.

Procedure

Step 1 Upload the software package, deep learning framework, host Ascend software package installation information file, and driver package version information file to the same directory (for example, **/home/test**) on the server.

- Ascend-cann-nnae_*{version}*_linux-*{arch}*.run
- mindspore_ascend-*{version}*-cp37-cp37m-linux_*{arch}*.whl
- ascend_install.info
- version.info

Step 2 Log in to the server as the **root** user.

Step 3 Perform the following steps to prepare the **prebuild.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **prebuild.sh** file:

   vim prebuild.sh

2. For details about the content to be written, see the **prebuild.sh compilation example (Ubuntu)**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 4 Perform the following steps to prepare the **install_ascend_pkgs.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **install_ascend_pkgs.sh** file:

   vim install_ascend_pkgs.sh

2. For details about the content to be written, see the **install_ascend_pkgs.sh compilation example**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 5 Perform the following steps to prepare the **postbuild.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **postbuild.sh** file:

 Copyright © Huawei Technologies Co., Ltd.

vim postbuild.sh

2. For details about the content to be written, see the **postbuild.sh compilation example (Ubuntu)**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

Step 6 Perform the following steps to create a Dockerfile:

1. Go to the directory where the software package is stored and run the following command to create the **Dockerfile** file:

   vim Dockerfile

2. For details about the content to be written, see the **Dockerfile in the Ubuntu ARM system**. After writing the content, run the **:wq** command to save the content. The following uses Ubuntu ARM as an example.

   NOTE

   To obtain the image **ubuntu:18.04**, you can also run the **docker pull ubuntu:18.04** command to obtain the image from Docker Hub.

Step 7 Go to the directory where the software packages are stored and run the following command to create a container image:

docker build -t *Image name_System architecture:Image tag* **.**

Example:

docker build -t *test_train_arm64:v1.0* **.**

Table 3-12 describes the commands.

Table 3-12 Command parameter description

Parameter	Description
-t	Specifies the image name.
Image name_System architecture:Image tag	Image name and tag. Change them based on the actual situation.

If "Successfully built xxx" is displayed, the image is successfully created. Do not omit **.** at the end of the command.

Step 8 After the image is created, run the following command to view the image information:

docker images

Example:

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
test_train_arm64    v1.0                d82746acd7f0        27 minutes ago      749MB
```

----End

 Copyright © Huawei Technologies Co., Ltd.

Compilation Examples

NOTE

Modify the software package version and architecture based on the actual situation.

1. Example of compiling the **prebuild.sh** script for the Ubuntu ARM OS

```
#!/bin/bash
#------------------------------------------------------------------------------
#Use the bash syntax to compile script code and prepare for the installation, for example, configuring the proxy.
# This script will be run before the formal creation process is started.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
# DNS settings
tee /etc/resolv.conf <<- EOF
nameserver xxx.xxx.xxx.xxx #IP address of the DNS server. You can enter multiple IP addresses based on the site requirements.
nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx
EOF
# APT proxy settings
tee /etc/apt/apt.conf.d/80proxy <<- EOF
Acquire::http::Proxy "http://xxx.xxx.xxx.xxx:xxx";  #IP address and port number of the HTTP proxy server.
Acquire::https::Proxy "http://xxx.xxx.xxx.xxx:xxx";  #IP address and port number of the HTTPS proxy server.
EOF
chmod 777 -R /tmp
rm /var/lib/apt/lists/*
#APT source settings (The following uses Ubuntu 18.04 ARM as an example. Set the information based on the site requirements.)
tee /etc/apt/sources.list <<- EOF
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu-ports/ bionic-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu-ports/ bionic-backports main restricted universe multiverse
EOF
```

Example of compiling the **prebuild.sh** script for the Ubuntu x86 OS

```
#!/bin/bash
#------------------------------------------------------------------------------

# Use the bash syntax to compile script code and prepare for the installation, for example, configuring the proxy.
# This script will be run before the formal creation process is started.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#------------------------------------------------------------------------------
# APT proxy settings
tee /etc/apt/apt.conf.d/80proxy <<- EOF
Acquire::http::Proxy "http://xxx.xxx.xxx.xxx:xxx";    #IP address and port number of the HTTP proxy server.
Acquire::https::Proxy "http://xxx.xxx.xxx.xxx:xxx";   #IP address and port number of the HTTPS proxy server.
EOF

#APT source settings (The following uses Ubuntu 18.04 x86 as an example. Set the information based on the site requirements.)
tee /etc/apt/sources.list <<- EOF
```

```
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-security main multiverse restricted universe
deb http://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-security main multiverse restricted universe
deb-src http://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main multiverse restricted universe
EOF
```

2. Compilation example of **install_ascend_pkgs.sh**

```
#!/bin/bash
#-------------------------------------------------------------------------------
# Use the bash syntax to compile script code and install the Ascend software package.
#
# Note: After this script is run, it will not be automatically cleared. If it does not need to be retained in the image, clear it from the postbuild.sh script.
#-------------------------------------------------------------------------------
# Copy the /etc/ascend_install.info file on the host to the current directory before creating the container image.
cp ascend_install.info /etc/
mkdir -p /usr/local/Ascend/driver/
cp version.info /usr/local/Ascend/driver/

# Ascend-cann-nnae_{version}_linux-{arch}.run
chmod +x Ascend-cann-nnae_{version}_linux-{arch}.run
./Ascend-cann-nnae_{version}_linux-{arch}.run --install-path=/usr/local/Ascend/ --install --quiet

# Only the nnae package is installed. Therefore, the nnae package needs to be cleared. When the container is started, the nnae package is mounted by the Ascend Docker.
rm -f version.info
rm -rf /usr/local/Ascend/driver/
```

3. Compilation example of **postbuild.sh** (Ubuntu)

```
#!/bin/bash
#-------------------------------------------------------------------------------
# Use the bash syntax to compile the script code and delete the installation packages, scripts, and proxy configurations that do not need to be retained in the container.
# This script will be run after the formal creation process ends.
#
# Note: After this script is run, it is automatically cleared and will not be left in the image. The scripts and Working Dir are stored in /root.
#-------------------------------------------------------------------------------

rm -f ascend_install.info
rm -f prebuild.sh
rm -f install_ascend_pkgs.sh
rm -f Dockerfile
rm -f version.info
rm -f Ascend-cann-nnae_{version}_linux-{arch}.run
rm -f mindspore_ascend-{version}-cp37-cp37m-linux_{arch}.whl
rm -f /etc/apt/apt.conf.d/80proxy

tee /etc/resolv.conf <<- EOF
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "systemd-resolve --status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
```

 Copyright © Huawei Technologies Co., Ltd.

```
# operation for /etc/resolv.conf.

options edns0

nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx
EOF
```

4. Dockerfile compilation sample
   - The following is an example of the Dockerfile in the Ubuntu ARM system.

```
FROM ubuntu:18.04

ARG HOST_ASCEND_BASE=/usr/local/Ascend
ARG INSTALL_ASCEND_PKGS_SH=install_ascend_pkgs.sh
ARG NNAE_PATH=/usr/local/Ascend/nnae/latest
ARG MINDSPORE_PKG=mindspore_ascend-{version}-cp37-cp37m-linux_aarch64.whl
ARG PREBUILD_SH=prebuild.sh
ARG POSTBUILD_SH=postbuild.sh
WORKDIR /tmp
COPY . ./

# Trigger prebuild.sh.
RUN bash -c "test -f $PREBUILD_SH && bash $PREBUILD_SH"

ENV http_proxy http://xxx.xxx.xxx.xxx:xxx
ENV https_proxy http://xxx.xxx.xxx.xxx:xxx


# System package
RUN apt update && \
    apt install --no-install-recommends python3.7 python3.7-dev curl g++ pkg-config unzip \
        libblas3 liblapack3 liblapack-dev libblas-dev gfortran libhdf5-dev libffi-dev libicu60 libxml2
-y

# Create a Python soft link.
RUN ln -s /usr/bin/python3.7 /usr/bin/python

# Configure the Python PIP source.
RUN mkdir -p ~/.pip \
&& echo '[global] \n\
index-url=https://pypi.doubanio.com/simple/\n\
trusted-host=pypi.doubanio.com' >> ~/.pip/pip.conf

# pip3.7
RUN curl -k https://bootstrap.pypa.io/get-pip.py -o get-pip.py && \
    cd /tmp && \
    apt-get download python3-distutils && \
    dpkg-deb -x python3-distutils_*.deb / && \
    rm python3-distutils_*.deb && \
    cd - && \
    python3.7 get-pip.py && \
    rm get-pip.py

# HwHiAiUser, hwMindX
RUN useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX && \
    useradd -d /home/HwHiAiUser -u 1000 -m -s /bin/bash HwHiAiUser && \
    usermod -a -G HwHiAiUser hwMindX

# Python package
RUN pip3.7 install numpy && \
    pip3.7 install decorator && \
    pip3.7 install sympy==1.4 && \
    pip3.7 install cffi==1.12.3 && \
    pip3.7 install pyyaml && \
    pip3.7 install pathlib2 && \
    pip3.7 install grpcio && \
    pip3.7 install grpcio-tools && \
    pip3.7 install protobuf && \
    pip3.7 install scipy && \
```

 Copyright © Huawei Technologies Co., Ltd.

```
    pip3.7 install requests

# Ascend package
RUN umask 0022 && bash $INSTALL_ASCEND_PKGS_SH

# MindSpore installation
RUN pip3.7 install $MINDSPORE_PKG

# Environment variable
ENV GLOG_v=2
ENV TBE_IMPL_PATH=$NNAE_PATH/opp/op_impl/built-in/ai_core/tbe
ENV FWK_PYTHON_PATH=$NNAE_PATH/fwkacllib/python/site-packages
ENV PATH=$NNAE_PATH/fwkacllib/ccec_compiler/bin:$PATH
ENV ASCEND_OPP_PATH=$NNAE_PATH/opp
ENV PYTHONPATH=\
$FWK_PYTHON_PATH:\
$FWK_PYTHON_PATH/auto_tune.egg:\
$FWK_PYTHON_PATH/schedule_search.egg:\
$TBE_IMPL_PATH:\
$PYTHONPATH
ENV LD_LIBRARY_PATH=$NNAE_PATH/fwkacllib/lib64/:\
/usr/local/Ascend/driver/lib64/common/:\
/usr/local/Ascend/driver/lib64/driver/:\
/usr/local/Ascend/add-ons/:\
/usr/local/Ascend/driver/tools/hccn_tool/:\
$LD_LIBRARY_PATH

# Create /lib64/ld-linux-aarch64.so.1.
RUN umask 0022 && \
    if [ ! -d "/lib64" ]; \
    then \
        mkdir /lib64 && ln -sf /lib/ld-linux-aarch64.so.1 /lib64/ld-linux-aarch64.so.1; \
    fi

ENV http_proxy ""
ENV https_proxy ""

# Trigger postbuild.sh.
RUN bash -c "test -f $POSTBUILD_SH && bash $POSTBUILD_SH" && \
    rm $POSTBUILD_SH
```

- Dockerfile example for Ubuntu x86

```
FROM ubuntu:18.04

ARG HOST_ASCEND_BASE=/usr/local/Ascend
ARG INSTALL_ASCEND_PKGS_SH=install_ascend_pkgs.sh
ARG NNAE_PATH=/usr/local/Ascend/nnae/latest
ARG MINDSPORE_PKG=mindspore_ascend-{version}-cp37-cp37m-linux_x86_64.whl
ARG PREBUILD_SH=prebuild.sh
ARG POSTBUILD_SH=postbuild.sh
WORKDIR /tmp
COPY . ./

# Trigger prebuild.sh.
RUN bash -c "test -f $PREBUILD_SH && bash $PREBUILD_SH"

ENV http_proxy http://xxx.xxx.xxx.xxx:xxx
ENV https_proxy http://xxx.xxx.xxx.xxx:xxx


# System package
RUN apt update && \
    apt install --no-install-recommends python3.7 python3.7-dev curl g++ pkg-config unzip \
        libblas3 liblapack3 liblapack-dev libblas-dev gfortran libhdf5-dev libffi-dev libicu60 libxml2
-y

# Create a Python soft link.
RUN ln -s /usr/bin/python3.7 /usr/bin/python

# Configure the Python PIP source.
```

 Copyright © Huawei Technologies Co., Ltd.

```
RUN mkdir -p ~/.pip \
&& echo '[global] \n\
index-url=https://pypi.doubanio.com/simple/\n\
trusted-host=pypi.doubanio.com' >> ~/.pip/pip.conf

# pip3.7
RUN curl -k https://bootstrap.pypa.io/get-pip.py -o get-pip.py && \
    cd /tmp && \
    apt-get download python3-distutils && \
    dpkg-deb -x python3-distutils_*.deb / && \
    rm python3-distutils_*.deb && \
    cd - && \
    python3.7 get-pip.py && \
    rm get-pip.py

# HwHiAiUser, hwMindX
RUN useradd -d /home/hwMindX -u 9000 -m -s /bin/bash hwMindX && \
    useradd -d /home/HwHiAiUser -u 1000 -m -s /bin/bash HwHiAiUser && \
    usermod -a -G HwHiAiUser hwMindX

# Python package
RUN pip3.7 install numpy && \
    pip3.7 install decorator && \
    pip3.7 install sympy==1.4 && \
    pip3.7 install cffi==1.12.3 && \
    pip3.7 install pyyaml && \
    pip3.7 install pathlib2 && \
    pip3.7 install grpcio && \
    pip3.7 install grpcio-tools && \
    pip3.7 install protobuf && \
    pip3.7 install scipy && \
    pip3.7 install requests

# Ascend package
RUN umask 0022 && bash $INSTALL_ASCEND_PKGS_SH

# MindSpore installation
RUN pip3.7 install $MINDSPORE_PKG

# Environment variable
ENV GLOG_v=2
ENV TBE_IMPL_PATH=$NNAE_PATH/opp/op_impl/built-in/ai_core/tbe
ENV FWK_PYTHON_PATH=$NNAE_PATH/fwkacllib/python/site-packages
ENV PATH=$NNAE_PATH/fwkacllib/ccec_compiler/bin:$PATH
ENV ASCEND_OPP_PATH=$NNAE_PATH/opp
ENV PYTHONPATH=\
$FWK_PYTHON_PATH:\
$FWK_PYTHON_PATH/auto_tune.egg:\
$FWK_PYTHON_PATH/schedule_search.egg:\
$TBE_IMPL_PATH:\
$PYTHONPATH
ENV LD_LIBRARY_PATH=$NNAE_PATH/fwkacllib/lib64/:\
/usr/local/Ascend/driver/lib64/common/:\
/usr/local/Ascend/driver/lib64/driver/:\
/usr/local/Ascend/add-ons/:\
/usr/local/Ascend/driver/tools/hccn_tool/:\
$LD_LIBRARY_PATH

ENV http_proxy ""
ENV https_proxy ""

# Trigger postbuild.sh.
RUN bash -c "test -f $POSTBUILD_SH && bash $POSTBUILD_SH" && \
    rm $POSTBUILD_SH
```

 Copyright © Huawei Technologies Co., Ltd.

3.6.5 Creating an Inference Image Using a Dockerfile

Prerequisites

Obtain the software packages of the corresponding operating system and the Dockerfile and script files required for packaging images by referring to **Table 3-13**.

In the name of the offline inference engine package, **{version}** indicates the version, **xxx** indicates the OS, and **{gcc_version}** indicates the GCC version.

Table 3-13 Required software

Software Package	Description	How to Obtain
Ascend-cann-nnrt_*{version}*_linux-*{arch}*.run	Offline inference engine package.	**Link**
Dockerfile	Required for creating an image	Prepared by users
install.sh	Script for installing the inference service.	
XXX.tar	Name of the inference service code package, which is prepared by users based on the inference service. This document uses **dvpp_resnet.tar** as an example.	
run.sh	Script for starting the inference service.	
NOTE Other software packages and code required for inference need to be prepared by users.		

This section uses Ubuntu x86 as an example.

Procedure

Step 1 Upload the software packages and files to the same directory (for example, **/home/infer**) on the server.

- Ascend-cann-nnrt_*{version}*_linux-*{arch}*.run
- Dockerfile
- install.sh
- run.sh

 Copyright © Huawei Technologies Co., Ltd.

- *XXX*.tar (Inference code or script prepared by users)

Step 2 Log in to the server as the **root** user.

Step 3 Perform the following steps to prepare the **install.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **install.sh** file:

   vim install.sh

2. Build a sample based on the service requirements and run the **:wq** command to save the content. For details, see the **install.sh compilation example**.

Step 4 Perform the following steps to prepare the **run.sh** file:

1. Go to the directory where the software package is stored and run the following command to create the **run.sh** file:

   vim run.sh

2. Build a sample based on the service requirements and run the **:wq** command to save the content. For details, see the **run.sh compilation example**.

Step 5 Perform the following steps to prepare the **Dockerfile** file:

1. Go to the directory where the software package is stored and run the following command to create the **Dockerfile** file:

   vim Dockerfile

2. Write the sample code by referring to the **Dockerfile compilation sample** and run the **:wq** command to save the file.

Step 6 Go to the directory where the software package is stored and run the following command to create a container image:

docker build --build-arg NNRT_VERSION=*{version}* **--build-arg NNRT_ARCH=***{arch}* **--build-arg DIST_PKG=***XXX*.**tar -t** *Image name_System architecture:Image tag* **.**

Example:

docker build --build-arg NNRT_VERSION=*20.1.rc1* **--build-arg NNRT_ARCH=***x86_64* **--build-arg DIST_PKG=***dvpp_resnet*.**tar -t** *ubuntu-infer:v1* **.**

Table 3-14 describes the commands.

Table 3-14 Command parameter description

Parameter	Description
--build-arg	Parameters in the Dockerfile.
{version}	Version of the offline inference acceleration package. Set it to the actual one.
{arch}	Architecture of the offline inference acceleration package. Set it to the actual one.
XXX.**tar**	Name of the inference service code package. Set it to the actual one.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Description
-t	Image name.
Image name_System architecture:Image tag	Image name and tag. Change them based on the actual situation.

If "Successfully built xxx" is displayed, the image is successfully created. Do not omit **.** at the end of the command.

Step 7 After the image is created, run the following command to view the image information:

docker images

Example:

```
REPOSITORY      TAG         IMAGE ID        CREATED          SIZE
ubuntu-infer    v1          fffbd83be42a    2 minutes ago    293MB
```

----End

Compilation Examples

1. Compilation example of **install.sh**

```
#!/bin/bash
#-------------------------------------------------------------------------------
# Install the inference service script. The following uses the inference service package dvpp_resnet.tar as an example. You can change the service package name as required.
#--------------------------------------
tar -xvf dvpp_resnet.tar
-------------------------------------------
```

2. Compilation example of **run.sh**

```
#!/bin/bash
# Runing the NPU inference driver daemon
mkdir -p /usr/slog
mkdir -p /var/log/npu/slog/slogd
chown -Rf HwHiAiUser:HwHiAiUser  /usr/slog
chown -Rf HwHiAiUser:HwHiAiUser  /var/log/npu/slog
/usr/local/Ascend/driver/tools/slogd
# Runing the service code
cd /home/out
numbers=`ls /dev/| grep davinci | grep -v davinci_manager | wc -l`
# Updating logs every 5 minutes
#./main $numbers|grep -nE '.*\[.*[[:digit:]]{2}:[[:digit:]]{1}[05]:00' >/home/log/log.txt
./main $numbers
```

3. Dockerfile compilation sample

```
FROM ubuntu:18.04

# Setting the parameters of the offline inference engine package
ARG NNRT_VERSION
ARG NNRT_ARCH
ARG NNRT_PKG=Ascend-cann-nnrt_${NNRT_VERSION}_linux-${NNRT_ARCH}.run
# Setting environment variables
ARG ASCEND_BASE=/usr/local/Ascend

ENV LD_LIBRARY_PATH=\
$LD_LIBRARY_PATH:\
$ASCEND_BASE/driver/lib64:\
$ASCEND_BASE/add-ons:\
$ASCEND_BASE/nnrt/latest/acllib/lib64
```

 Copyright © Huawei Technologies Co., Ltd.

```
# Setting the directory of the started container
WORKDIR /home
# Copy the driver package and offline inference engine package.
COPY $NNRT_PKG .

# Installing the offline inference engine package
RUN umask 0022 && \
    groupadd HwHiAiUser && \
    useradd -g HwHiAiUser -m -d /home/HwHiAiUser HwHiAiUser && \
    chmod +x ${NNRT_PKG} &&\
    ./${NNRT_PKG} --quiet --install &&\
    rm ${NNRT_PKG}

# Copying the service inference program package, installation script, and running script
ARG DIST_PKG
COPY $DIST_PKG .
COPY install.sh .
COPY run.sh .

# Running the installation script
RUN chmod +x run.sh install.sh && \
    sh install.sh && \
    rm $DIST_PKG && \
    rm install.sh

CMD sh run.sh
```

3.6.6 Creating the WHL Package of the TensorFlow Framework

Installation Preparations

- For the x86 architecture, skip this step.
- In the AArch64 architecture, TensorFlow depends on H5py, and H5py depends on HDF5. Therefore, you need to compile and install HDF5 first. Otherwise, an error is reported when you use pip to install H5py. Perform the following operations as the **root** user:

Step 1 Compile and install the HDF5.

1. Run the **wget** command to download the source code package of HDF5 to any directory of the installation environment. The command is as follows:
   ```
   wget https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.10/hdf5-1.10.5/src/hdf5-1.10.5.tar.gz --no-check-certificate
   ```
2. Run the following command to go to the download directory and decompress the source code package:
   ```
   tar -zxvf hdf5-1.10.5.tar.gz
   ```
   Go to the decompressed folder and run the following configuration, build, and installation commands:
   ```
   cd hdf5-1.10.5/
   ./configure --prefix=/usr/include/hdf5
   make install
   ```

Step 2 Configure environment variables and create a soft link to the dynamic link library (DLL).

1. Configure environment variables.
   ```
   export CPATH="/usr/include/hdf5/include/:/usr/include/hdf5/lib/"
   ```
2. Run the following command as the **root** user to create a soft link to the DLL. Add sudo before the following commands as a non-root user:

 Copyright © Huawei Technologies Co., Ltd.

```
ln -s /usr/include/hdf5/lib/libhdf5.so  /usr/lib/libhdf5.so
ln -s /usr/include/hdf5/lib/libhdf5_hl.so  /usr/lib/libhdf5_hl.so
```

Step 3 Install h5py.

Run the following command as the **root** user to install the h5py dependency:

```
pip3.7 install Cython
```

The h5py installation command is as follows:

```
pip3.7 install h5py==2.8.0
```

----End

Installing TensorFlow 1.15.0

TensorFlow 1.15 must be installed for operator development verification and training service development.

- For the x86 architecture, download the software package from the pip source. For details, see **https://www.tensorflow.org/install/pip?lang=python3**. Note that the instructions provided by the TensorFlow website are incorrect. To download the CPU version from the pip source, you need to explicitly specify **tensorflow-cpu**. Otherwise, the GPU version is downloaded by default. That is, change **tensorflow==1.15 —Release for CPU-only** to **tensorflow-cpu==1.15 —Release for CPU-only**. In addition, the installation command **pip3 install --user --upgrade tensorflow** described on the official website needs to be changed to **pip3.7 install tensorflow-cpu==1.15** as the **root** user and to **pip3.7 install tensorflow-cpu==1.15 --user** as a non-root user.
- For the AArch64 architecture, the pip source does not provide the corresponding version. Therefore, you need to use GCC 7.3.0 to compile TensorFlow 1.15.0. For details about the compilation procedure, see **https://www.tensorflow.org/install/source**. Pay attention to the following points:

After downloading the **tensorflow tag v1.15.0** source code, perform the following steps:

Step 1 Download the **nsync-1.22.0.tar.gz** source code package.

1. Go to the **tensorflow tag v1.15.0** source code directory, open the **tensorflow/workspace.bzl** file, and find the definition of **tf_http_archive** whose **name** is **nsync**.

```
  tf_http_archive(
      name = "nsync",
      sha256 = "caf32e6b3d478b78cff6c2ba009c3400f8251f646804bcb65465666a9cea93c4",
      strip_prefix = "nsync-1.22.0",
      system_build_file = clean_dep("//third_party/systemlibs:nsync.BUILD"),
      urls = [          "https://storage.googleapis.com/mirror.tensorflow.org/github.com/google/nsync/
archive/1.22.0.tar.gz",
          "https://github.com/google/nsync/archive/1.22.0.tar.gz",
      ],
  )
```

2. Download the **nsync-1.22.0.tar.gz** source code package from any path in **urls** and save it to any path.

Step 2 Modify the **nsync-1.22.0.tar.gz** source code package:

1. Go to the directory where **nsync-1.22.0.tar.gz** is stored and decompress the source code package. Find the decompressed **nsync-1.22.0** folder and the **pax_global_header** file.

 Copyright © Huawei Technologies Co., Ltd.

2. Edit the nsync-1.22.0/platform/c++11/atomic.h file.

   Add the following information to the end of the **NSYNC_CPP_START_** file:

```
#include "nsync_cpp.h"
#include "nsync_atomic.h"

NSYNC_CPP_START_

#define ATM_CB_() __sync_synchronize()

static INLINE int atm_cas_nomb_u32_ (nsync_atomic_uint32_ *p, uint32_t o, uint32_t n) {
  int result = (std::atomic_compare_exchange_strong_explicit (NSYNC_ATOMIC_UINT32_PTR_ (p),
&o, n, std::memory_order_relaxed, std::memory_order_relaxed));
  ATM_CB_();
  return result;
}
static INLINE int atm_cas_acq_u32_ (nsync_atomic_uint32_ *p, uint32_t o, uint32_t n) {
  int result = (std::atomic_compare_exchange_strong_explicit (NSYNC_ATOMIC_UINT32_PTR_ (p),
&o, n, std::memory_order_acquire, std::memory_order_relaxed));
  ATM_CB_();
  return result;
}
static INLINE int atm_cas_rel_u32_ (nsync_atomic_uint32_ *p, uint32_t o, uint32_t n) {
  int result = (std::atomic_compare_exchange_strong_explicit (NSYNC_ATOMIC_UINT32_PTR_ (p),
&o, n, std::memory_order_release, std::memory_order_relaxed));
  ATM_CB_();
  return result;
}
static INLINE int atm_cas_relacq_u32_ (nsync_atomic_uint32_ *p, uint32_t o, uint32_t n) {
  int result = (std::atomic_compare_exchange_strong_explicit (NSYNC_ATOMIC_UINT32_PTR_ (p),
&o, n, std::memory_order_acq_rel, std::memory_order_relaxed));
  ATM_CB_();
  return result;
}
```

Step 3 Compress the **nsync-1.22.0.tar.gz** source code package.

Compress the modified **nsync-1.22.0** folder and **pax_global_header** into a new **nsync-1.22.0.tar.gz** source code package (for example, **/tmp/nsync-1.22.0.tar.gz**).

Step 4 Generate a sha256sum verification code for the **nsync-1.22.0.tar.gz** source code package.

sha256sum /tmp/nsync-1.22.0.tar.gz

Run the preceding command to obtain the sha256sum verification code (a string of digits and letters).

Step 5 Change the **sha256sum** verification code and **urls**.

Go to the **tensorflow tag v1.15.0** source code directory, open the **tensorflow/workspace.bzl** file, find the definition of tf_http_archive whose name is nsync, enter the verification code obtained in **Step 4** after sha256=, and enter the first line of the list after urls=, enter the file:// index for storing the nsync-1.22.0.tar.gz file.

```
  tf_http_archive(
      name = "nsync",
      sha256 = "caf32e6b3d478b78cff6c2ba009c3400f8251f646804bcb65465666a9cea93c4",
      strip_prefix = "nsync-1.22.0",
      system_build_file = clean_dep("//third_party/systemlibs:nsync.BUILD"),
      urls = [
          "file:///tmp/nsync-1.22.0.tar.gz ",          "https://storage.googleapis.com/mirror.tensorflow.org/
github.com/google/nsync/archive/1.22.0.tar.gz",
          "https://github.com/google/nsync/archive/1.22.0.tar.gz",
      ],
  )
```

 Copyright © Huawei Technologies Co., Ltd.

Step 6 Continue to perform compilation from the official configuration build (**https://www.tensorflow.org/install/source**).

After the **./configure** command is executed, add the following build option to the **.tf_configure.bazelrc** configuration file.

```
build:opt --cxxopt=-D_GLIBCXX_USE_CXX11_ABI=0
```

Delete the following two lines:

```
build:opt --copt=-march=native
build:opt --host_copt=-march=native
```

Step 7 Proceed with the official compilation procedure (**https://www.tensorflow.org/install/source**).

----End

 Copyright © Huawei Technologies Co., Ltd.

4 API Reference

4.1 Overview

This document describes the external APIs of MindX DL. You can manage the vcjob jobs and view the NPU status.

Before calling Mind X DL APIs, ensure that you are familiar with the basic concepts and knowledge of Kubernetes, and that the Kubernetes authentication is implemented by yourself.

4.2 Description

4.2.1 API Communication Protocols

MindX DL APIs are invoked in Representational State Transfer (REST) mode. Volcano uses the URL of the Kubernetes API server. You are advised to enable HTTPS during Kubernetes installation. cAdvisor supports only HTTP, security hardening has been performed by default, and its port is open only in the Kubernetes cluster.

NOTE

You are advised to use the Kubernetes-encapsulated client to manage Volcano resources, which is more convenient. For details, see **Interconnection Programming Guide**.

4.2.2 Encoding Format

- Request and response packets are in JSON format (RFC 4627).

NOTE

Request packets received and response packets returned by APIs must be in JSON format. (The upload and download APIs are subject to API definitions.)

- The media type is **application/json**.
- All APIs use the UTF-8 encoding format.

 Copyright © Huawei Technologies Co., Ltd.

4.2.3 URLs

MindX DL APIs are designed based on the RESTful API architecture.

Representational State Transfer (REST) observes the entire network from the resource perspective. Resources are identified by Uniform Resource Identifiers (URIs) across the network and applications from a client obtain resources through Uniform Resource Locators (URLs).

NOTE

- A URL can contain path parameters. For example, in **https://localhost:26335/rest/uam/v1/roles/***{parm1}*/*{parm2}*, *parm1* and *parm2* are path parameters. You can also add query conditions by appending question marks (?) and ampersands (&) to the URL. For example, in **https://localhost:26335/rest/uam/v1/roles/{***parm1}/{parm2}?parm3=value3&parm4=value4*, *parm3* and *parm4* are query parameters, and *value3* and *value4* are their values, respectively.
- A URL cannot contain URL special characters (defined by RFC1738). Encode the URL if special characters are required.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Although a request URI is included in a request header, most programming languages or frameworks require the request URI to be separately transmitted, rather than being conveyed in a request message.

- **URI-scheme**:

  Protocol used to transmit requests. All APIs use HTTPS.

- **Endpoint**:

  Domain name or IP address of the server bearing the REST service endpoint. (Generally, the value is the IP address of the master node in the Kubernetes cluster, and the default port number is 6443.) You can obtain the value from the administrator.

- **resource-path**:

  Path in which the requested resource is located, that is, the API access path. For example, the **resource-path** of the API in **Reading All Volcano Jobs Under a Namespace** is **/apis/batch.volcano.sh/v1alpha1/namespaces/{namepace}/jobs**, in which *{namespace}* indicates the command space of the job.

- **query-string**:

  Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

The following shows the URI combination of the API in **Reading All Volcano Jobs Under a Namespace**. **{endpoint}** indicates the terminal node. Replace it with the actual value.

```
https://{endpoint}/apis/batch.volcano.sh/v1alpha1/{namespaces}/vcjob/jobs
```

 Copyright © Huawei Technologies Co., Ltd.

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

4.2.4 API Authentication

- For details about Kubernetes authentication, see **Kubernetes Documentation**.
- API authentication is disabled on cAdvisor by default. If you need to enable API authentication, see API **HTTP authentication**.

4.2.5 Requests

Request Methods

REST style regulates that resource-related operations must comply with HTTPS.

The request method can be GET, PUT, POST, or DELETE.

Table 4-1 Mapping between request methods and resource operations

Request Method	Resource Operation
GET	Obtain resources.
PUT	Update resources.
POST	Create resources.
DELETE	Delete resources.

NOTE

- If the requested URL does not support the operation, status code **405 (Method Not Allowed)** is returned.
- The PATCH request method is not supported.

For details about the constraints on each request method, see the following sections.

GET

[Application scenario] This method is used to obtain resources.

[Status code] If the request is successful, status code **200 (OK)** is returned.

[Constraints] The request method meets security and idempotence requirements.

NOTE

The security requirement means that the operation does not change the server resource status. The idempotence requirement means that relative status change is not allowed for the resource.

 Copyright © Huawei Technologies Co., Ltd.

POST

[Application scenario] This method is used to create resources in the scenario where operations cannot be expressed by CRUD (non-CRUD).

[Status code] If the resource is successfully created using POST, status code **200 (OK)**, **201 (Created)**, or **202 (Accepted)** is returned. If POST is performed successfully in non-CRUD scenarios, status code **200 (OK)** is returned.

[Constraints] The request method does not meet security or idempotence requirements.

PUT

[Application scenario] This method is used to fully update resources. If the object to be updated does not exist, the object will be created.

NOTE

For example, **PUT /users/admin** indicates that if user **admin** does not exist, the PUT method is used to create the user and set attributes for the user. If user **admin** exists, this method is used to replace all information about the user.

[Status code] If the resource is successfully created using PUT, status code **201 (Created)** or **202 (Accepted)** is returned. If the resource is successfully updated, status code **200 (OK)** is returned.

[Constraints] The request method meets idempotence requirements.

DELETE

[Application scenario] This method is used to delete resources.

[Status code] If the resource is successfully deleted, status code **204 (OK)** is returned. If the resource to be deleted does not exist, status code **404 (No Content)** is returned. If a service receives the request but the resource is not deleted immediately, status code **202 (Accepted)** is returned.

[Constraints] The request method meets idempotence requirements.

4.2.6 Responses

Unless otherwise specified, the returned result for each request is identified with a status code. For details, see **Status Codes**.

4.2.7 Status Codes

Status Code	Description
200 OK	The request is successful. The response header or message body will be returned with the response.
201 Created	When the POST or PUT operation is successful, status code **201 (Created)** is returned with the URI in the **Location** field of the new resource in the message header.

Copyright © Huawei Technologies Co., Ltd.

Status Code	Description
202 Accepted	The request has been accepted for processing, but the process has not been completed. A resource has been created according to the request and the request URI is returned with the **Location** field in the message header.
204 No Content	The server has processed the request but does not return any content.
206 Partial Content	The server has processed certain GET requests.
302 Found	The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client should continue to use the Request-URI for future requests.
303 See Other	The response to the request can be found under a different URI and should be retrieved using the GET method on that resource.
304 Not Modified	The document has not been modified as expected. The client has a cached document and has sent a conditional request (In most cases, the If-Modified-Since header is provided, indicating that the client uses only documents updated after a specified date). The server uses this error code to notify the client that the cached document is available.
400 Bad Request	1. The request cannot be understood by the server due to malformed syntax. The client should not send the request repeatedly without modifications. 2. Request parameters are incorrect.
401 Unauthorized	Used only in the HTTPS (BASIC authentication, DIGET authentication) authentication scenarios. If there are other authentication mechanisms, status code 403 is returned after authentication failure. The response must include the header WWW-Authenticate for user information query.
403 Forbidden	If the request method is not HEAD, the server must describe the reason for the refusal in the message body. If the cause cannot be disclosed, **404 Not Found** is returned.

 Copyright © Huawei Technologies Co., Ltd.

Status Code	Description
404 Not Found	The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent. The status code 410 should be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address. The status code 404 is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.
405 Method Not Allowed	You are not allowed to use the method specified in the request. The response must include an Allow header containing a list of valid methods for the requested resource. Since the PUT and DELETE methods will write resources on the server, the preceding methods are not supported or allowed by default by most servers. The server will return **405 (Method Not Allowed)** for this type of requests.
409 Conflict	The request cannot be completed due to a conflict with the current state of the resource. Conflicts are most likely to occur in response to a PUT request.
414 Request-URI Too Long	The URL can contain a maximum of 2083 characters.
429 Too Many Requests	The number of requests sent to the server in a given period of time has exceeded the threshold.
500 Internal Server Error	An unexpected error occurs on the server. As a result, the server cannot process the request.
502 Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
503 Service Unavailable	The server is currently unable to handle the request due to temporary overloading or maintenance of the server.

4.3 API Reference

MindX DL provides a job management component that depends on the Kubernetes platform.

 Copyright © Huawei Technologies Co., Ltd.

4.3.1 Data Structure

Table 4-2 Data structure of PodTemplate

Parameter	Mandatory	Type	Description
kind	Yes	String	A string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **PodTemplate**.
apiVersion	Yes	String	Versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
metadata	Yes	**ObjectMeta** object	-
template	Yes	**PodTemplateSpec** object	-

Table 4-3 Data structure of Pod

Parameter	Mandatory	Type	Description
kind	Yes	String	A string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **Pod**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	Versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
metadata	Yes	**ObjectMeta** object	-
spec	Yes	**podSpec** object	-
status	No	**PodStatus** object	Most recently observed status of the pod.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-4 Data structure of the PodStatus field

Parameter	Mandatory	Type	Description
phase	No	String	Current condition of the pod. **NOTE** Pod states include: - Pending: means the pod has been accepted by the system, but one or more of the containers has not been started. This includes time before being bound to a node, as well as time spent pulling images onto the host. - Running: The pod has been bound to a node and all of the containers have been started. At least one container is still running or is in the process of being restarted. Succeeded: All containers in the pod have voluntarily terminated with a container exit code of 0, and the system is not going to restart any of these containers. Failed: All containers in the pod have terminated, and at least one container has terminated in a failure (exited with a non-zero exit code or was stopped by the system). Unknown: The state of the pod could not be obtained for some reasons, typically due to an error in communicating with the host of the pod.
conditions	No	Array of **PodConditions** objects	Current service state of the pod.
message	No	String	A human readable message indicating details about why the pod is in this condition.
reason	No	String	A brief CamelCase message indicating details about why the pod is in this state. e.g. 'OutOfDisk'
hostIP	No	String	IP address of the host to which the pod is assigned. Empty if not yet scheduled.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
podIP	No	String	IP address allocated to the pod. Routable at least within the cluster. Empty if not yet allocated.
startTime	No	String	RFC 3339 date and time at which the object was acknowledged by the Kubelet. This is before the Kubelet pulled the container image(s) for the pod.
containerStatuses	No	Array of **containerStatuses** objects	The list has one entry per container in the manifest. Each entry is currently the output of container inspect.
initContainerStatuses	No	Array of **containerStatuses** objects	The list has one entry per init container in the manifest. The most recent successful init container will have ready = true, the most recently started container will have startTime set.
qosClass	No	String	The Quality of Service (QoS) classification assigned to the pod based on resource requirements. Can be: - Guaranteed - Burstable - BestEffort
podNetworks	No	Array of **PodNetworkInterface** objects	Complete list of Networks attached to this pod.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-5 Data structure of the PodConditions field

Parameter	Mandatory	Type	Description
type	No	String	Type of the condition. Currently only Ready. **Resizing** - An user trigger resize of pvc has been started **NOTE** Pod conditions include: - PodScheduled: represents status of the scheduling process for this pod. - Ready: pod is able to service requests and should be added to the load balancing pools of all matching services. - Initialized: all init containers in the pod have started successfully. - Unschedulable: the scheduler cannot schedule the pod right now, for example due to insufficient resources in the cluster.
status	No	String	Status of the condition. Can be **True**, **False**, or **Unknown**.
lastProbeTime	No	String	Last time we probed the condition.
lastTransitionTime	No	String	Last time the condition transitioned from one status to another.
reason	No	String	Unique, one-word, CamelCase reason for the condition's last transition.
message	No	String	Human-readable message indicating details about last transition.

Table 4-6 Data structure of the containerStatuses field

Parameter	Mandatory	Type	Description
name	Yes	String	This must be a DNS_LABEL. Each container in a pod must have a unique name. Cannot be updated.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
state	No	**ContainerState** object	Details about the container's current condition.
lastState	No	**ContainerState** object	Details about the container's last termination condition.
ready	No	Boolean	Specifies whether the container has passed its readiness probe.
restartCount	No	Integer	The number of times the container has been restarted, currently based on the number of dead containers that have not yet been removed. Note that this is calculated from dead containers. However, those containers are subject to garbage collection. This value will get capped at 5 by GC.
image	Yes	String	The image the container is running.
imageID	No	String	ID of the container's image.
containerID	No	String	Container's ID in the format 'docker://'.

Table 4-7 Data structure of the ContainerState field

Parameter	Mandatory	Type	Description
waiting	No	**ContainerStateWaiting** object	Details about a waiting container.
running	No	**ContainerStateRunning** object	Details about a running container.
terminated	No	**terminated** object	Details about a terminated container.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-8 Data structure of the ContainerStateWaiting field

Parameter	Mandatory	Type	Description
reason	No	String	(Brief) Reason the container is not yet running.
message	No	String	Message regarding why the container is not yet running.

Table 4-9 Data structure of the ContainerStateRunning field

Parameter	Mandatory	Type	Description
startedAt	No	String	Time at which the container was last (re-)started.

Table 4-10 Data structure of the terminated field

Parameter	Mandatory	Type	Description
exitCode	No	Integer	Exit status from the last termination of the container.
signal	No	Integer	Signal from the last termination of the container.
reason	No	String	(Brief) reason from the last termination of the container.
message	No	String	Message regarding the last termination of the container.
startedAt	No	String	Time at which previous execution of the container started.
finishedAt	No	String	Time at which the container last terminated.
containerID	No	String	Container's ID in the format 'docker://'

 Copyright © Huawei Technologies Co., Ltd.

Table 4-11 Data structure of the ObjectMeta field

Parameter	Mandatory	Type	Description
name	Yes	String	Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. The task name and job name could not be same. 0 characters < name length ≤ 63 characters. The name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
clusterName	No	String	The name of the cluster which the object belongs to. This is used to distinguish resources with same name and namespace in different clusters. This field is not set anywhere right now and apiserver is going to ignore it if set in create or update request.
initializers	No	**initializers** object	An initializer is a controller which enforces some system invariant at object creation time. This field is a list of initializers that have not yet acted on this object. If nil or empty, this object has been completely initialized. Otherwise, the object is considered uninitialized and is hidden (in list/watch and get calls) from clients that haven't explicitly asked to observe uninitialized objects. When an object is created, the system will populate this list with the current set of initializers. Only privileged users may set or modify this list. Once it is empty, it may not be modified further by any user.
enable	No	Boolean	Enable identify whether the resource is available.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
generateName	No	String	An optional prefix used by the server to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different from the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409. Instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. 0 characters < generated name length ≤ 253 characters. The generated name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
namespace	No	String	Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. 0 characters < namespace length ≤ 63 characters. The namespace must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
selfLink	No	String	A URL representing this object. Populated by the system. Read-only. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.
uid	No	String	UID is the unique in time and space value for this object. It is typically generated by the server on successful creation of a resource and is not allowed to change on PUT operations. Populated by the system. Read-only. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
resourceVersion	No	String	An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. Clients must treat these values as opaque and passed unmodified back to the server. They may only be valid for a particular resource or set of resources. Populated by the system. Read-only. Value must be treated as opaque by clients. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.
generation	No	Integer	A sequence number representing a specific generation of the desired state. Currently only implemented by replication controllers. Populated by the system. Read-only.
creationTimestamp	No	String	A timestamp representing the server time when this object was created. It is not guaranteed to be set in happens-before order across separate operations. Clients may not set this value. It is represented in RFC3339 form and is in UTC. Populated by the system. Read-only. Null for lists. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.
deletionTimestamp	No	String	RFC 3339 date and time at which this resource will be deleted. This field is set by the server when a graceful deletion is requested by the user, and is not directly settable by a client. The resource will be deleted (no longer visible from resource lists, and not reachable by name) after the time in this field. Once set, this value may not be unset or be set further into the future, although it may be shortened or the resource may be deleted prior to this time. For example, a user may request that a pod is deleted in 30 seconds. The Kubelet will react by sending a graceful termination signal to the containers in the pod. Once the resource is deleted in the API, the Kubelet will send a hard termination signal to the container. If not set, graceful deletion of the object has not been requested. Populated by the system when a graceful deletion is requested. Read-only.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
deletionGracePeriodSeconds	No	Integer	Number of seconds allowed for this object to gracefully terminate before it will be removed from the system. Only set when deletionTimestamp is also set. May only be shortened. Read-only.
labels	No	Object	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services.
annotations	No	**annotations** object	An unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. **NOTE** Each resource type has required annotations. For details, see the description in APIs of specific resources.
ownerReferences	No	Array of **ownerReferences** objects	List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.
finalizers	No	Array of strings	Must be empty before the object is deleted from the registry. Each entry is an identifier for the responsible component that will remove the entry from the list. If the deletionTimestamp of the object is non-nil, entries in this list can only be removed.

Table 4-12 Data structure of the annotations field

Parameter	Mandatory	Type	Description
obssidecar-injector-webhook/inject	No	Boolean	This parameter is required when a pod is created by mounting into an OBS bucket.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
pod.logcollection.kubernetes.io	No	Array of strings	List of containers whose standard output logs need to be collected. If this parameter is left blank, standard output logs of all containers will be collected. • Example 1: Collect standard output logs of all containers. pod annotation: log.stdoutcollection.kubernetes.io: {"collectionContainers":[]} • Example 2: Collect standard output logs of container0, where container0 is the container name. pod annotation: log.stdoutcollection.kubernetes.io: {"collectionContainers": ["container0"]}
paas.storage.io/cryptKeyId	No	String	Encryption key ID. This parameter is required only when the storage class is SFS or EVS and an encrypted volume needs to be created. You can obtain the key ID from the Security Console by choosing **Data Encryption Workshop** > **Key Management Service**.
paas.storage.io/cryptAlias	No	String	Encryption key alias. This parameter is required only when the storage class is SFS and an encrypted volume needs to be created. You can obtain the key alias from the Security Console by choosing **Data Encryption Workshop** > **Key Management Service**.
paas.storage.io/cryptDomainId	No	String	Domain ID of a tenant. This parameter is required only when the storage class is SFS and an encrypted volume needs to be created.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-13 Data structure of the initializers field

Parameter	Mandatory	Type	Description
pending	No	Array of **pending** objects	Pending is a list of initializers that must execute in order before this object is visible. When the last pending initializer is removed, and no failing result is set, the initializers struct will be set to nil and the object is considered as initialized and visible to all clients.
result	No	**status** object	If result is set with the Failure field, the object will be persisted to storage and then deleted, ensuring that other clients can observe the deletion.

Table 4-14 Data structure of the pending field

Parameter	Mandatory	Type	Description
name	No	String	Name of the process that is responsible for initializing this object.

Table 4-15 Data structure of the ownerReferences field

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version of the referent.
blockOwnerDeletion	No	Boolean	If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
kind	Yes	String	Kind of the referent.
name	Yes	String	Name of the referent.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
uid	No	String	UID of the referent.
controller	No	Boolean	If true, this reference points to the managing controller.

Table 4-16 Data structure of the spec field

Parameter	Mandatory	Type	Description
replicas	No	Integer	The number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Value range: ≥ 0. Default: 1
minReadySeconds	No	Integer	Minimum number of seconds for which a newly created pod should be ready without any of its containers crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
template	Yes	**PodTemplateSpec** object	-
selector	Yes	Object	A label query over pods that should match the Replicas count. If Selector is empty, it is defaulted to the labels present on the Pod template. Label keys and values that must match in order to be controlled by this replication controller, if empty defaulted to labels on Pod template.

Table 4-17 Data structure of the status field

Parameter	Mandatory	Type	Description
replicas	No	Integer	The most recently observed number of replicas.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
availableReplicas	No	Integer	The number of available replicas (ready for at least minReadySeconds) for this replication controller.
readyReplicas	No	Integer	The number of ready replicas for this replication controller.
conditions	No	**condition** object	Represents the latest available observations of a replication controller's current state.
observedGeneration	No	Integer	Reflects the generation of the most recently observed replication controller.
FullylabeledReplicas	No	Obeject	-

Table 4-18 Data structure of the PodTemplateSpec field

Parameter	Mandatory	Type	Description
metadata	No	**ObjectMeta** object	-
spec	No	**podSpec** object	-

Table 4-19 Data structure of the condition field

Parameter	Mandatory	Type	Description
lastTransitionTime	No	String	The last time the condition transitioned from one status to another.
message	No	String	A human readable message indicating details about the transition.
reason	No	String	The reason for the condition's last transition.
status	No	String	Status of the condition, one of True, False, Unknown.
type	No	String	Type of replication controller condition.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-20 Data structure of the podSpec field

Parameter	Mandatory	Type	Description
volumes	No	Array of **volumes** objects	List of volumes that can be mounted by containers belonging to the pod.
affinity	No	**affinity** object	If specified, the pod's scheduling constraints. **NOTE** Affinity settings cannot be configured. By default, the soft anti-affinity settings are used.
containers	Yes	Array of **containers** objects	List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a pod. Cannot be updated.
restartPolicy	No	String	Restart policy for all containers within the pod. Value: • Always • OnFailure • Never Default: Always.
priority	No	Integer	Pod priority. A larger value indicates a higher priority. The default value is 0. Value range: [-10, 10]

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
terminationGracePeriodSeconds	No	Integer	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be a non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. Defaults to 30 seconds.
activeDeadlineSeconds	No	Integer	Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer. Value range of this parameter: > 0.
dnsPolicy	No	String	Set DNS policy for containers within the pod. Value: • ClusterFirst • Default **NOTE** **dnsPolicy** cannot be set to **Default**. Default: ClusterFirst.
hostAliases	No	Array of **hostAliases** objects	HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified. This is only valid for non-hostNetwork pods.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
serviceAccountName	No	String	Name of the ServiceAccount used to run this pod. 0 characters < service account name length ≤ 253 characters. The service account name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?. **NOTE** This field cannot be set because **serviceaccount** is not supported.
serviceAccount	No	String	DeprecatedServiceAccount is a depreciated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead. **NOTE** This field cannot be set because **serviceaccount** is not supported.
schedulerName	No	String	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler. **NOTE** The scheduler name cannot be specified.
nodeName	No	String	A request to schedule this pod onto a specific node. If it is non-empty, the scheduler simply schedules this pod onto that node, assuming that it fits resource requirements. 0 characters < node name length ≤ 253 characters. The node name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?. **NOTE** The node name cannot be specified.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
nodeSelector	No	Object	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. **NOTE** The node selector cannot be configured.
automountServiceAccountToken	No	Boolean	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.
hostNetwork	No	Boolean	Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Defaults to false. (This parameter cannot be configured.) **NOTE** The host network cannot be used.
hostPID	No	Boolean	A flag indicating whether to use the host's pid namespace. This parameter is optional and defaults to false. **NOTE** The host PID namespaces cannot be used.
hostIPC	No	Boolean	A flag indicating whether to use the host's ipc namespace. This parameter is optional and defaults to false. **NOTE** The host IPC namespaces cannot be used.
securityContext	No	**PodSecurityContext** object	SecurityContext holds pod-level security attributes and common container settings. Defaults to empty.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
imagePullSecrets	No	Array of **imagePullSecrets** objects	An optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. **NOTE** If you select an image from the **My Images** tab page of the SWR console, this parameter is required.
initContainers	No	Array of **containers** objects	List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, or Liveness probes. The resourceRequirements of an init container is taken into account during scheduling by finding the highest request/limit for each resource type, and then using the maximum of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed.
hostname	No	String	Specifies the hostname of the Pod. If not specified, the pod's hostname will be set to a system-defined value.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
subdomain	No	String	If specified, the fully qualified Pod hostname will be"<hostname>.<subdomain>.<pod namespace>.svc<cluster domain>". If not specified, the pod will not have a domainname at all.
tolerations	No	**tolerations** object	If specified, the pod's tolerations. **NOTE** The **tolerations** field cannot be configured.
priorityClassName	No	String	If specified, indicates the pod's priority. "SYSTEM" is a special keyword which indicates the highest priority. Any other name must be defined by creating a PriorityClass object with that name. If not specified, the pod priority will be default or zero if there is no default.

Table 4-21 Data structure of the volumes field

Parameter	Mandatory	Type	Description
name	Yes	String	Volume name. Must be a DNS_LABEL and unique within the pod. 0 characters < volume name length ≤ 63 characters. The volume name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
secret	No	**SecretVolumeSource** object	Secret represents a secret that should populate this volume.
persistentVolumeClaim	No	**PersistentVolumeClaimVolumeSource** object	PersistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
localDir	No	**LocalDirVolumeSource** object	LocalDir represents a LocalDir volume that is created by LVM and mounted into the pod
emptyDir	No	**emptyDir** object	Used for creating a pod mounted into a local volume.

Table 4-22 Data structure of the containers field

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated. 0 characters < container name length ≤ 63 characters. The container name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?. **Cannot be updated.**
image	Yes	String	Container image address.
command	No	Array of strings	Entrypoint array. Not executed within a shell. The container image's entrypoint is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, for example, $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. **Cannot be updated.**

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
args	No	Array of strings	Arguments to the entrypoint. The container image's cmd is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, for example, $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. **Cannot be updated.**
workingDir	No	String	Container's working directory. Defaults to Container's default. Defaults to image's default. **Cannot be updated.**
ports	No	Array of **ContainerPort** objects	List of ports to expose from the container. **Cannot be updated.**
env	No	Array of **EnvVar** objects	List of environment variables to set in the container. **Cannot be updated.**
envFrom	No	Array of **EnvFromSource** objects	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. **Cannot be updated.**
resources	No	**ResourceRequirements** object	Minimum resources the volume should have. **Cannot be updated.**
volumeMounts	No	Array of **volumeMounts** objects	Pod volumes to mount into the container's filesystem. **Cannot be updated.**

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
volumeDevices	No	Array of **volumeDevice** objects	VolumeDevices is the list of block devices to be used by the container. This is an alpha feature and may change in the future.
livenessProbe	No	**Probe** object	Periodic probe of container liveness. Container will be restarted if the probe fails. **Cannot be updated.**
readinessProbe	No	**Probe** object	Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. **Cannot be updated.**
lifecycle	No	**lifecycle** object	Actions that the management system should take in response to container lifecycle events. **Cannot be updated.**
terminationMessagePath	No	String	Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Defaults to /dev/termination-log. Cannot be updated. **Cannot be updated.**

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
terminationMessagePolicy	No	String	Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. **Defaults to File.** **Cannot be updated.** **NOTE** Value options: -File: default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits. -FallbackToLogsOnError: will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents.
imagePullPolicy	No	String	Image pull policy. Defaults to Always if the :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. Value: • Always • Never • IfNotPresent **Cannot be updated.** **NOTE** Only **Always** is supported.
securityContext	No	**security Context** object	Security options the pod should run with
stdin	No	Boolean	A flag indicating whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. **Default is false.**

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
stdinOnce	No	Boolean	A flag indicating whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true, the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container process that reads from stdin will never receive an EOF. **Default is false.**
tty	No	Boolean	A flag indicating whether this container should allocate a TTY for itself, also requires 'stdin' to be true. **Default is false.**

Table 4-23 Data structure of the PodSecurityContext field

Parameter	Mandatory	Type	Description
seLinuxOptions	No	**seLinuxOptions** object	-
runAsUser	No	Integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Value length: > 0 characters.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
runAsNonRoot	No	Boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
supplementalGroups	No	Array of integers	A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.
fsGroup	No	Integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
fsOwner	No	Integer	A special supplemental owner that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning UID will be the FSOwner 2. The setgid bit is set (new files created in the volume will be owned by FSOwner). 3. The permission bits are OR'd with rw------- If unset, the Kubelet will not modify the ownership and permissions of any volume.

Table 4-24 Data structure of the imagePullSecrets field

Parameter	Mandatory	Type	Description
name	No	String	Name of the referent. **NOTICE** If you select an image from the **My Images** tab page of the SWR console, the value of this parameter must be set to **imagepull-secret**.

Table 4-25 Data structure of the SecretVolumeSource field

Parameter	Mandatory	Type	Description
secretName	No	String	Name of a secret in the pod's namespace.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
items	No	**items(KeyToPath)** object	If unspecified, each key-value pair in the Data field of the referenced. Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.
defaultMode	No	Integer	Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
optional	No	Boolean	Specify whether the Secret or it is keys must be defined

Table 4-26 Data structure of the PersistentVolumeClaimVolumeSource field

Parameter	Mandatory	Type	Description
claimName	No	String	Name of a PersistentVolume-Claim in the same namespace as the pod using this volume.
readOnly	No	Boolean	ReadOnly here will force the ReadOnly setting in VolumeMounts. Value: • true • false **Default: false.**

 Copyright © Huawei Technologies Co., Ltd.

Table 4-27 Data structure of the items(KeyToPath) field

Parameter	Mandatory	Type	Description
key	No	String	The key to project.
path	No	String	The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.
mode	No	Integer	Mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Table 4-28 Data structure of the ContainerPort field

Parameter	Mandatory	Type	Description
name	No	String	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services. 0 characters < name length ≤ 15 characters. The name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
hostPort	No	Integer	Number of the port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this. Value range: [1, 65535]. **NOTE** The **hostPort** field cannot be configured.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
containerPort	No	Integer	Number of the port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536. Value range: [1, 65535].
protocol	No	String	Protocol for port. Value: • TCP • UDP Default: TCP.
hostIP	No	String	What host IP to bind the external port to. **NOTE** The **hostIP** field cannot be configured.

Table 4-29 Data structure of the EnvVar field

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the environment variable. Must be a C_IDENTIFIER.
value	No	String	Variable references $(VAR_NAME) are expanded using the previous defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, for example, $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	No	EnvVarSource object	Source for the environment variable's value. Cannot be used if value is not empty.

Copyright © Huawei Technologies Co., Ltd.

Table 4-30 Data structure of the ResourceRequirements field

Parameter	Mandatory	Type	Description
limits	No	Array of ResouceName objects	Maximum amount of compute resources allowed. **NOTE** The values of **limits** and **requests** must be the same. Otherwise, an error is reported.
requests	No	Array of ResouceName objects	Minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Cloud Container Instance (CCI) has limitation on pod specifications. For details, see **Pod Specifications** in **Usage Constraints**.

Table 4-31 Available values of the ResourceName field

Parameter	Mandatory	Type	Description
storage	No	String	Volume size, in bytes (e,g. 5Gi = 5GiB = 5 * 1024 * 1024 * 1024)
cpu	No	String	CPU size, in cores. (500m = .5 cores)
memory	No	String	Memory size, in bytes. (500Gi = 500GiB = 500 * 1024 * 1024 * 1024)
localdir	No	String	Local Storage for LocalDir, in bytes. (500Gi = 500GiB = 500 * 1024 * 1024 * 1024)
nvidia.com/gpu-tesla-v100-16GB	No	String	NVIDIA GPU resource, the type may change in different environments, in production environment is nvidia.com/gpu-tesla-v100-16GB now. The value must be an integer and not less than 1.

Copyright © Huawei Technologies Co., Ltd.

Table 4-32 Data structure of the volumeMounts field

Parameter	Mandatory	Type	Description
name	Yes	String	This must match the Name of a Volume. 0 characters < name length ≤ 253 characters The name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
readOnly	No	Boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Value: ● true ● false Default: false.
mountPath	No	String	Path within the container at which the volume should be mounted. Value length: > 0 characters.
subPath	No	String	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
mountPropagation	No	String	MountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationHostTo-Container is used. This field is alpha in 1.8 and can be reworked or removed in a future release. **NOTE** The available values include: - HostToContainer: means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container will not be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - Bidirectional: means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology)

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
extendPathMode	No	String	Extend the volume path by appending the pod metadata to the path according to specified pattern, which provides a way of directory isolation and helps prevent the writing conflict between different pods. **NOTE** The available values include: - PodUID: Include PodUID in path - PodName: Include Pod full name in path - PodUID/ContainerName: Include Pod UID and container name in path - PodName/ContainerName: Include Pod full name and container name in path

Table 4-33 Data structure of volumeDevice

Parameter	Type	Description
name	String	Name must match the name of a persistentVolume-Claim in the pod.
devicePath	String	DevicePath is the path inside of the container that the device will be mapped to.

Table 4-34 Data structure of the Probe field

Parameter	Mandatory	Type	Description
exec	No	**exec** object	Only one option should be specified. Exec specifies the action to take.
initialDelaySeconds	No	Integer	Number of seconds after the container has started before liveness probes are initiated. Value range: ≥ 0.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
timeoutSeconds	No	Integer	Number of seconds after which the probe times out. Value range: ≥ 0. Default: 1.
periodSeconds	No	Integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1. Value range: ≥ 0. Default: 10.
successThreshold	No	Integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1. Value range: ≥ 0. Default: 1.
failureThreshold	No	Integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1. Value range: ≥ 0. Default: 3.

Table 4-35 Data structure of the lifecycle field

Parameter	Mandatory	Type	Description
postStart	No	**Handler** object	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
preStop	No	**Handler** object	PreStop is called immediately before a container is terminated. The container is terminated after the handler completes. The reason for termination is passed to the handler. Regardless of the outcome of the handler, the container is eventually terminated. Other management of the container blocks until the hook completes.

Table 4-36 Data structure of the securityContext field

Parameter	Mandatory	Type	Description
capabilities	No	**capabilities** object	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
privileged	No	Boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Value: • true • false Default: false. **NOTE** This parameter cannot be set to **True**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
seLinuxOptions	No	**seLinuxOptions** object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	No	Integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsNonRoot	No	Boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Value: • true • false
readOnlyRootFilesystem	No	Boolean	Whether this container has a read-only root filesystem. **Default is false.**

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
allowPrivilegeEscalation	No	Boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN

Table 4-37 Data structure of the seLinuxOptions field

Parameter	Mandatory	Type	Description
user	No	String	SELinux user label that applies to the container.
role	No	String	SELinux role label that applies to the container.
type	No	String	SELinux type label that applies to the container.
level	No	String	SELinux level label that applies to the container.

Table 4-38 Data structure of the items field

Parameter	Mandatory	Type	Description
path	No	String	Relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
fieldRef	No	**ObjectFieldSelector** object	-

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
resourceFieldRef	No	**ResourceFieldSelector** object	Selects a resource of the container: only resources limits and requests. (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Table 4-39 Data structure of the EnvVarSource field

Parameter	Mandatory	Type	Description
fieldRef	No	**ObjectFieldSelector** object	Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations,spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP.
resourceFieldRef	No	**ResourceFieldSelector** object	Selects a resource of the container: only resources limits and requests. (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.
configMapKeyRef	No	**ConfigMapKeySelector** object	Selects a key of a ConfigMap.
secretKeyRef	No	**SecretKeySelector** object	Selects a key of a secret in the pod's namespace
processResourceFieldRef	No	**ProcessResourceFieldSelector** object	Selects a resource of the process: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-40 Data structure of the exec field

Parameter	Mandatory	Type	Description
command	No	Array of strings	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply executed, it is not run inside a shell, so traditional shell instructions ('|', etc) do not work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Table 4-41 Data structure of Handler

Parameter	Mandatory	Type	Description
exec	No	**exec** object	Only one option should be specified. Exec specifies the action to take.

Table 4-42 Data structure of the capabilities field

Parameter	Mandatory	Type	Description
add	No	Array of strings	Added capabilities.
drop	No	Array of strings	Removed capabilities.

Table 4-43 Data structure of the ObjectFieldSelector field

Parameter	Mandatory	Type	Description
apiVersion	No	String	Version of the schema the FieldPath is written in terms of. Defaults to "v1".
fieldPath	No	String	Path of the field to select in the specified API version.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-44 Data structure of the ResourceFieldSelector field

Parameter	Mandatory	Type	Description
containerName	No	String	Container name: required for volumes, optional for env vars.
resource	Yes	String	Required: resource to select.
divisor	No	String	Specifies the output format of the exposed resources, defaults to "1".

Table 4-45 Data structure of the ConfigMapKeySelector field

Parameter	Mandatory	Type	Description
name	No	String	The ConfigMap name to select from
key	No	String	Key to be selected.
optional	No	String	Specify whether the ConfigMap or it is key must be defined

Table 4-46 Data structure of the SecretKeySelector field

Parameter	Mandatory	Type	Description
name	No	String	Secret name to be selected.
key	No	String	Key to be selected.
optional	No	String	Whether the secret or its key must be defined.

Table 4-47 Data structure of the ProcessResourceFieldSelector field

Parameter	Mandatory	Type	Description
processName	No	String	Process name: required for volumes, optional for env vars
resource	Yes	String	Required: resource to select.
divisor	No	Integer	Specifies the output format of the exposed resources, defaults to "1".

 Copyright © Huawei Technologies Co., Ltd.

Table 4-48 Data structure of the EnvFromSource field

Parameter	Mandatory	Type	Description
prefix	No	String	An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
configMapRef	No	**ConfigMapEnvSource** object	The ConfigMap to select from
secretRef	No	**SecretEnvSource** object	The Secret to select from

Table 4-49 Data structure of the ConfigMapEnvSource field

Parameter	Mandatory	Type	Description
name	No	String	The ConfigMap to select from
optional	No	String	Specify whether the ConfigMap must be defined

Table 4-50 Data structure of the SecretEnvSource field

Parameter	Mandatory	Type	Description
name	No	String	Secret name to be selected.
optional	No	String	Whether the secret must be defined.

Table 4-51 Data structure of the add field

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the resource.
namespaced	No	Boolean	A flag indicating whether a resource is namespaced or not. Default: false.
kind	No	String	Kind of the resource.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-52 Data structure of the affinity field

Parameter	Mandatory	Type	Description
nodeAffinity	No	**nodeAffinity** object	Describes node affinity scheduling rules for the pod.
podAffinity	No	**podAffinity** object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
podAntiAffinity	No	**podAffinity** object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Table 4-53 Data structure of the nodeAffinity field

Parameter	Mandatory	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	No	**preferredDuringSchedulingIgnoredDuringExecution** object	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
requiredDuringSchedulingIgnoredDuringExecution	No	**requiredDuringSchedulingIgnoredDuringExecution** object	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Table 4-54 Data structure of the podAffinity field

Parameter	Mandatory	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	No	**preferredDuringSchedulingIgnoredDuringExecution** object	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which match the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
requiredDuringSchedulingIgnoredDuringExecution	No	**podAffinityTerm** object	NOT YET IMPLEMENTED. TODO: Uncomment field once it is implemented. If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system will try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied. RequiredDuringSchedulingRequiredDuringExecution []PodAffinityTerm json:"requiredDuringSchedulingRequiredDuringExecution,omitempty" If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-55 Data structure of the preferredDuringSchedulingIgnoredDuringExecution field

Parameter	Mandatory	Type	Description
preference	No	**preference** object	A node selector term, associated with the corresponding weight.
weight	No	Integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

Table 4-56 Data structure of the requiredDuringSchedulingIgnoredDuringExecution field

Parameter	Mandatory	Type	Description
nodeSelectorTerms	No	**preference** object	Required. A list of node selector terms. The terms are ORed.

Table 4-57 Data structure of the preference field

Parameter	Mandatory	Type	Description
matchExpressions	No	**matchExpressions** object	Required. A list of node selector requirements. The requirements are ANDed.

Table 4-58 Data structure of the matchExpressions field

Parameter	Mandatory	Type	Description
key	No	String	The label key that the selector applies to.
operator	No	String	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
values	No	String	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Table 4-59 Data structure of the preferredDuringSchedulingIgnoredDuringExecution field

Parameter	Mandatory	Type	Description
podAffinityTerm	No	**podAffinityTerm** object	Required. A pod affinity term, associated with the corresponding weight.
weight	No	Integer	Weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

Table 4-60 Data structure of the podAffinityTerm field

Parameter	Mandatory	Type	Description
labelSelector	No	**labelSelector** object	A label query over a set of resources, in this case pods.
namespaces	No	Array of strings	Namespaces specifies which namespaces the labelSelector applies to (matches against); null or empty list means "this pod's namespace".

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
topologyKey	No	String	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. For PreferredDuringScheduling pod anti-affinity, empty topologyKey is interpreted as "all topologies" ("all topologies" here means all the topologyKeys indicated by scheduler command-line argument --failure-domains); for affinity and for RequiredDuringScheduling pod anti-affinity, empty topologyKey is not allowed.

Table 4-61 Data structure of the labelSelector field

Parameter	Mandatory	Type	Description
matchExpressions	No	Array of **LabelSelectorRequirement** objects	MatchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	No	Object	MatchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-62 Data structure of the LabelSelectorRequirement field

Parameter	Mandatory	Type	Description
key	No	String	Key is the label key that the selector applies to.
operator	No	String	Operator represents a key's relationship to a set of values. **Valid operators are In, NotIn, Exists and DoesNotExist.**
values	No	Array of strings	Values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Table 4-63 Data structure of the hostAliases field

Parameter	Mandatory	Type	Description
hostnames	No	Array of strings	Hostnames for the above IP address.
ip	No	String	IP address of the host file entry.

Table 4-64 Data structure of the tolerations field

Parameter	Mandatory	Type	Description
effect	No	String	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	No	String	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
operator	No	String	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
tolerationSeconds	No	Integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	No	String	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

Table 4-65 Data structure of DeleteOptions

Parameter	Mandatory	Type	Description
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **Namespace**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
gracePeriodSeconds	No	Integer	The duration in seconds before the object should be deleted. Value must be a non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. The value **0** indicates to delete immediately. Value range of this parameter: > 0.
preconditions	No	**preconditions** object	Must be fulfilled before a deletion is carried out. If not possible, a 409 Conflict status will be returned.
orphanDependents	No	Boolean	Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
propagationPolicy	No	String	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. **NOTICE** Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Table 4-66 Data structure of the preconditions field

Parameter	Mandatory	Type	Description
uid	No	String	Specifies the target UID.

Table 4-67 Data structure of PodNetworkInterface

Parameter	Type	Description
name	String	Name of the interface inside the pod
network	String	Name of the attached network
iP	Array of strings	IP address(both v4 and v6) of this interface

 Copyright © Huawei Technologies Co., Ltd.

Table 4-68 Data structure of v1.PodList

Parameter	Type	Description
kind	String	A string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
apiVersion	String	Versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadataString	**ListMeta** object	-
items	Array of **Pod** objects	List of pods.

Table 4-69 Data structure of v1.PodTemplateList

Parameter	Type	Description
kind	String	A string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
apiVersion	String	Versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadata	**ListMeta** object	-
items	Array of **PodTemplate** objects	List of pod templates.

Table 4-70 Data structure of the status field

Parameter	Type	Description
phase	String	Current condition of the pod.
conditions	**PodConditions** object	Current service state of the pod.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
message	String	A human readable message indicating details about why the pod is in this condition.
reason	String	A brief CamelCase message indicating details about why the pod is in this state. e.g. 'OutOfDisk'
hostIP	String	IP address of the host to which the pod is assigned. Empty if not yet scheduled.
podIP	String	IP address allocated to the pod. Routable at least within the cluster. Empty if not yet allocated.
startTime	String	RFC 3339 date and time at which the object was acknowledged by the Kubelet. This is before the Kubelet pulled the container image(s) for the pod.
containerStatuses	**containerStatuses** object	The list has one entry per container in the manifest. Each entry is currently the output of container inspect.

Table 4-71 Data structure of the metadata field

Parameter	Type	Description
selfLink	String	SelfLink is a URL representing this object. Populated by the system. Read-only.
resourceVersion	String	String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-72 Data structure of the objectReference field

Parameter	Type	Description
kind	String	Kind of the referent.
namespace	String	Namespace of the referent.
name	String	Name of the referent.
uid	String	UID of the referent.
apiVersion	String	API version of the referent.
resourceVersion	String	Specific resourceVersion to which this reference is made, if any.
fieldPath	String	Path of the field to select in the specified API version.

Table 4-73 Data structure of the status field

Parameter	Type	Description
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadata	**ListMeta** object	Standard list metadata.
status	String	Status of the operation. One of: "Success" or "Failure".
message	String	A human-readable description of the status of this operation.
reason	**reason** object	A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available. A Reason clarifies an HTTP status code but does not override it.
details	**StatusDetails** object	Extended data associated with the reason. Each reason may define its own extended details. This field is optional and the data returned is not guaranteed to conform to any schema except that defined by the reason type.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
code	Integer	Suggested HTTP return code for this status, 0 if not set.

Table 4-74 Data structure of StatusDetails

Parameter	Type	Description
causes	Array of **StatusCause** objects	The Causes array includes more details associated with the StatusReason failure. Not all StatusReasons may provide detailed causes.
group	String	The group attribute of the resource associated with the status StatusReason.
kind	String	The kind attribute of the resource associated with the status StatusReason. On some operations may differ from the requested resource Kind
name	String	The name attribute of the resource associated with the status StatusReason (when there is a single name which can be described).
retryAfterSeconds	Integer	If specified, the time in seconds before the operation should be retried. Some errors may indicate the client must take an alternate action - for those errors this field may indicate how long to wait before taking the alternate action.
uid	String	UID of the resource. (when there is a single resource which can be described)

Table 4-75 Data structure of StatusCause

Parameter	Type	Description
field	String	The field of the resource that has caused this error, as named by its JSON serialization. May include dot and postfix notation for nested attributes. Arrays are zero-indexed. Fields may appear more than once in an array of causes due to fields having multiple errors. Optional. Examples: "name" - the field "name" on the current resource "items[0].name" - the field "name" on the first array entry in "items"
message	String	A human-readable description of the cause of the error. This field may be presented as-is to a reader.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
reason	**StatusCause reason** object	A machine-readable description of the cause of the error. If this value is empty there is no information available.

Table 4-76 Value range of the reason field in StatusCause

Parameter	Description
FieldValueNotFound	CauseTypeFieldValueNotFound is used to report failure to find a requested value.(e.g. looking up an ID).
FieldValueRequired	CauseTypeFieldValueRequired is used to report required values that are not provided (e.g. empty strings, null values, or empty arrays).
FieldValueDuplicate	CauseTypeFieldValueDuplicate is used to report collisions of values that must be unique (e.g. unique IDs).
FieldValueInvalid	CauseTypeFieldValueInvalid is used to report malformed values (e.g. failed regex match).
FieldValueNotSupported	CauseTypeFieldValueNotSupported is used to report valid (as per formatting rules) values that cannot be handled (e.g. an enumerated string).
UnexpectedServerResponse	CauseTypeUnexpectedServerResponse is used to report when the server responded to the client without the expected return type. The presence of this cause indicates the error may be due to an intervening proxy or the server software malfunctioning.

Table 4-77 Data structure of ListMeta

Parameter	Type	Description
continue	String	Continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
resourceVersion	String	String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. **Read-only**
selfLink	String	SelfLink is a URL representing this object. Populated by the system. **Read-only**

Table 4-78 reason

Parameter	Value	Description
StatusReasonUnknown	""	StatusReasonUnknown means the server has declined to indicate a specific reason. The details field may contain other information about this error. **Status code 500**
StatusReasonUnauthorized	Unauthorized	StatusReasonUnauthorized means the server can be reached and understood the request, but requires the user to present appropriate authorization credentials (identified by the WWW-Authenticate header) in order for the action to be completed. If the user has specified credentials on the request, the server considers them insufficient. **Status code 401**
StatusReasonForbidden	Forbidden	StatusReasonForbidden means the server can be reached and understood the request, but refuses to take any further action. It is the result of the server being configured to deny access for some reason to the requested resource by the client. Details (optional): "kind" string - the kind attribute of the forbidden resource on some operations may differ from the requested resource. "id" string - the identifier of the forbidden resource **Status code 403**

Copyright © Huawei Technologies Co., Ltd.

Parameter	Value	Description
StatusReasonNotFound	NotFound	StatusReasonNotFound means one or more resources required for this operation could not be found. Details (optional): "kind" string - the kind attribute of the missing resource on some operations may differ from the requested resource. "id" string - the identifier of the missing resource **Status code 404**
StatusReasonAlreadyExists	AlreadyExists	StatusReasonAlreadyExists means the resource you are creating already exists. Details (optional): "kind" string - the kind attribute of the conflicting resource "id" string - the identifier of the conflicting resource **Status code 409**
StatusReasonConflict	Conflict	StatusReasonConflict means the requested operation cannot be completed due to a conflict in the operation. The client may need to alter the request. Each resource may define custom details that indicate the nature of the conflict. **Status code 409**
StatusReasonGone	Gone	StatusReasonGone means the item is no longer available at the server and no forwarding address is known. **Status code 410**
StatusReasonInvalid	Invalid	StatusReasonInvalid means the requested create or update operation cannot be completed due to invalid data provided as part of the request. The client may need to alter the request. When set, the client may use the StatusDetailsmessage field as a summary of the issues encountered. Details (optional): "kind" string - the kind attribute of the invalid resource "id" string - the identifier of the invalid resource "causes" - one or more StatusCause entries indicating the data in the provided resource that was invalid. The code, message, and field attributes will be set. **Status code 422**

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Value	Description
StatusReasonServerTimeout	ServerTimeout	StatusReasonServerTimeout means the server can be reached and understood the request, but cannot complete the action in a reasonable time. The client should retry the request. This is probably due to temporary server load or a transient communication issue with another server. Status code 500 is used because the HTTP spec provides no suitableserver-requested client retry and the 5xx class represents actionable errors. Details (optional): "kind" string - the kind attribute of the resource being acted on. "id" string - the operation that is being attempted. "retryAfterSeconds" Integer - the number of seconds before the operation should be retried **Status code 500**
StatusReasonTimeout	Timeout	StatusReasonTimeout means that the request could not be completed within the given time. Clients can get this response only when they specified a timeout param in the request, or if the server cannot complete the operation within a reasonable amount of time. The request might succeed with an increased value of timeout param. The client *should*wait at least the number of seconds specified by the retryAfterSeconds field. Details (optional):"retryAfterSeconds" int32 - the number of seconds before the operation should be retried **Status code 504**
StatusReasonTooManyRequests	TooManyRequests	StatusReasonTooManyRequests means the server experienced too many requests within a given window and that the client must wait to perform the action again. A client may always retry the request that led to this error, although the client should wait at least the number of seconds specified by the retryAfterSeconds field. Details (optional): "retryAfterSeconds" int32 - the number of seconds before the operation should be retried **Status code 429**

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Value	Description
StatusReasonBadRequest	BadRequest	StatusReasonBadRequest means that the request itself was invalid, because the request does not make any sense, for example deleting a read-only object. This is different from StatusReasonInvalid above which indicates that the API call could possibly succeed, but the data was invalid. API calls that return BadRequest can never succeed.
StatusReasonMethodNotAllowed	MethodNotAllowed	StatusReasonMethodNotAllowed means that the action the client attempted to perform on the resource was not supported by the code - for instance, attempting to delete a resource that can only be created. API calls that return MethodNotAllowed can never succeed.
StatusReasonInternalError	InternalError	StatusReasonInternalError indicates that an internal error occurred, it is unexpected and the outcome of the call is unknown. Details (optional): "causes" - The original error **Status code 500**
StatusReasonExpired	Expired	StatusReasonExpired indicates that the request is invalid because the content you are requesting has expired and is no longer available. It is typically associated with watches that cannot be serviced. **Status code 410 (gone)**
StatusReasonServiceUnavailable	ServiceUnavailable	StatusReasonServiceUnavailable means that the request itself was valid, but the requested service is unavailable at this time. Retrying the request after some time might succeed. **Status code 503**

Table 4-79 Data structure of WatchEvent

Parameter	Type	Description
type	String	Type of Event. Can be: **- Added** **- Modified** **- Deleted** **- Error**

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
Object	String	Object is: - If Type is Added or Modified: the new state of the object. - If Type is Deleted: the state of the object immediately before deletion. - If Type is Error: Status is recommended; - other types may make sense depending on context.

Table 4-80 Data structure of Deployment

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	Yes	**ObjectMeta** object	Standard object metadata.
spec	Yes	**Deployment Spec** object	Specification of the desired behavior of the Deployment.
status	No	**Deployment Status** object	Most recently observed status of the Deployment.

Copyright © Huawei Technologies Co., Ltd.

Table 4-81 Data structure of the DeploymentSpec field

Parameter	Mandatory	Type	Description
minReadySeconds	No	Integer	Minimum number of seconds for which a newly created pod should be ready without any of its containers crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
paused	No	Boolean	Indicates that the deployment is paused.
progressDeadlineSeconds	No	Integer	The maximum time in seconds for a deployment to make progress before it is considered to be failed. The deployment controller will continue to process failed deployments and a condition with a ProgressDeadlineExceeded reason will be surfaced in the deployment status. Once autoRollback is implemented, the deployment controller will automatically rollback failed deployments. Note that progress will not be estimated during the time a deployment is paused. Defaults to 600s.
replicas	No	Integer	Number of desired pods. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1. The value of **1** indicates one pod, meaning low availability. You are advised to set this parameter to a value greater than 1.
priority	No	Integer	Workload priority. A larger value indicates a higher priority. The default value is 0. Value range: [-10, 10]
revisionHistoryLimit	No	Integer	The number of old ReplicaSets to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 2.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
selector	No	**labelSelector** object	Label selector for pods. Existing ReplicaSets whose pods are selected by this will be the ones affected by this deployment.
strategy	No	**DeploymentStrategy** object	The deployment strategy to use to replace existing pods with new ones.
template	Yes	**PodTemplateSpec** object	Template describes the pods that will be created.

Table 4-82 Data structure of the DeploymentStatus field

Parameter	Mandatory	Type	Description
availableReplicas	No	Integer	Total number of available pods (ready for at least minReadySeconds) targeted by this deployment.
collisionCount	No	Integer	Count of hash collisions for the Deployment. The Deployment controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ReplicaSet.
conditions	No	Array of **DeploymentCondition** objects	Represents the latest available observations of a deployment's current state.
observedGeneration	No	Integer	The generation observed by the deployment controller
readyReplicas	No	Integer	Total number of ready pods targeted by this deployment
replicas	No	Integer	Total number of non-terminated pods targeted by this deployment (their labels match the selector).
unavailableReplicas	No	Integer	Total number of unavailable pods targeted by this Deployment.
updatedReplicas	No	Integer	Total number of non-terminated pods targeted by this Deployment that have the desired template spec.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-83 Data structure of the DeploymentStrategy field

Parameter	Mandatory	Type	Description
rollingUpdate	Yes	RollingUpateDeployment object	Rolling update config params. Present only if DeploymentStrategy-Type = RollingUpdate.
type	No	String	Type of deployment. Can be "Recreate" or "RollingUpdate". Default is RollingUpdate.

Table 4-84 Data structure of the DeploymentCondition field

Parameter	Mandatory	Type	Description
lastTransitionTime	No	String	Last time the condition transitioned from one status to another.
lastUpdateTime	No	String	The last time this condition was updated.
message	No	String	A human readable message indicating details about the transition.
reason	No	String	The reason for the condition's last transition.
status	No	String	Status of the condition, one of True, False, Unknown.
type	No	String	Type of deployment condition. Can be "Available", "Progressing", "ReplicaFailure"

 Copyright © Huawei Technologies Co., Ltd.

Table 4-85 Data structure of the RollingUpateDeployment field

Parameter	Mandatory	Type	Description
maxSurge	No	Integer	The maximum number of pods that can be scheduled above the desired number of pods. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). This cannot be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 25%. Example: when this is set to 30%, the new RC can be scaled up immediately when the rolling update starts, such that the total number of old and new pods do not exceed 130% of desired pods. Once old pods have been killed, new RC can be scaled up further, ensuring that total number of pods running at any time during the update is at most 130% of desired pods.
maxUnavailable	No	Integer	The maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). Absolute number is calculated from percentage by rounding down. This cannot be 0 if MaxSurge is 0. Defaults to 25%. Example: when this is set to 30%, the old RC can be scaled down to 70% of desired pods immediately when the rolling update starts. Once new pods are ready, old RC can be scaled down further, followed by scaling up the new RC, ensuring that the total number of pods available at all times during the update is at least 70% of desired pods.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-86 Data structure of the apps field in DeploymentList v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	No	**ListMeta** object	Standard object metadata.
items	Yes	Array of **Deployment** objects	Items is the list of Deployments

Table 4-87 Data structure of StatefulSet

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
metadata	Yes	**ObjectMeta** object	Standard list metadata.
spec	Yes	**StatefulSetSpec** object	Spec defines the desired identities of pods in this set.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
status	No	**StatefulSetStatus** object	Status is the current status of Pods in this StatefulSet. This data may be out of date by some window of time.

Table 4-88 Data structure of the StatefulSetStatus field

Parameter	Mandatory	Type	Description
observedGeneration	No	Integer	Most recent generation observed by this autoscaler.
replicas	No	Integer	Replicas is the number of actual replicas.
currentReplicas	No	Integer	CurrentReplicas is the number of Pods created by the StatefulSet controller from the StatefulSet version indicated by currentRevision.
currentRevision	No	String	CurrentRevision, if not empty, indicates the version of the StatefulSet used to generate Pods in the sequence [0,currentReplicas).
readyReplicas	No	Integer	ReadyReplicas is the number of Pods created by the StatefulSet controller that have a Ready Condition.
updateRevision	No	String	UpdateRevision, if not empty, indicates the version of the StatefulSet used to generate Pods in the sequence [replicas-updatedReplicas,replicas).
updatedReplicas	No	Integer	UpdatedReplicas is the number of Pods created by the StatefulSet controller from the StatefulSet version indicated by updateRevision.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
collisionCount	No	Integer	CollisionCount is the count of hash collisions for the StatefulSet. The StatefulSet controller uses this field as a collision avoidance mechanism when it needs to create the name for the newest ControllerRevision
conditions	No	Array of **StatefulSetCondition** objects	Represents the latest available observations of a statefulset's current state.

Table 4-89 Data structure of the StatefulSetSpec field

Parameter	Mandatory	Type	Description
replicas	No	Integer	Replicas is the desired number of replicas of the given Template. These are replicas in the sense that they are instantiations of the same Template, but individual replicas also have a consistent identity. If unspecified, defaults to 1. The value of **1** indicates one pod, meaning low availability. You are advised to set this parameter to a value greater than 1.
priority	No	Integer	Workload priority. A larger value indicates a higher priority. The default value is 0. Value range: [-10, 10]

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
podManagementPolicy	No	String	PodManagementPolicy controls how pods are created during initial scale up, when replacing pods on nodes, or when scaling down. Values can be: **OrderedReady, Parallel**. **The default policy is OrderedReady,** where pods are created in increasing order (pod-0, then pod-1, etc) and the controller will wait until each pod is ready before continuing. When scaling down, the pods are removed in the opposite order. The alternative policy is Parallel which will create pods in parallel to match the desired scale without waiting, and on scale down will delete all pods at once.
revisionHistory-Limit	No	Integer	RevisionHistoryLimit is the maximum number of revisions that will be maintained in the StatefulSet's revision history. The revision history consists of all revisions not represented by a currently applied StatefulSetSpec version. The default value is 10.
updateStrategy	No	**StatefulSetUpdateStrategy** object	UpdateStrategy indicates the StatefulSetUpdateStrategy that will be employed to update Pods in the StatefulSet when a revision is made to Template.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
serviceName	Yes	String	ServiceName is the name of the service that governs this StatefulSet. This service must exist before the StatefulSet, and is responsible for the network identity of the set. Pods get DNS/hostnames that follow the pattern: pod-specific-string.serviceName.default.svc.cluster.local where "pod-specific-string" is managed by the StatefulSet controller.
volumeClaimTemplates	No	**PersistentVolumeClaim** object	VolumeClaimTemplates is a list of claims that pods are allowed to reference. The StatefulSet controller is responsible for mapping network identities to claims in a way that maintains the identity of a pod. Every claim in this list must have at least one matching (by name) volumeMount in one container in the template. A claim in this list takes precedence over any volumes in the template, with the same name. Currently, only EVS disks can be mounted.
selector	Yes	**labelSelector** object	Selector is a label query over pods that should match the replica count. If empty, defaulted to labels on the pod template.
template	Yes	**PodTemplateSpec** object	Template is the object that describes the pod that will be created if insufficient replicas are detected. Each pod stamped out by the StatefulSet will fulfill this Template, but have a unique identity from the rest of the StatefulSet.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-90 Data structure of the StatefulSetUpdateStrategy field

Parameter	Mandatory	Type	Description
rollingUpdate	No	**RollingUpdateStatefulSetStrategy** object	RollingUpdate is used to communicate parameters when Type is RollingUpdateStatefulSetStrategyType.
type	No	String	Type indicates the type of the StatefulSetUpdateStrategy. can be: - **RollingUpdate**: indicates that update will be applied to all Pods in the StatefulSet with respect to the StatefulSet ordering constraints. When a scale operation is performed with this strategy, new Pods will be created from the specification version indicated by the StatefulSet's updateRevision. - **OnDelete**: triggers the legacy behavior. Version tracking and ordered rolling restarts are disabled. Pods are recreated from the StatefulSetSpec when they are manually deleted. When a scale operation is performed with this strategy, specification version indicated by the StatefulSet's currentRevision.

Table 4-91 Data structure of the RollingUpdateStatefulSetStrategy field

Parameter	Mandatory	Type	Description
partition	No	Integer	Partition indicates the ordinal at which the StatefulSet should be partitioned.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-92 Data structure of the StatefulSetCondition field

Parameter	Mandatory	Type	Description
type	No	String	Type of the condition. Currently only Ready.
status	No	String	Status of the condition. Can be **True**, **False**, or **Unknown**.
lastTransitionTime	No	String	Last time the condition transitioned from one status to another.
reason	No	String	Unique, one-word, CamelCase reason for the condition's last transition.
message	No	String	Human-readable message indicating details about last transition.

Table 4-93 Data structure of PersistentVolumeClaim

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
metadata	**ObjectMeta** object	Standard object's metadata.
spec	**PersistentVolumeClaimSpec** object	Spec defines the desired characteristics of a volume requested by a pod author.
status	**PersistentVolumeClaimStatus** object	Status represents the current information/status of a persistent volume claim. Read-only.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-94 Data structure of the PersistentVolumeClaimStatus field

Parameter	Mandatory	Type	Description
accessModes	No	Array of strings	AccessModes contains the actual access modes the volume backing the PVC has. **ReadWriteOnce** - can be mounted read/write mode to exactly 1 host **ReadOnlyMany** - can be mounted in read-only mode to many hosts **ReadWriteMany** - can be mounted in read/write mode to many hosts
capacity	No	Array of **ResouceName** objects	Represents the actual resources of the underlying volume.
phase	No	String	Phase represents the current phase of PersistentVolume-Claim. pending - used for PersistentVolumeClaims that are not yet bound Bound - used for PersistentVolumeClaims that are bound Lost - used for PersistentVolumeClaims that lost their underlying. PersistentVolume. The claim was bound to a PersistentVolume and this volume does not exist any longer and all data on it was lost.
conditions	No	Array of **PersistentVolumeClaimCondition** objects	Current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to ResizeStarted.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-95 Data structure of the PersistentVolumeClaimCondition field

Parameter	Mandatory	Type	Description
type	No	String	Type of the condition. Currently only Ready. **Resizing** - An user trigger resize of pvc has been started
status	No	String	Status of the condition. Can be **True**, **False**, or **Unknown**.
lastProbeTime	No	String	Last time we probed the condition.
lastTransitionTime	No	String	Last time the condition transitioned from one status to another.
reason	No	String	Unique, one-word, CamelCase reason for the condition's last transition.
message	No	String	Human-readable message indicating details about last transition.

Table 4-96 Data structure of the PersistentVolumeClaimSpec field

Parameter	Mandatory	Type	Description
volumeName	No	String	VolumeName is the binding reference to the PersistentVolume backing this claim.
accessModes	Yes	Array of strings	AccessModes contains the desired access modes the volume should have. **ReadWriteOnce** – the volume can be mounted as read-write by a single node **ReadOnlyMany** – the volume can be mounted read-only by many nodes **ReadWriteMany** – the volume can be mounted as read-write by many nodes
resources	Yes	**ResourceRequirements** object	Resources represents the minimum resources the volume should have.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
selector	No	labelSelector object	A label query over volumes to consider for binding.
storageClassName	Yes	String	Name of the StorageClass required by the claim. The following fields are supported: • EVS Currently, EVS disks of high I/O (SAS disks), ultra-high I/O (SSD disks), and common I/O (SATA disks) types are supported. • SFS Currently, nfs-rw is supported.
volumeMode	No	String	volumeMode defines what type of volume is required by the claim. Can be: - **Block**: the volume will not be formatted with a filesystem and will remain a raw block device - **Filesystem**: the volume will be or is formatted with a filesystem

Table 4-97 Data structure of the apps field in StatefulsetList v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
-	No	ListMeta object	-

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
items	Yes	Array of **StatefulSet** objects	Items is the list of StatefulSets.

Table 4-98 Data structure of Job

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
metadata	Yes	**ObjectMeta** object	Standard list metadata.
spec	Yes	**JobSpec** object	Specification of the desired behavior of a job.
status	No	**JobStatus** object	Current status of a job.

Table 4-99 Data structure of the JobStatus field

Parameter	Mandatory	Type	Description
active	No	Integer	The number of actively running pods.
completionTime	No	Time	Replicas is the number of actual replicas.
conditions	No	Array of **JobCondition** objects	The latest available observations of an object's current state. More info:

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
failed	No	Integer	The number of pods which reached phase Failed.
startTime	No	Time	Represents time when the job was acknowledged by the job controller. It is not guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC.
succeeded	No	Integer	The number of pods which reached phase Succeeded.

Table 4-100 Data structure of the JobSpec field

Parameter	Mandatory	Type	Description
activeDeadlineSeconds	No	Integer	Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
backoffLimit	No	Integer	Specifies the number of retries before marking this job failed. Defaults to 6.
priority	No	Integer	Job priority. A larger value indicates a higher priority. The default value is 0. Value range: [-10, 10]
completions	No	Integer	Specifies the desired number of successfully finished pods the job should be run with. Setting to nil means that the success of any pod signals the success of all pods, and allows parallelism to have any positive value. Setting to 1 means that parallelism is limited to 1 and the success of that pod signals the success of the job.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
manualSelector	No	Boolean	ManualSelector controls generation of pod labels and pod selectors. Leave manualSelector unset unless you are certain what you are doing. When false or unset, the system picks labels unique to this job and appends those labels to the pod template. When true, the user is responsible for picking unique labels and specifying the selector. Failure to pick a unique label may cause this and other jobs to not function correctly. However, You may see manualSelector=true in jobs that were created with the old extensions/v1beta1 API.
parallelism	No	Integer	Specifies the maximum desired number of pods the job should run at any given time. The actual number of pods running in steady state will be less than this number when ((.spec.completions - .status.successful) < .spec.parallelism), i.e. when the work left to do is less than max parallelism.
selector	Yes	**labelSelector** object	Selector is a label query over pods that should match the replica count. If empty, defaulted to labels on the pod template.
template	Yes	**PodTemplateSpec** object	Template is the object that describes the pod that will be created if insufficient replicas are detected. Each pod stamped out by the StatefulSet will fulfill this Template, but have a unique identity from the rest of the StatefulSet.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-101 Data structure of the JobCondition field

Parameter	Mandatory	Type	Description
lastProbeTime	No	String	Last time the condition was checked.
lastTransitionTime	No	String	Last time the condition transitioned from one status to another.
message	No	String	Human-readable message indicating details about last transition.
reason	No	String	(Brief) Reason for the condition's last transition.
status	No	String	Status of the condition, **one of True, False, Unknown.**
type	No	String	Type of job condition, **Complete or Failed**.

Table 4-102 Data structure of the core field in Service v1

Parameter	Mandatory	Type	Description
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **Service**.
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
metadata	Yes	**ObjectMeta** object	-

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
spec	Yes	**ServiceSpec** object	-
status	No	**ServiceStatus** object	-

Table 4-103 Data structure of the ServiceSpec field

Parameter	Mandatory	Type	Description
ports	Yes	Array of **ServicePort** objects	The list of ports that are exposed by this service.
selector	No	Object	This service will route traffic to pods having labels matching this selector. Label keys and values that must match in order to receive traffic for this service. If empty, all pods are selected, if not specified, endpoints must be manually specified.
clusterIP	No	String	ClusterIP is the IP address of the service and is usually assigned randomly by the master. If an address is specified manually and is not in use by others, it will be allocated to the service; otherwise, creation of the service will fail. This field **cannot be changed through updates. Valid values are "None", empty string (""), or a valid IP address**. "None" can be specified for headless services when proxying is not required. Only applies to types **ClusterIP, NodePort, and LoadBalancer**. Ignored if type is ExternalName.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
type	No	String	Type determines how the Service is exposed. **Defaults to ClusterIP**. **Valid options are ExternalName, ClusterIP and LoadBalancer**. **"ExternalName"** maps to the specified externalName. **"ClusterIP"** allocates a cluster-internal IP address for load-balancing to endpoints. Endpoints are determined by the selector or if that is not specified, by manual construction of an Endpoints object. If clusterIP is "None", no virtual IP is allocated and the endpoints are published as a set of endpoints rather than a stable IP. **"LoadBalancer"** builds on NodePort and creates an external load-balancer (if supported in the current cloud) which routes to the clusterIP. **NOTE** The nodePort service is supported in the community version but not supported in CCI scenarios.
externalIPs	No	Array of strings	ExternalIPs is a list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
externalTraffic-Policy	No	String	ExternalTrafficPolicy denotes if this Service desires to route external traffic to node-local or cluster-wide endpoints. **valid values are "Local" and "Cluster"** - **"Local"** preserves the client source IP and avoids a second hop for LoadBalancer and Nodeport type services, but risks potentially imbalanced traffic spreading. - **"Cluster"** obscures the client source IP and may cause a second hop to another node, but should have good overall load-spreading.
healthCheckNodePort	No	Integer	HealthCheckNodePort specifies the healthcheck nodePort for the service. If not specified, HealthCheckNodePort is created by the service api backend with the allocated nodePort. Will use user-specified nodePort value if specified by the client. **Only effects when Type is set to LoadBalancer and ExternalTrafficPolicy is set to Local.**
externalName	No	String	ExternalName is the external reference that kubedns or equivalent will return as a CNAME record for this service. No proxying will be involved. Must be a valid DNS name and requires Type to be ExternalName.
sessionAffinity	No	String	Used to maintain session affinity. Enable client IP based session affinity. **Must be ClientIP or None.** **Defaults to None.**

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
loadBalancerIP	No	String	**Only applies to Service Type: LoadBalancer** LoadBalancer will get created with the IP specified in this field. This feature depends on whether the underlying cloud-provider supports specifying the loadBalancerIP when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature.
loadBalancerSourceRanges	No	Array of strings	Optional: If specified and supported by the platform, this will restrict traffic through the cloud-provide.load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature.
publishNotReadyAddresses	No	Boolean	PublishNotReadyAddresses, when set to true, indicates that DNS implementations must publish the notReadyAddresses of subsets for the Endpoints associated with the Service. The default value is false. The primary use case for setting this field is to use a StatefulSet's Headless Service to propagate SRV records for its Pods without respect to their readiness for purpose of peer discovery. This field will replace the service.alpha.kubernetes.io/tolerate-unready-endpoints when that annotation is deprecated and all clients have been converted to use this field.
sessionAffinity-Config	No	**SessionAffinityConfig** object	SessionAffinityConfig contains the configurations of session affinity.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-104 Data structure of the ServiceStatus field

Parameter	Mandatory	Type	Description
loadBalancer	No	**loadBalancerStatus** object	LoadBalancer contains the current status of the load-balancer, if one is present

Table 4-105 Data structure of the ServicePort field

Parameter	Mandatory	Type	Description
name	No	String	The name of this port within the service. This must be a DNS_LABEL. All ports within a ServiceSpec must have unique names. This maps to the 'Name' field in EndpointPort objects. Optional if only one ServicePort is defined on this service. Value length: 0 character < String length ≤ 63 characters. The string must comply with regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
protocol	No	String	The IP protocol for this port. Supports "TCP" and "UDP". This parameter can be set to: • **TCP** • **UDP**
port	Yes	Integer	The port that will be exposed by this service. Value range: (0,65535].

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
targetPort	No	String	Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME. If this is a string, it will be looked up as a named port in the target Pod's container ports. If this is not specified, the value of Port is used (an identity map). Defaults to the service port. **Value range: (0,65535].**
nodePort	No	Integer	The port on each node on which this service is exposed when type=NodePort or LoadBalancer. Usually assigned by the system. If specified, it will be allocated to the service if unused or else creation of the service will fail. Default is to auto-allocate a port if the ServiceType of this Service requires one. **Value range: [30000,32767].**

Table 4-106 Data structure of loadBalancerStatus

Parameter	Mandatory	Type	Description
ingress	No	Array of **LoadBalancerIngress** objects	Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

Table 4-107 Data structure of the LoadBalancerIngress field

Parameter	Mandatory	Type	Description
ip	No	String	IP is set for load-balancer ingress points that are IP based.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
hostname	No	String	Hostname is set for load-balancer ingress points that are DNS based.

Table 4-108 Data structure of the SessionAffinityConfig field

Parameter	Mandatory	Type	Description
clientIP	No	**ClientIPConfig** object	ClientIP contains the configurations of Client IP based session affinity.

Table 4-109 Data structure of the ClientIPConfig field

Parameter	Mandatory	Type	Description
timeoutSeconds	No	Integer	TimeoutSeconds specifies the seconds of ClientIP type session sticky time. The value must be >0 && <=86400(for 1 day) if ServiceAffinity == "ClientIP". Default value is 10800(for 3 hours).

Table 4-110 Data structure of ServiceList

Parameter	Type	Description
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadata	**ListMeta** object	Standard list metadata.
items	Array of **Service** objects	List of services.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-111 Data structure of extensions in Ingress v1 beta1

Parameter	Mandatory	Type	Description
apiVersion	No	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	No	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
metadata	No	**ObjectMeta** object	Standard object's metadata.
spec	No	**IngressSpec** object	Spec is the desired state of the Ingress.
status	No	IngressStatus	Status is the current state of the Ingress.

Table 4-112 Data structure of the IngressSpec field

Parameter	Mandatory	Type	Description
backend	No	**IngressBackend** object	A default backend capable of servicing requests that do not match any rule. At least one of *backend* or *rules* must be specified. This field is optional to allow the loadbalancer controller or defaulting logic to specify a global default.
rules	No	Array of **IngressRule** objects	A list of host rules used to configure the Ingress. If unspecified, or no rule matches, all traffic is sent to the default backend.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
tls	No	Array of **IngressTLS** objects	TLS configuration. Currently the Ingress only supports a single TLS port, 443. If multiple members of this list specify different hosts, they will be multiplexed on the same port according to the hostname specified through the SNI TLS extension, if the ingress controller fulfilling the ingress supports SNI.

Table 4-113 Data structure of the IngressStatus field

Parameter	Mandatory	Type	Description
loadBalancer	No	**loadBalancerStatus** object	LoadBalancer contains the current status of the load-balancer.

Table 4-114 Data structure of the IngressBackend field

Parameter	Mandatory	Type	Description
serviceName	No	String	Specifies the name of the referenced service.
servicePort	No	String	Specifies the port of the referenced service.

Table 4-115 Data structure of IngressTLS

Parameter	Mandatory	Type	Description
hosts	No	Array of strings	Hosts are a list of hosts included in the TLS certificate. The values in this list must match the name/s used in the tlsSecret. Defaults to the wildcard host setting for the loadbalancer controller fulfilling this Ingress, if left unspecified.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
secretName	No	String	SecretName is the name of the secret used to terminate SSL traffic on 443. Field is left optional to allow SSL routing based on SNI hostname alone. If the SNI host in a listener conflicts with the "Host" header field used by an IngressRule, the SNI host is used for termination and value of the Host header is used for routing

Table 4-116 Data structure of IngressRule

Parameter	Mandatory	Type	Description
host	No	String	Host is the fully qualified domain name of a network host, as defined by RFC 3986. Note the following deviations from the "host" part of the URI as defined in the RFC: 1. IPs are not allowed. Currently an IngressRuleValue can only apply to the IP in the Spec of the parent Ingress. 2. The : delimiter is not respected because ports are not allowed. Currently the port of an Ingress is implicitly :80 for http and :443 for https. Both these may change in the future. Incoming requests are matched against the host before the IngressRuleValue. If the host is unspecified, the Ingress routes all traffic based on the specified IngressRuleValue
http	No	**IngressRuleValue** object	IngressRuleValue represents a rule to route requests for this IngressRule. If unspecified, the rule defaults to an http catch-all. Whether that sends just traffic matching the host to the default backend or all traffic to the default backend, is left to the controller fulfilling the Ingress. Http is currently the only supported IngressRuleValue.

Table 4-117 Data structure of IngressRuleValue

Parameter	Mandatory	Type	Description
http	No	**HTTPIngressRuleValue** object	HTTP ingress rule.

Copyright © Huawei Technologies Co., Ltd.

Table 4-118 Data structure of HTTPIngressRuleValue

Parameter	Mandatory	Type	Description
paths	No	Array of **HTTPIngressPath** objects	A collection of paths that map requests to backends.

Table 4-119 Data structure of HTTPIngressPath

Parameter	Mandatory	Type	Description
path	No	String	Path is an extended POSIX regex as defined by IEEE Std 1003.1, (i.e this follows the egrep/unix syntax, not the perl syntax) matched against the path of an incoming request. Currently it can contain characters disallowed from the conventional "path" part of a URL as defined by RFC 3986. Paths must begin with a '/'. If unspecified, the path defaults to a catch all sending traffic to the backend
backend	Yes	**IngressBackend** object	Backend defines the referenced service endpoint to which the traffic will be forwarded to
property	Yes	Object	Extension property on the path

Table 4-120 Data structure of the loadBalancerStatus field

Parameter	Mandatory	Type	Description
ingress	No	Array of **LoadBalancerIngress** objects	Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

Copyright © Huawei Technologies Co., Ltd.

Table 4-121 Data structure of IngressList

Parameter	Type	Description
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadata	**ListMeta** object	Standard list metadata.
items	Array of **Ingress v1beta1 extensions** objects	List of Ingress.

Table 4-122 Request parameters of core in Configmap v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
metadata	Yes	**ObjectMeta** object	Standard list metadata.
data	Yes	Object	Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '.'. **The value cannot exceed 512 characters.**

 Copyright © Huawei Technologies Co., Ltd.

Table 4-123 Data structure of ConfigmapList

Parameter	Type	Description
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadata	**ListMeta** object	Standard list metadata.
items	Array of **Configmap v1 core** objects	List of ConfigMaps.

Table 4-124 Data structure of core in Secret v1

Parameter	Mandatory	Type	Description
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **Secret**.
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
metadata	Yes	**ObjectMeta** object	-
data	No	Object	Data contains the secret data. Each key must consist of alphanumeric characters, '-', '_' or '.'. The serialized form of the secret data is a **base64 encoded string**, representing the arbitrary (possibly non-string) data value here

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
stringData	No	Object	StringData allows specifying non-binary secret data in string form. It is provided as a write-only convenience method. All keys and values are merged into the data field on write, overwriting any existing values. It is never output when reading from the API
type	No	String	Used to facilitate programmatic handling of secret data. The primitive k8s supports the following secret types, for details, see **Table 4-125**. • **Opaque** • **kubernetes.io/dockercfg** • **kubernetes.io/dockerconfigjson** • **kubernetes.io/tls**

Table 4-125 Key restrictions of data for different types of secrets

Secret Type	Required Key	Description
Opaque	N/A	Secret type Opaque is the default; arbitrary user-defined data.
kubernetes.io/dockercfg	.dockercfg	Secret type kubernetes.io/dockercfg contains a dockercfg file that follows the same format rules as ~/.dockercfg.
kubernetes.io/tls	tls.key tls.crt	Secret type kubernetes.io/tls contains information about a TLS client or server secret. It is primarily used with TLS termination of the Ingress resource, but may be used in other types.
kubernetes.io/dockerconfigjson	.dockerconfigjson	SecretTypeDockerConfigJson contains a dockercfg file that follows the same format rules as ~/.docker/config.json

Copyright © Huawei Technologies Co., Ltd.

Table 4-126 Data structure of ServiceList

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
metadata	**ListMeta** object	Standard list metadata.
items	Array of **Secret v1 core** objects	List of Secrets.

Table 4-127 Data structure of core in PersistentVolumeClaimList v1

Parameter	Type	Description
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
metadata	**ListMeta** object	Standard list metadata.
items	Array of **PersistentVolume-Claim** object	A list of persistent volume claims.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-128 Data structure of core in Event v1

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
count	integer	The number of times this event has occurred.
firstTimestamp	Time	The time at which the event was first recorded. (Time of server receipt is in TypeMeta.)
involvedObject	**involvedObject** object	The object that this event is about.
lastTimestamp	Time	The time at which the most recent occurrence of this event was recorded.
message	String	A human-readable description of the status of this operation.
metadata	metadata object	Standard object's metadata.
reason	String	This should be a short, machine understandable string that gives the reason for the transition into the object's current status.
source	**EventSource** object	The component reporting this event. Should be a short machine understandable string.
type	String	Type of this event (Normal, Warning), new types could be added in the future
eventTime	time.Time	Time when this Event was first observed.
Series	**EventSerie** object	Data about the Event series this event represents or nil if it is a singleton Event.
action	String	What action was taken/failed regarding to the Regarding object.
related	**ObjectReference** object	Optional secondary object for more complex actions.
reportingComponent	String	Name of the controller that emitted this Event, e.g. `kubernetes.io/kubelet`

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
reportingInstance	String	ID of the controller instance, e.g. `kubelet-xyzf`.

Table 4-129 Data structure of the involvedObject field

Parameter	Type	Description
kind	String	Kind of the referent.
namespace	String	Namespace of the referent.
name	String	Name of the referent.
uid	String	UID of the referent.
apiVersion	String	API version of the referent.
resourceVersion	String	Specific resourceVersion to which this reference is made.
fieldPath	String	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.

Table 4-130 Data structure of the EventSource field

Parameter	Type	Description
component	String	Component from which the event is generated.
host	String	Node name on which the event is generated.

Copyright © Huawei Technologies Co., Ltd.

Table 4-131 Data structure of EventSeries

Parameter	Type	Description
count	*integer*	Number of occurrences in this series up to the last heartbeat time
lastObservedTime	time.Time	Time of the last occurrence observed
state	String	State of this Series: **Ongoing**, **Finished**, or **Unknown**

Table 4-132 Data structure of the ObjectReference field

Parameter	Type	Description
apiVersion	String	API version of the referent.
fieldPath	String	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
kind	String	Kind of the referent.
name	String	Name of the referent.
namespace	String	Namespace of the referent.
resourceVersion	String	Specific resourceVersion to which this reference is made, if any.
uid	String	UID of the referent.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-133 Data structure of core in EventList v1

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
items	Array of **Event v1 core** objects	List of services.
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	**ListMeta** object	Standard list metadata.

Table 4-134 Data structure of LocalDirVolumeSource

Parameter	Type	Description
sizeLimit	Object	Storage space of the localDir in use.

Table 4-135 Data structure of emptyDir

Parameter	Type	Description
sizeLimit	integer	Storage space of the emptyDir in use. Value range: (0, 2147483647] Unit: Gi
medium	String	Medium type. The options are as follows: • **LocalVolume**: ultra-high I/O EVS disks • **LocalSSD**: local SSDs **NOTE** If this parameter is not set, ultra high I/O EVS disks are used by default.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-136 Data structure of Endpoints

Parameter	Type	Description
kind	String	String Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **Secret**.
apiVersion	String	String APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
metadata	**ObjectMeta** object	-
subsets	Array of **EndpointSubset** objects	The set of all endpoints is the union of all subsets. Addresses are placed into subsets according to the IPs they share. A single address with multiple ports, some of which are ready and some of which are not (because they come from different containers) will result in the address being displayed in different subsets for the different ports. No address will appear in both Addresses and NotReadyAddresses in the same subset. Sets of addresses and ports that comprise a service.

Table 4-137 Data structure of EndpointSubset

Parameter	Type	Description
addresses	Array of **EndpointAddress** objects	IP addresses which offer the related ports that are marked as ready. These endpoints should be considered safe for load balancers and clients to utilize.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
notReadyAddresses	Array of **EndpointAddress** objects	IP addresses which offer the related ports but are not currently marked as ready because they have not yet finished starting, have recently failed a readiness check, or have recently failed a liveness check.
ports	Array of **EndpointPort** objects	Port numbers available on the related IP addresses.

Table 4-138 Data structure of EndpointAddress

Parameter	Type	Description
ip	String	The IP of this endpoint. IPv6 is also accepted but not fully supported on all platforms. Also, certain kubernetes components, like kube-proxy, are not IPv6 ready
hostname	String	The Hostname of this endpoint
nodename	String	Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.
targetRef	**ObjectReference** object	Reference to object providing the endpoint.
nodeAvailableZone	String	Optional: The availability zone of the endpoint's host node

Table 4-139 Data structure of EndpointPort

Parameter	Type	Description
name	String	The name of this port (corresponds to ServicePort.Name). Must be a DNS_LABEL. Optional only if one port is defined.
port	Integer	The port number of the endpoint.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
protocol	String	The IP protocol for this port. **Must be UDP or TCP.** Default is TCP

Table 4-140 Data structure of core in EndpointsList v1

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
items	Array of **Endpoints** objects	List of endpoints
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	**ListMeta** object	Standard list metadata.

Table 4-141 Data structure of ReplicaSet

Parameter	Type	Description
kind	String	String Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **Secret**.
apiVersion	String	String APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
metadata	**ObjectMeta** object	-
spec	**ReplicaSetSpec** object	Spec defines the specification of the desired behavior of the ReplicaSet.
status	**ReplicaSetStatus** object	Status is the most recently observed status of the ReplicaSet. This data may be out of date by some window of time. Populated by the system. Read-only.

Table 4-142 Data structure of ReplicaSetSpec

Parameter	Type	Description
replicas	Integer	Replicas is the number of desired replicas. This is a pointer to distinguish between explicit zero and unspecified. Defaults to 1.
minReadySeconds	Integer	Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
selector	**labelSelector** object	Selector is a label query over pods that should match the replica count. Label keys and values that must match in order to be controlled by this replica set. It must match the pod template's labels.
template	**PodTemplateSpec** object	Template is the object that describes the pod that will be created if insufficient replicas are detected.

Copyright © Huawei Technologies Co., Ltd.

Table 4-143 Data structure of ReplicaSetStatus

Parameter	Type	Description
replicas	Integer	Replicas is the most recently observed number of replicas.
fullyLabeledReplicas	Integer	The number of pods that have labels matching the labels of the pod template of the replicaset.
readyReplicas	Integer	The number of ready replicas for this replica set.
availableReplicas	Integer	The number of available replicas (ready for at least minReadySeconds) for this replica set.
observedGeneration	Integer	ObservedGeneration reflects the generation of the most recently observed ReplicaSet.
conditions	**ReplicaSetCondition** object	Represents the latest available observations of a replica set's current state.

Table 4-144 Data structure of ReplicaSetCondition

Parameter	Type	Description
type	String	Type of replica set condition. Available values: -- ReplicaFailure: ReplicaSetReplicaFailure is added in a replica set when one of its pods fails to be created due to insufficient quota, limit ranges, pod security policy, node selectors, etc. or deleted due to kubelet being down or finalizers are failing.
status	String	Status of the condition, one of True, False, Unknown.
lastTransitionTime	Object	The last time the condition transitioned from one status to another.
reason	String	The reason for the condition's last transition.
message	String	A human readable message indicating details about the transition.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-145 Data structure of core in ReplicaSetList v1

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
items	Array of **ReplicaSet** objects	List of endpoints
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	**ListMeta** object	Standard list metadata.

Table 4-146 Data structure of Volcano Job batch_v1alpha1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#resources.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds.
metadata	Yes	**ObjectMeta** object	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
spec	Yes	**VolcanoJobSpec** object	Specification of the desired behavior of a cron job, including the minAvailable.
status	No	**VolcanoJobStatus** object	Current status of Job.

Table 4-147 Data structure of the VolcanoJobSpec field

Parameter	Mandatory	Type	Description
maxRetry	No	Integer	The limit for retrying submitting job. The default value is 3.
minAvailable	Yes	Integer	The minimal available pods to run for this Job. This value should above 0 and no more than existing pod numebr. If the value below 0, will use defaule value(The whole number of this job's pod numebr).
plugins	No	**VolcanoPlugin** object	Enabled task plugins when creating job.
policies	No	Array of **VolcanoJobPolicy** objects	Specifies the default lifecycle of tasks.
queue	No	String	The name of the queue on which job should been created.
schedulerName	No	String	SchedulerName is the default value of `tasks.template.spec.schedulerName`.
tasks	Yes	Array of **VolcanoJobTask** objects	Tasks specifies the task specification of Job.
volumes	No	Array of **VolcanoJobVolume** objects	The volumes for Job.

Copyright © Huawei Technologies Co., Ltd.

Table 4-148 Data structure of the VolcanoJobStatus field

Parameter	Mandatory	Type	Description
ControlledResources	Yes	Object	All of the resources that are controlled by this job. {"plugin-env":"env","plugin-ssh":"ssh","plugin-svc":"svc"}.
Succeeded	No	Integer	The number of pods which reached phase Succeeded.
failed	No	Integer	The number of pods which reached phase Failed.
minAvailable	Yes	Integer	The minimal available pods to run for this Job.
pending	No	Integer	The number of pending pods.
retryCount	No	Integer	The number that volcano retried to submit the job.
running	No	Integer	The number of running pods.
version	No	Integer	Job's current version.
state	Yes	**VolcanoJobStatusState** object	Current state of Job.

Table 4-149 Data structure of the VolcanoJobPolicy field

Parameter	Mandatory	Type	Description
action	No	String	The action that will be taken to the PodGroup according to Event. One of \"Restart\", \"None\". Default to None.
event	No	String	The Event recorded by scheduler; the controller takes actions according to this Event.
timeout	No	Object	Timeout is the grace period for controller to take actions. Default to nil (take action immediately).

 Copyright © Huawei Technologies Co., Ltd.

Table 4-150 Data structure of the VolcanoJobTask field

Parameter	Mandatory	Type	Description
name	Yes	String	Name specifies the name of tasks.
policies	No	**VolcanoJobPolicy** object	Specifies the lifecycle of task.
replicas	No	Integer	Replicas specifies the replicas of this TaskSpec in Job.
template	No	**PodTemplateSpec** object	Specifies the pod that will be created for this TaskSpec when executing a Job.

Table 4-151 Data structure of the VolcanoJobVolume field

Parameter	Mandatory	Type	Description
mountPath	Yes	String	Path within the container at which the volume should be mounted. Must not contain ':'.
volumeClaim	No	**VolcanoTaskVolumeClaimSpec** object	VolumeClaim defines the PVC used by the VolumeMount.
volumeClaimName	No	String	The name of the volume claim.

Table 4-152 Data structure of the VolcanoJobStatusState field

Parameter	Mandatory	Type	Description
message	No	String	Human-readable message indicating details about last transition.
phase	Yes	String	The phase of Job.
reason	No	String	Unique, one-word, CamelCase reason for the condition's last transition.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-153 Data structure of the VolcanoTaskVolumeClaimSpec field

Parameter	Mandatory	Type	Description
accessModes	Yes	Array of strings	AccessModes contains the desired access modes the volume should have. **ReadWriteOnce** – the volume can be mounted as read-write by a single node **ReadOnlyMany** – the volume can be mounted read-only by many nodes **ReadWriteMany** – the volume can be mounted as read-write by many nodes
resources	Yes	**ResourceRequirements** object	Resources represents the minimum resources the volume should have.
storageClassName	Yes	String	Name of the StorageClass required by the claim. The following fields are supported: • EVS Currently, EVS disks of high I/O (SAS disks), ultra-high I/O (SSD disks), and common I/O (SATA disks) types are supported. • SFS Currently, nfs-rw is supported.

Table 4-154 Data structure of the VolcanoPlugin field

Parameter	Mandatory	Type	Description
ssh	No	Array of strings	Set VK_TASK_INDEX to each container, which is an index for giving the identity to container. The value **no-root** indicates logging in as a non-root user using SSH.
svc	No	Array of strings	Create Service and *.host to enable pod communication.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
ssh	No	Array of strings	Sign in ssh without password, e.g. use command mpirun or mpiexec.

Table 4-155 Data structure of TFJob kubeflow_v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **TFJob**.
metadata	Yes	**ObjectMeta** object	Standard object's metadata.
spec	Yes	**TFJobSpec** object	Specification of the desired behavior of a TFJob.
status	No	**JobStatus** object	Current status of TFJob.

Table 4-156 Data structure of the TFJobSpec field

Parameter	Mandatory	Type	Description
activeDeadlineSecond	No	Integer	Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
backoffLimit	No	Integer	Optional number of retries before marking this job failed.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
cleanPodPolicy	No	**CleanPodPolicy** object	CleanPodPolicy defines the policy to kill pods after TFJob is succeeded. Default to Running.
ttlSecondsAfter-Finished	No	Integer	TTLSecondsAfterFinished is the TTL to clean up tf-jobs (temporary before kubernetes adds the cleanup controller). It may take extra ReconcilePeriod seconds for the cleanup, since reconcile gets called periodically.

tfReplicaSpecs	Yes	Array of **ReplicaSpec** objects	TFReplicaSpecs is map of TFReplicaType and ReplicaSpec specifies the TF replicas to run.
			For example,
			{
			"PS": ReplicaSpec,
			"Worker": ReplicaSpec,
			}

Table 4-157 Available values of the CleanPodPolicy field

Available Value	Description
All	When the job is finished, kill all pods that the job created.
Running	When the job is finished, kill pods that the job created and is in running phase.
None	When the job is finished, do not kill any pods that the job created.

Table 4-158 Available values of the TFReplicaType field

Available Value	Description
PS	PS is the type for parameter servers of distributed TensorFlow.

 Copyright © Huawei Technologies Co., Ltd.

Available Value	Description
Worker	Worker is the type for workers of distributed TensorFlow. This is also used for non-distributed TensorFlow.
Chief	Chief is the type for chief worker of distributed TensorFlow. If there is "chief" replica type, it's the "chief worker". Else, worker:0 is the chief worker.
Evaluator	Evaluator is the type for evaluation replica in TensorFlow.

Table 4-159 Data structure of the ReplicaSpec field

Parameter	Mandatory	Type	Description
replicas	No	Integer	Replicas is the desired number of replicas of the given template. If unspecified, defaults to 1.
template	Yes	**PodTemplateSpec** object	Template is the object that describes the pod that will be created for this replica. RestartPolicy in PodTemplateSpec will be overridden by RestartPolicy in ReplicaSpec.
restartPolicy	No	String	Restart policy for all replicas within the job. One of Always, OnFailure, Never and ExitCode. Default to Never.

Table 4-160 Data structure of the JobStatus field

Parameter	Mandatory	Type	Description
conditions	No	Array of **JobCondition** objects	Conditions is an array of current observed job conditions.
replicaStatuses	No	Array of **ReplicaStatus** objects	ReplicaStatuses is map of ReplicaType and ReplicaStatus, specifies the status of each replica.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
startTime	No	Time	Represents time when the job was acknowledged by the job controller. It is not guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC.
completionTime	No	Time	Represents time when the job was completed. It is not guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC.
lastReconcileTime	No	Time	Represents last time when the job was reconciled. It is not guaranteed to be set in happens-before order across separate operations. It is represented in RFC3339 form and is in UTC.

Table 4-161 Data structure of the JobCondition field

Parameter	Mandatory	Type	Description
type	No	String	Type of job condition. **NOTE** Job states include: -Created: means the job has been accepted by the system, but one or more of the pods/services has not been started. This includes time before pods being scheduled and launched. -Running: means all sub-resources (e.g. services/pods) of this job have been successfully scheduled and launched. The training is running without error. -Restarting: means one or more sub-resources (e.g. services/pods) of this job reached phase failed but maybe restarted according to it's restart policy which specified by user in v1.PodTemplateSpec. The training is freezing/pending. -Succeeded: means all sub-resources (e.g. services/pods) of this job reached phase have terminated in success. The training is complete without error. -Failed: means one or more sub-resources (e.g. services/pods) of this job reached phase failed with no restarting. The training has failed its execution.
status	No	String	Status of the condition, one of True, False, Unknown.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
reason	No	String	(Brief) Reason for the condition's last transition.
message	No	String	Human readable message indicating details about last transition.
lastUpdateTime	No	Time	The last time this condition was updated.
lastTransitionTime	No	Time	Last time the condition transitioned from one status to another.

Table 4-162 Data structure of the ReplicaStatus field

Parameter	Mandatory	Type	Description
active	No	Integer	The number of actively running pods.
succeeded	No	Integer	The number of pods which reached phase Succeeded.
failed	No	Integer	The number of pods which reached phase Failed.

Table 4-163 Data structure of MXJob kubeflow_v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **MXJob**.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
metadata	Yes	**ObjectMeta** object	Standard object's metadata.
spec	Yes	MXJobSpec	Specification of the desired behavior of a MXJob.
status	No	**JobStatus** object	Current status of MXJob.

Table 4-164 Data structure of the MXJobSpec field

Parameter	Mandatory	Type	Description
activeDeadlineSecond	No	Integer	Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
backoffLimit	No	Integer	Optional number of retries before marking this job failed.
cleanPodPolicy	No	**CleanPodPolicy** object	CleanPodPolicy defines the policy to kill pods after TFJob is succeeded. Default to Running.
ttlSecondsAfter-Finished	No	Integer	TTLSecondsAfterFinished is the TTL to clean up tf-jobs (temporary before kubernetes adds the cleanup controller). It may take extra ReconcilePeriod seconds for the cleanup, since reconcile gets called periodically.

mxReplicaSpecs	Yes	Array of **ReplicaSpec** objects	MXReplicaSpecs is map of MXReplicaType and MXReplicaSpec specifies the MX replicas to run.
			For example,
			{
			"Scheduler": MXReplicaSpec,
			"Server": MXReplicaSpec,
			"Worker": MXReplicaSpec,
			}

 Copyright © Huawei Technologies Co., Ltd.

Table 4-165 Available values of the MXReplicaType field

Available Value	Description
Scheduler	Scheduler is the type for scheduler replica in MXNet.
Worker	Worker is the type for workers of distributed MXNet.
Server	Server is the type for parameter servers of distributed MXNet.

Table 4-166 Data structure of PyTorchJob kubeflow_v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1**.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **PyTorchJob**.
metadata	Yes	**ObjectMeta** object	Standard object's metadata.
spec	Yes	**PyTorchJobSpec** object	Specification of the desired behavior of a pytorchjob.
status	No	**JobStatus** object	Current status of pytorchJob.

Copyright © Huawei Technologies Co., Ltd.

Table 4-167 Data structure of the PyTorchJobSpec field

Parameter	Mandatory	Type	Description
activeDeadlineSecond	No	Integer	Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
backoffLimit	No	Integer	Optional number of retries before marking this job failed.
cleanPodPolicy	No	**CleanPodPolicy** object	CleanPodPolicy defines the policy to kill pods after TFJob is succeeded. Default to Running.
ttlSecondsAfter-Finished	No	Integer	TTLSecondsAfterFinished is the TTL to clean up tf-jobs (temporary before kubernetes adds the cleanup controller). It may take extra ReconcilePeriod seconds for the cleanup, since reconcile gets called periodically.

ReplicaSpecs	Yes	Array of **ReplicaSpec** objects	A map of PyTorchReplicaType (type) to ReplicaSpec (value). Specifies the PyTorch cluster configuration.
			For example,
			{
			"Master": PyTorchReplicaSpec,
			"Worker": PyTorchReplicaSpec,
			}

Table 4-168 Available values of the PytorchReplicaType field

Available Value	Description
Master	Master is the type of Master of distributed PyTorch.
Worker	Worker is the type for workers of distributed PyTorch.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-169 Data structure of TFJobList kubeflow_v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	No	**ListMeta** object	Standard type metadata.
items	Yes	Array of **TFJob kubeflow_v1** objects	Lists of TFJobs.

Table 4-170 Data structure of MXJobList kubeflow_v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	No	**ListMeta** object	Standard type metadata.
items	Yes	Array of **MXJob kubeflow_v1** objects	Lists of mxjobs.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-171 Data structure of PyTorchJobList kubeflow_v1

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	No	**ListMeta** object	Standard type metadata.
items	Yes	Array of **PyTorchJob kubeflow_v1** objects	Lists of pytorchjobs.

Table 4-172 Data structure of MPIJob kubeflow_v1alpha2

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. The value of this parameter is **v1alpha2**.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. The value of this parameter is **MPIJob**.
metadata	Yes	**ObjectMeta** object	Standard object's metadata.
spec	Yes	**MPIJobSpec** object	Specification of the desired behavior of an MPIJob.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
status	No	**JobStatus** object	Current status of MPIJob.

Table 4-173 Data structure of the MPIJobSpec field

Parameter	Mandatory	Type	Description
activeDeadlineSecond	No	Integer	Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
backoffLimit	No	Integer	Optional number of retries before marking this job failed.
cleanPodPolicy	No	**CleanPodPolicy** object	CleanPodPolicy defines the policy to kill pods after TFJob is succeeded. Default to Running.
slotsPerWorker	No	Integer	Specifies the number of slots per worker used in hostfile. Defaults to 1.

mpiReplicaSpecs	Yes	Array of **ReplicaSpec** objects	MPIReplicaSpecs is map of MPIReplicaType and MPIReplicaSpec specifies the MPI replicas to run.
			For example,
			{
			"Launcher": MPIReplicaSpec,
			"Worker": MPIReplicaSpec,
			}

Table 4-174 Available values of the MPIReplicaType field

Available Value	Description
Launcher	Launcher is the type for launcher replica in MPI.

Copyright © Huawei Technologies Co., Ltd.

Available Value	Description
Worker	Worker is the type for workers of distributed MPI.

Table 4-175 Data structure of MPIJobList kubeflow_v1alpha2

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	No	**ListMeta** object	Standard type metadata.
items	Yes	Array of **MPIJob kubeflow_v1alpha2** objects	List of MPIJobs.

Table 4-176 Data structure of PersistentVolumeClaim v1

Parameter	Type	Description
apiVersion	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Type	Description
metadata	**ListMeta v1 meta** object	Standard object's metadata.
spec	**PersistentVolumeClaimSpec** object	Spec defines the desired characteristics of a volume requested **PersistentVolumeClaimStatus** by a pod author.
status	**PersistentVolumeClaimStatus** object	Status represents the current information/status of a persistent volume claim. Read-only.

Table 4-177 Data structure of the meta field in ListMeta v1

Name	Type	Description
continue	string	Continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response
resourceVersion	string	String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. **Read-only**
selfLink	string	SelfLink is a URL representing this object. Populated by the system. **Read-only**

Table 4-178 Data structure of the PersistentVolumeClaimSpec field

Parameter	Mandatory	Type	Description
volumeName	No	String	VolumeName is the binding reference to the PersistentVolume backing this claim.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
accessModes	Yes	Array of strings	AccessModes contains the desired access modes the volume should have. **ReadWriteOnce** – the volume can be mounted as read-write by a single node **ReadOnlyMany** – the volume can be mounted read-only by many nodes **ReadWriteMany** – the volume can be mounted as read-write by many nodes
resources	Yes	**ResourceRequirements** object	Resources represents the minimum resources the volume should have.
selector	No	**labelSelector** object	A label query over volumes to consider for binding.
storageClassName	Yes	String	Name of the StorageClass required by the claim. The following fields are supported: • EVS Currently, EVS disks of high I/O (SAS disks), ultra-high I/O (SSD disks), and common I/O (SATA disks) types are supported. • SFS Currently, nfs-rw is supported. • SFS Turbo Currently, SFS Turbo volumes of high-performance (efs-performance) and standard (efs-standard) types are supported. • OBS Currently, OBS volumes are supported.
volumeMode	No	String	volumeMode defines what type of volume is required by the claim. Can be: - **Block**: the volume will not be formatted with a filesystem and will remain a raw block device - **Filesystem**: the volume will be or is formatted with a filesystem

 Copyright © Huawei Technologies Co., Ltd.

Table 4-179 Data structure of the PersistentVolumeClaimStatus field

Parameter	Mandatory	Type	Description
accessModes	No	Array of strings	AccessModes contains the actual access modes the volume backing the PVC has. **ReadWriteOnce** - can be mounted read/write mode to exactly 1 host **ReadOnlyMany** - can be mounted in read-only mode to many hosts **ReadWriteMany** - can be mounted in read/write mode to many hosts
capacity	No	Array of **ResourceRequirements** objects	Represents the actual resources of the underlying volume.
phase	No	String	Phase represents the current phase of PersistentVolumeClaim. pending - used for PersistentVolume-Claims that are not yet bound Bound - used for PersistentVolume-Claims that are bound Lost - used for PersistentVolume-Claims that lost their underlying. PersistentVolume. The claim was bound to a PersistentVolume and this volume does not exist any longer and all data on it was lost.
conditions	No	Array of **PersistentVolumeClaimCondition** objects	Current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to ResizeStarted.

Table 4-180 Data structure of the PersistentVolumeClaimCondition field

Name	Mandatory	Type	Description
type	No	String	Type of the condition. Currently only Ready. **Resizing** - An user trigger resize of pvc has been started
status	No	String	Status of the condition. Can be **True**, **False**, or **Unknown**.

 Copyright © Huawei Technologies Co., Ltd.

Name	Mandatory	Type	Description
lastProbeTime	No	String	Last time we probed the condition.
lastTransitionTime	No	String	Last time the condition transitioned from one status to another.
reason	No	String	Unique, one-word, CamelCase reason for the condition's last transition.
message	No	String	Human-readable message indicating details about last transition.

Table 4-181 Data structure of the ResourceRequirements field

Parameter	Mandatory	Type	Description
limits	No	Array of **ResourceName** objects	Maximum amount of compute resources allowed. **NOTE** The values of **limits** and **requests** must be the same. Otherwise, an error is reported.
requests	No	Array of **ResourceName** objects	Minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. CCI has limitation on pod specifications. For details, see **Pod Specifications** in **Usage Constraints**.

Table 4-182 Available values of the ResourceName field

Type	Mandatory	Type	Description
storage	No	String	Volume size, in bytes (e,g. 5Gi = 5GiB = 5 * 1024 * 1024 * 1024)
cpu	No	String	CPU size, in cores. (500m = .5 cores)
memory	No	String	Memory size, in bytes. (500Gi = 500GiB = 500 * 1024 * 1024 * 1024)

Copyright © Huawei Technologies Co., Ltd.

Type	Mandatory	Type	Description
localdir	No	String	Local Storage for LocalDir, in bytes. (500Gi = 500GiB = 500 * 1024 * 1024 * 1024)
nvidia.com/gpu-tesla-v100-16GB	No	String	NVIDIA GPU resource, the type may change in different environments, in production environment is nvidia.com/gpu-tesla-v100-16GB now. The value must be an integer and not less than 1.

Table 4-183 Data structure of the labelSelector field

Parameter	Mandatory	Type	Description
matchExpressions	No	Array of **LabelSelectorRequirement** objects	MatchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	No	Object	MatchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Table 4-184 Data structure of the LabelSelectorRequirement field

Parameter	Mandatory	Type	Description
key	No	String	Key is the label key that the selector applies to.
operator	No	String	Operator represents a key's relationship to a set of values. **Valid operators are In, NotIn, Exists and DoesNotExist.**

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
values	No	Array of strings	Values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Table 4-185 Data structure of PersistentVolume

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	Yes	**metadata** object	Standard object's metadata.
spec	Yes	**spec** object	Spec defines a specification of a persistent volume owned by the cluster. Provisioned by an administrator.
status	No	**status** object	Status represents the current information/status for the persistent volume. Populated by the system. Read-only.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-186 Data structure of the spec field

Parameter	Mandatory	Type	Description
accessModes	Yes	Array of strings	Access mode. Options: ReadWriteOnce: can be read and written by a single node. ReadOnlyMany: can only be read by multiple nodes. ReadWriteMany: can be read and written by multiple nodes.
capacity	Yes	Object	A description of the persistent volume's resources and capacity.
claimRef	No	**claimRef** object	ClaimRef is part of a bi-directional binding between PersistentVolume and PersistentVolumeClaim. Expected to be non-nil when bound. claim. VolumeName is the authoritative bind between PV and PVC.
hostPath	No	**hostPath** object	HostPath represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster.
nfs	No	**nfs** object	NFS represents an NFS mount on the host. Provisioned by an admin.
persistentVolumeReclaimPolicy	No	String	What happens to a persistent volume when released from its claim. Valid options are Retain (default) and Recycle. Recycling must be supported by the volume plugin underlying this persistent volume.
storageClassName	No	String	Name of StorageClass to which this persistent volume belongs. Empty value means that this volume does not belong to any StorageClass.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-187 Data structure of the status field

Parameter	Mandatory	Type	Description
message	No	String	A human-readable message indicating details about why the volume is in this state.
phase	No	String	Phase indicates if a volume is available, bound to a claim, or released by a claim.
reason	No	String	Reason is a brief CamelCase string that describes any failure and is meant for machine parsing and tidy display in the CLI.

Table 4-188 Data structure of the claimRef field

Parameter	Mandatory	Type	Description
apiVersion	No	String	API version of the referent.
fieldPath	No	String	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
kind	No	String	Kind of the referent.
name	No	String	Name of the referent.
namespace	No	String	Namespace of the referent.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
resourceVersion	No	String	Specific resourceVersion to which this reference is made, if any.
uid	No	String	UID of the referent.

Table 4-189 Data structure of the hostPath field

Parameter	Mandatory	Type	Description
path	No	String	Path of the directory on the host.

Table 4-190 Data structure of the nfs field

Parameter	Mandatory	Type	Description
path	No	String	Path that is exported by the NFS server.
readOnly	No	Boolean	ReadOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false.
server	No	String	Server is the hostname or IP address of the NFS server.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-191 Data structure of the metadata field

Name	Mandatory	Type	Description
name	Yes	String	Name must be unique within a namespace. Is required when creating resources, although some resources may allow a client to request the generation of an appropriate name automatically. Name is primarily intended for creation idempotence and configuration definition. Cannot be updated. 0 characters < name length ≤ 253 characters. The name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
clusterName	No	String	The name of the cluster which the object belongs to. This is used to distinguish resources with same name and namespace in different clusters. This field is not set anywhere right now and apiserver is going to ignore it if set in create or update request.
initializers	No	**initializers** object	An initializer is a controller which enforces some system invariant at object creation time. This field is a list of initializers that have not yet acted on this object. If nil or empty, this object has been completely initialized. Otherwise, the object is considered uninitialized and is hidden (in list/watch and get calls) from clients that haven't explicitly asked to observe uninitialized objects. When an object is created, the system will populate this list with the current set of initializers. Only privileged users may set or modify this list. Once it is empty, it may not be modified further by any user.
enable	No	Boolean	Enable identify whether the resource is available.

 Copyright © Huawei Technologies Co., Ltd.

Name	Mandatory	Type	Description
generateName	No	String	An optional prefix used by the server to generate a unique name ONLY IF the Name field has not been provided. If this field is used, the name returned to the client will be different from the name passed. This value will also be combined with a unique suffix. The provided value has the same validation rules as the Name field, and may be truncated by the length of the suffix required to make the value unique on the server. If this field is specified and the generated name exists, the server will NOT return a 409. Instead, it will either return 201 Created or 500 with Reason ServerTimeout indicating a unique name could not be found in the time allotted, and the client should retry (optionally after the time indicated in the Retry-After header). Applied only if Name is not specified. 0 characters < generated name length ≤ 253 characters. The generated name must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.

 Copyright © Huawei Technologies Co., Ltd.

Name	Mandatory	Type	Description
namespace	No	String	Namespace defines the space within each name must be unique. An empty namespace is equivalent to the "default" namespace, but "default" is the canonical representation. Not all objects are required to be scoped to a namespace - the value of this field for those objects will be empty. Must be a DNS_LABEL. Cannot be updated. 0 characters < namespace length ≤ 63 characters. The namespace must be a regular expression [a-z0-9]([-a-z0-9]*[a-z0-9])?.
selfLink	No	String	A URL representing this object. Populated by the system. Read-only. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.
uid	No	String	UID is the unique in time and space value for this object. It is typically generated by the server on successful creation of a resource and is not allowed to change on PUT operations. Populated by the system. Read-only. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.

 Copyright © Huawei Technologies Co., Ltd.

Name	Mandatory	Type	Description
resourceVersion	No	String	An opaque value that represents the internal version of this object that can be used by clients to determine when objects have changed. May be used for optimistic concurrency, change detection, and the watch operation on a resource or set of resources. Clients must treat these values as opaque and passed unmodified back to the server. They may only be valid for a particular resource or set of resources. Populated by the system. Read-only. Value must be treated as opaque by clients. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.
generation	No	Integer	A sequence number representing a specific generation of the desired state. Currently only implemented by replication controllers. Populated by the system. Read-only.
creationTimestamp	No	String	A timestamp representing the server time when this object was created. It is not guaranteed to be set in happens-before order across separate operations. Clients may not set this value. It is represented in RFC3339 form and is in UTC. Populated by the system. Read-only. Null for lists. **NOTE** This field is automatically generated. Do not assign any value to this field. Otherwise, API calls would fail.

 Copyright © Huawei Technologies Co., Ltd.

Name	Mandatory	Type	Description
deletionTimestamp	No	String	RFC 3339 date and time at which this resource will be deleted. This field is set by the server when a graceful deletion is requested by the user, and is not directly settable by a client. The resource will be deleted (no longer visible from resource lists, and not reachable by name) after the time in this field. Once set, this value may not be unset or be set further into the future, although it may be shortened or the resource may be deleted prior to this time. For example, a user may request that a pod is deleted in 30 seconds. The Kubelet will react by sending a graceful termination signal to the containers in the pod. Once the resource is deleted in the API, the Kubelet will send a hard termination signal to the container. If not set, graceful deletion of the object has not been requested. Populated by the system when a graceful deletion is requested. Read-only.
deletionGracePeriodSeconds	No	Integer	Number of seconds allowed for this object to gracefully terminate before it will be removed from the system. Only set when deletionTimestamp is also set. May only be shortened. Read-only.
labels	Yes	Object	Map of string keys and values that can be used to organize and categorize (scope and select) objects. May match selectors of replication controllers and services. **NOTE** This field should be filled in to create the real storage dynamically. The value of the field is according to the real region and zone.

 Copyright © Huawei Technologies Co., Ltd.

Name	Mandatory	Type	Description
annotations	No	**annotations** object	An unstructured key value map stored with a resource that may be set by external tools to store and retrieve arbitrary metadata. They are not queryable and should be preserved when modifying objects. **NOTE** This field should be filled in to create the real storage dynamically. This field indicates the storage plugin and the StorageClass.
ownerReferences	No	**ownerReferences** object	List of objects depended by this object. If ALL objects in the list have been deleted, this object will be garbage collected. If this object is managed by a controller, then an entry in this list will point to this controller, with the controller field set to true. There cannot be more than one managing controller.
finalizers	No	Array of strings	Must be empty before the object is deleted from the registry. Each entry is an identifier for the responsible component that will remove the entry from the list. If the deletionTimestamp of the object is non-nil, entries in this list can only be removed.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-192 Data structure of the annotations field

Parameter	Mandatory	Type	Description
volume.beta.kubernetes.io/storage-class	Yes	String	Storage class. • EVS Currently, EVS disks of high I/O (SAS disks), ultra-high I/O (SSD disks), and common I/O (SATA disks) types are supported. • SFS Currently, nfs-rw is supported. • OBS Currently, OBS buckets of standard (obs-standard) and low-frequency (obs-standard-ia) types are supported.
volume.beta.kubernetes.io/storage-provisioner	Yes	String	Mount path. • If the storage class is **EVS**, set this parameter to **flexvolume-huawei.com/fuxivol**. • If the storage class is **SFS**, set this parameter to **flexvolume-huawei.com/fuxinfs**. • If the storage class is **OBS**, set this parameter to **flexvolume-huawei.com/fuxiobs**.

Table 4-193 Data structure of the initializers field

Parameter	Mandatory	Type	Description
pending	No	**pending** object	Pending is a list of initializers that must execute in order before this object is visible. When the last pending initializer is removed, and no failing result is set, the initializers struct will be set to nil and the object is considered as initialized and visible to all clients.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
result	No	**result** object	If result is set with the Failure field, the object will be persisted to storage and then deleted, ensuring that other clients can observe the deletion.

Table 4-194 Data structure of the pending field

Parameter	Mandatory	Type	Description
name	No	String	Name of the process that is responsible for initializing this object.

Table 4-195 Data structure of the result field

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
code	No	Integer	Suggested HTTP return code for this status, 0 if not set.
details	No	**details** object	Extended data associated with the reason. Each reason may define its own extended details. This field is optional and the data returned is not guaranteed to conform to any schema except that defined by the reason type.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated.
message	No	String	A human-readable description of the status of this operation.
metadata	Yes	**ListMeta** object	Standard list metadata.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
reason	No	String	A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available. A Reason clarifies an HTTP status code but does not override it.
status	No	String	Status of the operation. One of: "Success" or "Failure".

Table 4-196 Data structure of the details field

Parameter	Mandatory	Type	Description
causes	No	**causes** object	The Causes array includes more details associated with the StatusReason failure. Not all StatusReasons may provide detailed causes.
group	No	String	The group attribute of the resource associated with the status StatusReason.
kind	No	String	The kind attribute of the resource associated with the status StatusReason. On some operations may differ from the requested resource Kind.
name	No	String	The name attribute of the resource associated with the status StatusReason (when there is a single name which can be described).
retryAfterSeconds	No	Integer	If specified, the time in seconds before the operation should be retried.
uid	No	String	UID of the resource. (when there is a single resource which can be described).

Copyright © Huawei Technologies Co., Ltd.

Table 4-197 Data structure of the ListMeta field

Parameter	Mandatory	Type	Description
resourceVersion	No	String	String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only.
Continue	No	String	Continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response.
selfLink	No	String	SelfLink is a URL representing this object. Populated by the system. Read-only.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-198 Data structure of the causes field

Parameter	Mandatory	Type	Description
field	No	String	The field of the resource that has caused this error, as named by its JSON serialization. May include dot and postfix notation for nested attributes. Arrays are zero-indexed. Fields may appear more than once in an array of causes due to fields having multiple errors. Optional. Examples: "name" - the field "name" on the current resource "items[0].name" - the field "name" on the first array entry in "items"
message	No	String	A human-readable description of the cause of the error. This field may be presented as-is to a reader.
reason	No	String	A machine-readable description of the cause of the error. If this value is empty there is no information available.

Table 4-199 Data structure of the ownerReferences field

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	API version of the referent.
blockOwnerDeletion	No	Boolean	If true, AND if the owner has the "foregroundDeletion" finalizer, then the owner cannot be deleted from the key-value store until this reference is removed. Defaults to false. To set this field, a user needs "delete" permission of the owner, otherwise 422 (Unprocessable Entity) will be returned.
kind	Yes	String	Kind of the referent.
name	Yes	String	Name of the referent.
uid	No	String	UID of the referent.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
controller	No	Boolean	If true, this reference points to the managing controller.

Table 4-200 Data structure of volume

Parameter	Mandatory	Type	Description
apiVersion	Yes	String	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values.
kind	Yes	String	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase.
metadata	Yes	**metadata** object	Standard object's metadata.
spec	Yes	**spec** object	Spec defines a specification of a volume owned by the cluster.
status	No	**status** object	Status represents the current information/status for the volume. Populated by the system. Read-only.

Table 4-201 Data structure of the spec field

Parameter	Mandatory	Type	Description
name	Yes	String	Name of this volume.
size	Yes	Integer	Size of this volume.
description	No	String	Description of this volume.
storageclassname	No	String	StorageclassName new add, use to get az and type from k8s.

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
inresourcepool	Yes	Boolean	Whether the volume is in the resource pool.
availability_zone	No	String	AvailabilityZone of this volume.
volume_type	No	String	VolumeType of this volume.
snapshot_id	No	String	SnapshotId of this volume.
multiattach	Yes	Boolean	Multiattach defines whether to attach by multiple containers.
storage_type	No	String	Optional values: BS(Block Storage), OS(Object Storage), NFS(Network File System).Default: BS
share_proto	No	String	When storage_type NFS is required, effective value is NFS.
is_public	No	Boolean	When storage_type is NFS, the visibility of sharing is expressed. Set to true, public visible, set to false, private visible. Default:false
access_to	No	String	When storage_type NFS is required, the definition of the access rule. The length of 1~255, is VPC ID.
access_level	No	String	When storage_type NFS is required, said sharing permission level to access the value of RO (read only), RW (read and write).
pvc_name	No	String	pvcName of volume
access	Yes	Array of **access** object	sfs access.
vpc_id	Yes	String	efs vpc
enterprise_project_id	No	String	enterprise_project_id
volume_id	No	String	volume_id

 Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Type	Description
auto_expand	No	Boolean	When storage_type NFS is required, when value is true capacity expansion is not supported.

Table 4-202 Data structure of the status field

Parameter	Mandatory	Type	Description
id	No	String	A human-readable message indicating details about why the volume is in this state.
status	Yes	String	Phase indicates if a volume is available, bound to a claim, or released by a claim.
created_at	No	String	Reason is a brief CamelCase string that describes any failure and is meant for machine parsing and tidy display in the CLI.
attachments	Yes	Array of **attachment** objects	Attachments information of this volume.
app_info	Yes	Array of **app_info** objects	volume using info.
access_state	No	String	Access state
access_id	No	String	Vpc id
export_location	No	String	Export location
export_locations	No	String	Export locations
x-obs-fs-file-interface	No	Bool	True is obs posix bucket

 Copyright © Huawei Technologies Co., Ltd.

Table 4-203 Data structure of the access field

Parameter	Mandatory	Type	Description
share_id	No	String	uuid of share.
access_type	Yes	String	access rule type.
access_to	Yes	String	vpc id.
access_level	Yes	String	access level.
id	Yes	String	uuid of access rule.
state	Yes	String	status of access rule.should be active or error.

Table 4-204 Data structure of the attachment field

Parameter	Mandatory	Type	Description
attachment_id	No	String	AttachmentId.
server_id	No	String	Server of Attached device.
host_name	No	String	Host name of the attached machine.
device	No	String	Attached device.

Table 4-205 Data structure of the app_info field

Parameter	Mandatory	Type	Description
app_name	Yes	String	App name.
namespace	Yes	String	namespace.
mount_path	Yes	String	Mount path.
app_type	Yes	String	App type.

4.3.2 Volcano Job

 Copyright © Huawei Technologies Co., Ltd.

4.3.2.1 Reading All Volcano Jobs Under a Namespace

Function

This API is used to read all Volcano jobs under a specified namespace.

URI

GET /apis/batch.volcano.sh/v1alpha1/namespaces/{namespace}/jobs

Table 4-206 Path parameter

Parameter	Mandatory	Description
namespace	Yes	Object name and auth scope, such as for teams and projects.

Table 4-207 Query parameters

Parameter	Mandatory	Description
fieldSelector	No	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	No	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Description
limit	No	Limit is a maximum number of responses to return for a list call. If more items exist, the server will set the continue field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
resourceVersion	No	When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it is 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv.
timeoutSeconds	No	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.
watch	No	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

 Copyright © Huawei Technologies Co., Ltd.

Request

N/A

Response

Response parameters

For the description about response parameters, see **Table 4-146**.

Example response


```
{
   "apiVersion": "batch.volcano.sh/v1alpha1",
   "items": [
      {
         "apiVersion": "batch.volcano.sh/v1alpha1",
         "kind": "Job",
         "metadata": {
            "creationTimestamp": "2019-06-26T03:16:26Z",
            "generation": 1,
            "name": "openmpi-hello-2-com",
            "namespace": "cci-namespace-42263891",
            "resourceVersion": "7625538",
            "selfLink": "/apis/batch.volcano.sh/v1alpha1/namespaces/cci-namespace-42263891/jobs/
openmpi-hello-2-com",
            "uid": "c84d86f0-97c0-11e9-9d09-dc9914fb58e0"
         },
         "spec": {
            "minAvailable": 1,
            "plugins": {
               "env": [],
               "ssh": [],
               "svc": []
            },
            "queue": "default",
            "schedulerName": "volcano",
            "tasks": [
               {
                  "name": "mpimaster",
                  "policies": [
                     {
                        "action": "CompleteJob",
                        "event": "TaskCompleted"
                     }
                  ],
                  "replicas": 1,
                  "template": {
                     "spec": {
                        "containers": [
                           {
                              "command": [
                                 "/bin/sh",
                                 "-c",
                                 "MPI_HOST=`cat /etc/volcano/mpiworker.host | tr \"\\n\" \",\"`;\nmkdir -
p /var/run/sshd; /usr/sbin/sshd;\nmpiexec --allow-run-as-root --host ${MPI_HOST} -np 2 mpi_hello_world
003e /home/re\n"
                              ],
                              "image": "*.*.5.235:20202/swr/openmpi-hello:3.28",
                              "name": "mpimaster",
                              "ports": [
                                 {
                                    "containerPort": 22,
                                    "name": "mpijob-port"
                                 }
                              ],
                              "resources": {
                                 "limits": {
```


 Copyright © Huawei Technologies Co., Ltd.

```
                                "cpu": "250m",
                                "memory": "1Gi"
                            },
                            "requests": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            }
                        },
                        "workingDir": "/home"
                    }
                ],
                "imagePullSecrets": [
                    {
                        "name": "default-secret"
                    }
                ],
                "restartPolicy": "OnFailure"
            }
        }
    },
    {
        "name": "mpiworker",
        "replicas": 2,
        "template": {
            "spec": {
                "containers": [
                    {
                        "command": [
                            "/bin/sh",
                            "-c",
                            "mkdir -p /var/run/sshd; /usr/sbin/sshd -D;\n"
                        ],
                        "image": "*.*.*.*:20202/swr/openmpi-hello:3.28",
                        "name": "mpiworker",
                        "ports": [
                            {
                                "containerPort": 22,
                                "name": "mpijob-port"
                            }
                        ],
                        "resources": {
                            "limits": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            },
                            "requests": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            }
                        },
                        "workingDir": "/home"
                    }
                ],
                "imagePullSecrets": [
                    {
                        "name": "default-secret"
                    }
                ],
                "restartPolicy": "OnFailure"
            }
        }
    }
  ]
},
"status": {
    "controlledResources": {
        "plugin-env": "env",
        "plugin-ssh": "ssh",
        "plugin-svc": "svc"
```

 Copyright © Huawei Technologies Co., Ltd.

```
            },
            "minAvailable": 1,
            "pending": 3,
            "state": {
                "lastTransitionTime": "2019-06-26T03:16:27Z",
                "phase": "Inqueue"
            }
        }
    }
  ],
  "kind": "JobList",
  "metadata": {
    "continue": "",
    "resourceVersion": "7678090",
    "selfLink": "/apis/batch.volcano.sh/v1alpha1/namespaces/cci-namespace-42263891/jobs"
  }
}
```


Status Code

Table 4-208 Status codes

Status Code	Description
200	OK
401	Unauthorized
404	Not found
500	Internal error

4.3.2.2 Creating a Volcano Job

Function

This API is used to create a Volcano job.

URI

POST /apis/batch.volcano.sh/v1alpha1/namespaces/{namespace}/jobs

Table 4-209 Path parameter

Parameter	Mandatory	Description
namespace	Yes	object name and auth scope, such as for teams and projects. The namespace need be existent before using this url.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-210 Query parameter

Parameter	Mandatory	Description
pretty	No	If 'true', then the output is pretty printed.

Request

For the description about request parameters, see **Table 4-146**.

Example request

```
{
   "apiVersion": "batch.volcano.sh/v1alpha1",
   "kind": "Job",
   "metadata": {
      "name": "openmpi-hello-2-com"
   },
   "spec": {
      "minAvailable": 1,
      "schedulerName": "volcano",
      "plugins": {
         "ssh": [],
         "env": [],
         "svc": []
      },
      "tasks": [
         {
            "replicas": 1,
            "name": "mpimaster",
            "policies": [
               {
                  "event": "TaskCompleted",
                  "action": "CompleteJob"
               }
            ],
            "template": {
               "spec": {
                  "imagePullSecrets": [
                     {
                        "name": "default-secret"
                     }
                  ],
                  "containers": [
                     {
                        "command": [
                           "/bin/sh",
                           "-c",
                           "MPI_HOST=`cat/etc/volcano/mpiworker.host|tr\"\\n\"\",\"`;\nmkdir-p/var/run/sshd;/usr/sbin/sshd;\nmpiexec--allow-run-as-root--host${\t\t\t\t\t\t\tMPI_HOST\t\t\t\t\t\t}-np2mpi_hello_world>/home/re\n"
                        ],
                        "image": "*.*.*.*: 20202/swr/openmpi-hello:3.28",
                        "name": "mpimaster",
                        "ports": [
                           {
                              "containerPort": 22,
                              "name": "mpijob-port"
                           }
                        ],
                        "resources": {
                           "requests": {
                              "cpu": "250m",
                              "memory": "1Gi"
```

 Copyright © Huawei Technologies Co., Ltd.

```
                        },
                        "limits": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        }
                    },
                    "workingDir": "/home"
                }
            ],
            "restartPolicy": "OnFailure"
        }
    }
},
{
    "replicas": 2,
    "name": "mpiworker",
    "template": {
        "spec": {
            "imagePullSecrets": [
                {
                    "name": "default-secret"
                }
            ],
            "containers": [
                {
                    "command": [
                        "/bin/sh",
                        "-c",
                        "mkdir-p/var/run/sshd;/usr/sbin/sshd-D;\n"
                    ],
                    "image": "*.*.*.*:20202/swr/openmpi-hello:3.28",
                    "name": "mpiworker",
                    "ports": [
                        {
                            "containerPort": 22,
                            "name": "mpijob-port"
                        }
                    ],
                    "workingDir": "/home",
                    "resources": {
                        "requests": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        },
                        "limits": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        }
                    }
                }
            ],
            "restartPolicy": "OnFailure"
        }
    }
}
]
}
}
```

Response

Response parameters:

For the description about response parameters, see **Table 4-146**.

Example response

```
{
    "apiVersion": "batch.volcano.sh/v1alpha1",
```

 Copyright © Huawei Technologies Co., Ltd.

```
  "kind": "Job",
  "metadata": {
    "creationTimestamp": "2019-06-26T06:24:50Z",
    "generation": 1,
    "name": "openmpi-hello-3-com",
    "namespace": "cci-namespace-42263891",
    "resourceVersion": "7681331",
    "selfLink": "/apis/batch.volcano.sh/v1alpha1/namespaces/cci-namespace-42263891/jobs/openmpi-
hello-3-com",
    "uid": "1a32a8c4-97db-11e9-9d09-dc9914fb58e0"
  },
  "spec": {
    "minAvailable": 1,
    "plugins": {
      "env": [],
      "ssh": [],
      "svc": []
    },
    "queue": "default",
    "schedulerName": "volcano",
    "tasks": [
      {
        "name": "mpimaster",
        "policies": [
          {
            "action": "CompleteJob",
            "event": "TaskCompleted"
          }
        ],
        "replicas": 1,
        "template": {
          "spec": {
            "containers": [
              {
                "command": [
                  "/bin/sh",
                  "-c",
                  "MPI_HOST=`cat /etc/volcano/mpiworker.host | tr \"\\n\" \",\"`;\nmkdir -p /var/run/
sshd; /usr/sbin/sshd;\nmpiexec --allow-run-as-root --host ${MPI_HOST} -np 2 mpi_hello_world 003e /
home/re\n"
                ],
                "image": "*.*.*.*:20202/swr/openmpi-hello:3.28",
                "name": "mpimaster",
                "ports": [
                  {
                    "containerPort": 22,
                    "name": "mpijob-port"
                  }
                ],
                "resources": {
                  "limits": {
                    "cpu": "250m",
                    "memory": "1Gi"
                  },
                  "requests": {
                    "cpu": "250m",
                    "memory": "1Gi"
                  }
                },
                "workingDir": "/home"
              }
            ],
            "imagePullSecrets": [
              {
                "name": "default-secret"
              }
            ],
            "restartPolicy": "OnFailure"
          }
```


 Copyright © Huawei Technologies Co., Ltd.

```
            }
        },
        {
            "name": "mpiworker",
            "replicas": 2,
            "template": {
                "spec": {
                    "containers": [
                        {
                            "command": [
                                "/bin/sh",
                                "-c",
                                "mkdir -p /var/run/sshd; /usr/sbin/sshd -D;\n"
                            ],
                            "image": "*.*.*.*:20202/swr/openmpi-hello:3.28",
                            "name": "mpiworker",
                            "ports": [
                                {
                                    "containerPort": 22,
                                    "name": "mpijob-port"
                                }
                            ],
                            "resources": {
                                "limits": {
                                    "cpu": "250m",
                                    "memory": "1Gi"
                                },
                                "requests": {
                                    "cpu": "250m",
                                    "memory": "1Gi"
                                }
                            },
                            "workingDir": "/home"
                        }
                    ],
                    "imagePullSecrets": [
                        {
                            "name": "default-secret"
                        }
                    ],
                    "restartPolicy": "OnFailure"
                }
            }
        }
    ]
  }
}
```

Status Code

Table 4-211 Status codes

Status Code	Description
200	OK
201	Created
202	Accepted
401	Unauthorized
400	Badrequest
500	Internal error

 Copyright © Huawei Technologies Co., Ltd.

Status Code	Description
403	Forbidden

4.3.2.3 Deleting All Volcano Jobs Under a Namespace

Function

This API is used to delete all Volcano jobs under a specified namespace.

URI

DELETE /apis/batch.volcano.sh/v1alpha1/namespaces/{namespace}/jobs

Table 4-212 Path parameter

Parameter	Mandatory	Description
namespace	Yes	Object name and auth scope, such as for teams and projects.

Table 4-213 Query parameters

Parameter	Mandatory	Description
fieldSelector	No	A selector to restrict the list of returned objects by their fields. Defaults to everything.
labelSelector	No	A selector to restrict the list of returned objects by their labels. Defaults to everything.

Copyright © Huawei Technologies Co., Ltd.

Parameter	Mandatory	Description
limit	No	Limit is a maximum number of responses to return for a list call. If more items exist, the server will set the continue field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true. The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
resourceVersion	No	When specified with a watch call, shows changes that occur after that particular version of a resource. Defaults to changes from the beginning of history. When specified for list: - if unset, then the result is returned from remote storage based on quorum-read flag; - if it is 0, then we simply return what we currently have in cache, no guarantee; - if set to non zero, then the result is at least as fresh as given rv.
timeoutSeconds	No	Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.
watch	No	Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

 Copyright © Huawei Technologies Co., Ltd.

Request

N/A

Response

Response parameters

For the description about response parameters, see **Table 4-146**.

Example response


```
{
   "apiVersion": "batch.volcano.sh/v1alpha1",
   "items": [
      {
         "apiVersion": "batch.volcano.sh/v1alpha1",
         "kind": "Job",
         "metadata": {
            "creationTimestamp": "2019-06-26T03:16:26Z",
            "generation": 1,
            "labels": {
               "app": "patchlabel"
            },
            "name": "openmpi-hello-2-com",
            "namespace": "cci-namespace-42263891",
            "resourceVersion": "7695210",
            "selfLink": "/apis/batch.volcano.sh/v1alpha1/namespaces/cci-namespace-42263891/jobs/
openmpi-hello-2-com",
            "uid": "c84d86f0-97c0-11e9-9d09-dc9914fb58e0"
         },
         "spec": {
            "minAvailable": 1,
            "plugins": {
               "env": [],
               "ssh": [],
               "svc": []
            },
            "queue": "default",
            "schedulerName": "volcano",
            "tasks": [
               {
                  "name": "mpimaster",
                  "policies": [
                     {
                        "action": "CompleteJob",
                        "event": "TaskCompleted"
                     }
                  ],
                  "replicas": 1,
                  "template": {
                     "spec": {
                        "containers": [
                           {
                              "command": [
                                 "/bin/sh",
                                 "-c",
                                 "MPI_HOST=`cat /etc/volcano/mpiworker.host | tr \"\\n\" \",\"`;\nmkdir -
p /var/run/sshd; /usr/sbin/sshd;\nmpiexec --allow-run-as-root --host ${MPI_HOST} -np 2 mpi_hello_world
003e /home/re\n"
                              ],
                              "image": "*.*.*.*:20202/l00427178/openmpi-hello:3.28",
                              "name": "mpimaster",
                              "ports": [
                                 {
                                    "containerPort": 22,
                                    "name": "mpijob-port"
                                 }
```


 Copyright © Huawei Technologies Co., Ltd.

```
                        ],
                        "resources": {
                            "limits": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            },
                            "requests": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            }
                        },
                        "workingDir": "/home"
                    }
                ],
                "imagePullSecrets": [
                    {
                        "name": "default-secret"
                    }
                ],
                "restartPolicy": "OnFailure"
            }
        }
    },
    {
        "name": "mpiworker",
        "replicas": 2,
        "template": {
            "spec": {
                "containers": [
                    {
                        "command": [
                            "/bin/sh",
                            "-c",
                            "mkdir -p /var/run/sshd; /usr/sbin/sshd -D;\n"
                        ],
                        "image": "*.*.*.*:20202/l00427178/openmpi-hello:3.28",
                        "name": "mpiworker",
                        "ports": [
                            {
                                "containerPort": 22,
                                "name": "mpijob-port"
                            }
                        ],
                        "resources": {
                            "limits": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            },
                            "requests": {
                                "cpu": "250m",
                                "memory": "1Gi"
                            }
                        },
                        "workingDir": "/home"
                    }
                ],
                "imagePullSecrets": [
                    {
                        "name": "default-secret"
                    }
                ],
                "restartPolicy": "OnFailure"
            }
        }
    }
]
},
"status": {
    "controlledResources": {
```


 Copyright © Huawei Technologies Co., Ltd.

```
                "plugin-env": "env",
                "plugin-ssh": "ssh",
                "plugin-svc": "svc"
            },
            "minAvailable": 1,
            "pending": 3,
            "state": {
                "lastTransitionTime": "2019-06-26T03:16:27Z",
                "phase": "Inqueue"
            }
        }
    }
  ],
  "kind": "JobList",
  "metadata": {
    "continue": "",
    "resourceVersion": "7732232",
    "selfLink": "/apis/batch.volcano.sh/v1alpha1/namespaces/cci-namespace-42263891/jobs"
  }
}
```


Status Code

Table 4-214 Status codes

Status Code	Description
200	OK
401	Unauthorized
500	Internal error

4.3.2.4 Reading the Details of a Volcano Job

Function

This API is used to read the details about a specified Volcano job.

URI

GET /apis/batch.volcano.sh/v1alpha1/namespaces/{namespace}/jobs/{name}

Table 4-215 Path parameters

Parameter	Mandatory	Description
name	No	name of the Volcano Job. Null name means all jobs.
namespace	Yes	Object name and auth scope, such as for teams and projects.

Copyright © Huawei Technologies Co., Ltd.

Table 4-216 Query parameter

Parameter	Mandatory	Description
pretty	No	If 'true', then the output is pretty printed.

Request

N/A

Response

Response parameters

For the description about response parameters, see **Table 4-146**.

Example response


```
{
   "apiVersion": "batch.volcano.sh/v1alpha1",
   "kind": "Job",
   "metadata": {
      "creationTimestamp": "2019-06-26T06:24:50Z",
      "generation": 1,
      "name": "openmpi-hello-3-com",
      "namespace": "cci-namespace-42263891",
      "resourceVersion": "7681358",
      "selfLink": "/apis/batch.volcano.sh/v1alpha1/namespaces/cci-namespace-42263891/jobs/openmpi-hello-3-com",
      "uid": "1a32a8c4-97db-11e9-9d09-dc9914fb58e0"
   },
   "spec": {
      "minAvailable": 1,
      "plugins": {
         "env": [],
         "ssh": [],
         "svc": []
      },
      "queue": "default",
      "schedulerName": "volcano",
      "tasks": [
         {
            "name": "mpimaster",
            "policies": [
               {
                  "action": "CompleteJob",
                  "event": "TaskCompleted"
               }
            ],
            "replicas": 1,
            "template": {
               "spec": {
                  "containers": [
                     {
                        "command": [
                           "/bin/sh",
                           "-c",
                           "MPI_HOST=`cat /etc/volcano/mpiworker.host | tr \"\\n\" \",\"`;\nmkdir -p /var/run/sshd; /usr/sbin/sshd;\nmpiexec --allow-run-as-root --host ${MPI_HOST} -np 2 mpi_hello_world 003e /home/re\n"
                        ],
                        "image": "*.*.*.*:20202/swr/openmpi-hello:3.28",
                        "name": "mpimaster",
```


 Copyright © Huawei Technologies Co., Ltd.

```
                    "ports": [
                        {
                            "containerPort": 22,
                            "name": "mpijob-port"
                        }
                    ],
                    "resources": {
                        "limits": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        },
                        "requests": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        }
                    },
                    "workingDir": "/home"
                }
            ],
            "imagePullSecrets": [
                {
                    "name": "default-secret"
                }
            ],
            "restartPolicy": "OnFailure"
        }
    }
},
{
    "name": "mpiworker",
    "replicas": 2,
    "template": {
        "spec": {
            "containers": [
                {
                    "command": [
                        "/bin/sh",
                        "-c",
                        "mkdir -p /var/run/sshd; /usr/sbin/sshd -D;\n"
                    ],
                    "image": "*.*.*.*:20202/swr/openmpi-hello:3.28",
                    "name": "mpiworker",
                    "ports": [
                        {
                            "containerPort": 22,
                            "name": "mpijob-port"
                        }
                    ],
                    "resources": {
                        "limits": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        },
                        "requests": {
                            "cpu": "250m",
                            "memory": "1Gi"
                        }
                    },
                    "workingDir": "/home"
                }
            ],
            "imagePullSecrets": [
                {
                    "name": "default-secret"
                }
            ],
            "restartPolicy": "OnFailure"
        }
    }
```

 Copyright © Huawei Technologies Co., Ltd.

```
            }
        ]
    },
    "status": {
        "controlledResources": {
            "plugin-env": "env",
            "plugin-ssh": "ssh",
            "plugin-svc": "svc"
        },
        "minAvailable": 1,
        "pending": 3,
        "state": {
            "lastTransitionTime": "2019-06-26T06:24:51Z",
            "phase": "Inqueue"
        }
    }
}
```


Status Code

Table 4-217 Status codes

Status Code	Description
200	OK
401	Unauthorized
404	Not found
500	Internal error
403	Forbidden

4.3.2.5 Deleting a Volcano Job

Function

This API is used to delete a specified Volcano job.

URI

DELETE /apis/batch.volcano.sh/v1alpha1/namespaces/{namespace}/jobs/{name}

Table 4-218 Path parameters

Parameter	Mandatory	Description
name	Yes	name of the Volcano Job. When name is null, it means delete all the job belong this namespace.
namespace	Yes	Object name and auth scope, such as for teams and projects.

 Copyright © Huawei Technologies Co., Ltd.

Table 4-219 Query parameters

Parameter	Mandatory	Description
dryRun	No	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
gracePeriodSeconds	No	The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. The value **0** indicates to delete immediately.
orphanDependents	No	Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
propagationPolicy	No	Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.
pretty	No	If 'true', then the output is pretty printed.

Request

N/A

Response

Response parameters

For the description about response parameters, see **Table 4-73**.

Example response

```
{
    "kind": "Status",
    "apiVersion": "v1",
```

 Copyright © Huawei Technologies Co., Ltd.

```
  "metadata": {},
  "status": "Success",
  "details": {
     "name": "openmpi-hello-3-com",
     "group": "batch.volcano.sh",
     "kind": "jobs",
     "uid": "1a32a8c4-97db-11e9-9d09-dc9914fb58e0"
  }
}
```

Status Code

Table 4-220 Status codes

Status Code	Description
200	OK
202	Accepted
401	Unauthorized
500	Internal Error
403	Forbidden

4.3.3 cAdvisor

4.3.3.1 Obtaining Ascend AI Processor Monitoring Information

Function

The original cAdvisor machine information API is added to obtain the basic Ascend AI Processor information, as listed in **Table 4-221**.

Table 4-221 Ascend AI Processor information

NPU quantity
NPU device list
Device health status
Device error code
Device usage
Device frequency
Device temperature
Device power
Device voltage

 Copyright © Huawei Technologies Co., Ltd.

Device HBM memory information (for Ascend 910 AI Processor only)
Device memory information

URL

GET http://ip:port/api/v1.0/machine

NOTE

- **ip**: IP address of the cAdvisor container. By default, the IP address is not exposed to the host port and cannot be accessed using the host and port number.
- **port**: port number. The default value is 8081. If you need to change the port number, change the value of **ports** in **deploy/kubernetes/base/daemonset.yaml**. After **hostPort** is added, the port number can be exposed to the host. You can also change the port number. To change the port number, change the value of **--port** in **deploy/kubernetes/overlays/huawei/cadvisor-args.yaml**.

Request Parameters

N/A

Response Description

```
{
   "num_cores": 72,
   "cpu_frequency_khz": 2601000,
   "memory_capacity": 607924887552,
   "hugepages": [],
   "machine_id": "4b8a3236aa884f7fa1aa2ce868205768",
   "system_uuid": "E1C5D866-0018-8CA3-B211-D21D0CDA1A24",
   "boot_id": "dc356dae-d541-4c81-ad37-9f3beda84f3a",
   "filesystems": [],
   "disk_map": {},
   "network_devices": [],
   "topology": [],
   "cloud_provider": "Unknown",
   "instance_type": "Unknown",
   "instance_id": "None",
   "npu_list": [                              ## List of new Ascend AI Processors
      {
         "device_id": 0,                      ## NPU device ID
         "device_list": [
            {
               "health_status": "Healthy",      ## NPU device monitoring status, Healthy or Unhealthy
               "error_code": 0,                 ## NNPU error code. 0 indicates normal.
               "utilization": 0,                ## NPU AI Core Usage
               "temperature": 63,                  ## Device temperature (°C)
               "power": 76.4,                   ## Device power consumption, in W
               "voltage": 12.24,                ## Device voltage, in V
               "frequency": 2000,                 ## NPU AI core working frequency, in MHz
               "memory_info": {                 ## NPU device memory information
                  "memory_size": 15307,         ## Memory size, in MB
                  "memory_frequency": 1200,     ## Memory frequency, in MHz
                  "memory_utilization": 1       ## Memory usage (%)
               },
               "chip_info": {                   ## Processor information
                  "chip_type": "Ascend",        ## Processor type
                  "chip_name": "910own",        ## Processor name
                  "chip_version": "V1"          ## Processor version
               },
```

 Copyright © Huawei Technologies Co., Ltd.

```
                "hbm_info": {                ## HBM information (for Ascend 910 AI Processor only)
                    "memory_size": 32255,        ## HBM memory size, in MB
                    "hbm_frequency": 1200,       ## HBM working frequency, in MHx
                    "memory_usage": 0,           ## Used HBM memory, in MB
                    "hbm_temperature": 62,       ## HBM temperature (°C)
                    "hbm_bandwidth_util": 0      ## HBM bandwidth usage (%)
                }
            }
        ],
        "timestamp": "2020-09-24T16:33:53.673903765+08:00"
    },
     ...                                ##Other processor information is omitted.
  ]
}
```

NOTE

For details about other APIs, see **official cAdvisor documentation**.

Status Code

Table 4-222 Status code

Status Code	Description
200	Normal
307	Temporary redirection
500	Internal server error

4.3.3.2 cAdvisor Prometheus Metrics API

Function

The Metrics API is available for Prometheus to call and integrate.

URL

GET http://ip:port/metrics

NOTE

For security purposes, cAdvisor enables the container-level port by default, and the request IP address is the IP address of the Kubernetes container.

Request Parameters

N/A

Response Description

The data is returned in the Prometheus-specific format.

```
# HELP cadvisor_version_info A metric with a constant '1' value labeled by kernel version, OS version, docker version, cadvisor version & cadvisor revision. # TYPE cadvisor_version_info gauge
cadvisor_version_info{cadvisorRevision="unknown",cadvisorVersion="v0.34.0-r40",dockerVersion="18.06.3-
```

 Copyright © Huawei Technologies Co., Ltd.

```
ce",kernelVersion="4.15.0-29-generic",osVersion="Alpine Linux v3.12"} 1
...
# HELP container_accelerator_duty_cycle Percent of time over the past sample period during which the accelerator was actively processing.
# TYPE container_accelerator_duty_cycle gauge
container_accelerator_duty_cycle{acc_id="davinci1",...
...
# HELP container_accelerator_memory_total_bytes Total accelerator memory.
# TYPE container_accelerator_memory_total_bytes gauge
container_accelerator_memory_total_bytes{acc_id="davinci1",...
...
# HELP container_accelerator_memory_used_bytes Total accelerator memory allocated.
# TYPE container_accelerator_memory_used_bytes gauge
container_accelerator_memory_used_bytes{acc_id="davinci1",...
...
# TYPE machine_npu_nums gauge
machine_npu_nums 8
# HELP npu_chip_info_health_status the npu health status
# TYPE npu_chip_info_health_status gauge
npu_chip_info_health_status{id="0"} 1 1597126711464
npu_chip_info_health_status{id="1"} 1 1597126711472
npu_chip_info_health_status{id="2"} 1 1597126711479
npu_chip_info_health_status{id="3"} 1 1597126711487
npu_chip_info_health_status{id="4"} 1 1597126711493
npu_chip_info_health_status{id="5"} 1 1597126711502
npu_chip_info_health_status{id="6"} 1 1597126711509
npu_chip_info_health_status{id="7"} 1 1597126711517
...
...
```

Table 4-223 Prometheus labels

Label	Description	Unit
container_accelerator_duty_cycle	Accelerator usage in a container.	%
container_accelerator_memory_total_bytes	Total accelerator memory size in a container.	Byte
container_accelerator_memory_used_bytes	Used accelerator memory size in a container.	Byte
machine_npu_nums	Number of Ascend AI Processors.	Number
machine_npu_name	Name of the Ascend AI Processor.	-
npu_chip_info_error_code	Error code of the Ascend AI Processor.	-
npu_chip_info_health_status	Health status of an Ascend AI Processor.	• 1: healthy • 0: unhealthy
npu_chip_info_power	Power consumption of an Ascend AI Processor.	W
npu_chip_info_temperature	Temperature of an Ascend AI Processor.	°C

 Copyright © Huawei Technologies Co., Ltd.

Label	Description	Unit
npu_chip_info_used_memory	Used memory of the Ascend AI Processor.	MB
npu_chip_info_total_memory	Total memory of the Ascend AI Processor.	MB
npu_chip_info_hbm_used_memory	Used HBM memory dedicated for the Ascend 910 AI Processor.	MB
npu_chip_info_hbm_total_memory	Total HBM memory dedicated for the Ascend 910 AI Processor.	MB
npu_chip_info_utilization	AI Core usage of an Ascend AI Processor.	%
npu_chip_info_voltage	Voltage of an Ascend AI Processor.	V

NOTE

- For details about other Prometheus labels, see the **official cAdvisor documentation**.
- The accelerator tag in the cAdvisor container is displayed only when an NPU or GPU accelerator is mounted to a container.

Status Code

Table 4-224 Status code

Status Code	Description
200	Normal
307	Temporary redirection
500	Internal server error

4.3.3.3 Other cAdvisor APIs

For details about other APIs, see **official cAdvisor documentation**.

Copyright © Huawei Technologies Co., Ltd.

5 FAQs

5.1 Pod Remains in the Terminating State After vcjob Is Manually Deleted

Symptom

After vcjob is deleted using **kubectl delete -f xxx.yaml**, the pod remains in the **Terminating** state.

```
root@ubuntu:/home/btg# kubectl get pod --all-namespaces
NAMESPACE        NAME                                       READY   STATUS        RESTARTS   AGE
default          hccl-controller-7cf7945574-w2lqr           1/1     Running       0          4d17h
kube-system      ascend-device-plugin-daemonset-wgb2m       1/1     Running       0          47h
kube-system      calico-kube-controllers-8464785d6b-8ck4q   1/1     Running       0          4d18h
kube-system      calico-node-vb4t2                          1/1     Running       1          4d18h
kube-system      coredns-6955765f44-hxlzl                   1/1     Running       0          4d18h
kube-system      coredns-6955765f44-jpznk                   1/1     Running       40         4d18h
kube-system      etcd-ubuntu                                1/1     Running       0          4d18h
kube-system      kube-apiserver-ubuntu                      1/1     Running       48         4d18h
kube-system      kube-controller-manager-ubuntu             1/1     Running       0          4d18h
kube-system      kube-proxy-g8sgs                           1/1     Running       0          4d18h
kube-system      kube-scheduler-ubuntu                      1/1     Running       0          4d18h
kube-system      resnet-default-test-0                      1/1     Terminating   0          52m
volcano-system   volcano-admission-5867848568-ft8p2         1/1     Running       5          4d18h
volcano-system   volcano-admission-init-5g6cw               0/1     Error         0          4d18h
volcano-system   volcano-admission-init-ttrnc               0/1     Error         0          4d18h
volcano-system   volcano-admission-init-vrh65               0/1     Error         0          4d18h
volcano-system   volcano-admission-init-z9pjh               0/1     Error         0          4d18h
volcano-system   volcano-controllers-6f84f9f8d-gdnj4        1/1     Running       0          4d18h
volcano-system   volcano-scheduler-5f8fdc7996-vps7h         1/1     Running       1          4d18h
root@ubuntu:/home/btg#
```

Possible Causes

N/A

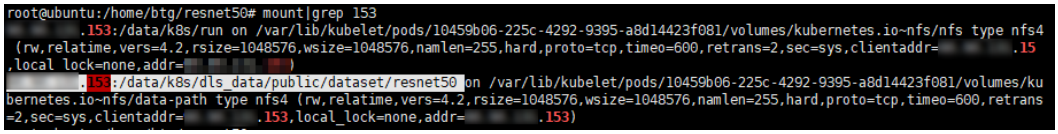
Method 1: Unmounting the NFS Mounting Paths of the Pod

Step 1 Run the following command to check the NFS mounting paths of the pod:

mount|grep *NFS share IP address*

 Copyright © Huawei Technologies Co., Ltd.

Figure 5-1 Query result



As shown in the figure, **xxx.xxx.xxx.xxx:/data/k8s/run** and **xxx.xxx.xxx.xxx:/data/k8s/dls_data/public/dataset/resnet50** are the NFS mounting paths of the pod.

Step 2 Run the following command to unmount each NFS mounting path of the pod:

umount -f *NFS mounting path*

Step 3 Run the following command to check whether the NFS mounting paths of the pod have been unmounted:

mount|grep *NFS share IP address*

- If yes, no further action is required.
- If no, go to **Method 2: Deleting the Docker Process to Which the Pod Belongs**.

----End

Method 2: Deleting the Docker Process to Which the Pod Belongs

Step 1 Run the following command to query the Docker process to which the pod belongs:

docker ps |grep *Pod name*

Step 2 Run the following command to check the files occupied by the Docker process:

ll /var/lib/docker/containers |grep *Docker process ID*

The following is an example of the query result.

```
root@ubuntu:/data/k8s/run# ll /var/lib/docker/containers |grep 95aeeafe2db8
drwx------ 4 root root 4096 Jun 24 16:00
95aeeafe2db898065094dd34dbfbeca04734d5248316aa802d43a36b4d8b99df/
```

Step 3 Run the following command to delete the files occupied by the Docker process:

rm -rf */var/lib/docker/container/*
95aeeafe2db898065094dd34dbfbeca04734d5248316aa802d43a36b4d8b99df/

Step 4 Run the following command to query the ID of the Docker process that occupies the files:

lsof |grep *95aeeafe2db8*

 Copyright © Huawei Technologies Co., Ltd.

Figure 5-2 Query result

Step 5 Run the following command to kill the process:

kill -9 *Process ID*

Step 6 Run the following command to check whether the process has been deleted:

- If yes, go to **Step 7**.
- If no, query and stop the process again by referring to **Step 4** and **Step 7**.

Step 7 Run the following command to delete the Docker to which the pod belongs:

docker rm *95aeeafe2db8*

After the pod is deleted, wait for about 1 minute and then view the pod information again.

----End

5.2 The Training Task Is in the Pending State Because "nodes are unavailable"

Symptom

After being delivered, the vcjob training job is not running.

Step 1 Run the **kubectl get pod --all-namespaces** command to check whether the pod to which the training job belongs is in the **Pending** state, as shown in the following figure.

 Copyright © Huawei Technologies Co., Ltd.

```
[root@bc-linux-3 sasa]# kubectl get pod --all-namespaces
NAMESPACE              NAME                                        READY   STATUS      RESTARTS   AGE
cadvisor               cadvisor-2glzt                              1/1     Running     0          13m
cadvisor               cadvisor-7f2h2                              1/1     Running     4          3d15h
default                hccl-controller-8b44b64d9-qlk6c             1/1     Running     0          3d21h
job                    lky-eval-resnet50-infer1-001-6n2pv          0/1     Completed   0          51m
kube-system            ascend-device-plugin-daemonset-9tztf        1/1     Running     1          19h
kube-system            ascend-device-plugin2-daemonset-kvdn5       1/1     Running     0          3m54s
kube-system            calico-kube-controllers-7756d8ddd6-hkppt    1/1     Running     2          24d
kube-system            calico-node-gbcgs                           1/1     Running     2          24d
kube-system            calico-node-j4t9z                           1/1     Running     7          16d
kube-system            calico-node-nsxsv                           1/1     Running     0          15h
kube-system            coredns-559688597c-2ngj7                    1/1     Running     2          24d
kube-system            coredns-559688597c-464ph                    1/1     Running     3          24d
kube-system            etcd-bc-linux-60.3                          1/1     Running     2          24d
kube-system            kube-apiserver-bc-linux-60.3                1/1     Running     2          24d
kube-system            kube-controller-manager-bc-linux-60.3       1/1     Running     11         24d
kube-system            kube-proxy-hgq54                            1/1     Running     8          16d
kube-system            kube-proxy-jpl6r                            1/1     Running     2          24d
kube-system            kube-proxy-lkv7x                            1/1     Running     0          15h
kube-system            kube-scheduler-bc-linux-60.3                1/1     Running     3          24d
kube-system            resnet1-1-default-test-0                    1/1     Running     0          18h
kubernetes-dashboard   dashboard-metrics-scraper-c89857586-b46nq   1/1     Running     0          4d20h
kubernetes-dashboard   kubernetes-dashboard-84fd684c8d-vlh74       1/1     Running     0          4d20h
vcjob                  lky0815-resnet1-009-default-test-0          0/1     Completed   0          35m
vcjob                  lky0815-resnet4-001-default-test-0          0/1     Completed   0          71m
vcjob                  sasa-resnet1-acc-default-test-0             0/1     Pending     0          1s
volcano-system         volcano-admission-6fb8cc9bd-zfqv4           1/1     Running     0          3d20h
```

Step 2 Run the **kubectl describe pod sasa-resnet1-acc-default-test-0 -n vcjob** command to view the pod details. In the **event** field, the following error is reported: all nodes are unavailable: 1 node annotations(7) not same node idle(8).

```
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type     Reason            Age   From     Message
  ----     ------            ----  ----     -------
  Warning  FailedScheduling  12m   volcano  all nodes are unavailable: 1 node annotations(7) not same node idle(8).
```

----End

Possible Causes

The number of unused NPUs on the node was different from the number of unused NPUs displayed in **Annotations**. Volcano considered that the system was unstable and could not allocate NPU resources.

The **kubectl describe nodes** command was run to check the **huawei.com/Ascend910:** field in **Allocated resources** and **Annotations** of the node.

```
root@ubuntu-217-94:~# kubectl describe nodes
Name:               ubuntu-217-94
Roles:              master
Labels:             accelerator=huawei-Ascend910
                    beta.kubernetes.io/arch=arm64
                    beta.kubernetes.io/fluentd-ds-ready=true
                    beta.kubernetes.io/os=linux
                    host-arch=huawei-arm
                    kubernetes.io/arch=arm64
                    kubernetes.io/hostname=ubuntu-217-94
                    kubernetes.io/os=linux
                    masterselector=dls-master-node
                    node-role.kubernetes.io/master=
                    workerselector=dls-worker-node
Annotations:        huawei.com/Ascend910: Ascend910-4,Ascend910-5,Ascend910-6,Ascend910-7,Ascend910-1,Ascend910-2,As
cend910-3
                    kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
```

According to the command output, the Ascend Device Plugin startup mode was incorrect, and Kubernetes run slowly when the number of jobs was large.

 Copyright © Huawei Technologies Co., Ltd.

Solution

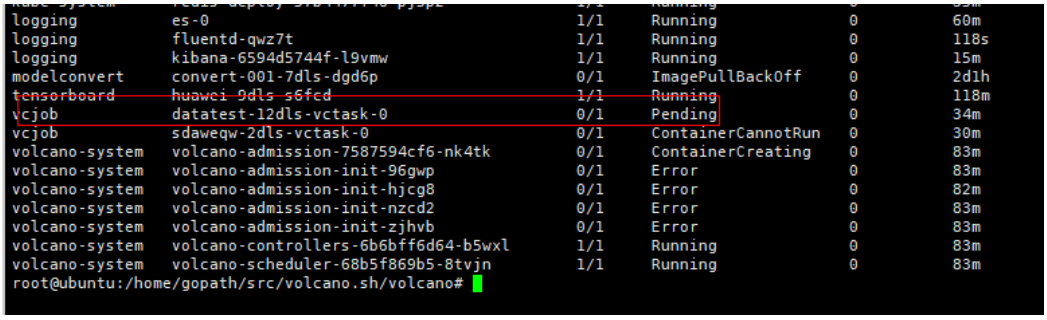
Reinstall Ascend Device Plugin. For details, see **MindX DL Installation**.

5.3 A Job Was Pending Due to Insufficient Volcano Resources

Symptom

When the resources requested by a job met the model requirements but the actual resources did not meet the requirements, the job could not be scheduled and was in the pending state. However, the waiting was not terminated in a timely manner or timed out. The job keeps waiting in some cases if the resource requirement cannot be met.

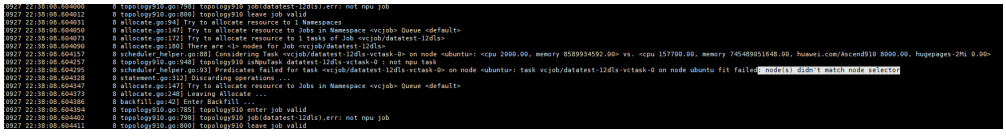
Figure 5-3 A job was in the **Pending** state.



Possible Causes

- When resources were insufficient, volcano-scheduler did not terminate job scheduling.
- A job could not continue due to the lack of a label, but the job was not terminated yet. **Figure 5-4** shows the volcano-schedule log.

Figure 5-4 Job pending due to the lack of a node selector



Solution

Step 1 Ensure that resources are sufficient before using.

Step 2 Ensure that the request body (or YAML file) and node label of the job contain the corresponding labeling commands. For details about the labeling commands, see **Creating a Node Label**.

You can delete the current vcjob using the following method to resolve the problem:

 Copyright © Huawei Technologies Co., Ltd.

If a vcjob is in the **Pending** state due to insufficient resources, run the **kubectl delete vcjob** *job-zdd-001-7dls* **-n** *vcjob* command to delete the vcjob.

```
modelconvert     convert-001-7dls-dgd6p                    0/1     ImagePullBackOff   0
vcjob            job-zdd-001-7dls-vctask-0                 0/1     Completed          0
volcano-system   volcano-admission-7587594cf6-n8sgn        1/1     Running            0
volcano-system   volcano-controllers-6b6bff6d64-dchpj      1/1     Running            0
volcano-system   volcano-scheduler-68b5f869b5-5cr47        1/1     Running            0
root@ubuntu:/home/btg/yaml#
root@ubuntu:/home/btg/yaml#
root@ubuntu:/home/btg/yaml# kubectl delete vcjob job-zdd-001-7dls -n vcjob
job.batch.volcano.sh "job-zdd-001-7dls" deleted
root@ubuntu:/home/btg/yaml#
```

NOTE

- *job-zdd-001-7dls*: name of the vcjob.
- *vcjob*: namespace to which the vcjob belongs.

----End

5.4 Failed to Generate the hcl.json File

Symptom

After a training job is started, the **hcl.json** file in the training job container is in the initializing state. The default file path is **/user/serverid/devindex/config/hccl.json**.

Run the **kubectl exec -it** *XXX* **bash** command to access the container. If the pod is not in the default namespace, add **-n** *XXX* to specify the namespace, for example, **kubectl exec -it** *XXX* **-n** *XXX* **bash**.

```
root@ubuntu:/home/edmund/yamls# kubectl exec -it    resnet1-1-default-test-0  bash
dls-user@resnet1-1-default-test-0:/tmp$ cat /user/serverid/devindex/config/hccl.json
{
    "status":"initializing"
}
dls-user@resnet1-1-default-test-0:/tmp$
```

Possible Causes

- Cause 1: HCCL-Controller is not started properly.
- Cause 2: The HCCL-Controller version does not match the Ascend Device Plugin version.
- Cause 3: Ascend Device Plugin does not correctly generate the annotation of the pod. To view the annotation, run the **kubectl describe** *XXX* **-n** *XXX* command.

  In normal cases, the command output contains **ascend.kubectl.kubernetes.io/ascend-910-configuration** or **atlas.kubectl.kubernetes.io/ascend-910-configuration** (20.1.0 and earlier versions).

 Copyright © Huawei Technologies Co., Ltd.

```
root@ubuntu-65-153:/home# kubectl describe pod mnist-2dls-vctask-0  -n vcjob
Name:         mnist-2dls-vctask-0
Namespace:    vcjob
Priority:     0
Node:         ubuntu-65-153/51.[redacted]
Start Time:   Thu, 10 Dec 2020 16:54:40 +0800
Labels:       app=mnist-2dls
              ring-controller.atlas=ascend-910
              volcano.sh/job-name=mnist-2dls
              volcano.sh/job-namespace=vcjob
Annotations:  ascend.kubectl.kubernetes.io/ascend-910-configuration:
                {"pod_name":"0","server_id":"51.[redacted]","devices":[{"device_id":"0","device_ip":"192.168.10.100"}]}
              cni.projectcalico.org/podIP: 192.168.5.146/32
              cni.projectcalico.org/podIPs: 192.168.5.146/32
              huawei.com/Ascend910: Ascend910-0
              predicate-time: 18446744073709551615
              scheduling.k8s.io/group-name: mnist-2dls
              volcano.sh/job-name: mnist-2dls
              volcano.sh/job-version: 0
              volcano.sh/task-spec: vctask
Status:       Running
```

Solution

For cause 1: Reinstall HCCL-Controller by referring to the **installation and deployment guide**.

For cause 2: Reinstall HCCL-Controller and Ascend Device Plugin by referring to **Table 1-2**.

For cause 3: The corresponding annotation is not found. The possible cause is that Ascend Device Plugin does not obtain the correct device IP address. Ensure that the device IP address is correctly configured after the driver is installed. For details, see " Development Environment Installation (Training) > Changing NPU IP Addresses" in the ***CANN Software Installation Guide***.

5.5 Calico Network Plugin Not Ready

Symptom

In the output of the **kubectl get pod -A** command for checking the calico network plugin, the value in the **READY** column is **0/1**.

```
[root@centos-187 package]# kubectl get pod -A
NAMESPACE       NAME                                       READY   STATUS
cadvisor        cadvisor-gcldn                             1/1     Running
default         dls-cec-deploy-589f8ffdcb-zxc25            1/1     Running
default         dls-mms-deploy-84d4bf87c5-cf8d7            1/1     Running
default         hccl-controller-648c468d57-zw8rb           1/1     Running
default         tjm-74c747dd5c-sdv45                       1/1     Running
kube-system     ascend-device-plugin2-daemonset-9pslp      1/1     Running
kube-system     calico-kube-controllers-8464785d6b-txm4l   [redacted]
kube-system     calico-node-hvpjt                          0/1     Running
```

Possible Causes

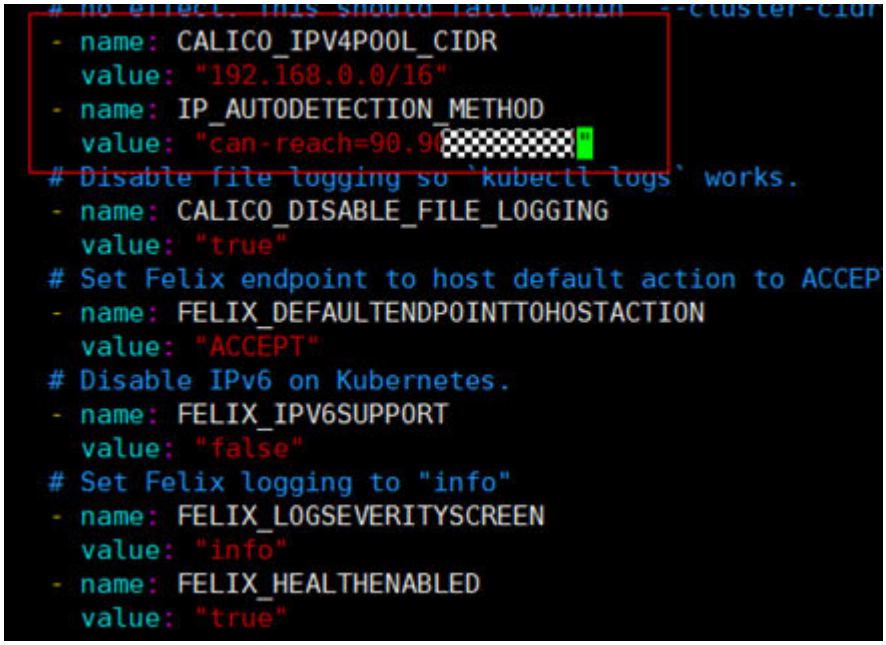
If the network segment of a physical machine conflicts with that of the configured container or the physical machine is in a complex network environment, Calico cannot correctly identify valid NICs of the managment and compute nodes.

 Copyright © Huawei Technologies Co., Ltd.

Solution

Check whether the network segment of the physical machine is the same as that of the container. If yes, initialize the Kubernetes cluster again and change the value of **pod-network-cidr** to a network segment that does not conflict with the container network segment. After the initialization, change the value of calico accordingly.

Modify the default container network segment parameter **CALICO_IPV4POOL_CIDR** in the YAML file for starting Calico. In addition, you are advised to add the **IP_AUTODETECTION_METHOD** configuration. The value is **can-reach={*masterIP*}**, where *masterIP* indicates the IP address of the physical machine on the Kubernetes management node. The following figure shows the content that needs to be modified in the YAML file for starting Calico. For details about how to reset and install the Kubernetes, see the **official Kubernetes website**.



5.6 Error Message "admission webhook "validatejob.volcano.sh" denied the request" Is Displayed When a YAML Job Is Running

Symptom

When **kubectl apply -f** *XXX*.**yaml** is used to start a job, the following error message is displayed:

```
Error from server: error when creating "resnet50_hccl_1-acc-pytorch.yaml": admission webhook "validatejob.volcano.sh" denied the request: spec.task[0].template.spec.volumes[5].hostPath.path: Required value. template.spec.containers[0].volumeMounts[5].name: Not found: "ascend-add-ons".
```

 Copyright © Huawei Technologies Co., Ltd.

Possible Causes

An error occurs in the YAML file during job startup. In this example, the mount path of **ascend-add-ons** is misspelled.

An example is provided as follows.

```
root@ubuntu-60-19:/data/hrp#
root@ubuntu-60-19:/data/hrp# grep -C 5 ascend-add-ons resnet50_hccl_1-acc-pytorch.yaml
            mountPath: /job/data
          - name: output
            mountPath: /job/output
          - name: ascend-dirver
            mountPath: /usr/local/Ascend/driver
          - name: ascend-add-ons
            mountPath: /usr/local/Ascend/add-ons
          - mountPath: /dev/shm
            name: dshm
        nodeSelector:
          host-arch: huawei-x86
--
            server: [redacted]
            path: "/data/atlas_dls/public/output"
```

The mount path is misspelled as **ipath**. As a result, YAML fails to run the job.

```
      - name: ascend-dirver
        hostPath:
          path: /usr/local/Ascend/driver
      - name: ascend-add-ons
        hostPath:
          ipath: /usr/local/Ascend/add-ons
      - name: dshm
        emptyDir:
          medium: Memory
    env:
      - name: hrp-pytorch
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
    restartPolicy: OnFailure
```

Solution

Check the YAML file and rectify any incorrect fields.

5.7 Kubernetes Fails to Be Restarted After the Server Is Restarted

Symptom

After the server is restarted, the Kubernetes fails to be restarted. After the **kubectl get pod** command is run, the following error information is displayed:

```
The connection to the server xxx.xxx.xxx.xxx:6443 was refused - did you specify the right host or port?
```

Run the **free -m** command to check whether the swap is disabled. The following information is displayed:

```
              total        used        free      shared  buff/cache   available
Mem:         773737        5373      766172           4        2192      765453
Swap:        38146            0       38146
```

 Copyright © Huawei Technologies Co., Ltd.

Possible Causes

The swap is not disabled.

Solution

Step 1 Run the following command to disable the swap partition:

swapoff -a

Step 2 After the Kubernetes is started, run the following command:

kubectl get pod

If information similar to the following is displayed, the process is normal:

```
NAME                                READY    STATUS    RESTARTS    AGE
hccl-controller-767f45c6b5-srkr4    1/1      Running   10          46h
tjm-6ff5f74865-wh6nm                1/1      Running   0           5s
```

NOTE

To make the configuration take effect permanently, perform the following operations:

Step 3 Run the following commands to create a .sh file:

mkdir -p /usr/local/scripts/

vim /usr/local/scripts/dls_swap_check.sh

Add the following content to the file:

```
#!/bin/bash

function check_swap() {
    swap_total=1
    sleep 10
    while [ "$swap_total" != "0" ]; do
        swap_total=$(free -m | grep -i swap | awk  '{ print $2 }')
        echo "The swap total: $swap_total." > /dev/kmsg
        swapoff -a
        sleep 5
    done
}

check_swap &
echo 0
```

Step 4 Run the following command to change the .sh file permission:

chmod 750 /usr/local/scripts/dls_swap_check.sh

Step 5 Run the following command to edit the **rc.local** file:

vi /etc/rc.local

Add the following information before **exit 0**:

```
/usr/local/scripts/dls_swap_check.sh
exit 0
```

----End

 Copyright © Huawei Technologies Co., Ltd.

5.8 Message "certificate signed by unknown authority" Is Displayed When a kubectl Command Is Run

Symptom

When a kubectl command, for example, **kubectl get pods --all-namespaces**, is run, the following error message is displayed:

```
Unable to connect to the server: x509: certificate signed by unknown authority
```

Possible Causes

A proxy has been configured in the environment.

Solution

Run the following command to cancel the proxy:

unset http_proxy https_proxy

 Copyright © Huawei Technologies Co., Ltd.

6 Communication Matrix

Table 6-1 Communication Matrix

Source Device	Other nodes in the Kubernetes cluster.
Source IP Address	IP addresses of other nodes in the Kubernetes cluster.
Source Port	Uncertain
Destination Device	Kubernetes cluster Worker node.
Destination IP Address	IP address of the Kubernetes cluster Worker node.
Destination Port (Listening)	8081
Protocol	TCP
Port Description	Used by management nodes in the Kubernetes cluster to query cluster Worker node real-time monitoring and performance data collection, including CPU usage, memory usage, network throughput, and file system usage. This port is enabled by default. **NOTE** This is a reference design. This port is available only after you compile and install it in the system during secondary development.
Listening Port Configurable	Mandatory
Authentication Mode	N/A
Encryption Mode	N/A
Plane	Management plane
Version	All versions

 Copyright © Huawei Technologies Co., Ltd.

Special Scenario	N/A
Remarks	cAdvisor process, which is used for communication between nodes on the container network.

 Copyright © Huawei Technologies Co., Ltd.

A Change History

Date	Description
2021-03-22	• Added the description of the **hwMindX** user. • Optimized MindSpore image creation.
2021-01-25	This issue is the first official release.

 Copyright © Huawei Technologies Co., Ltd.