ModelArts

Lite Cluster User Guide

Issue	01
Date	2024-12-31

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Huawei Cloud Computing Technologies Co., Ltd.

Address:	Huawei Cloud Data Center Jiaoxinggong Road Qianzhong Avenue Gui'an New District Gui Zhou 550029 People's Republic of China
Website:	https://www.huaweicloud.com/intl/en-us/

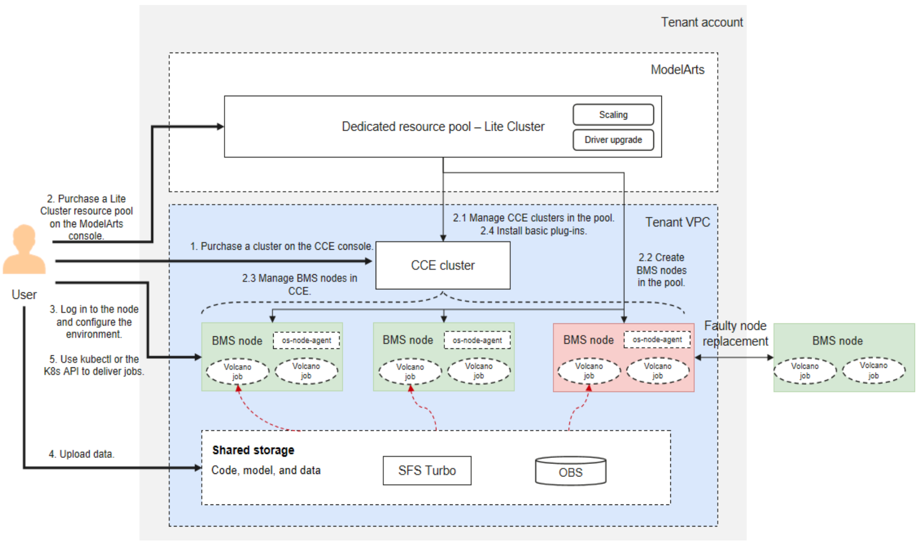
Contents

1 Before You Start

1.1 Usage Process

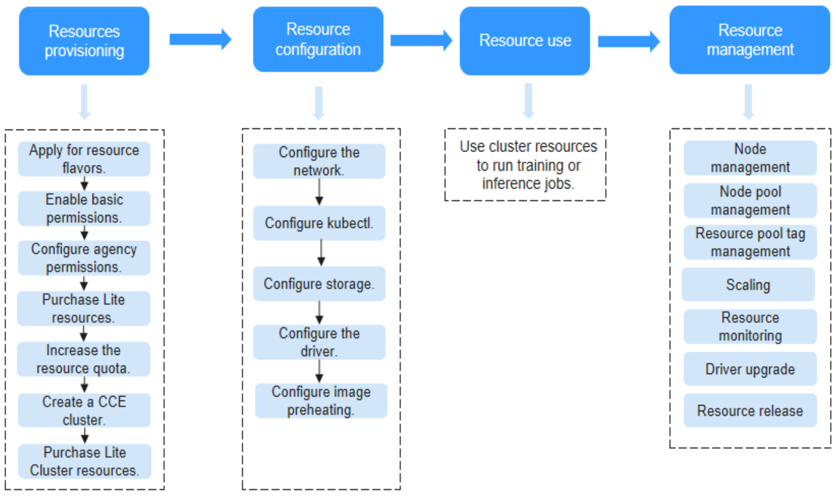
ModelArts Lite Cluster offers hosted Kubernetes clusters with pre-installed AI development and acceleration plug-ins. These elastic clusters allow you to access AI resources and tasks in a cloud-native environment. You can directly manage nodes and Kubernetes clusters within the resource pools. This document shows how to get started.

Figure 1-1 Resource pool architecture


Tenant account
ModelArts
Dedicated resource pool – Lite Cluster
Scaling
Driver upgrade
2. Purchase a Lite Cluster resource pool on the ModelArts console.
2.1 Manage CCE clusters in the pool.
2.4 Install basic plug-ins.
Tenant VPC
1. Purchase a cluster on the CCE console.
CCE cluster
2.3 Manage BMS nodes in CCE.
2.2 Create BMS nodes in the pool.
User
3. Log in to the node and configure the environment.
5. Use kubectl or the K8s API to deliver jobs.
BMS node
os-node-agent
Volcano job
Volcano job
BMS node
os-node-agent
Volcano job
Volcano job
BMS node
os-node-agent
Volcano job
Volcano job
Faulty node replacement
BMS node
Volcano job
Volcano job
4. Upload data.
Shared storage
Code, model, and data
SFS Turbo
OBS

This figure shows Lite Cluster architecture. To use Lite Cluster, start by purchasing a CCE cluster. Lite Cluster then manages resource nodes within this CCE cluster. After you purchase a Lite cluster on the ModelArts console, ModelArts manages the CCE cluster within a resource pool and creates compute nodes (BMSs/ECSs) based on the specifications you set. These nodes are then managed by CCE, and ModelArts installs necessary plug-ins (such as npuDriver and os-node-agent) in the CCE cluster. Once you have acquired a Lite Cluster resource pool, you can configure resources and upload data to the cloud storage service. When you require cluster resources, you can use the kubectl tool or Kubernetes APIs to submit jobs. Additionally, ModelArts offers scaling and driver upgrade to streamline cluster resource management.

Figure 1-2 Usage process



To use Lite Cluster, follow these steps:

1. Resource subscription: Apply for the required specifications, configure permissions, and purchase Lite Cluster resources on the ModelArts console. For details, see **Enabling Lite Cluster Resources**.
2. Resource configuration: After acquiring resources, set up network, storage, and drivers. For details, see **Configuring Lite Cluster Resources**.
3. Resource usage: Once configured, use cluster resources for training and inference. For details, see **Using Lite Cluster Resources**.
4. Resource management: Lite Cluster provides scaling and driver upgrades. You can manage resources on the ModelArts console. For details, see **Managing Lite Server Resources**.

Table 1-1 Terms

Term	Description
Container	Containers, rooted in Linux, are lightweight virtualization technologies that isolate processes and resources. Docker popularized containers by making them portable across different machines. It simplifies the packaging of both applications and the applications' repository and dependencies. Even an OS file system can be packaged into a simple portable package that can be used on any other machine that runs Docker.
Kubernetes	Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. To use Lite Cluster, familiarity with Kubernetes is essential. For details, see **Kubernetes Basics**.

Term	Description
CCE	Cloud Container Engine (CCE) is a Kubernetes cluster hosting service for enterprises. It manages containerized applications and offers scalable, high-performance solutions for deploying and managing cloud native applications. For details, see **What Is CCE?**.
BMS	Combining VM scalability with physical server performance, BMS provides dedicated cloud servers. These servers are designed to meet the demands of computing performance and data security for core databases, critical applications, high-performance computing (HPC), and big data.
ECS	Elastic Cloud Server (ECS) provides scalable, on-demand cloud servers for secure, flexible, and efficient application environments, ensuring reliable, uninterrupted services.
os-node-agent	The os-node-agent plug-in is installed by default on ModelArts Lite Kubernetes cluster nodes, allowing for node management. For example: ● Driver upgrades: The plug-in downloads and updates or rolls back driver versions. ● Fault detection: It periodically checks for node faults. ● Metric collection: The plug-in gathers key monitoring data, such as GPU and NPU usage, and sends it to AOM on the tenant side. ● Node O&M: After authorization, the plug-in runs diagnosis scripts for fault identification and demarcation.

1.2 High-Risk Operations

When you perform operations on ModelArts Lite Cluster resources on the CCE, ECS, or BMS console, certain resource pool functions may be abnormal. The table below shows common risky operations.

Risky operations fall into three levels:

- High: Such operations may cause service failures, data loss, system maintenance failures, and system resource exhaustion.
- Medium: Such operations may cause security risks and reduce service reliability.
- Low: Such operations include high-risk operations other than those of a high or medium risk level.

Table 1-2 Operations and risks

Object	Operation	Risk	Severity	Solution
Cluster	Upgrade, modify, hibernate, or delete clusters.	These operations may impact basic ModelArts functions, including resource pool management, node management, scaling, and driver upgrades	High	These operations cannot be undone.
Node	Unsubscribe, remove, shut down, manage taints, or switch or reinstall OS.	These operations may impact basic ModelArts functions, including node management, scaling, driver upgrades, and data loss of local disks.	High	These operations cannot be undone.
	Modify a network security group.	These operations may impact basic ModelArts functions, including node management, scaling, and driver upgrades	Medium	If needed, revert back to the original data.
Network	Modify or delete the CIDR block associated with a cluster.	These operations impact basic ModelArts functions, including node management, scaling, and driver upgrades	High	These operations cannot be undone.
Plug-in	Upgrade or uninstall the gpu-beta plug-in.	The GPU driver may be abnormal.	Medium	Roll back the version and reinstall the plug-in.
	Upgrade or uninstall the huawei-npu plug-in.	The NPU driver may be abnormal.	Medium	Roll back the version and reinstall the plug-in.
	Upgrade or uninstall the volcano plug-in.	Job scheduling may be abnormal.	Medium	Roll back the version and reinstall the plug-in.
	Uninstall the ICAgent plug-in.	Logging and monitoring may be abnormal.	Medium	Roll back the version and reinstall the plug-in.

Object	Operation	Risk	Severity	Solution
helm	Upgrade, roll back, or uninstall os-node-agent.	Driver upgrades, fault detection, metric collection, and node O&M are abnormal.	High	Contact Huawei Cloud technical support to reinstall os-node-agent.
	Upgrade, roll back, or uninstall rdma-sriov-dev-plugin.	The use of RDMA NICs in containers may be affected.	High	Contact Huawei Cloud technical support to reinstall rdma-sriov-dev-plugin.

1.3 Software Versions Required by Different Models

A resource pool for elastic clusters can use either Elastic Bare Metal Servers (BMSs) or Elastic Cloud Servers (ECSs) as nodes. Each node model has its own operating system (OS) and compatible CCE cluster versions. This document outlines the necessary software versions for each model to simplify image creation and software upgrades.

Software Versions Required by BMSs

Table 1-3 BMS

Type	Card Type	RDMA Network Protocol	OS	Applicable Scope	Dependent Plug-in
NPU	ascend-snt9b	RoCE	• OS: EulerOS 2.10 64-bit (recommended) • Kernel version: 4.19.90-vhulk2211.3.0.h1543.eulerosv2r10.aarch64 • Architecture type: aarch64	• Cluster type: CCE Standard • Cluster version: v1.23 (v1.23.5-r0 or later) or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC • Cluster forwarding mode: iptables or ipvs	• huawei-npu • npu-driver • volcano For details about the plug-in version mapping, see **Table 1-5**.
		RoCE	• OS: Huawei Cloud EulerOS 2.0 64-bit • Kernel version: 5.10.0-60.18.0.50.r865_35.hce2.aarch64 • Architecture type: aarch64	• Cluster type: CCE Turbo • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: ENI • Cluster forwarding mode: iptables or ipvs	

Type	Card Type	RDMA Network Protocol	OS	Applicable Scope	Dependent Plug-in
	ascend-snt9	RoCE	• OS: EulerOS 2.8 64-bit • Kernel version: 4.19.36-vhulk1907.1.0.h619.eulerosv2r8.aarch64 • Architecture type: aarch64	• Cluster type: CCE Standard or Turbo • Cluster version: v1.23 (v1.23.5-r0 or later) and v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network, VPC, or ENI • Cluster forwarding mode: iptables or ipvs	

Type	Card Type	RDMA Network Protocol	OS	Applicable Scope	Dependent Plug-in
GPU	gp-ant8	RoCE	• OS: EulerOS 2.10 64-bit • Kernel version: 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC Distributed training only supports container tunnel network. • Cluster forwarding mode: iptables or ipvs	• gpu-beta • gpu-driver • rdma-sriov-dev-plugin For details about the plug-in version mapping, see **Table 1-5**.

Type	Card Type	RDMA Network Protocol	OS	Applicable Scope	Dependent Plug-in
	gp-ant1	RoCE	• OS: EulerOS 2.10 64-bit • 4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC Distributed training only supports container tunnel network. • Cluster forwarding mode: iptables or ipvs	

Type	Card Type	RDMA Network Protocol	OS	Applicable Scope	Dependent Plug-in
	gp-vnt1	RoCE|IB	• OS: EulerOS 2.9 64-bit (used only for p6 and p6s flavors in Shanghai 1) • Kernel version: 147.5.1.6.h1099.eulerosv2r9.x86_64 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC Distributed training only supports container tunnel network. • Cluster forwarding mode: iptables or ipvs	
			• OS: EulerOS 2.9 64-bit (recommended) • Kernel version: 4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64 • Architecture type: x86		
• Remote direct memory access (RDMA) is a direct memory access from the memory of one computer into that of another without involving either one's operating system. • RDMA over Converged Ethernet (RoCE) is a network protocol which allows RDMA over an Ethernet network. • InfiniBand (IB) is a computer networking communications standard used in high-performance computing. It is used for data interconnect both among and within computers.					

Software Versions Required by ECSs

Table 1-4 ECS

Type	Card Type	OS	Applicable Scope	Dependent Plug-in
NPU	ascend-snt3p-300i	• OS: EulerOS 2.9 • Architecture type: x86	• Cluster type: CCE Standard or Turbo • Cluster version: v1.23 (v1.23.5-r0 or later) and v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network, VPC, or ENI • Cluster forwarding mode: iptables or ipvs	• huawei-npu • npu-driver • volcano For details about the plug-in version mapping, see **Table 1-5**.
	ascend-snt3	• OS: EulerOS 2.5 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC • Cluster forwarding mode: iptables or ipvs	

Type	Card Type	OS	Applicable Scope	Dependent Plug-in
		• OS: EulerOS 2.8 • Architecture type: Arm	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC Cluster forwarding mode: iptables or ipvs	
GPU	gp-vnt1	• OS: EulerOS 2.9 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC • Cluster forwarding mode: iptables or ipvs	• gpu-beta • gpu-driver • rdma-sriov-dev-plugin For details about the plug-in version mapping, see **Table 1-5**.
	gp-ant03	• OS: EulerOS 2.9 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC • Cluster forwarding mode: iptables or ipvs	

Type	Card Type	OS	Applicable Scope	Dependent Plug-in
	gp-ant1-pcie40	• OS: EulerOS 2.9 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC • Cluster forwarding mode: iptables or ipvs	
	gp-tnt004	• OS: EulerOS 2.9 • Architecture type: x86	• Cluster type: CCE Standard • Cluster version: v1.23 or v1.25 (recommended) • Cluster scale: 50, 200, 1000, or 2000 • Cluster network mode: container tunnel network or VPC • Cluster forwarding mode: iptables or ipvs	

Mapping Between Plug-in Versions and CCE Cluster Versions

Table 1-5 Mapping between plug-in versions and CCE cluster versions

Type	Plug-in	Plug-in Version	Matched CCE Cluster Version	Applicable Scope	Plug-in Function
ccePlugin	gpu-beta	2.0.48 (recommended)	v1.(23|25).*	GPU	Allows containers to use GPU devices.
		1.2.15	v1.23.*		

Type	Plug-in	Plug-in Version	Matched CCE Cluster Version	Applicable Scope	Plug-in Function
	huawei-npu	2.1.5 (recommended)	v1.(23|25).*	NPU	Allows containers to use Huawei NPU devices.
	volcano	1.11.9 (recommended)	v1.(23|25).*	NPU	Kubernetes-based batch processing platform.
npuDriver	npu-driver	7.1.0.7.220-23.0.5 (recommended) 7.1.0.5.220-23.0.3	None	NPU	Upgrades and rolls back NPU drivers.
helm	rdma-sriov-dev-plugin	0.1.0	None	Used for BMS and RDMA (non-ascend-1980)	Allows containers to use RDMA NICs.
	memarts	3.23.6-r002	None	None	Near-compute distributed cache plug-in, which is used for storage acceleration.
	os-node-agent	6.5.0-20240529142433	None	None	OS plug-in, which is used for fault detection.

Type	Plug-in	Plug-in Version	Matched CCE Cluster Version	Applicable Scope	Plug-in Function
icAgent	icagent	default	The matched CCE version is installed by default.	None	CCE basic component, which is used for logging and monitoring.
gpuDriver	gpu-driver	515.65.01 (recommended) 510.47.03 470.182.03 470.57.02 gpu-driver is related to the system kernel version. For details, see **Table 1-6**.			Upgrades and rolls back GPU drivers. The plug-in depends on the gpu-beta version.

Mapping Between the Kernel and gpu-driver

Table 1-6 Mapping between the kernel and gpu-driver

Image Tag	Kernel Version	Matched CCE	gpu-driver Version
EulerOS 2.10	4.18.0-147.5.2.15.h1109.eulerosv2r10.x86_64	v1.(23|25|27|28).* Container tunnel network, VPC, or ENI	470.57.02
	4.18.0-147.5.2.5.h805.eulerosv2r10.x86_64	v1.(23|25|27).* Container tunnel network, VPC, or ENI	470.57.02
EulerOS 2.9	4.18.0-147.5.1.6.h841.eulerosv2r9.x86_64	v1.(23|25|27|28).* Container tunnel network or VPC	470.57.02
EulerOS 2.3	3.10.0-514.44.5.10.h193.x86_64	v1.(23|25).* Container tunnel network or VPC	470.57.02
	3.10.0-514.44.5.10.h254.x86_64	v1.(23|25).* Container tunnel network or VPC	470.57.02

2 Enabling Lite Cluster Resources

Process

The following figure shows the process of enabling cluster resources.

Figure 2-1 Process

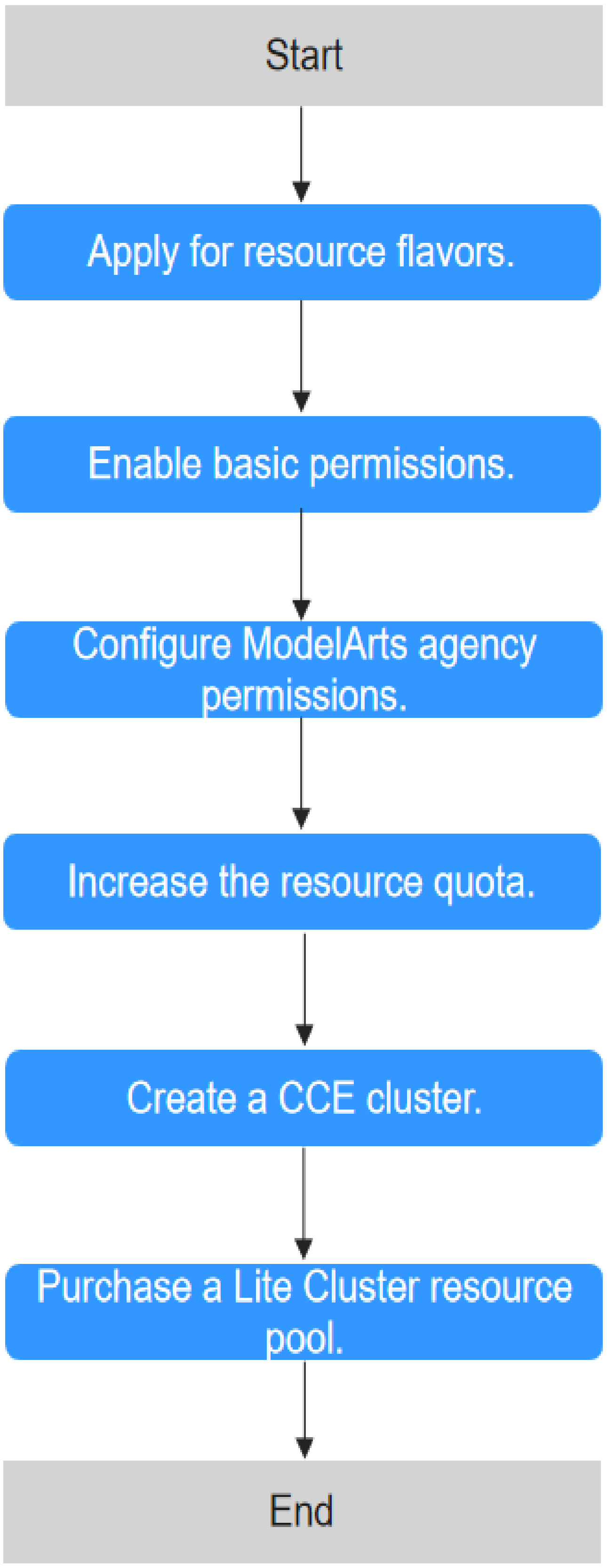

Start
Apply for resource flavors.
Enable basic permissions.
Configure ModelArts agency permissions.
Increase the resource quota.
Create a CCE cluster.
Purchase a Lite Cluster resource pool.
End

Table 2-1 Enabling cluster resources

Step	Description
Step1 Enabling Resource Specifications	Contact your account manager to request resource specifications in advance. They will enable the specifications within one to three working days. If there is no account manager, submit a service ticket.
Step 2: Enabling Basic Permissions	Assign the necessary permissions to the target IAM user to use resource pools.
Step 3 Creating an Agency in ModelArts	Create an agency in ModelArts to authorize access to other cloud services. If you already have an agency, update its permissions.
Step 4 Applying for a Higher Resource Quota	To run clusters, you will need more resources than Huawei Cloud's default quotas provided. This includes more ECS instances, memory, CPU cores, and EVS disk space. You will need to request a higher quota to meet these needs. Contact your customer manager for information on the quota solution. Increase the quota before purchasing and provisioning the resource, ensuring it exceeds the resource's requirements.
Step 5 Buying a CCE Cluster	When buying a Lite Cluster resource pool, choose a CCE cluster. If none is available, create one on the CCE console beforehand.
Step 6 Buying Lite Cluster Resources	Purchase Lite Cluster resources on the ModelArts console.

Step1 Enabling Resource Specifications

Contact your account manager to request restricted specifications (such as modelarts.bm.npu.arm.8snt9b3.d) in advance. They will enable the specifications within one to three working days. If there is no account manager, submit a service ticket.

Step 2: Enabling Basic Permissions

Log in to the administrator account and grant the target IAM account basic permissions to use resource pools.

Step 1 Log in to the IAM console.

Step 2 In the navigation pane, choose **User Groups** and click **Create User Group** in the upper right corner.

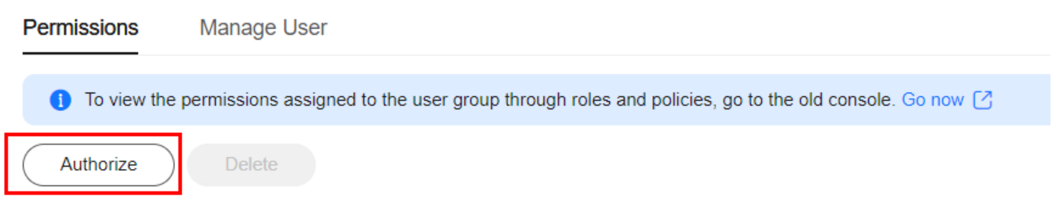
Step 3 Enter a group name and click **OK**.

Step 4 Click **Manage User** in the **Operation** column and add the users for which you want to assign permissions to the user group.

Step 5 Click the name of the user group to go to the group details page.

Step 6 In the **Permissions** tab, click **Authorize**.

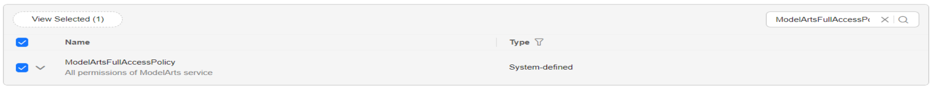
Figure 2-2 Assigning permissions



To view the permissions assigned to the user group through roles and policies, go to the old console. Go now

Step 7 Search for **ModelArts FullAccess** in the search box and select it.

Figure 2-3 ModelArts FullAccess



Name	Type
ModelArtsFullAccessPolicy All permissions of ModelArts service	System-defined

Repeat this step to select the following permissions:

- ModelArts FullAccess
- CTS Administrator
- CCE Administrator
- BMS FullAccess
- IMS FullAccess
- DEW KeypairReadOnlyAccess
- VPC FullAccess
- ECS FullAccess
- SFS Turbo FullAccess
- OBS Administrator
- AOM FullAccess
- TMS FullAccess
- BSS Administrator

Step 8 Click **Next** and set **Scope** to **All resources**.

Step 9 Click **OK**.

----End

Step 3 Creating an Agency in ModelArts

- Creating an agency

  Create an agency in ModelArts to authorize access to other cloud services.

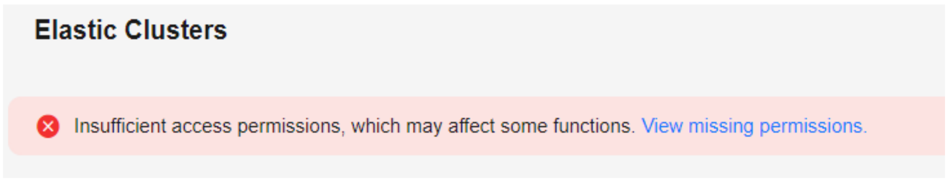
  To do so, log in to the ModelArts console. In the navigation pane on the left, choose **Permission Management**. On the displayed page, click **Add Authorization**.

- Updating an agency

  Update the permissions for your existing ModelArts agency.

  a. Log in to the ModelArts console. In the navigation pane on the left, choose **Resource Management** > **AI Dedicated Resource Pools** > **Elastic Clusters**. On the displayed page, check whether a message is reported, indicating that the authorization is insufficient.

  Figure 2-4 Insufficient permission on elastic clusters

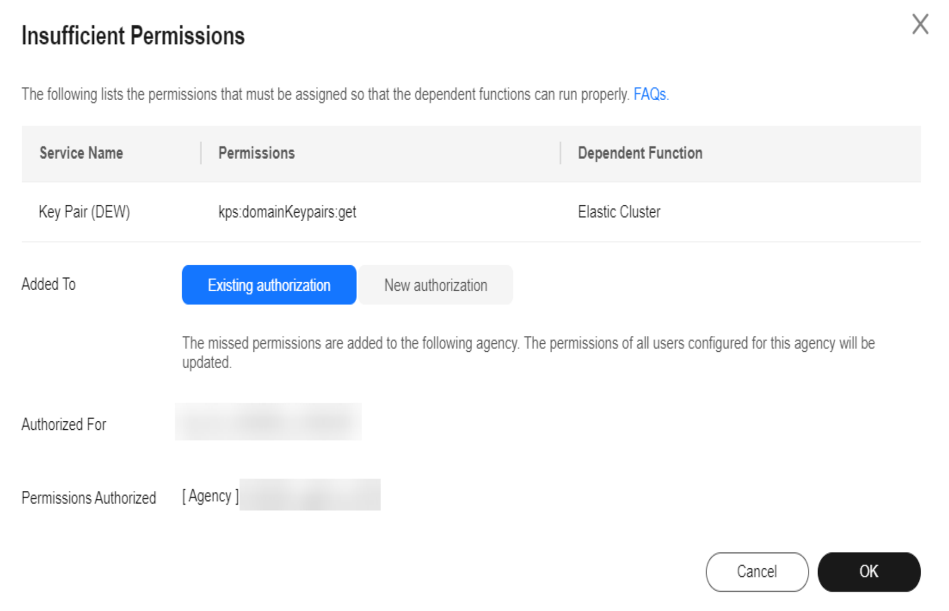
  

  Elastic Clusters

  Insufficient access permissions, which may affect some functions. View missing permissions.

  b. Click **Authorize access** to update the agency if needed. Select **Append to Existing Entitlement** and click **OK**. The system shows the permission update is successful.

  Figure 2-5 Adding authorization

  Insufficient Permissions

  The following lists the permissions that must be assigned so that the dependent functions can run properly. FAQs.

Service Name	Permissions	Dependent Function
Key Pair (DEW)	kps:domainKeypairs:get	Elastic Cluster

  Added To: Existing authorization | New authorization

  The missed permissions are added to the following agency. The permissions of all users configured for this agency will be updated.

  Authorized For

  Permissions Authorized: [Agency]

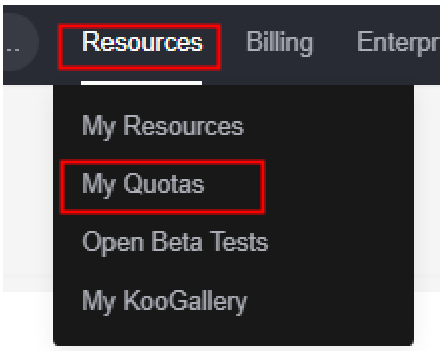
Step 4 Applying for a Higher Resource Quota

To run AI workloads in resource pools, you will need more resources than Huawei Cloud's default quotas provided. This includes more ECS instances, memory, CPU cores, and EVS disk space. To access these extra resources, request a higher quota. Confirm the solution with the customer manager, then apply for a higher resource quota by following these steps.

Step 1 Log in to Huawei Cloud console.

Step 2 Hover over **Resources** from the top navigation bar and choose **My Quotas**.

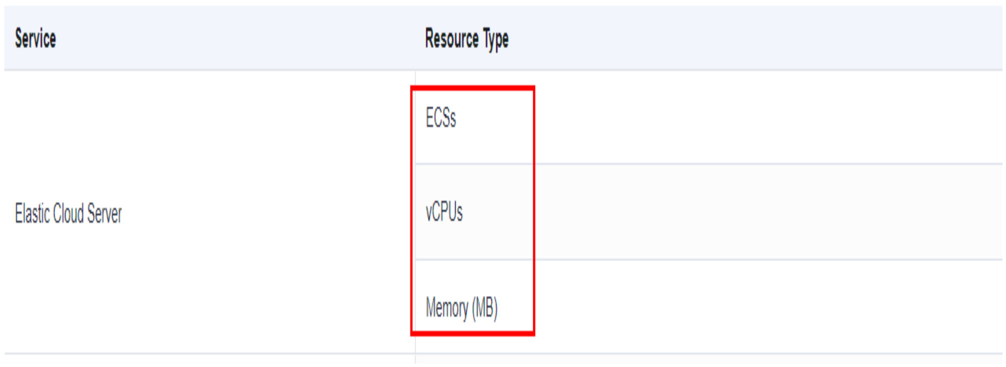
Figure 2-6 My Quotas



Step 3 On the **Quotas** page, click **Increase Quota** in the upper right corner and submit a service ticket.

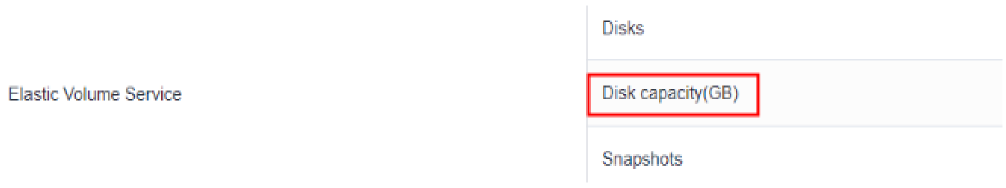
Request the required number of ECS instances, CPU cores, RAM capacity (memory size), and EVS disk capacity. Contact your customer manager for quota details.

Figure 2-7 ECS resource type



Service	Resource Type
Elastic Cloud Server	ECSs
	vCPUs
	Memory (MB)

Figure 2-8 EVS resource type



Elastic Volume Service	Disks
	Disk capacity(GB)
	Snapshots

NOTE

Increase the quota before purchasing and provisioning the resource, ensuring it exceeds the resource's requirements.

----End

Step 5 Buying a CCE Cluster

When buying a Lite Cluster resource pool, choose a CCE cluster. If none is available, follow the instructions in **Buying a CCE Standard/Turbo Cluster** to acquire one. For details about the required cluster version, see **Software Versions Required by Different Models**.

Create a Lite Cluster resource pool only when the CCE cluster is running.

NOTE

- CCE clusters of versions 1.23, 1.25, and 1.28 are supported.
- If no CCE cluster is available, create one. Create CCE clusters of version 1.28 using either the console or APIs. Create CCE clusters of versions 1.23 and 1.25 using APIs only. For details about how to create CCE clusters of different versions, see **Kubernetes Version Policy**.
- Upgrade your CCE cluster to version 1.28 if it is running an earlier version, such as 1.23 or lower. For details, see **Process and Method of Upgrading a Cluster**.

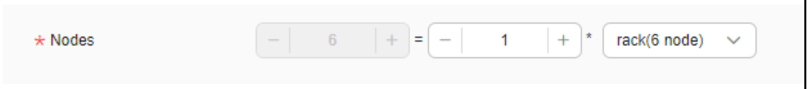
Step 6 Buying Lite Cluster Resources

1. Log in to the ModelArts console. From the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**.
2. On the **Elastic Clusters** page, click **Buy Dedicated AI Cluster**.

Table 2-2 Parameters

Parameter	Sub-Parameter	Description
Name	N/A	Enter a name. Only lowercase letters, digits, and hyphens (-) are allowed. The value must start with a lowercase letter and cannot end with a hyphen (-).
Description	N/A	Enter a brief description of the dedicated resource pool.
Product Version	N/A	Select **ModelArts Lite**.
Billing Mode	N/A	Select **Pay-per-use** or **Yearly/Monthly**. ● Yearly/Monthly Yearly/Monthly is a prepaid billing mode in which your subscription is billed based on the required duration. This mode is more cost-effective when the usage duration is predictable. ● Pay-per-use Pay-per-use is a postpaid billing mode in which your resources are billed based on usage duration. You can create or delete your resources at any time.

Parameter	Sub-Parameter	Description
CCE cluster	N/A	Choose an existing CCE cluster from the drop-down list. Click **Create Cluster** on the right to create a cluster if none is available. For details about the required cluster version, see **Software Versions Required by Different Models**. Create a Lite Cluster resource pool only when the CCE cluster is running. **NOTE** ● CCE clusters of versions 1.23, 1.25, and 1.28 are supported. ● If no CCE cluster is available, create one. Create CCE clusters of version 1.28 using either the console or APIs. Create CCE clusters of versions 1.23 and 1.25 using APIs only. For details about how to create CCE clusters of different versions, see **Kubernetes Version Policy**. ● Upgrade your CCE cluster to version 1.28 if it is running an earlier version, such as 1.23 or lower. For details, see **Process and Method of Upgrading a Cluster**.
User-defined node name	N/A	Choose whether to enable this function to add a node name prefix. ● After a prefix is added, a node name consists of a prefix and a random number. ● The value can contain 1 to 64 characters. ● The prefix starts with a lowercase letter and only contains lowercase letters and digits. It is separated from the node name by a hyphen (-), for example, **node-com**.
Specification Management	N/A	You can add multiple specifications. Restrictions: ● Selecting multiple same specifications allows you to specify a node pool name by clicking **Advanced Configuration**. Only one node pool name can be left unspecified. ● The CPU architectures of all specifications must be identical, being either x86 or Arm. ● When selecting multiple GPU or NPU specifications, distributed training speed is impacted because different specifications' parameter network planes are not connected. For distributed training, it is recommended that you choose only one GPU or NPU specification. ● You can add up to 10 specifications to a resource pool.
	Specifications	Select required specifications. Due to system loss, the available resources are less than those specified in the specifications. After a dedicated resource pool is created, view the available resources in the **Nodes** tab on the details page.

Parameter	Sub-Parameter	Description
	AZ	Select **Automatic** or **Manual**. An AZ is a physical region where resources use independent power supplies and networks. AZs are physically isolated but interconnected over an intranet. ● **Automatic**: AZs are automatically allocated. ● **Manual**: Specify AZs for resource pool nodes. To ensure system disaster recovery, deploy all nodes in the same AZ. You can set the number of nodes in an AZ.
	Nodes	Select the number of nodes in a dedicated resource pool. More nodes mean higher computing performance. If **AZ** is set to **Manual**, you do not need to configure **Nodes**. **NOTE** It is good practice to create no more than 30 nodes at a time. Otherwise, the creation may fail due to traffic limiting. You can purchase nodes by rack for certain specifications. The total number of nodes is the number of racks multiplied by the number of nodes per rack. Purchasing a full rack allows you to isolate tasks physically, preventing communication conflicts and maintaining linear computing performance as task scale increases. All nodes in a rack must be created or deleted together. **Figure 2-9** Purchasing a rack of instances 

Parameter	Sub-Parameter	Description
	Advanced Configuration	Configure the following parameters if you enable advanced configuration: ● **Container Engine Space Size**: The default value is 50 GiB. The default and minimum values are 50 GiB. The maximum value depends on the specifications, and can be found in the console prompt. ● **Container Engine**: Container engines, one of the most important components of Kubernetes, manage the lifecycle of images and containers. kubelet interacts with a container engine through the Container Runtime Interface (CRI) to manage images and containers. When creating a resource pool, you can choose a container engine. Alternatively, you can change the container engine on the scaling page after the resource pool is created. Containerd has a shorter call chain, fewer components, and lower resource requirements, making it more stable. For details about the differences between Containerd and Docker, see **Container Engines**. The CCE cluster version determines the available container engines. If it is earlier than 1.23, only Docker is supported. If it is 1.27 or later, only containerd is supported. For all other versions, both containerd and Docker are options. ● **Node Pool Name**: You can customize the name of the new node pool. If you do not specify a name, the default name *Specification*-**default** is used. When selecting same specifications for multiple nodes, only one node pool name can be left unspecified. ● **Virtual Private Cloud**: Specifies the VPC network where the CCE cluster is located and cannot be changed. ● **Node subnet**: Choose a subnet within the same VPC. New nodes will be created using this subnet. ● **Associated Security Group**: Specifies the security group used by nodes created in the node pool. A maximum of four security groups can be selected. Traffic needs to pass through certain ports in the node security group to ensure node communications. If no security group is associated, the cluster's default rules are applied. ● **Resource Tag**: Add resource tags to classify resources. ● **Kubernetes Label**: Add key/value pairs that are attached to Kubernetes objects, such as Pods. A maximum of 20 labels can be added. Labels can be used to distinguish nodes. With workload affinity settings, container pods can be scheduled to a specified node.

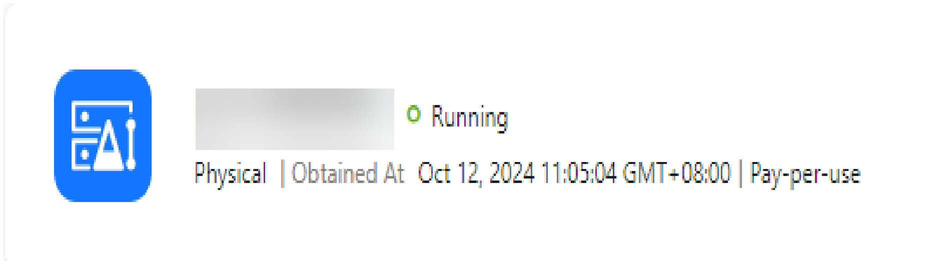
Parameter	Sub-Parameter	Description
		● **Taint**: This parameter is left blank by default. Configure anti-affinity by adding taints to nodes, with a maximum of 20 taints per node. ● **Post-installation Command**: Enter the script command, which cannot include Chinese characters. The Base64-encoded script must be transferred. The encoded script should not exceed 2,048 characters. The script will be executed after Kubernetes software is installed, which does not affect the installation. **NOTE** ● The name of an existing node pool in a resource pool cannot be changed. ● Do not run the **reboot** command in the post-installation script to restart the system immediately. To restart the system, run the **shutdown -r 1** command to restart with a delay of one minute.
Custom Driver	N/A	This function is disabled by default. Some GPU and Ascend resource pools allow custom driver installation. The driver is automatically installed in the cluster by default. Enable this function only if you need to specify the driver version. Determine the required driver version and choose the matching driver when buying Lite Cluster resources.
GPU/Ascend Driver	N/A	This parameter is displayed if **Custom Driver** is enabled. You can select a GPU or Ascend driver. The value depends on the driver you choose. For details about the required gpu-driver version, see **Software Versions Required by Different Models**.
Required Duration	N/A	Select the time length for which you want to use the resource pool. This parameter is mandatory only when the **Yearly/Monthly** billing mode is selected.
Login Mode	N/A	Choose a cluster login mode, **Password** or **Key pair**. ● **Password**: The default username is **root**, and you can set a password. ● **Key pair**: Select an existing key pair or click **Create Key Pair** to create one.

Parameter	Sub-Parameter	Description
Advanced Configuration	N/A	You can select **Configure Now** to configure tag information. ModelArts can work with Tag Management Service (TMS). When creating resource-consuming tasks in ModelArts, for example, training jobs, configure tags for these tasks so that ModelArts can use tags to manage resources by group. For details about how to use tags, see **Using TMS Tags to Manage Resources by Group**.

Click **Next** and confirm the settings. Then, click **Submit** to create the dedicated resource pool.

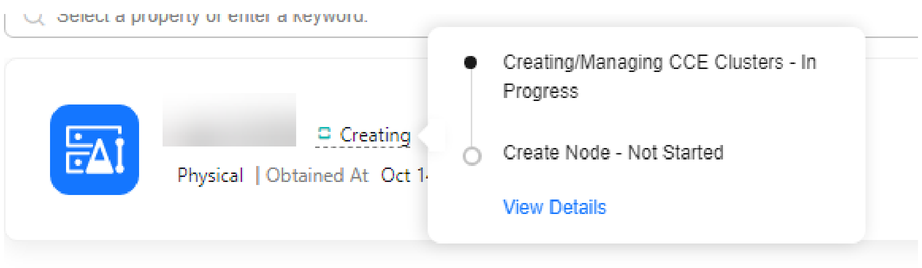
- After a resource pool is created, its status changes to **Running**. Only when the number of available nodes is greater than 0, tasks can be delivered to this resource pool.

Figure 2-10 Viewing a resource pool


Running
Physical | Obtained At Oct 12, 2024 11:05:04 GMT+08:00 | Pay-per-use

- Hover over **Creating** to view the details about the creation process. Click **View Details** to go the operation record page.

Figure 2-11 Creating


Select a property or enter a keyword.
Creating
Physical | Obtained At Oct 1
Creating/Managing CCE Clusters - In Progress
Create Node - Not Started
View Details

- You can view the task records of the resource pool by clicking **Records** in the upper left corner of the resource pool list.

Figure 2-12 Operation records

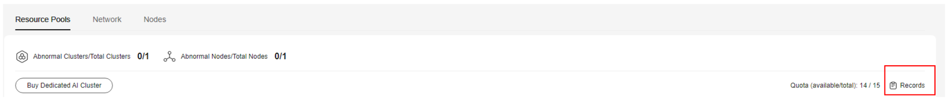
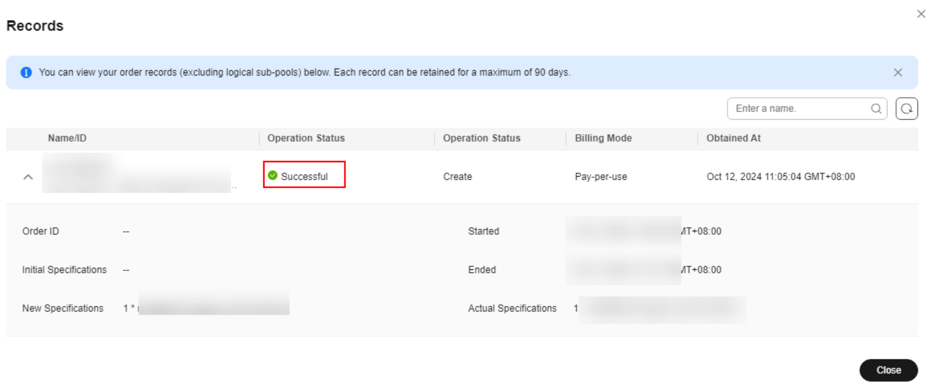

Resource Pools
Network
Nodes
Abnormal Clusters/Total Clusters 0/1
Abnormal Nodes/Total Nodes 0/1
Buy Dedicated AI Cluster
Quota (available/total): 14 / 15
Records

Figure 2-13 Viewing the resource pool status



Records

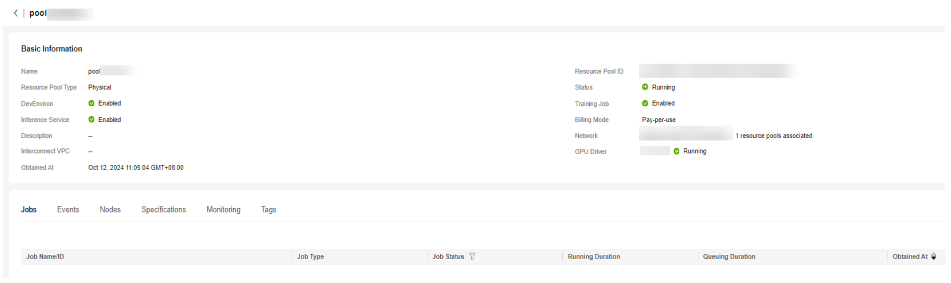
You can view your order records (excluding logical sub-pools) below. Each record can be retained for a maximum of 90 days.

Name/ID	Operation Status	Operation Status	Billing Mode	Obtained At
[redacted]	✓ Successful	Create	Pay-per-use	Oct 12, 2024 11:05:04 GMT+08:00

Order ID	--	Started	[redacted]MT+08:00
Initial Specifications	--	Ended	[redacted]MT+08:00
New Specifications	1 * [redacted]	Actual Specifications	1 [redacted]

After a resource pool is created, its status changes to **Running**. Click the cluster resource name to go to the resource details page. Check whether the purchased specifications are correct.

Figure 2-14 Viewing resource details



pool[redacted]

Basic Information

Name	pool[redacted]	Resource Pool ID	[redacted]
Resource Pool Type	Physical	Status	✓ Running
DevEnviron	✓ Enabled	Training Job	✓ Enabled
Inference Service	✓ Enabled	Billing Mode	Pay-per-use
Description	--	Network	[redacted] 1 resource pools associated
Interconnect VPC	--	GPU Driver	[redacted] ✓ Running
Obtained At	Oct 12, 2024 11:05:04 GMT+08:00		

Job Name/ID	Job Type	Job Status	Running Duration	Queuing Duration	Obtained At

3 Configuring Lite Cluster Resources

3.1 Configuring the Lite Cluster Environment

Configure the Lite Cluster environment by following this section, which applies to the accelerator card environment setup.

Prerequisites

- You have purchased and enabled cluster resources. For details, see **Enabling Lite Cluster Resources**.
- To configure and use a cluster, you need to have a solid understanding of **Kubernetes Basics**, as well as basic knowledge of networks, storage, and images.

Configuration Process

Figure 3-1 Flowchart

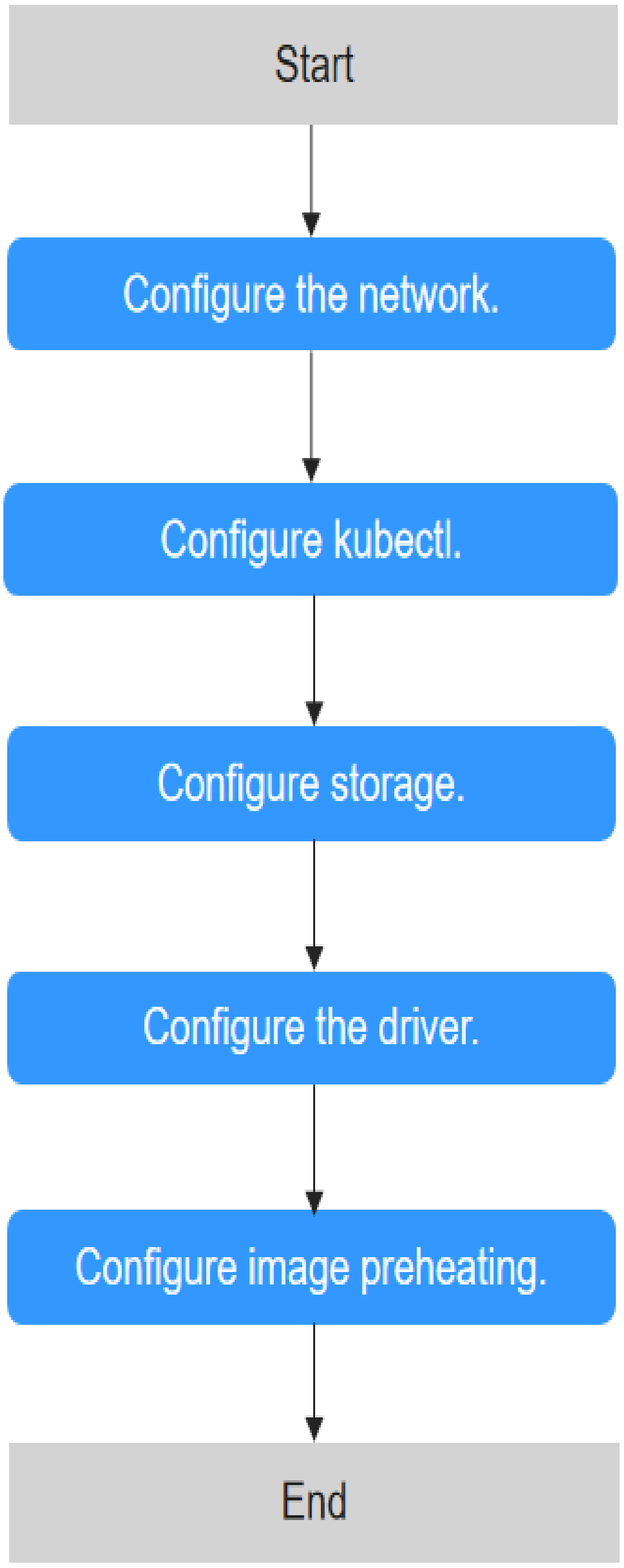


Table 3-1 Configuration process

Step	Task	Description
1	**Configuring the Lite Cluster Network**	After purchasing a resource pool, create an elastic IP (EIP) and configure the network. Once the network is set up, you can access cluster resources through the EIP.
2	**Configuring kubectl**	With kubectl configured, you can use the command line tool to manage your Kubernetes clusters by running kubectl commands.
3	**Configuring Lite Cluster Storage**	The available storage space is determined by dockerBaseSize when no external storage is mounted. However, the accessible storage space is limited. It is recommended that you mount external storage to overcome this limitation. You can mount storage to a container in various methods. The recommended method depends on the scenario, and you can choose one that meets your service needs.
4	**(Optional) Configuring the Driver**	Configure the corresponding driver to ensure proper use of GPU/Ascend resources in nodes within a dedicated resource pool. If no custom driver is configured and the default driver does not meet service requirements, upgrade the default driver to the required version.
5	**(Optional) Configuring Image Pre-provisioning**	Lite Cluster resource pools enable image pre-provisioning, which pulls images from nodes in the pools beforehand, accelerating image pulling during inference and large-scale distributed training.

Quick Configuration of Lite Cluster Resources

This section shows how to configure Lite Cluster resources quickly to log in to nodes and view accelerator cards, then complete a training job. Before you start, you need to purchase resources. For details, see **Enabling Lite Cluster Resources**.

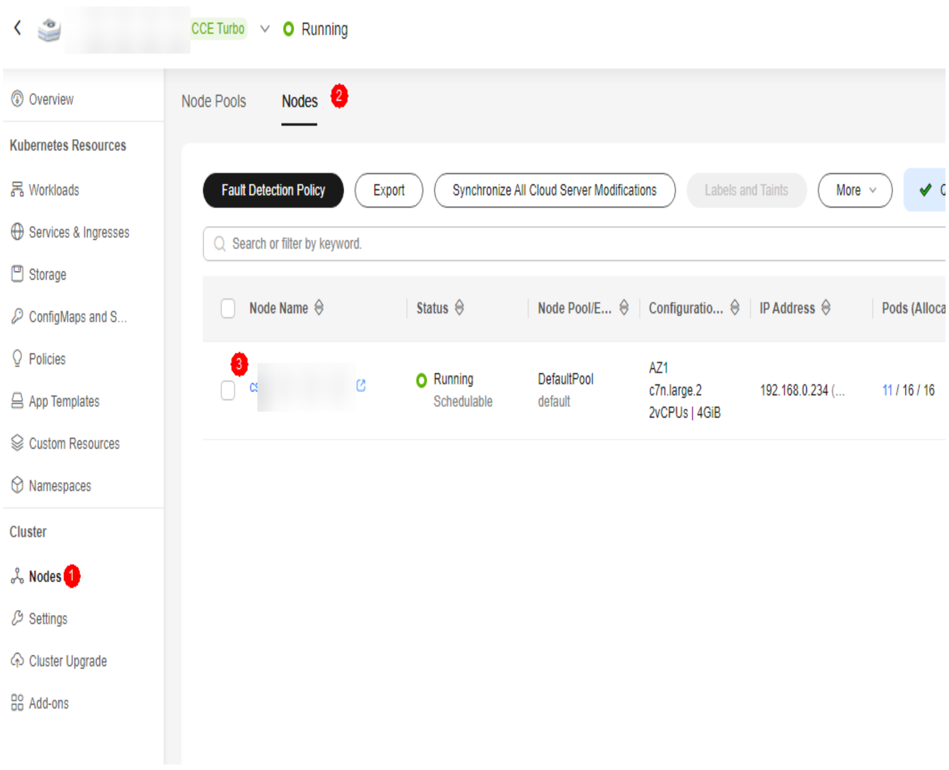
Step 1 Log in to a node.

(Recommended) Method 1: Binding an EIP

Bind an EIP to the node and use Bash tools such as Xshell and MobaXterm to log in to the node.

1. Log in to the CCE console.
2. On the CCE cluster details page, click **Nodes**. In the **Nodes** tab, click the name of the target node to go to the ECS page.

Figure 3-2 Node management

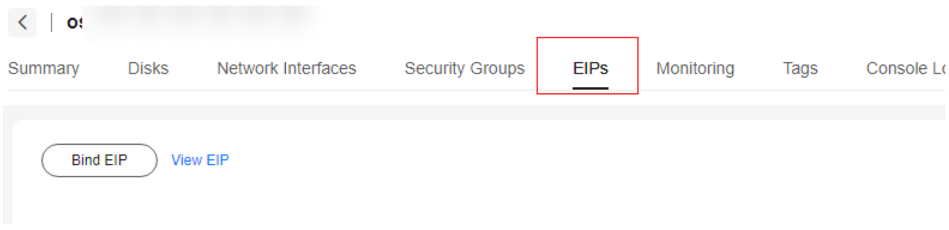


Node Name	Status	Node Pool/E...	Configuratio...	IP Address	Pods (Alloca
cs	Running Schedulable	DefaultPool default	AZ1 c7n.large.2 2vCPUs | 4GiB	192.168.0.234 (...	11 / 16 / 16

3. Bind an EIP.

   Choose or create one.

Figure 3-3 EIP



Click **Buy EIP**.

Figure 3-4 Binding an EIP

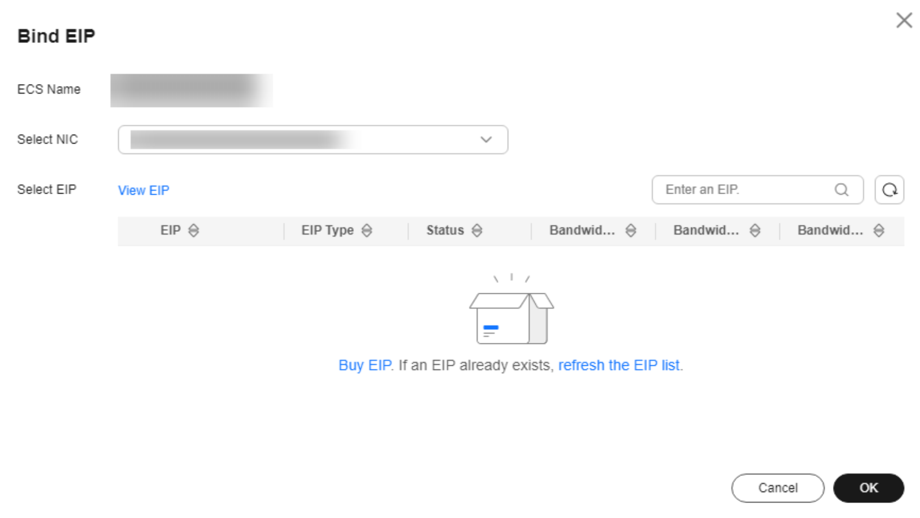
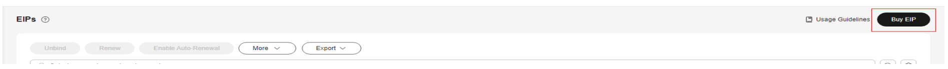


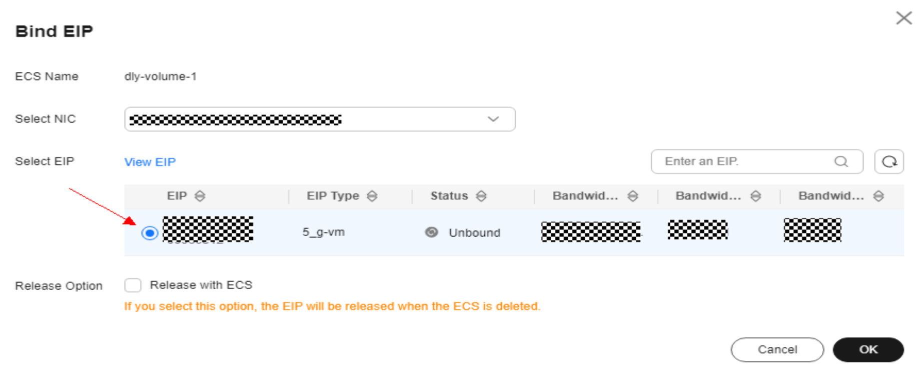
Figure 3-5 Buying an EIP



Refresh the list on the ECS page after completing the purchase.

Select the new EIP and click **OK**.

Figure 3-6 Binding an EIP



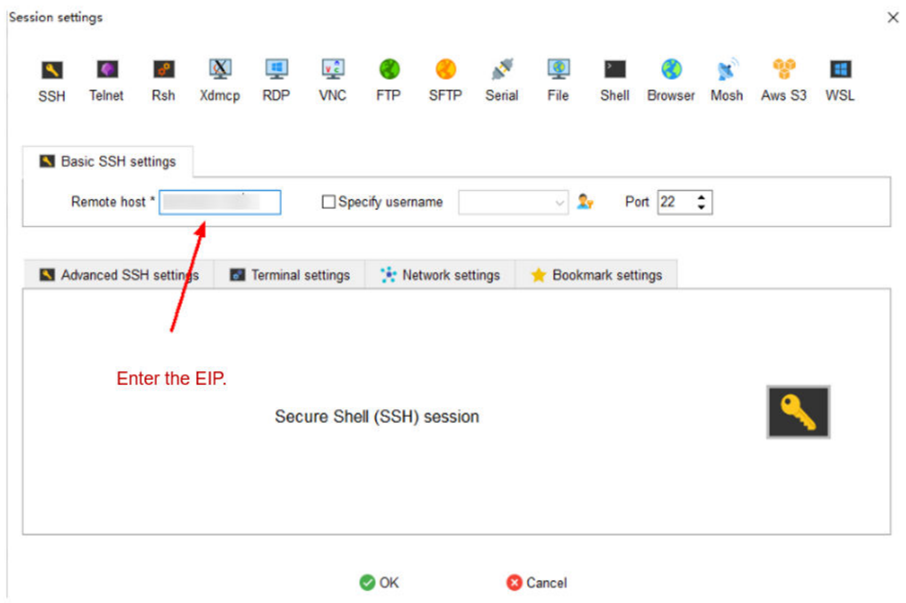
ECS Name	dly-volume-1
Select NIC	[redacted]
Release Option	Release with ECS

EIP	EIP Type	Status	Bandwid...	Bandwid...	Bandwid...
[redacted]	5_g-vm	Unbound	[redacted]	[redacted]	[redacted]

If you select this option, the EIP will be released when the ECS is deleted.

4. Log in to the node using MobaXterm or Xshell. To log in using MobaXterm, enter the EIP.

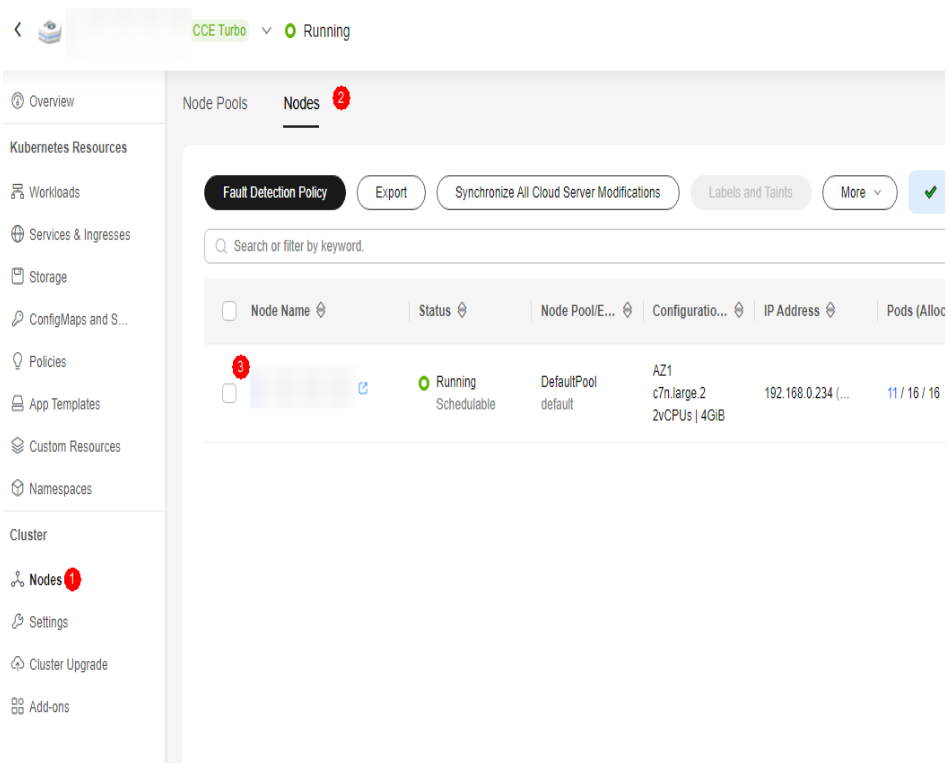
Figure 3-7 Logging in to a node



Method 2: Using Huawei Cloud Remote Login

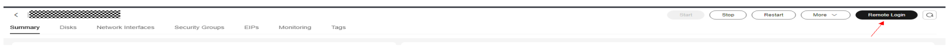
1. Log in to the CCE console.
2. On the CCE cluster details page, click **Nodes**. In the **Nodes** tab, click the name of the target node to go to the ECS page.

Figure 3-8 Node management



3. Click **Remote Login**. In the displayed dialog box, click **Log In**.

Figure 3-9 Remote login



4. After setting parameters such as the password in CloudShell, click **Connect** to log in to the node. For details about CloudShell, see **Logging In to a Linux ECS Using CloudShell**.

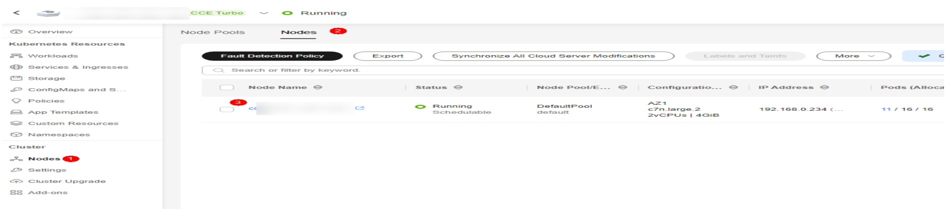
Step 2 Configure the kubectl tool.

Log in to the ModelArts console. From the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**.

Click the new dedicated resource pool to access its details page. Click the CCE cluster to access its details page.

On the CCE cluster details page, locate **Connection Information** in the cluster information.

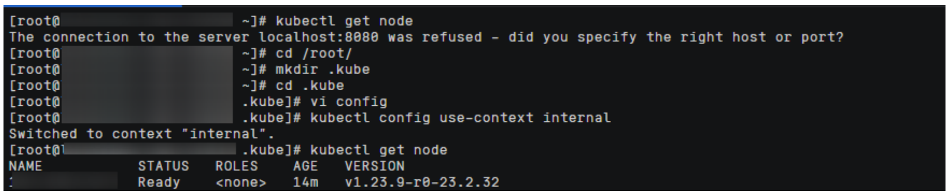
Figure 3-10 Connection Information



Use kubectl.

- To use kubectl through the intranet, install it on a node within the same VPC as the cluster. Click **Configure** next to **kubectl** to use the kubectl tool.

Figure 3-11 Using kubectl through the intranet

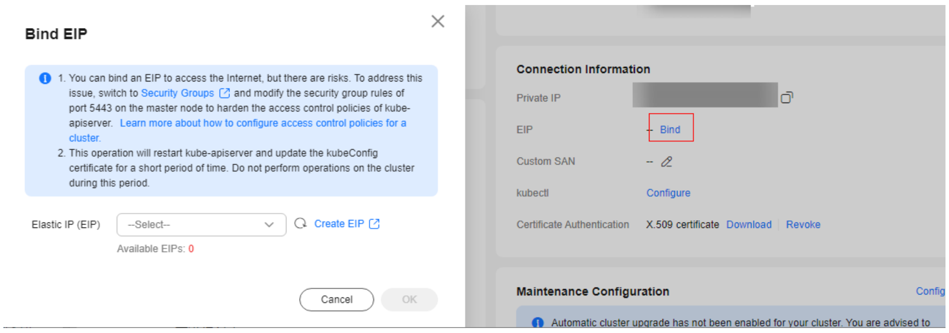


```
[root@[redacted] ~]# kubectl get node
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@[redacted] ~]# cd /root/
[root@[redacted] ~]# mkdir .kube
[root@[redacted] ~]# cd .kube
[root@[redacted] .kube]# vi config
[root@[redacted] .kube]# kubectl config use-context internal
Switched to context "internal".
[root@[redacted] .kube]# kubectl get node
NAME             STATUS   ROLES    AGE   VERSION
[redacted]       Ready    <none>   14m   v1.23.9-r0-23.2.32
```

- To use kubectl through an EIP, install it on any node that associated with the EIP.

  To bind an EIP, click **Bind** next to **EIP**.

Figure 3-12 Binding an EIP



Bind EIP

1. You can bind an EIP to access the Internet, but there are risks. To address this issue, switch to Security Groups and modify the security group rules of port 5443 on the master node to harden the access control policies of kube-apiserver. Learn more about how to configure access control policies for a cluster.
2. This operation will restart kube-apiserver and update the kubeConfig certificate for a short period of time. Do not perform operations on the cluster during this period.

Elastic IP (EIP): --Select--
Available EIPs: 0

Connection Information

Private IP	[redacted]
EIP	-- Bind
Custom SAN	--
kubectl	Configure
Certificate Authentication	X.509 certificate Download | Revoke

Maintenance Configuration

Automatic cluster upgrade has not been enabled for your cluster. You are advised to

Select an EIP and click **OK**. If no EIP is available, click **Create EIP** to create one.

After the binding is complete, click **Configure** next to **kubectl** and use kubectl as prompted.

Step 3 Start a task using **docker run**.

Snt9B clusters managed in CCE automatically install container runtime. The following uses Docker as an example and is only for testing and verification. You can start the container for testing without creating a deployment or volcano job. The BERT NLP model is used in the training test cases.

1. Pull the image. The test image is **bert_pretrain_mindspore:v1**, which contains the test data and code.

```
docker pull swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
docker tag swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
bert_pretrain_mindspore:v1
```

2. Start the container.

```
docker run -tid --privileged=true \
-u 0 \
-v /dev/shm:/dev/shm \
--device=/dev/davinci0 \
--device=/dev/davinci1 \
--device=/dev/davinci2 \
--device=/dev/davinci3 \
--device=/dev/davinci4 \
--device=/dev/davinci5 \
--device=/dev/davinci6 \
--device=/dev/davinci7 \
--device=/dev/davinci_manager \
--device=/dev/devmm_svm \
--device=/dev/hisi_hdc \
-v /usr/local/Ascend/driver:/usr/local/Ascend/driver  \
-v /usr/local/sbin/npu-smi:/usr/local/sbin/npu-smi \
-v /etc/hccn.conf:/etc/hccn.conf \
bert_pretrain_mindspore:v1 \
bash
```

Parameters:

– --privileged=true //Privileged container, which can access all devices connected to the host.

– -u 0 //root user

– -v /dev/shm:/dev/shm //Prevents the training task from failing due to insufficient shared memory.

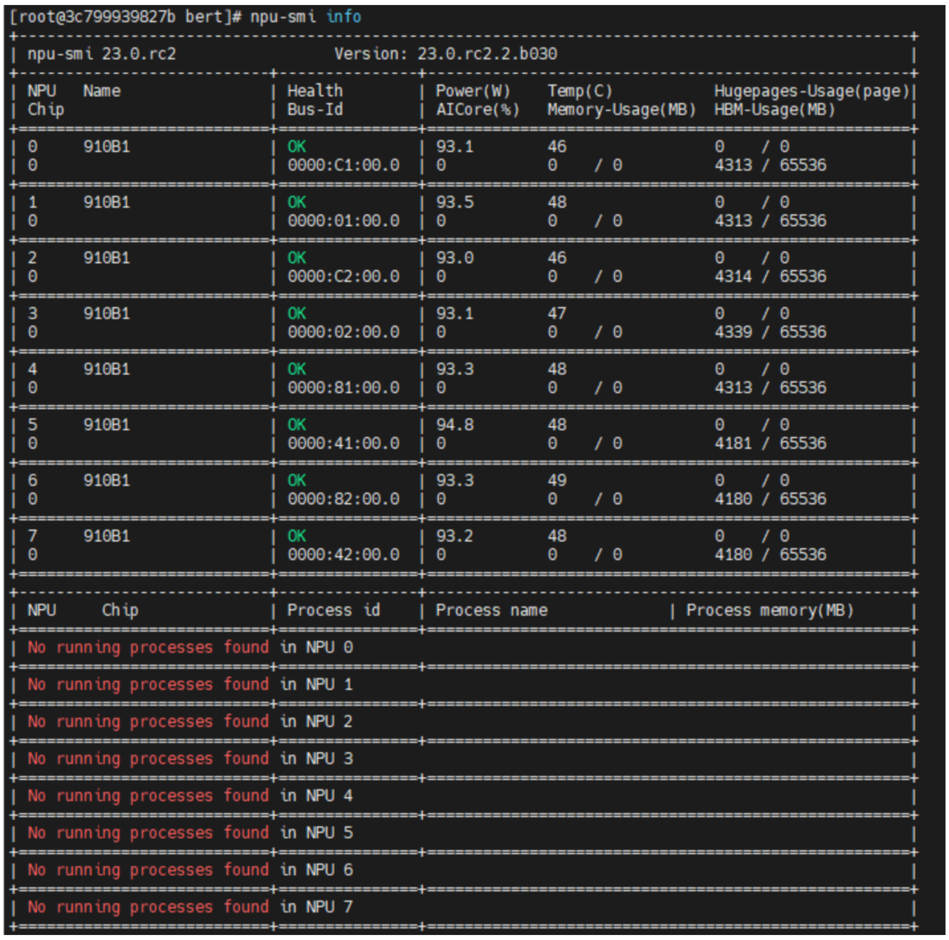
– --device=/dev/davinci0 //NPU card device

- --device=/dev/davinci1 //NPU card device
- --device=/dev/davinci2 //NPU card device
- --device=/dev/davinci3 //NPU card device
- --device=/dev/davinci4 //NPU card device
- --device=/dev/davinci5 //NPU card device
- --device=/dev/davinci6 //NPU card device
- --device=/dev/davinci7 //NPU card device
- --device=/dev/davinci_manager //Da Vinci-related management device
- --device=/dev/devmm_svm //Management device
- --device=/dev/hisi_hdc //Management device
- -v /usr/local/Ascend/driver:/usr/local/Ascend/driver //NPU card driver mounting
- -v /usr/local/sbin/npu-smi:/usr/local/sbin/npu-smi //npu-smi tool mounting
- -v /etc/hccn.conf:/etc/hccn.conf //hccn.conf configuration mounting

3. Access the container and view the card information.

```
docker exec -it xxxxxxx bash    //Access the container. Replace xxxxxxx with the container ID.
npu-smi info    //View card information.
```

Figure 3-13 Viewing NPU information

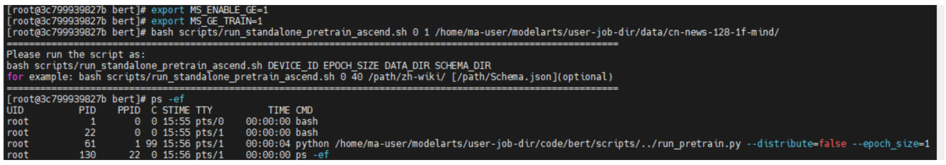


```
[root@3c799939827b bert]# npu-smi info
+------------------------------------------------------------------------------------------------+
| npu-smi 23.0.rc2                 Version: 23.0.rc2.2.b030                                      |
+---------------------------+---------------+----------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+====================================================+
| 0     910B1               | OK            | 93.1        46                0    / 0             |
| 0                         | 0000:C1:00.0  | 0           0    / 0          4313 / 65536         |
+===========================+===============+====================================================+
| 1     910B1               | OK            | 93.5        48                0    / 0             |
| 0                         | 0000:01:00.0  | 0           0    / 0          4313 / 65536         |
+===========================+===============+====================================================+
| 2     910B1               | OK            | 93.0        46                0    / 0             |
| 0                         | 0000:C2:00.0  | 0           0    / 0          4314 / 65536         |
+===========================+===============+====================================================+
| 3     910B1               | OK            | 93.1        47                0    / 0             |
| 0                         | 0000:02:00.0  | 0           0    / 0          4339 / 65536         |
+===========================+===============+====================================================+
| 4     910B1               | OK            | 93.3        48                0    / 0             |
| 0                         | 0000:81:00.0  | 0           0    / 0          4313 / 65536         |
+===========================+===============+====================================================+
| 5     910B1               | OK            | 94.8        48                0    / 0             |
| 0                         | 0000:41:00.0  | 0           0    / 0          4181 / 65536         |
+===========================+===============+====================================================+
| 6     910B1               | OK            | 93.3        49                0    / 0             |
| 0                         | 0000:82:00.0  | 0           0    / 0          4180 / 65536         |
+===========================+===============+====================================================+
| 7     910B1               | OK            | 93.2        48                0    / 0             |
| 0                         | 0000:42:00.0  | 0           0    / 0          4180 / 65536         |
+===========================+===============+====================================================+
+---------------------------+---------------+----------------------------------------------------+
| NPU     Chip              | Process id    | Process name             | Process memory(MB)      |
+===========================+===============+====================================================+
| No running processes found in NPU 0                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 1                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 2                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 3                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 4                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 5                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 6                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 7                                                            |
+===========================+===============+====================================================+
```

4. Start the training task:

```
cd /home/ma-user/modelarts/user-job-dir/code/bert/
export MS_ENABLE_GE=1
export MS_GE_TRAIN=1
bash scripts/run_standalone_pretrain_ascend.sh 0 1 /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
```

Figure 3-14 Training process

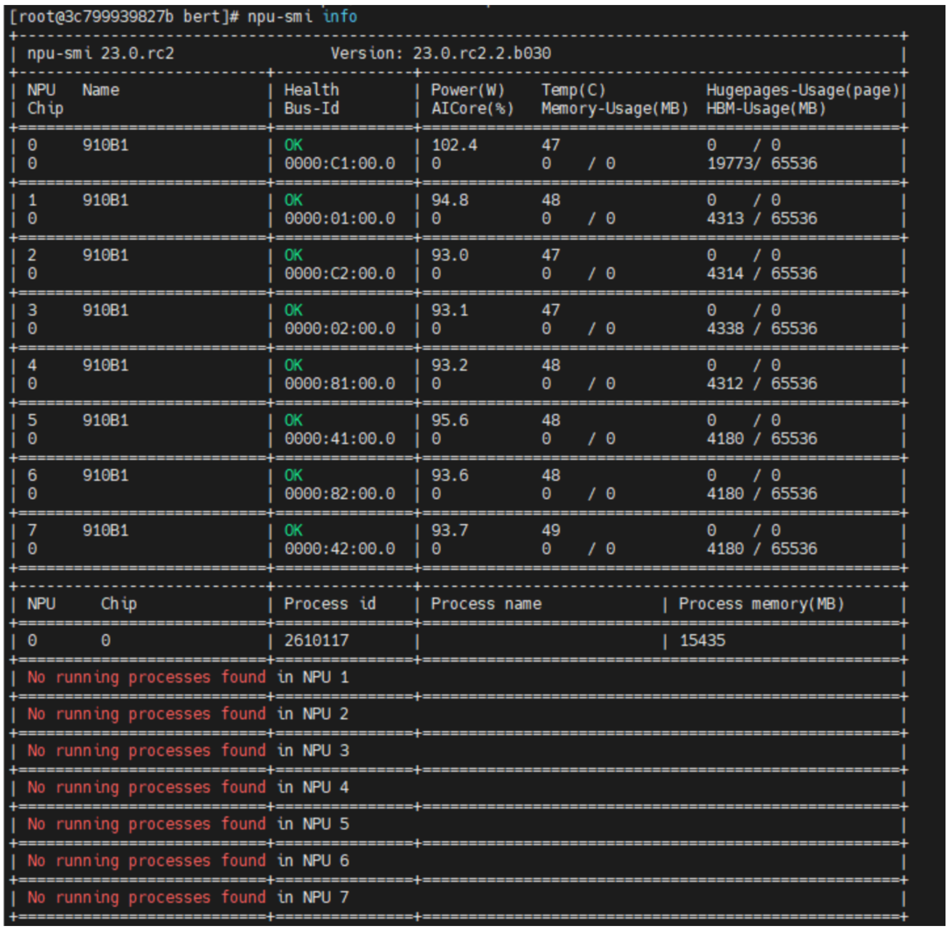


```
[root@3c799939827b bert]# export MS_ENABLE_GE=1
[root@3c799939827b bert]# export MS_GE_TRAIN=1
[root@3c799939827b bert]# bash scripts/run_standalone_pretrain_ascend.sh 0 1 /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
==============================================================================================================
Please run the script as:
bash scripts/run_standalone_pretrain_ascend.sh DEVICE_ID EPOCH_SIZE DATA_DIR SCHEMA_DIR
for example: bash scripts/run_standalone_pretrain_ascend.sh 0 40 /path/zh-wiki/ [/path/Schema.json](optional)
==============================================================================================================
[root@3c799939827b bert]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 15:55 pts/0    00:00:00 bash
root          22       0  0 15:55 pts/1    00:00:00 bash
root          61       1 99 15:56 pts/1    00:00:04 python /home/ma-user/modelarts/user-job-dir/code/bert/scripts/../run_pretrain.py --distribute=false --epoch_size=1
root         130      22  0 15:56 pts/1    00:00:00 ps -ef
```

Check the card usage. The card 0 is in use, as expected.

```
npu-smi info    //View card information.
```

Figure 3-15 Viewing NPU information

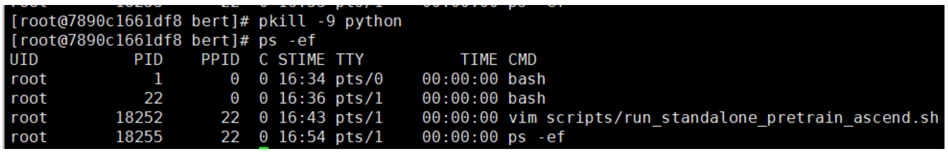


```
[root@3c799939827b bert]# npu-smi info
+------------------------------------------------------------------------------------------------+
| npu-smi 23.0.rc2                 Version: 23.0.rc2.2.b030                                      |
+---------------------------+---------------+----------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+====================================================+
| 0     910B1               | OK            | 102.4       47                0    / 0             |
| 0                         | 0000:C1:00.0  | 0           0    / 0          19773/ 65536         |
+===========================+===============+====================================================+
| 1     910B1               | OK            | 94.8        48                0    / 0             |
| 0                         | 0000:01:00.0  | 0           0    / 0          4313 / 65536         |
+===========================+===============+====================================================+
| 2     910B1               | OK            | 93.0        47                0    / 0             |
| 0                         | 0000:C2:00.0  | 0           0    / 0          4314 / 65536         |
+===========================+===============+====================================================+
| 3     910B1               | OK            | 93.1        47                0    / 0             |
| 0                         | 0000:02:00.0  | 0           0    / 0          4338 / 65536         |
+===========================+===============+====================================================+
| 4     910B1               | OK            | 93.2        48                0    / 0             |
| 0                         | 0000:81:00.0  | 0           0    / 0          4312 / 65536         |
+===========================+===============+====================================================+
| 5     910B1               | OK            | 95.6        48                0    / 0             |
| 0                         | 0000:41:00.0  | 0           0    / 0          4180 / 65536         |
+===========================+===============+====================================================+
| 6     910B1               | OK            | 93.6        48                0    / 0             |
| 0                         | 0000:82:00.0  | 0           0    / 0          4180 / 65536         |
+===========================+===============+====================================================+
| 7     910B1               | OK            | 93.7        49                0    / 0             |
| 0                         | 0000:42:00.0  | 0           0    / 0          4180 / 65536         |
+===========================+===============+====================================================+
+---------------------------+---------------+----------------------------------------------------+
| NPU     Chip              | Process id    | Process name             | Process memory(MB)      |
+===========================+===============+====================================================+
| 0       0                 | 2610117       |                          | 15435                   |
+===========================+===============+====================================================+
| No running processes found in NPU 1                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 2                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 3                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 4                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 5                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 6                                                            |
+===========================+===============+====================================================+
| No running processes found in NPU 7                                                            |
+===========================+===============+====================================================+
```

The training task takes about two hours to complete and then automatically stops. To stop a training task, run the commands below:

```
pkill -9 python
ps -ef
```

Figure 3-16 Stopping the training process



```
[root@7890c1661df8 bert]# pkill -9 python
[root@7890c1661df8 bert]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 16:34 pts/0    00:00:00 bash
root          22       0  0 16:36 pts/1    00:00:00 bash
root       18252      22  0 16:43 pts/1    00:00:00 vim scripts/run_standalone_pretrain_ascend.sh
root       18255      22  0 16:54 pts/1    00:00:00 ps -ef
```

----End

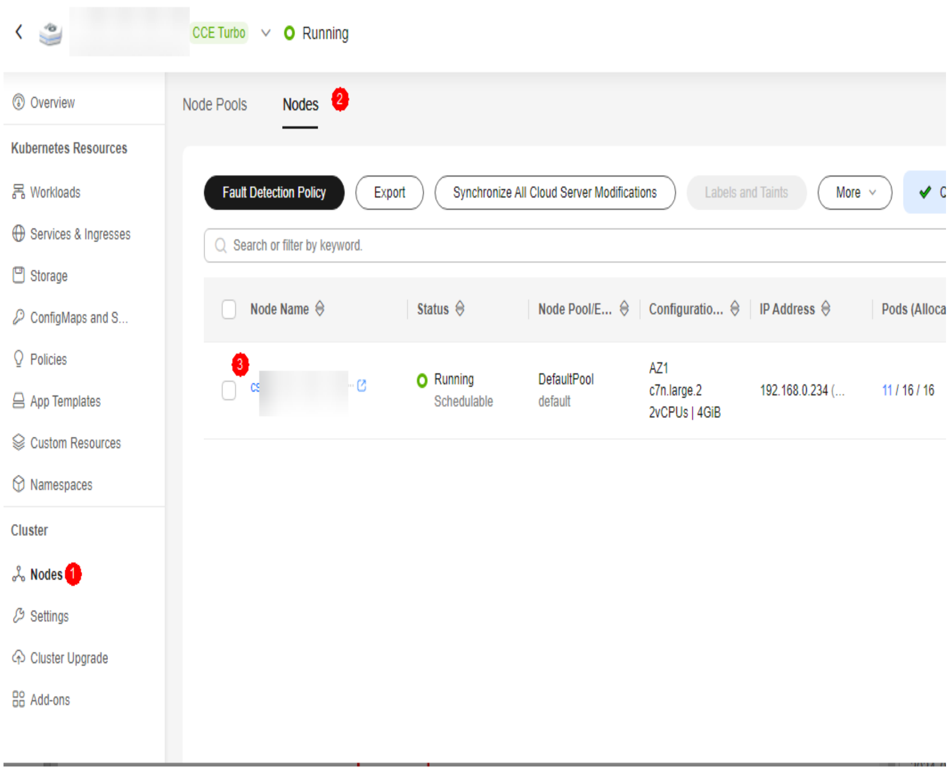
3.2 Configuring the Lite Cluster Network

Apply for an EIP and bind it to an ECS to enable the ECS to access the Internet.

Step 1 Log in to **the CCE console**.

Step 2 Locate the CCE cluster you select when you purchase Lite Cluster resources. Click the cluster name to access the CCE cluster details page. From there, click **Nodes**. On the **Nodes** tab, click the node to log in to. The ECS page will appear.

Figure 3-17 Node management

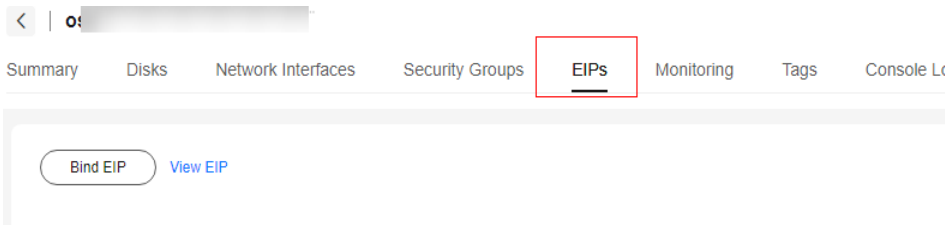


Node Name	Status	Node Pool/E...	Configuratio...	IP Address	Pods (Alloca
cs	Running Schedulable	DefaultPool default	AZ1 c7n.large.2 2vCPUs | 4GiB	192.168.0.234 (...	11 / 16 / 16

Step 3 Bind an EIP.

Choose or buy one. For details about how to buy an EIP, see **Setting Up a Network in a VPC and Enabling Internet Access Using an EIP**.

Figure 3-18 EIP



Click **Buy EIP**.

Figure 3-19 Binding an EIP

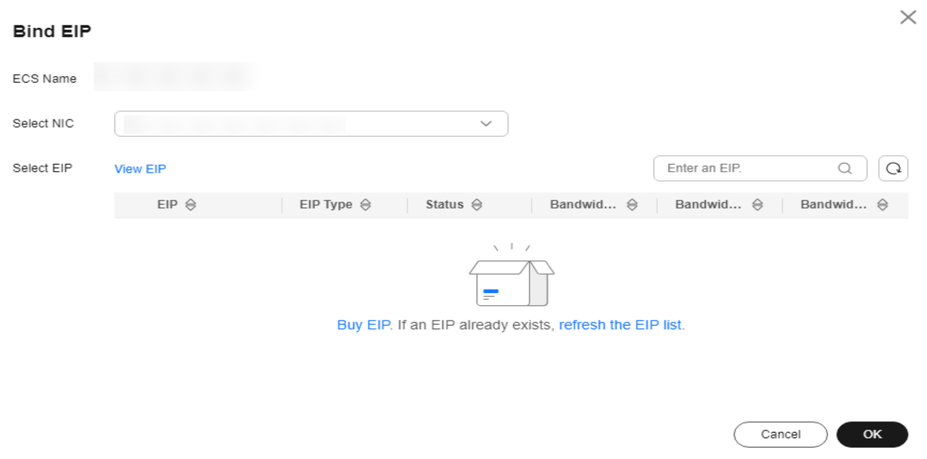
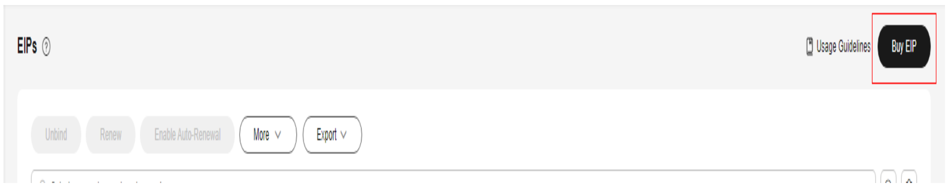


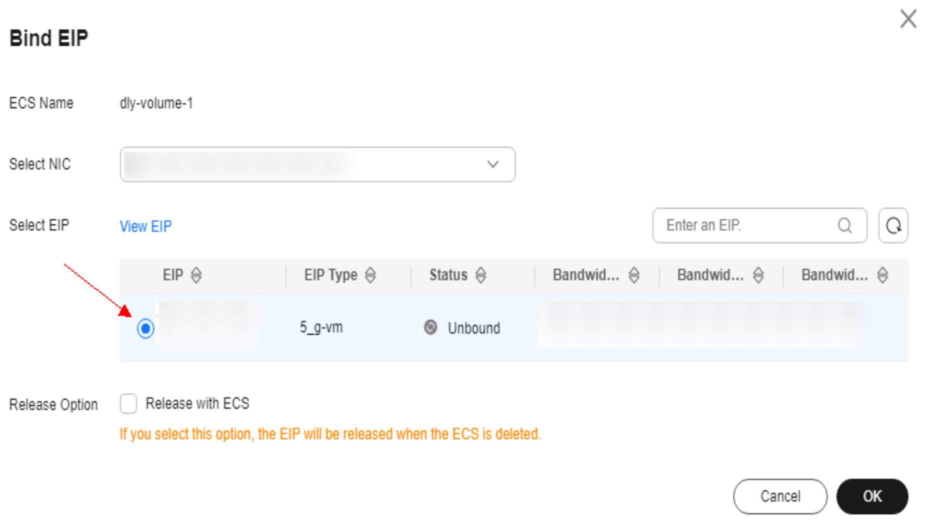
Figure 3-20 Buying an EIP



Refresh the list on the ECS page after completing the purchase.

Select the new EIP and click **OK**.

Figure 3-21 Binding an EIP



Bind EIP

ECS Name: dly-volume-1

Select NIC

Select EIP: View EIP

EIP	EIP Type	Status	Bandwid...	Bandwid...	Bandwid...
●	5_g-vm	Unbound			

Release Option: ☐ Release with ECS

If you select this option, the EIP will be released when the ECS is deleted.

Step 4 Access cluster resources remotely using SSH with a password or key pair.

- To use a key pair, see **Logging In to a Linux ECS Using an SSH Key Pair**.
- To use a key pair, see **Logging In to a Linux ECS Using an SSH Password**.

----End

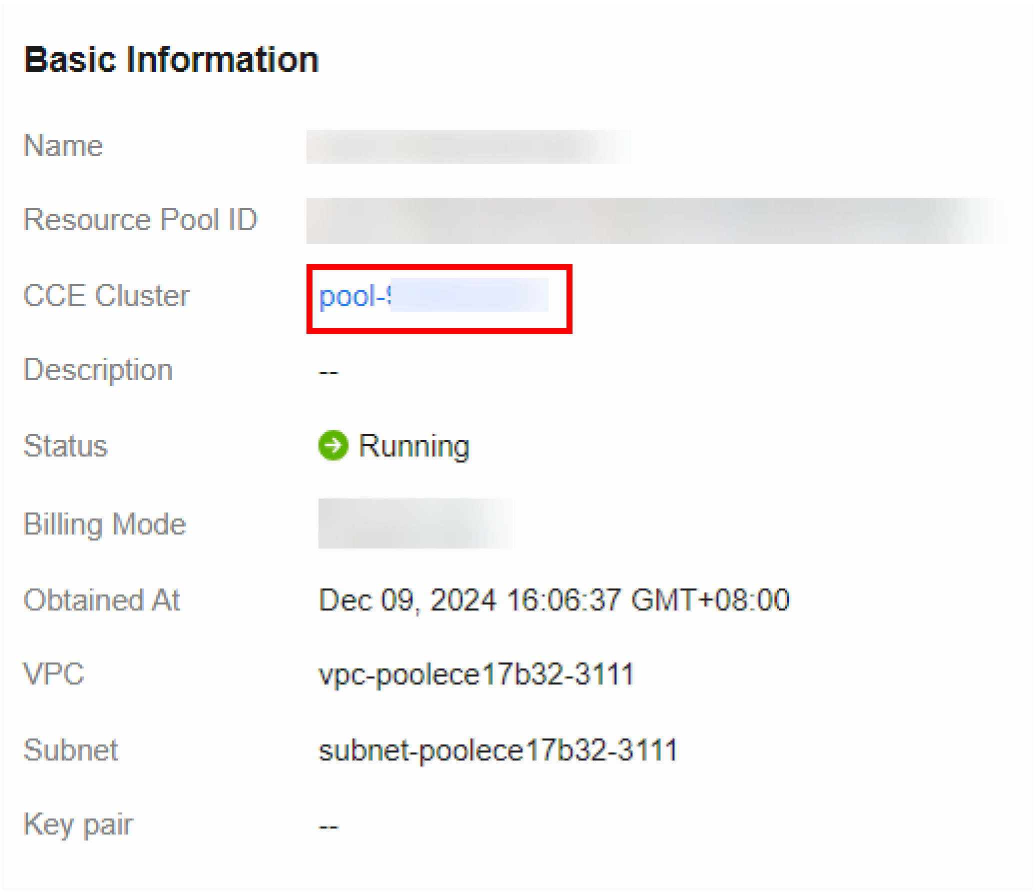
3.3 Configuring kubectl

With **kubectl** configured, you can use the command line tool to manage your Kubernetes clusters by running kubectl commands. Follow these steps to configure kubectl.

Step 1 Log in to the ModelArts console. From the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. Click the **ModelArts Lite** tab.

Step 2 Click the created dedicated resource pool to access its details page.

Figure 3-22 Basic information

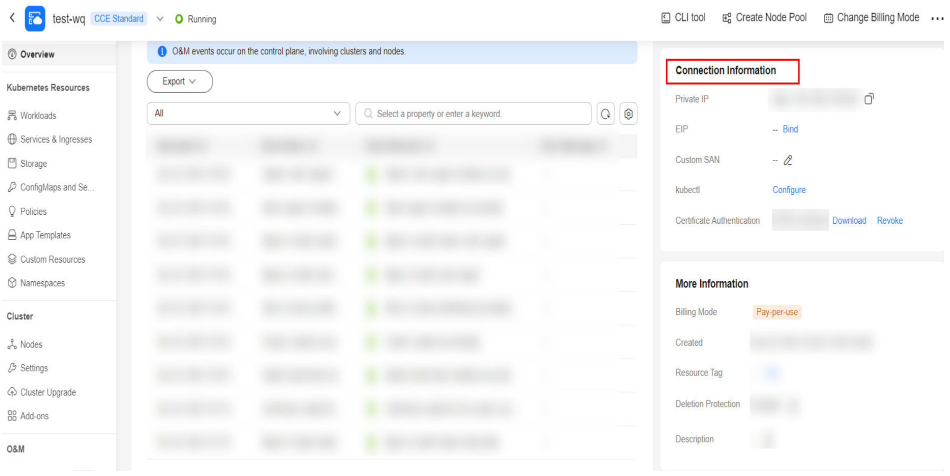


Basic Information

Name	[redacted]
Resource Pool ID	[redacted]
CCE Cluster	pool-[redacted]
Description	--
Status	Running
Billing Mode	[redacted]
Obtained At	Dec 09, 2024 16:06:37 GMT+08:00
VPC	vpc-poolece17b32-3111
Subnet	subnet-poolece17b32-3111
Key pair	--

Step 3 Click the CCE cluster to access its details page. From there, locate **Connection Information** in the cluster information.

Figure 3-23 Connection Information



O&M events occur on the control plane, involving clusters and nodes.

Connection Information

Private IP	[redacted]
EIP	-- Bind
Custom SAN	--
kubectl	Configure
Certificate Authentication	[redacted] Download Revoke

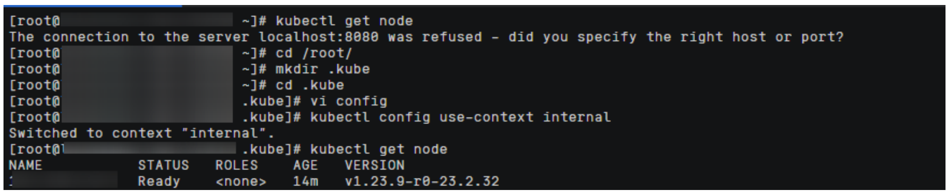
More Information

Billing Mode	Pay-per-use
Created	[redacted]
Resource Tag	[redacted]
Deletion Protection	[redacted]
Description	[redacted]

Step 4 Use kubectl.

- To use kubectl through the intranet, install it on a node within the same VPC as the cluster. Click **Configure** next to **kubectl**. Perform operations as prompted.

Figure 3-24 Using kubectl through the intranet

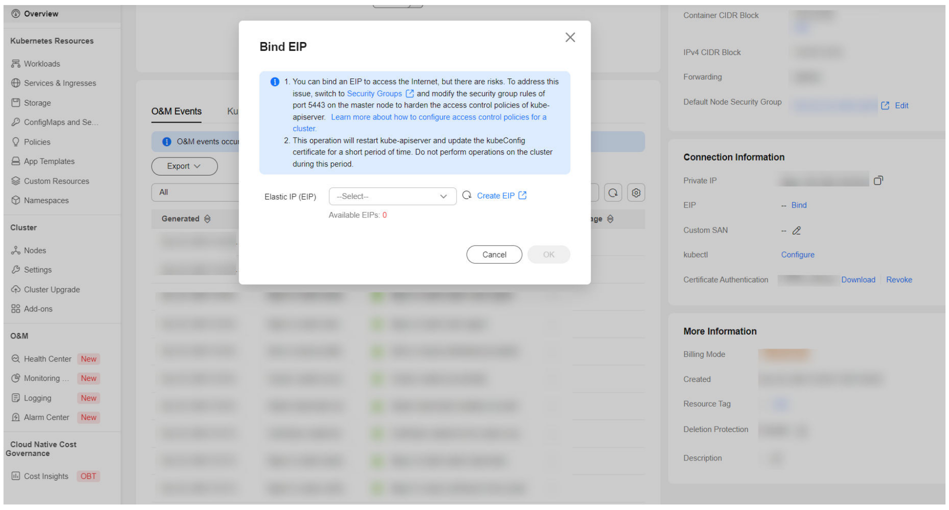


```
[root@[redacted] ~]# kubectl get node
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@[redacted] ~]# cd /root/
[root@[redacted] ~]# mkdir .kube
[root@[redacted] ~]# cd .kube
[root@[redacted] .kube]# vi config
[root@[redacted] .kube]# kubectl config use-context internal
Switched to context "internal".
[root@[redacted] .kube]# kubectl get node
NAME            STATUS   ROLES    AGE   VERSION
[redacted]      Ready    <none>   14m   v1.23.9-r0-23.2.32
```

- To use kubectl through an EIP, install it on any node that associated with the EIP.

  To bind an EIP, click **Bind** next to **EIP**.

Figure 3-25 Binding an EIP



Bind EIP

1. You can bind an EIP to access the Internet, but there are risks. To address this issue, switch to Security Groups and modify the security group rules of port 5443 on the master node to harden the access control policies of kube-apiserver. Learn more about how to configure access control policies for a cluster.
2. This operation will restart kube-apiserver and update the kubeConfig certificate for a short period of time. Do not perform operations on the cluster during this period.

Elastic IP (EIP) --Select-- Create EIP

Available EIPs: 0

Cancel OK

Container CIDR Block [redacted]
IPv4 CIDR Block [redacted]
Forwarding [redacted]
Default Node Security Group [redacted] Edit

Connection Information

Private IP [redacted]
EIP -- Bind
Custom SAN --
kubectl Configure
Certificate Authentication [redacted] Download Revoke

More Information

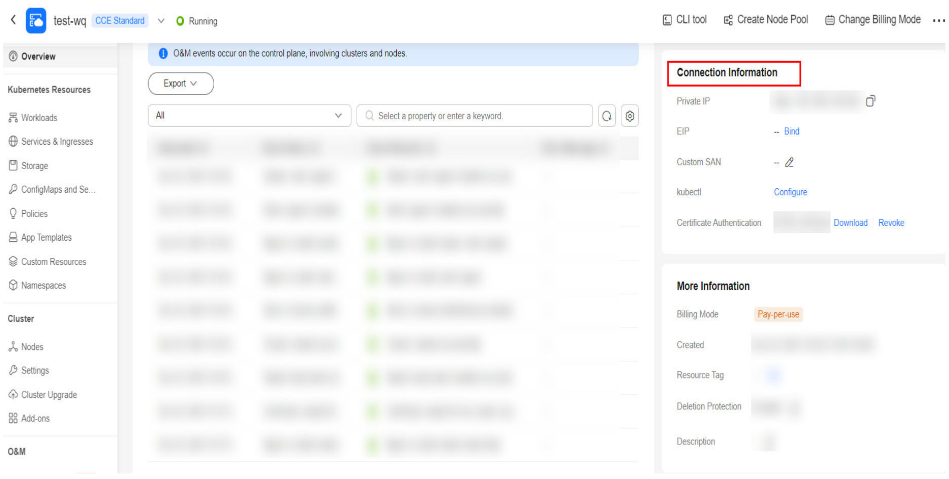
Billing Mode [redacted]
Created [redacted]
Resource Tag [redacted]
Deletion Protection [redacted]
Description [redacted]

Choose or create an EIP.

After the EIP is bound, locate **Connection Information** in the cluster information and click **Configure** next to **kubectl**.

Perform operations as prompted.

Figure 3-26 Configuring kubectl



O&M events occur on the control plane, involving clusters and nodes.

Connection Information

Private IP [redacted]
EIP -- Bind
Custom SAN --
kubectl Configure
Certificate Authentication [redacted] Download Revoke

More Information

Billing Mode Pay-per-use
Created [redacted]
Resource Tag [redacted]
Deletion Protection [redacted]
Description [redacted]

Step 5 Verify the configuration.

Run this command on the node where kubectl is installed. If the cluster node is displayed, the configuration is successful.

```
kubectl get node
```

----End

3.4 Configuring Lite Cluster Storage

The available storage space is determined by dockerBaseSize when no external storage is mounted. However, the accessible storage space is limited. It is recommended that you mount external storage to overcome this limitation.

You can mount storage to a container in various methods. The recommended method depends on the scenario. For details, see **Table 3-2**. **Storage Basics** helps you understand this section. **Data Disk Space Allocation** helps you understand how to configure data disk size based on service needs.

Table 3-2 Different methods of mounting storage to a container

Method	Scenario	Description	Operation Reference
EmptyDir	Training cache	Kubernetes ephemeral volumes, which are created and deleted together with Pods following the Pod lifecycle.	**Using a Temporary Path**
HostPath	This method is suitable for: 1. Containerized workload logs that need to be saved permanently 2. Containerized workloads that need to access internal data structure of the Docker engine in the host	Node storage. Multiple containers may share the storage, causing write conflicts. Deleting a Pod does not clear its storage.	**hostPath**

Method	Scenario	Description	Operation Reference
OBS	Training dataset storage	Object storage. The OBS SDKs are used to download sample data. Due to the large storage capacity being far from nodes, direct training speed is slow. To improve this, data is typically pulled to a local cache before training.	• **Using an Existing OBS Bucket Through a Static PV** • **Using an OBS Bucket Through a Dynamic PV**
SFS Turbo	Massive amounts of small files	• POSIX file system • Shared or interconnected VPC between the file system and resource pool • High costs	• **Using an Existing SFS Turbo File System Through a Static PV** • Dynamic mounting: not supported
SFS	Persistent storage for frequent reads and writes	This method applies to cost-sensitive workloads which require large-capacity scalability, such as media processing, content management, big data analytics, and workload analysis. SFS Capacity-Oriented file systems are not suitable for services with massive amounts of small files.	• **Using an Existing SFS File System Through a Static PV** • **Using an SFS File System Through a Dynamic PV**
EVS	Data persistence for notebook-based development	Each volume can be mounted to only one node. The storage size depends on the size of the EVS disk.	• **Using an Existing EVS Disk Through a Static PV** • **Using an EVS Disk Through a Dynamic PV**

3.5 (Optional) Configuring the Driver

Configure the corresponding driver to ensure proper use of GPU/Ascend resources in nodes within a dedicated resource pool.

Lite Cluster supports two driver configuration methods:

- **Method 1: Configuring a Custom Driver When Buying a Resource Pool**
- **Method 2: Upgrading the Existing Resource Pool Driver**

Method 1: Configuring a Custom Driver When Buying a Resource Pool

Some GPU and Ascend resource pools allow custom drivers. Enable **Custom Driver** and select the required driver version.

Method 2: Upgrading the Existing Resource Pool Driver

If no custom driver is configured and the default driver does not meet service requirements, upgrade the default driver to the required version. For details, see **Upgrading the Lite Cluster Resource Pool Driver**.

3.6 (Optional) Configuring Image Pre-provisioning

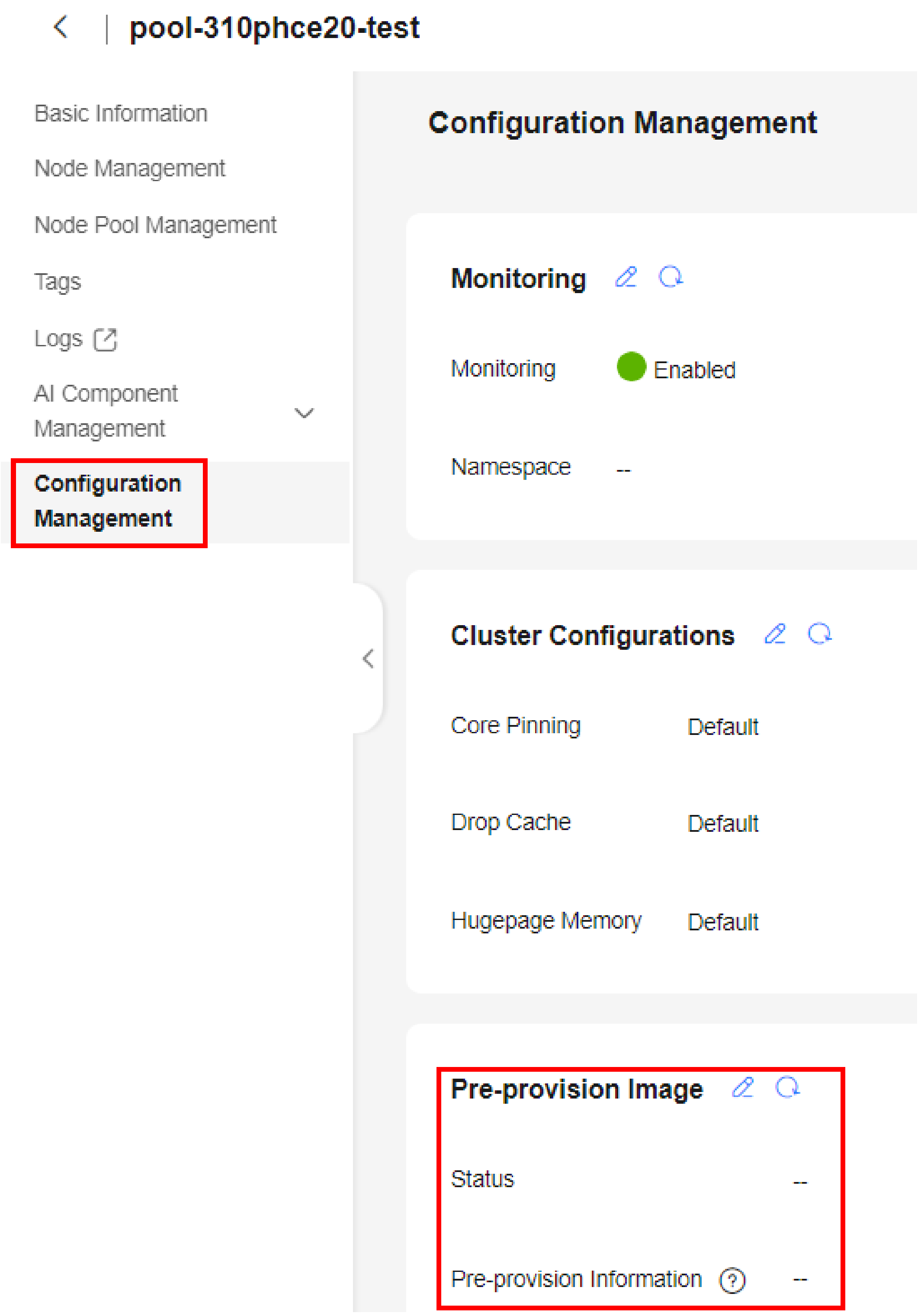
Lite Cluster resource pools enable image pre-provisioning, which pulls images from nodes in the pools beforehand, accelerating image pulling during inference and large-scale distributed training.

Procedure

Step 1 In the navigation pane of the ModelArts console, choose **Resource Management** > **AI Dedicated Resource Pools** > **Elastic Clusters**. In the **ModelArts Lite** tab, click the target resource pool to access its details page.

Step 2 Click **Configuration Management** on the left.

Figure 3-27 Configuration Management



Monitoring

Monitoring	● Enabled
Namespace	--

Cluster Configurations

Core Pinning	Default
Drop Cache	Default
Hugepage Memory	Default

Pre-provision Image

Status	--
Pre-provision Information	--

Step 3 In **Pre-provision Image**, click ✎ and configure parameters.

Table 3-3 Parameters

Parameter	Description
Image Source	Select **Preset** or **Custom**. ● **Preset**: Select an image on SWR or a shared image. ● **Custom**: Enter an image path.
Image Key	To pre-provision an image that you do not have permissions on, you will need to add an image key. Once enabled, select the namespace and key. For details about how to create a key, see **Creating a Secret**. The key type must be kubernetes.io/dockerconfigjson. To add multiple keys, click **+**.
Add	To add multiple images, click this button.

Figure 3-28 Pre-provisioning a preset image

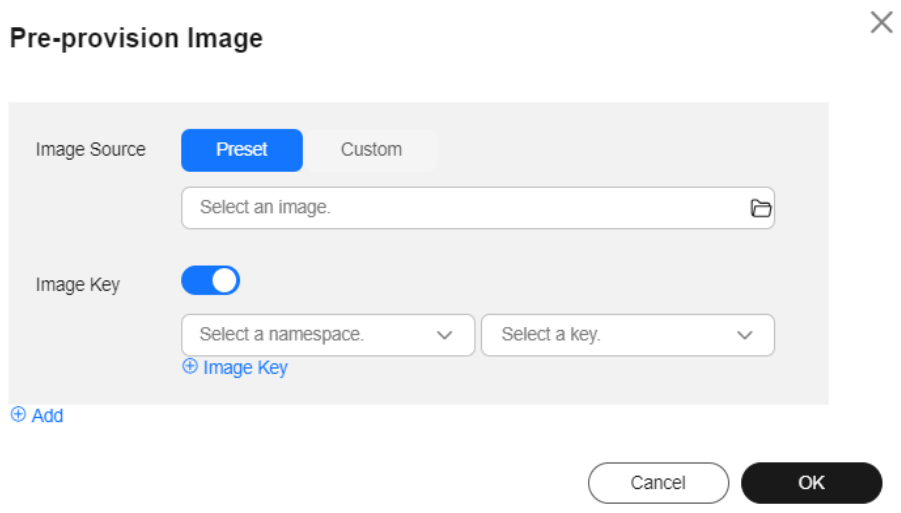
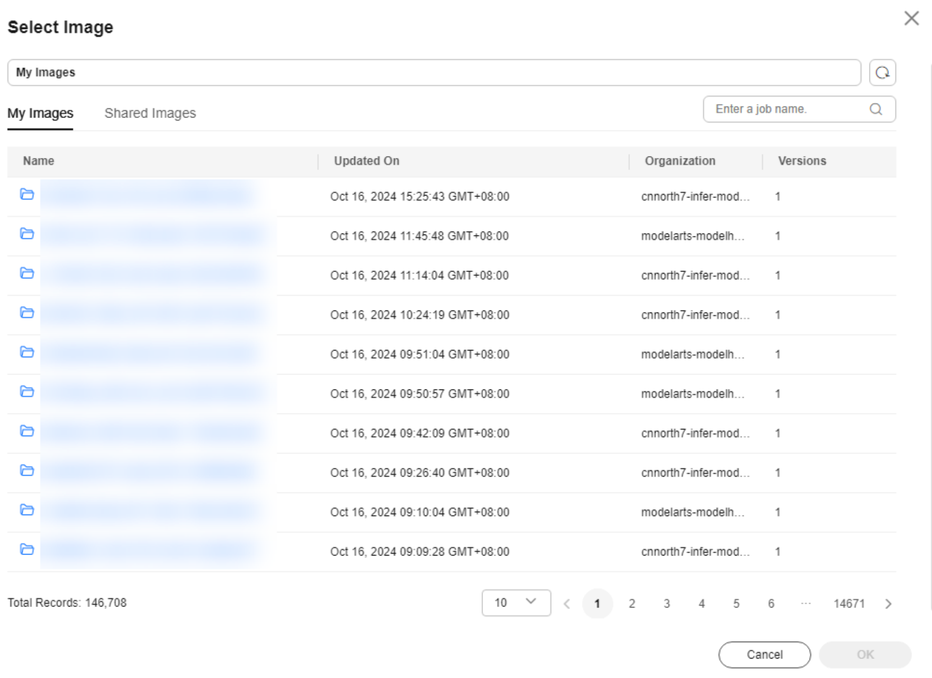


Figure 3-29 Selecting a preset image



Select Image

My Images

My Images | Shared Images

Enter a job name.

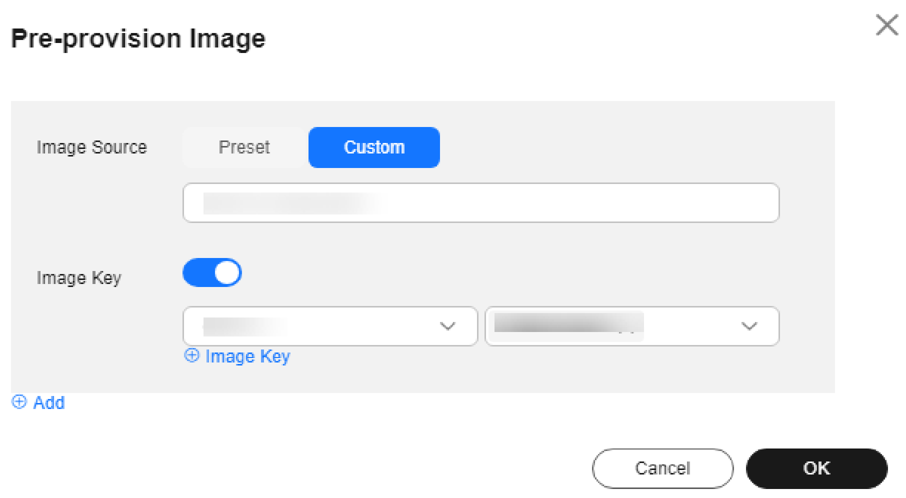
Name	Updated On	Organization	Versions
[redacted]	Oct 16, 2024 15:25:43 GMT+08:00	cnnorth7-infer-mod...	1
[redacted]	Oct 16, 2024 11:45:48 GMT+08:00	modelarts-modelh...	1
[redacted]	Oct 16, 2024 11:14:04 GMT+08:00	cnnorth7-infer-mod...	1
[redacted]	Oct 16, 2024 10:24:19 GMT+08:00	cnnorth7-infer-mod...	1
[redacted]	Oct 16, 2024 09:51:04 GMT+08:00	modelarts-modelh...	1
[redacted]	Oct 16, 2024 09:50:57 GMT+08:00	modelarts-modelh...	1
[redacted]	Oct 16, 2024 09:42:09 GMT+08:00	cnnorth7-infer-mod...	1
[redacted]	Oct 16, 2024 09:26:40 GMT+08:00	cnnorth7-infer-mod...	1
[redacted]	Oct 16, 2024 09:10:04 GMT+08:00	modelarts-modelh...	1
[redacted]	Oct 16, 2024 09:09:28 GMT+08:00	cnnorth7-infer-mod...	1

Total Records: 146,708

10 | 1 2 3 4 5 6 ... 14671

Cancel | OK

Figure 3-30 Pre-provisioning a custom image



Pre-provision Image

Image Source: Preset | **Custom**

[redacted]

Image Key: on

[redacted] | [redacted]

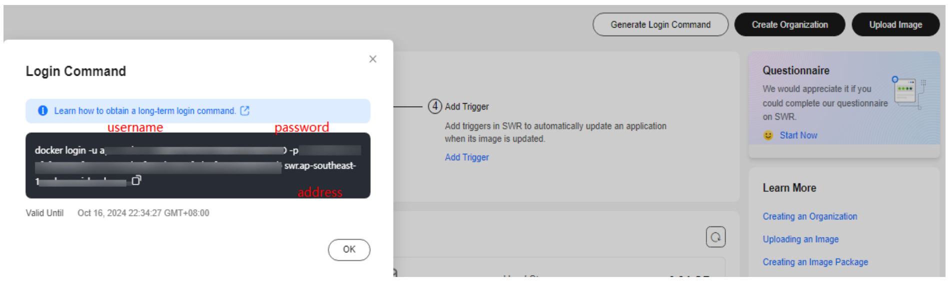
⊕ Image Key

⊕ Add

Cancel | OK

To create a key, refer to the tenant's SWR login command for the repository address, username, and password.

Figure 3-31 Login Command



Login Command

Learn how to obtain a long-term login command.

```
docker login -u a[redacted] -p
[redacted] swr.ap-southeast-
1[redacted]
```

Valid Until Oct 16, 2024 22:34:27 GMT+08:00

④ Add Trigger

Add triggers in SWR to automatically update an application when its image is updated.

Questionnaire

We would appreciate it if you could complete our questionnaire on SWR.

NOTE

The preceding figure shows a temporary login command. To obtain a long-term valid login command, click **Learn how to obtain a long-term login command**.

Step 4 Click **OK**. Then, you can see the information about the image that is pre-provisioned.

NOTE

If pre-provisioning an image failed, check whether the image path and key are correct.

----End

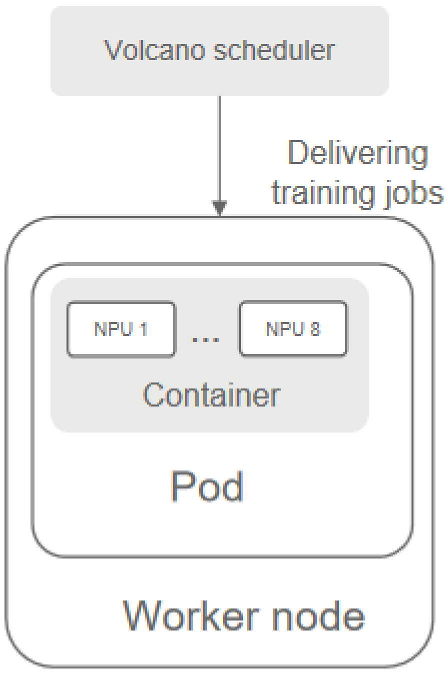
4 Using Lite Cluster Resources

4.1 Using Snt9B for Distributed Training in a Lite Cluster Resource Pool

Description

This case guides you through distributed training on Snt9B. By default, Lite Cluster resource pools come with the volcano scheduler, which delivers training jobs to clusters in volcano job mode. The BERT NLP model is used in the training test cases.

Figure 4-1 Delivering training jobs



Procedure

Step 1 Pull the image. The test image is bert_pretrain_mindspore:v1, which contains the test data and code.

```
docker pull swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
docker tag swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
bert_pretrain_mindspore:v1
```

Step 2 Create the **config.yaml** file on the host.

Configure Pods using this file. For debugging, start a Pod with the **sleep** command. Alternatively, replace the command with the boot command for your job (for example, **python train.py**). The job will run once the container starts.

The file content is as follows:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap1980-yourvcjobname     #The prefix is configmap1980-, followed by the vcjob name.
  namespace: default                      #Namespace, which is optional and must be in the same namespace as
vcjob.
  labels:
    ring-controller.cce: ascend-1980   # Retain the default settings.
data:             # The data content remains unchanged. After the initialization is complete, the data content is
automatically modified by the Volcano plug-in.
  jobstart_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: yourvcjobname                  # Job name, which must be the same as that in configmap.
namespace: default             # The value must be the same as that of ConfigMap.
  labels:
    ring-controller.cce: ascend-1980        # Retain the default settings.
    fault-scheduling: "force"
spec:
  minAvailable: 1                        # The value of minAvailable is 1 in a single-node scenario and N in an N-
node distributed scenario.
schedulerName: volcano         # Retain the default settings. Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    configmap1980:
    - --rank-table-version=v2           # Retain the default settings. The ranktable file of the v2 version is
generated.
    env: []
    svc:
    - --publish-not-ready-addresses=true
  maxRetry: 3
  queue: default
  tasks:
  - name: "yourvcjobname-1"
    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: mindspore
ring-controller.cce: ascend-1980       # Retain the default value. The value must be the same as the label in
ConfigMap and cannot be changed.
      spec:
        affinity:
          podAntiAffinity:
            requiredDuringSchedulingIgnoredDuringExecution:
              - labelSelector:
                  matchExpressions:
                    - key: volcano.sh/job-name
                      operator: In
                      values:
                        - yourvcjobname
                  topologyKey: kubernetes.io/hostname
        containers:
- image: bert_pretrain_mindspore:v1        # Training framework image path, which can be modified.
          imagePullPolicy: IfNotPresent
          name: mindspore
          env:
          - name: name                               # The value must be the same as that of Jobname.
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: ip                                   # IP address of the physical node, which is used to identify the
node where the pod is running
            valueFrom:
              fieldRef:
                fieldPath: status.hostIP
          - name: framework
            value: "MindSpore"
          command:
          - "sleep"
          - "1000000000000000000"
          resources:
            requests:
huawei.com/ascend-1980: "1"      # Number of required NPUs. The maximum value is 16. You can add lines
below to configure resources such as memory and CPU. The key remains unchanged.
            limits:
huawei.com/ascend-1980: "1"        # Limits the number of cards. The key remains unchanged. The value
must be consistent with that in requests.
          volumeMounts:
          - name: ascend-driver               # Mount driver. Retain the settings.
            mountPath: /usr/local/Ascend/driver
          - name: ascend-add-ons           # Mount driver. Retain the settings.
            mountPath: /usr/local/Ascend/add-ons
          - name: localtime
            mountPath: /etc/localtime
          - name: hccn                            #   HCCN configuration of the driver. Retain the settings.
            mountPath: /etc/hccn.conf
          - name: npu-smi                            #npu-smi
            mountPath: /usr/local/sbin/npu-smi
        nodeSelector:
          accelerator/huawei-npu: ascend-1980
        volumes:
        - name: ascend-driver
          hostPath:
            path: /usr/local/Ascend/driver
        - name: ascend-add-ons
          hostPath:
            path: /usr/local/Ascend/add-ons
        - name: localtime
          hostPath:
            path: /etc/localtime                      # Configure the Docker time.
        - name: hccn
          hostPath:
            path: /etc/hccn.conf
        - name: npu-smi
          hostPath:
            path: /usr/local/sbin/npu-smi
        restartPolicy: OnFailure
```

Step 3 Create a pod based on the **config.yaml** file.

```
kubectl apply -f config.yaml
```

Step 4 Run the following command to check the pod startup status. If **1/1 running** is displayed, the startup is successful.

```
kubectl get pod -A
```

Step 5 Go to the container, replace {pod_name} with your pod name (displayed by the **get pod** command), and replace {namespace} with your namespace (default).

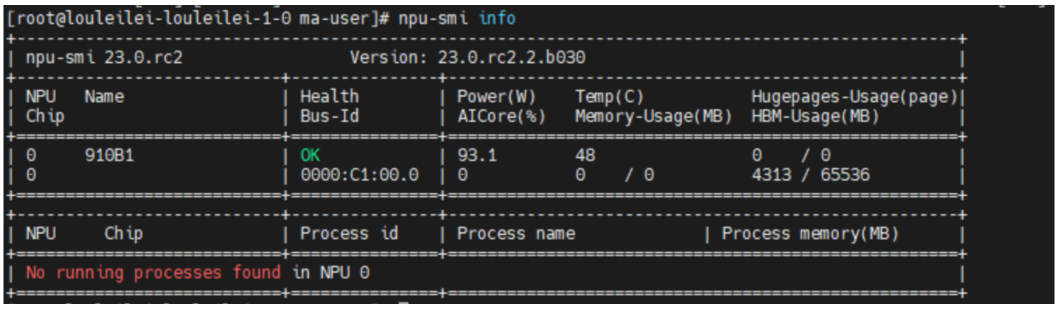
```
kubectl exec -it {pod_name} bash -n {namespace}
```

Step 6 Run the following command to view the NPU information:

```
npu-smi info
```

Kubernetes allocates resources to pods according to the number of NPUs specified in the **config.yaml** file. As illustrated in the figure below, only one NPU is displayed in the container, reflecting the single NPU configuration. This confirms that the configuration is effective.

Figure 4-2 Viewing NPU information



```
[root@louleilei-louleilei-1-0 ma-user]# npu-smi info
+------------------------------------------------------------------------------------------------+
| npu-smi 23.0.rc2                 Version: 23.0.rc2.2.b030                                      |
+---------------------------+---------------+----------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+====================================================+
| 0     910B1               | OK            | 93.1        48                0    / 0             |
| 0                         | 0000:C1:00.0  | 0           0    / 0          4313 / 65536         |
+===========================+===============+====================================================+
+---------------------------+---------------+----------------------------------------------------+
| NPU     Chip              | Process id    | Process name             | Process memory(MB)      |
+===========================+===============+====================================================+
| No running processes found in NPU 0                                                            |
+===========================+===============+====================================================+
```

Step 7 Change the number of NPUs in the pod. In this example, distributed training is used. The number of required NPUs is changed to 8.

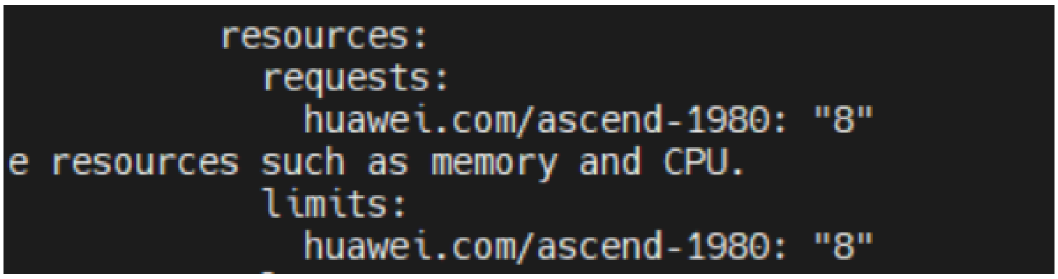
Delete the created pod.

```
kubectl delete -f config.yaml
```

Change the values of **limit** and **request** in the **config.yaml** file to 8.

```
vi config.yaml
```

Figure 4-3 Modify the number of NPUs



```
          resources:
            requests:
              huawei.com/ascend-1980: "8"
e resources such as memory and CPU.
            limits:
              huawei.com/ascend-1980: "8"
```

Re-create a pod.

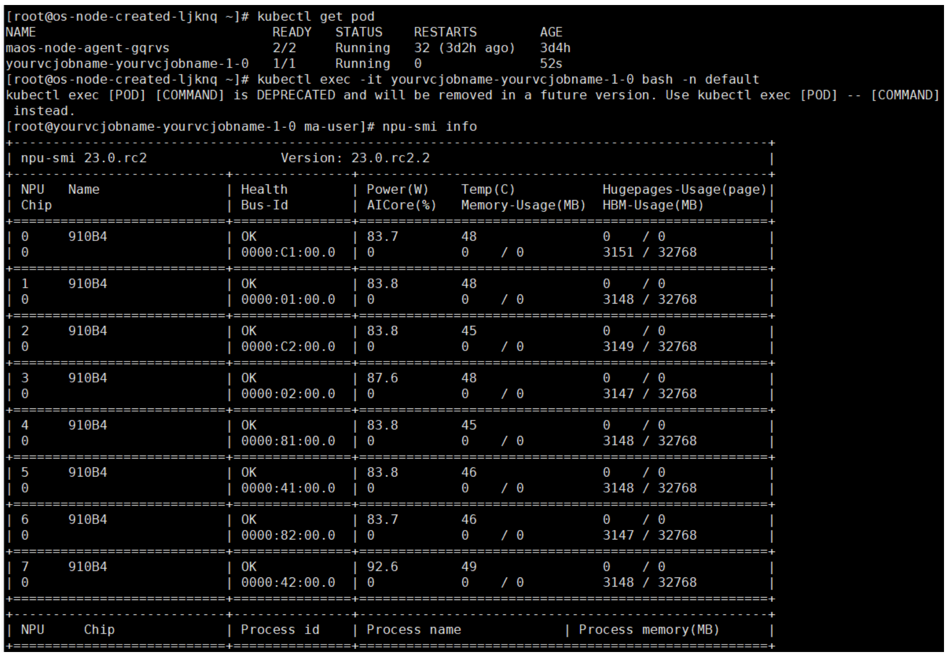
```
kubectl apply -f config.yaml
```

Go to the container and view the NPU information. Replace {pod_name} with your pod name and {namespace} with your namespace (default).

```
kubectl exec -it {pod_name} bash -n {namespace}
npu-smi info
```

As shown in the following figure, 8 NPUs are used and the pod is successfully configured.

Figure 4-4 Viewing NPU information



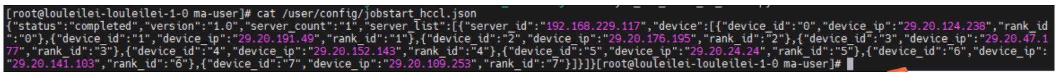
```
[root@os-node-created-ljknq ~]# kubectl get pod
NAME                                 READY   STATUS    RESTARTS        AGE
maos-node-agent-gqrvs                2/2     Running   32 (3d2h ago)   3d4h
yourvcjobname-yourvcjobname-1-0      1/1     Running   0               52s
[root@os-node-created-ljknq ~]# kubectl exec -it yourvcjobname-yourvcjobname-1-0 bash -n default
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND]
 instead.
[root@yourvcjobname-yourvcjobname-1-0 ma-user]# npu-smi info
+------------------------------------------------------------------------------------------------+
| npu-smi 23.0.rc2                 Version: 23.0.rc2.2                                           |
+---------------------------+---------------+----------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+====================================================+
| 0     910B4               | OK            | 83.7        48                0    / 0             |
| 0                         | 0000:C1:00.0  | 0           0    / 0          3151 / 32768         |
+===========================+===============+====================================================+
| 1     910B4               | OK            | 83.8        48                0    / 0             |
| 0                         | 0000:01:00.0  | 0           0    / 0          3148 / 32768         |
+===========================+===============+====================================================+
| 2     910B4               | OK            | 83.8        45                0    / 0             |
| 0                         | 0000:C2:00.0  | 0           0    / 0          3149 / 32768         |
+===========================+===============+====================================================+
| 3     910B4               | OK            | 87.6        48                0    / 0             |
| 0                         | 0000:02:00.0  | 0           0    / 0          3147 / 32768         |
+===========================+===============+====================================================+
| 4     910B4               | OK            | 83.8        45                0    / 0             |
| 0                         | 0000:81:00.0  | 0           0    / 0          3148 / 32768         |
+===========================+===============+====================================================+
| 5     910B4               | OK            | 83.8        46                0    / 0             |
| 0                         | 0000:41:00.0  | 0           0    / 0          3148 / 32768         |
+===========================+===============+====================================================+
| 6     910B4               | OK            | 83.7        46                0    / 0             |
| 0                         | 0000:82:00.0  | 0           0    / 0          3147 / 32768         |
+===========================+===============+====================================================+
| 7     910B4               | OK            | 92.6        49                0    / 0             |
| 0                         | 0000:42:00.0  | 0           0    / 0          3148 / 32768         |
+===========================+===============+====================================================+
+---------------------------+---------------+----------------------------------------------------+
| NPU     Chip              | Process id    | Process name             | Process memory(MB)      |
+===========================+===============+====================================================+
```

Step 8 Run the following command to view the inter-NPU communication configuration file:

```
cat /user/config/jobstart_hccl.json
```

During multi-NPU training, the **rank_table_file** configuration file is essential for inter-NPU communication. This file is automatically generated and provides the file address once the pod is initiated. It takes a period of time to generate the **/user/config/jobstart_hccl.json** and **/user/config/jobstart_hccl.json** configuration files. The service process can generate the inter-NPU communication information only after the status field in **/user/config/jobstart_hccl.json** is **completed**. The process is shown in the figure below.

Figure 4-5 Inter-NPU communication configuration file

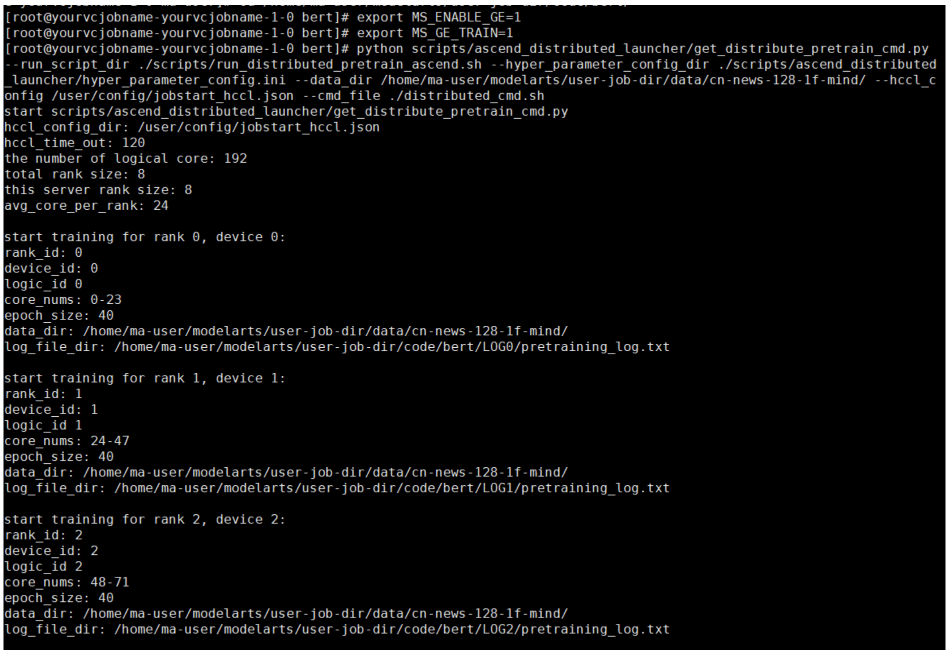


```
[root@louleilei-louleilei-1-0 ma-user]# cat /user/config/jobstart_hccl.json
{"status":"completed","version":"1.0","server_count":"1","server_list":[{"server_id":"192.168.229.117","device":[{"device_id":"0","device_ip":"29.20.124.238","rank_id
":"0"},{"device_id":"1","device_ip":"29.20.191.49","rank_id":"1"},{"device_id":"2","device_ip":"29.20.176.195","rank_id":"2"},{"device_id":"3","device_ip":"29.20.47.1
77","rank_id":"3"},{"device_id":"4","device_ip":"29.20.152.143","rank_id":"4"},{"device_id":"5","device_ip":"29.20.24.24","rank_id":"5"},{"device_id":"6","device_ip":
"29.20.141.103","rank_id":"6"},{"device_id":"7","device_ip":"29.20.109.253","rank_id":"7"}]}]}[root@louleilei-louleilei-1-0 ma-user]#
```

Step 9 Start a training job.

```
cd /home/ma-user/modelarts/user-job-dir/code/bert/
export MS_ENABLE_GE=1
export MS_GE_TRAIN=1
python scripts/ascend_distributed_launcher/get_distribute_pretrain_cmd.py --run_script_dir ./scripts/
run_distributed_pretrain_ascend.sh --hyper_parameter_config_dir ./scripts/ascend_distributed_launcher/
hyper_parameter_config.ini --data_dir /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/ --
hccl_config /user/config/jobstart_hccl.json --cmd_file ./distributed_cmd.sh
bash scripts/run_distributed_pretrain_ascend.sh /home/ma-user/modelarts/user-job-dir/data/cn-
news-128-1f-mind/ /user/config/jobstart_hccl.json
```

Figure 4-6 Starting a training job

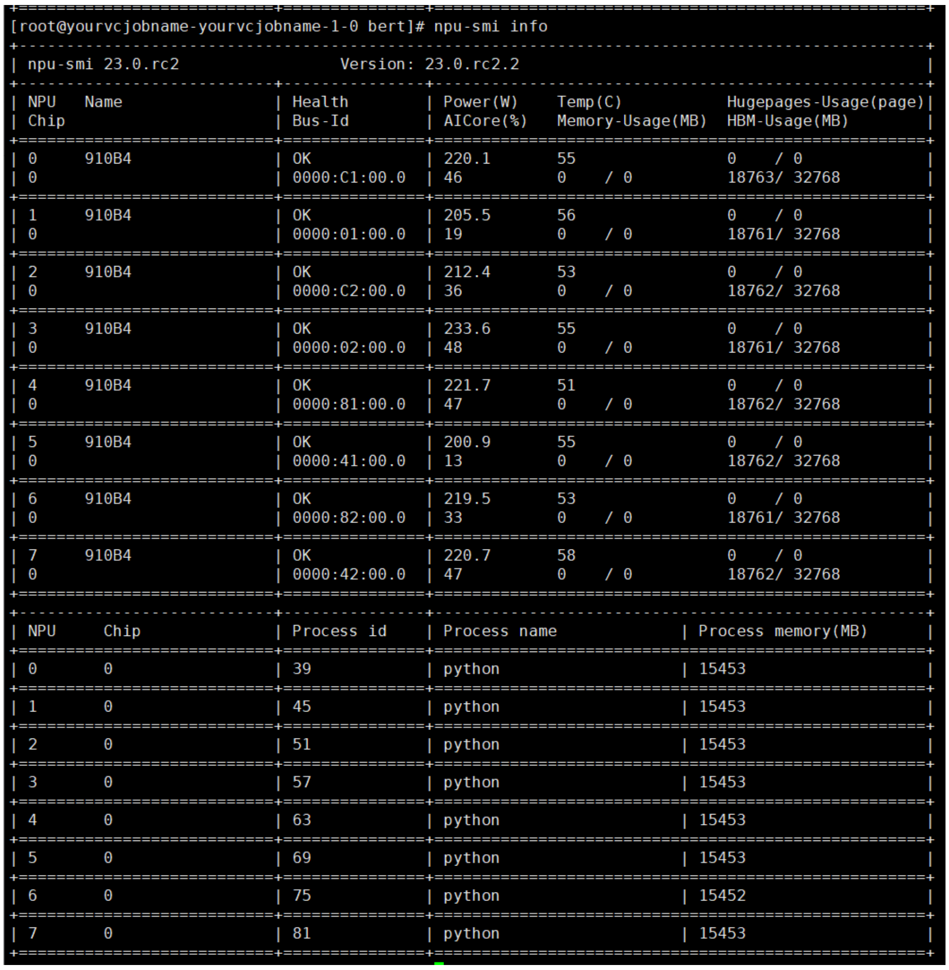


```
[root@yourvcjobname-yourvcjobname-1-0 bert]# export MS_ENABLE_GE=1
[root@yourvcjobname-yourvcjobname-1-0 bert]# export MS_GE_TRAIN=1
[root@yourvcjobname-yourvcjobname-1-0 bert]# python scripts/ascend_distributed_launcher/get_distribute_pretrain_cmd.py
--run_script_dir ./scripts/run_distributed_pretrain_ascend.sh --hyper_parameter_config_dir ./scripts/ascend_distributed
_launcher/hyper_parameter_config.ini --data_dir /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/ --hccl_c
onfig /user/config/jobstart_hccl.json --cmd_file ./distributed_cmd.sh
start scripts/ascend_distributed_launcher/get_distribute_pretrain_cmd.py
hccl_config_dir: /user/config/jobstart_hccl.json
hccl_time_out: 120
the number of logical core: 192
total rank size: 8
this server rank size: 8
avg_core_per_rank: 24

start training for rank 0, device 0:
rank_id: 0
device_id: 0
logic_id 0
core_nums: 0-23
epoch_size: 40
data_dir: /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
log_file_dir: /home/ma-user/modelarts/user-job-dir/code/bert/LOG0/pretraining_log.txt

start training for rank 1, device 1:
rank_id: 1
device_id: 1
logic_id 1
core_nums: 24-47
epoch_size: 40
data_dir: /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
log_file_dir: /home/ma-user/modelarts/user-job-dir/code/bert/LOG1/pretraining_log.txt

start training for rank 2, device 2:
rank_id: 2
device_id: 2
logic_id 2
core_nums: 48-71
epoch_size: 40
data_dir: /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
log_file_dir: /home/ma-user/modelarts/user-job-dir/code/bert/LOG2/pretraining_log.txt
```

It takes some time to load a training job. After several minutes, run the following command to view the NPU information. As shown in the following figure, all the eight NPUs are occupied, indicating that the training task is in progress.

```
npu-smi info
```

Figure 4-7 Viewing NPU information

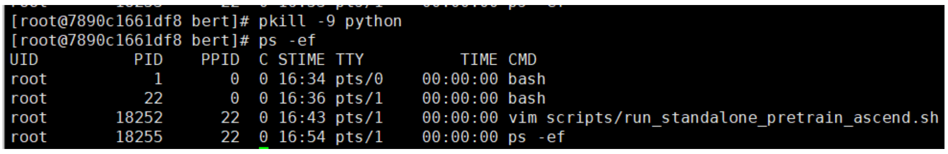


```
[root@yourvcjobname-yourvcjobname-1-0 bert]# npu-smi info
+------------------------------------------------------------------------------------------------+
| npu-smi 23.0.rc2                 Version: 23.0.rc2.2                                           |
+---------------------------+---------------+----------------------------------------------------+
| NPU   Name                | Health        | Power(W)    Temp(C)           Hugepages-Usage(page)|
| Chip                      | Bus-Id        | AICore(%)   Memory-Usage(MB)  HBM-Usage(MB)        |
+===========================+===============+====================================================+
| 0     910B4               | OK            | 220.1       55                0    / 0             |
| 0                         | 0000:C1:00.0  | 46          0    / 0          18763/ 32768         |
+===========================+===============+====================================================+
| 1     910B4               | OK            | 205.5       56                0    / 0             |
| 0                         | 0000:01:00.0  | 19          0    / 0          18761/ 32768         |
+===========================+===============+====================================================+
| 2     910B4               | OK            | 212.4       53                0    / 0             |
| 0                         | 0000:C2:00.0  | 36          0    / 0          18762/ 32768         |
+===========================+===============+====================================================+
| 3     910B4               | OK            | 233.6       55                0    / 0             |
| 0                         | 0000:02:00.0  | 48          0    / 0          18761/ 32768         |
+===========================+===============+====================================================+
| 4     910B4               | OK            | 221.7       51                0    / 0             |
| 0                         | 0000:81:00.0  | 47          0    / 0          18762/ 32768         |
+===========================+===============+====================================================+
| 5     910B4               | OK            | 200.9       55                0    / 0             |
| 0                         | 0000:41:00.0  | 13          0    / 0          18762/ 32768         |
+===========================+===============+====================================================+
| 6     910B4               | OK            | 219.5       53                0    / 0             |
| 0                         | 0000:82:00.0  | 33          0    / 0          18761/ 32768         |
+===========================+===============+====================================================+
| 7     910B4               | OK            | 220.7       58                0    / 0             |
| 0                         | 0000:42:00.0  | 47          0    / 0          18762/ 32768         |
+===========================+===============+====================================================+
+---------------------------+---------------+----------------------------------------------------+
| NPU     Chip              | Process id    | Process name             | Process memory(MB)      |
+===========================+===============+====================================================+
| 0       0                 | 39            | python                   | 15453                   |
+===========================+===============+====================================================+
| 1       0                 | 45            | python                   | 15453                   |
+===========================+===============+====================================================+
| 2       0                 | 51            | python                   | 15453                   |
+===========================+===============+====================================================+
| 3       0                 | 57            | python                   | 15453                   |
+===========================+===============+====================================================+
| 4       0                 | 63            | python                   | 15453                   |
+===========================+===============+====================================================+
| 5       0                 | 69            | python                   | 15453                   |
+===========================+===============+====================================================+
| 6       0                 | 75            | python                   | 15452                   |
+===========================+===============+====================================================+
| 7       0                 | 81            | python                   | 15453                   |
+===========================+===============+====================================================+
```

To stop a training task, run the commands below:

```
pkill -9 python
ps -ef
```

Figure 4-8 Stopping the training process



```
[root@7890c1661df8 bert]# pkill -9 python
[root@7890c1661df8 bert]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 16:34 pts/0    00:00:00 bash
root          22       0  0 16:36 pts/1    00:00:00 bash
root       18252      22  0 16:43 pts/1    00:00:00 vim scripts/run_standalone_pretrain_ascend.sh
root       18255      22  0 16:54 pts/1    00:00:00 ps -ef
```

NOTE

Set **limit** and **request** to proper values to restrict the number of CPUs and memory size. A single Snt9B node is equipped with eight Snt9B cards and 192u1536g. Properly plan the CPU and memory allocations to avoid task failures due to insufficient CPU and memory limits.

----End

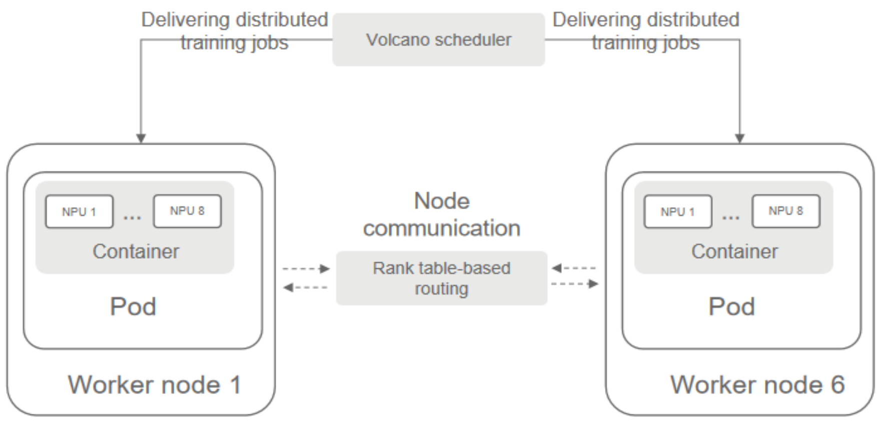
4.2 Performing PyTorch NPU Distributed Training In a ModelArts Lite Resource Pool Using Ranktable-based Route Planning

Description

The ranktable route planning is a communication optimization capability used in distributed parallel training. When NPUs are used, network route affinity planning can be performed for communication paths between nodes based on the actual switch topology, improving the communication speed between nodes.

This case describes how to complete a PyTorch NPU distributed training task in ModelArts Lite using ranktable route planning. By default, training tasks are delivered to the Lite resource pool cluster in Volcano job mode.

Figure 4-9 Job delivering



Constraints

- This function is available only in CN Southwest-Guiyang1. If you want to use it in another region, contact technical support.
- The Huawei Cloud Volcano plug-ins of 1.10.12 or later must be installed in the CCE cluster corresponding to the ModelArts Lite resource pool. For details about how to install and upgrade a Volcano scheduler, see **Volcano Scheduler**. Only Huawei Cloud Volcano plug-ins support route acceleration.
- Python 3.7 or 3.9 must be used for training. Otherwise, the ranktable route cannot be used for accelerating.
- There must be at least three task nodes in a training job. Otherwise, the ranktable route will be skipped. Use ranktable route in large model scenarios, that is, there are 512 cards or more.
- The script execution directory cannot be a shared directory. Otherwise, the ranktable route will fail.

- To use ranktable route is to change the rank number. Therefore, the rank in codes must be unified. Otherwise, the training will be abnormal.

Procedure

Step 1 Enable the cabinet plug-in of the CCE cluster corresponding to the ModelArts Lite resource pool.

1. In the ModelArts Lite dedicated resource pool list, click the resource pool name to view its details.
2. On the displayed page, click the CCE cluster.
3. In the navigation pane on the left, choose **Add-ons**, and search for **Volcano Scheduler**.
4. Click **Edit** and check whether **{"name":"cabinet"}** exists in the **plugins** parameter.
   - If **{"name":"cabinet"}** exists, go to **Step 2**.
   - If **{"name":"cabinet"}** does not exist, add it to the **plugins** parameter in the advanced settings, and click **Install**.

Step 2 Modify the **torch_npu** training startup script.

NOTICE

You can only run the **torch.distributed.launch/run** command to start up the script. Otherwise, the ranktable route cannot be used for accelerating.

During Pytorch training, you need to set **NODE_RANK** to the value of the environment variable **RANK_AFTER_ACC**. The following shows an example of a training startup script (*xxx*_**train.sh**): **MASTER_ADDR** and **NODE_RANK** must retain these values.

```
#!/bin/bash

# MASTER_ADDR
MASTER_ADDR="${MA_VJ_NAME}-${MA_TASK_NAME}-${MA_MASTER_INDEX}.${MA_VJ_NAME}"
NODE_RANK="$RANK_AFTER_ACC"
NNODES="$MA_NUM_HOSTS"
NGPUS_PER_NODE="$MA_NUM_GPUS"
# self-define, it can be changed to >=10000 port
MASTER_PORT="39888"

# replace ${MA_JOB_DIR}/code/torch_ddp.py to the actutal training script
PYTHON_SCRIPT=${MA_JOB_DIR}/code/torch_ddp.py
PYTHON_ARGS=""

# set hccl timeout time in seconds
export HCCL_CONNECT_TIMEOUT=1800

# replace ${ANACONDA_DIR}/envs/${ENV_NAME}/bin/python to the actual python
CMD="${ANACONDA_DIR}/envs/${ENV_NAME}/bin/python -m torch.distributed.launch \
    --nnodes=$NNODES \
    --node_rank=$NODE_RANK \
    --nproc_per_node=$NGPUS_PER_NODE \
    --master_addr $MASTER_ADDR \
    --master_port=$MASTER_PORT \
    $PYTHON_SCRIPT \
    $PYTHON_ARGS
"
```

```
echo $CMD
$CMD
```

Step 3 Create the **config.yaml** file on the host.

The **config.yaml** file is used to configure pods. The following shows a code example. ***xxxx_train.sh*** indicates the modified training startup script in **Step 2**.

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: yourvcjobname                  # Job name
  namespace: default            # Namespace
  labels:
    ring-controller.cce: ascend-1980   # Retain the default settings.
    fault-scheduling: "force"
spec:
  minAvailable: 6                       # Number of nodes used for distributed training
  schedulerName: volcano                # Retain the default settings.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    configmap1980:
    - --rank-table-version=v2           # Retain the default settings. The ranktable file of the v2 version is
generated.
    env: []
    svc:
    - --publish-not-ready-addresses=true   # Retain the default settings. It is used for the communication
between pods. Certain required environment variables are generated.
  maxRetry: 1
  queue: default
  tasks:
  - name: "worker"   # Retain the default settings.
    replicas: 6                           # Number of tasks, which is the number of nodes in PyTorch. Set this to
the value of minAvailable.
      template:
        metadata:
          annotations:
          cabinet: "cabinet"  # Retain the default settings. Enable tor-topo delivery.
          labels:
          app: pytorch-npu   # Tag
          ring-controller.cce: ascend-1980  # Retain the default settings.
        spec:
          affinity:
            podAntiAffinity:
              requiredDuringSchedulingIgnoredDuringExecution:
                - labelSelector:
                    matchExpressions:
                      - key: volcano.sh/job-name
                        operator: In
                        values:
                        - yourvcjobname   # Job name
                  topologyKey: kubernetes.io/hostname
          containers:
        - image:  swr.xxxxxx.com/xxxx/custom_pytorch_npu:v1             # Image address
              imagePullPolicy: IfNotPresent
          name: pytorch-npu          # Container name
              env:
          - name: OPEN_SCRIPT_ADDRESS     # Open script address. Set region-id based on the actual-life
scenario, for example, cn-southwest-2.
                  value: "https://mtest-bucket.obs.{region-id}.myhuaweicloud.com/acc/rank"
                - name: NAME
                  valueFrom:
                    fieldRef:
                      fieldPath: metadata.name
          - name: MA_CURRENT_HOST_IP                        # Retain the default settings. This indicates the IP
address of the node where the current pod is deployed.
                  valueFrom:
                    fieldRef:
```

```
                fieldPath: status.hostIP
          - name: MA_NUM_GPUS      # Number of NPUs used by each pod
              value: "8"
          - name: MA_NUM_HOSTS   # Number of nodes used in the distributed training. Set this to the value
of minAvailable.
              value: "6"
- name: MA_VJ_NAME               # Name of the volcano job.
              valueFrom:
                fieldRef:
                  fieldPath: metadata.annotations['volcano.sh/job-name']
- name: MA_TASK_NAME         # Name of the task.
              valueFrom:
                fieldRef:
                  fieldPath: metadata.annotations['volcano.sh/task-spec']
            command:
              - /bin/bash
              - -c
- Replace "wget ${OPEN_SCRIPT_ADDRESS}/bootstrap.sh -q && bash bootstrap.sh; export
RANK_AFTER_ACC=${VC_TASK_INDEX}; rank_acc=$(cat /tmp/RANK_AFTER_ACC 2>/dev/null); [ -n \"$
{rank_acc}\" ] && export RANK_AFTER_ACC=${rank_acc};export MA_MASTER_INDEX=$(cat /tmp/
MASTER_INDEX 2>/dev/null || echo 0); bash xxxx_train.sh"          # Replace xxxx_train.sh with the actual
training script path.
            resources:
              requests:
            huawei.com/ascend-1980: "8"                  # Number of cards required by each node. The key
remains the same. Set this to the value of MA_NUM_GPUS.
              limits:
            huawei.com/ascend-1980: "8"                  # Maximum number of cards on each node. The key
remains the same. Set this to the value of MA_NUM_GPUS.
            volumeMounts:
          - name: ascend-driver               #Mount driver. Retain the settings.
              mountPath: /usr/local/Ascend/driver
          - name: ascend-add-ons           #Mount driver. Retain the settings.
              mountPath: /usr/local/Ascend/add-ons
            - name: localtime
              mountPath: /etc/localtime
          - name: hccn                             # HCCN configuration of the driver. Retain the settings.
              mountPath: /etc/hccn.conf
            - name: npu-smi
              mountPath: /usr/local/sbin/npu-smi
          nodeSelector:
            accelerator/huawei-npu: ascend-1980
          volumes:
            - name: ascend-driver
              hostPath:
                path: /usr/local/Ascend/driver
            - name: ascend-add-ons
              hostPath:
                path: /usr/local/Ascend/add-ons
            - name: localtime
              hostPath:
                path: /etc/localtime
            - name: hccn
              hostPath:
                path: /etc/hccn.conf
            - name: npu-smi
              hostPath:
                path: /usr/local/sbin/npu-smi
          restartPolicy: OnFailure
```

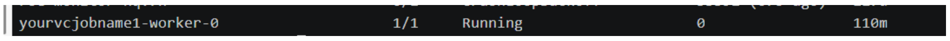
Step 4 Run the following command to create and start the pod based on **config.yaml**. After the container is started, the training job is automatically executed.

```
kubectl apply -f config.yaml
```

Step 5 Run the following command to check the pod startup status. If **1/1 running** is displayed, the startup is successful.

```
kubectl get pod
```

Figure 4-10 Command output of successful startup



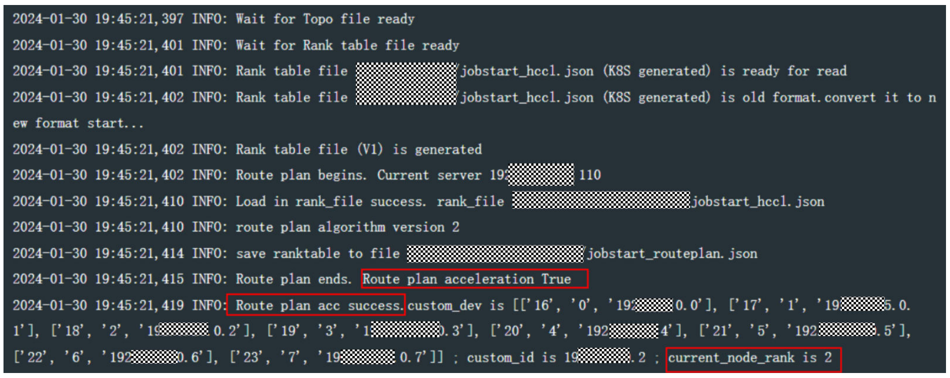
```
yourvcjobname1-worker-0                 1/1     Running                 0                   110m
```

Step 6 Run the following command to view the logs. If **Figure 4-11** is displayed, the route is executed.

```
kubectl logs {pod-name}
```

Replace *{pod-name}* with the actual pod name, which can be obtained from the output in **Step 5**.

Figure 4-11 Command output of executed dynamic route



```
2024-01-30 19:45:21,397 INFO: Wait for Topo file ready
2024-01-30 19:45:21,401 INFO: Wait for Rank table file ready
2024-01-30 19:45:21,401 INFO: Rank table file [redacted]/jobstart_hccl.json (K8S generated) is ready for read
2024-01-30 19:45:21,402 INFO: Rank table file [redacted]/jobstart_hccl.json (K8S generated) is old format.convert it to n
ew format start...
2024-01-30 19:45:21,402 INFO: Rank table file (V1) is generated
2024-01-30 19:45:21,402 INFO: Route plan begins. Current server 19[redacted]110
2024-01-30 19:45:21,410 INFO: Load in rank_file success. rank_file [redacted]jobstart_hccl.json
2024-01-30 19:45:21,410 INFO: route plan algorithm version 2
2024-01-30 19:45:21,414 INFO: save ranktable to file [redacted]/jobstart_routeplan.json
2024-01-30 19:45:21,415 INFO: Route plan ends. Route plan acceleration True
2024-01-30 19:45:21,419 INFO: Route plan acc success.custom_dev is [['16', '0', '19[redacted]0.0'], ['17', '1', '19[redacted]5.0.
1'], ['18', '2', '19[redacted]0.2'], ['19', '3', '1[redacted].3'], ['20', '4', '192[redacted]4'], ['21', '5', '192[redacted].5'],
['22', '6', '192[redacted]0.6'], ['23', '7', '19[redacted]0.7']] ; custom_id is 19[redacted].2 ; current_node_rank is 2
```

NOTE

- Dynamic routing can be executed only if there are at least three training nodes in a training task.
- If the execution fails, rectify the fault by referring to **Troubleshooting: ranktable Route Optimization Fails**.

----End

Troubleshooting: ranktable Route Optimization Fails

Symptom

There is error information in the container logs.

Possible Causes

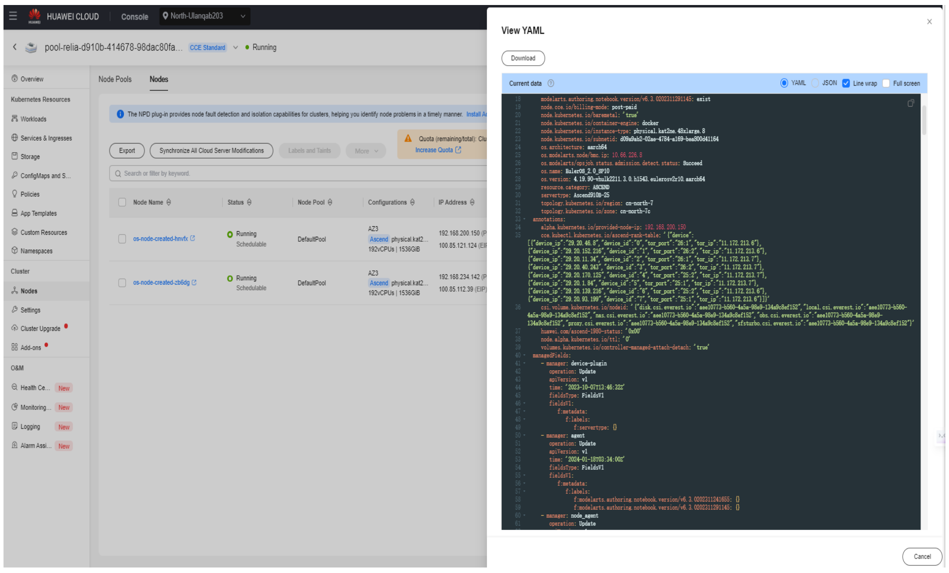
The cluster node does not deliver the **topo** file and **ranktable** file.

Procedure

1. In the ModelArts Lite dedicated resource pool list, click the resource pool name to view its details.
2. On the displayed page, click the CCE cluster.
3. In the navigation pane on the left, choose **Nodes**, and go to the **Nodes** tab.
4. In the node list, locate the target node, and choose **More** > **View YAML** in the **Operation** column.
5. Check whether the **cce.kubectl.kubernetes.io/ascend-rank-table** field in the **yaml** file has a value.

As shown in the following figure, if there is a value, delivering the **topo** file and **ranktable** file has been enabled on the node. Otherwise, contact technical support.

Figure 4-12 Viewing the **YAML** file of a node



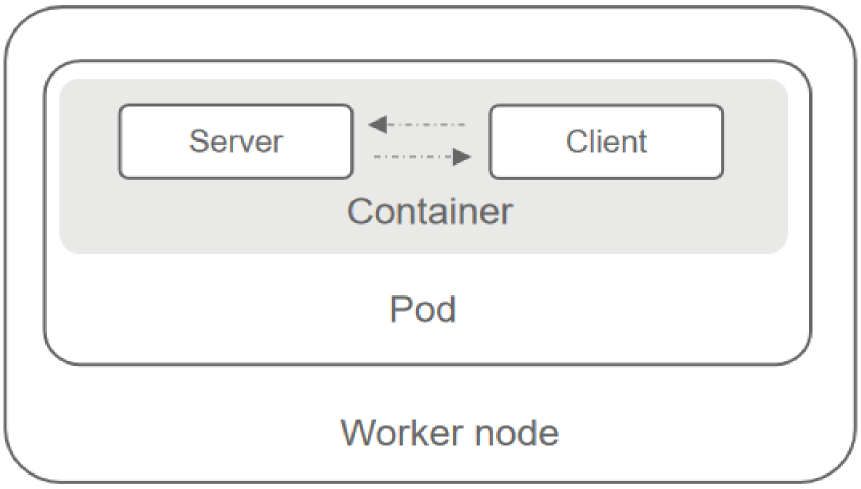
```
    modelarts.authoring.notebook.version/v6.3.0202311291145: exist
    node.cce.io/billing-mode: post-paid
    node.kubernetes.io/baremetal: 'true'
    node.kubernetes.io/container-engine: docker
    node.kubernetes.io/instance-type: physical.kat2ne.48xlarge.8
    node.kubernetes.io/subnetid: d09a9ab2-02ae-4784-a169-bea800d41164
    os.architecture: aarch64
    os.modelarts.node/bmc.ip: 10.66.226.8
    os.modelarts/opsjob.status.admission.detect.status: Succeed
    os.name: EulerOS_2.0_SP10
    os.version: 4.19.90-vhulk2211.3.0.h1543.eulerosv2r10.aarch64
    resource.category: ASCEND
    servertype: Ascend910B-25
    topology.kubernetes.io/region: cn-north-7
    topology.kubernetes.io/zone: cn-north-7c
  annotations:
    alpha.kubernetes.io/provided-node-ip: 192.168.200.150
    cce.kubectl.kubernetes.io/ascend-rank-table: '{"device":
[{"device_ip":"29.20.46.8","device_id":"0","tor_port":"26:1","tor_ip":"11.172.213.6"},
{"device_ip":"29.20.152.216","device_id":"1","tor_port":"26:2","tor_ip":"11.172.213.6"},
{"device_ip":"29.20.11.34","device_id":"2","tor_port":"26:1","tor_ip":"11.172.213.7"},
{"device_ip":"29.20.40.243","device_id":"3","tor_port":"26:2","tor_ip":"11.172.213.7"},
{"device_ip":"29.20.170.125","device_id":"4","tor_port":"25:2","tor_ip":"11.172.213.7"},
{"device_ip":"29.20.1.84","device_id":"5","tor_port":"25:1","tor_ip":"11.172.213.7"},
{"device_ip":"29.20.139.216","device_id":"6","tor_port":"25:2","tor_ip":"11.172.213.6"},
{"device_ip":"29.20.93.199","device_id":"7","tor_port":"25:1","tor_ip":"11.172.213.6"}]}'
    csi.volume.kubernetes.io/nodeid: '{"disk.csi.everest.io":"aee10773-b560-4a5a-98e9-134a9c8ef152","local.csi.everest.io":"aee10773-b560-4a5a-98e9-134a9c8ef152","nas.csi.everest.io":"aee10773-b560-4a5a-98e9-134a9c8ef152","obs.csi.everest.io":"aee10773-b560-4a5a-98e9-134a9c8ef152","proxy.csi.everest.io":"aee10773-b560-4a5a-98e9-134a9c8ef152","sfsturbo.csi.everest.io":"aee10773-b560-4a5a-98e9-134a9c8ef152"}'
    huawei.com/ascend-1980-status: '0x00'
    node.alpha.kubernetes.io/ttl: '0'
    volumes.kubernetes.io/controller-managed-attach-detach: 'true'
  managedFields:
    - manager: device-plugin
      operation: Update
      apiVersion: v1
      time: '2023-10-07T13:46:32Z'
      fieldsType: FieldsV1
      fieldsV1:
        f:metadata:
          f:labels:
            f:servertype: {}
    - manager: agent
      operation: Update
      apiVersion: v1
      time: '2024-01-18T03:34:00Z'
      fieldsType: FieldsV1
      fieldsV1:
        f:metadata:
          f:labels:
            f:modelarts.authoring.notebook.version/v6.3.0202311241655: {}
            f:modelarts.authoring.notebook.version/v6.3.0202311291145: {}
    - manager: node_agent
      operation: Update
```

4.3 Using Snt9B for Inference in a Lite Cluster Resource Pool

Description

This case outlines the process of using the Deployment mechanism to deploy a real-time inference service in the Snt9B environment. Create a pod to host the service, log in to the pod container to deploy the real-time service, and create a terminal as the client to access the service to test its functions.

Figure 4-13 Task diagram



Procedure

Step 1 Pulls the image. The test image is **bert_pretrain_mindspore:v1**, which contains the test data and code.

```
docker pull swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
docker tag swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
bert_pretrain_mindspore:v1
```

Step 2 Create the **config.yaml** file on the host.

Configure Pods using this file. For debugging, start a Pod with the **sleep** command. Alternatively, replace the command with the boot command for your job (for example, **python inference.py**). The job will run once the container starts.

The file content is as follows:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: yourapp
  labels:
     app: infers
spec:
  replicas: 1
  selector:
    matchLabels:
      app: infers
  template:
    metadata:
      labels:
         app: infers
    spec:
      schedulerName: volcano
      nodeSelector:
        accelerator/huawei-npu: ascend-1980
      containers:
      - image: bert_pretrain_mindspore:v1                  # Inference image name
        imagePullPolicy: IfNotPresent
        name: mindspore
        command:
        - "sleep"
        - "1000000000000000000"
        resources:
          requests:
huawei.com/ascend-1980: "1"      # Number of required NPUs. The maximum value is 16. You can add lines
below to configure resources such as memory and CPU. The key remains unchanged.
          limits:
huawei.com/ascend-1980: "1"        # Limits the number of cards. The key remains unchanged. The value
must be consistent with that in requests.
        volumeMounts:
          - name: ascend-driver               # Mount driver. Retain the settings.
          mountPath: /usr/local/Ascend/driver
          - name: ascend-add-ons           # Mount driver. Retain the settings.
          mountPath: /usr/local/Ascend/add-ons
          - name: hccn                            # HCCN configuration of the driver. Retain the settings.
          mountPath: /etc/hccn.conf
        - name: npu-smi                            #npu-smi
          mountPath: /usr/local/sbin/npu-smi
        - name: localtime                        #The container time must be the same as the host time.
          mountPath: /etc/localtime
      volumes:
      - name: ascend-driver
        hostPath:
          path: /usr/local/Ascend/driver
      - name: ascend-add-ons
        hostPath:
```

```
        path: /usr/local/Ascend/add-ons
    - name: hccn
      hostPath:
        path: /etc/hccn.conf
    - name: npu-smi
      hostPath:
        path: /usr/local/sbin/npu-smi
    - name: localtime
      hostPath:
        path: /etc/localtime
```

Step 3 Create a pod based on the **config.yaml** file.

```
kubectl apply -f config.yaml
```

Step 4 Run the following command to check the pod startup status. If **1/1 running** is displayed, the startup is successful.

```
kubectl get pod -A
```

Step 5 Go to the container, replace {pod_name} with your pod name (displayed by the **get pod** command), and replace {namespace} with your namespace (default).

```
kubectl exec -it {pod_name} bash -n {namespace}
```

Step 6 Activate the conda mode.

```
su - ma-user //Switch the user identity.
conda activate MindSpore //Activate the MindSpore environment.
```

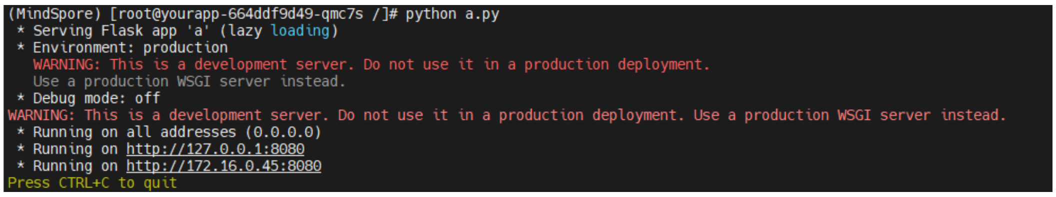
Step 7 Create test code **test.py**.

```
from flask import Flask, request
import json
app = Flask(__name__)

@app.route('/greet', methods=['POST'])
def say_hello_func():
    print("----------- in hello func ----------")
    data = json.loads(request.get_data(as_text=True))
    print(data)
    username = data['name']
    rsp_msg = 'Hello, {}!'.format(username)
    return json.dumps({"response":rsp_msg}, indent=4)

@app.route('/goodbye', methods=['GET'])
def say_goodbye_func():
    print("----------- in goodbye func ----------")
    return '\nGoodbye!\n'


@app.route('/', methods=['POST'])
def default_func():
    print("----------- in default func ----------")
    data = json.loads(request.get_data(as_text=True))
    return '\n called default func !\n {} \n'.format(str(data))

# host must be "0.0.0.0", port must be 8080
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=8080)
```

Execute the code. After the code is executed, a real-time service is deployed. The container is the server.

```
python test.py
```

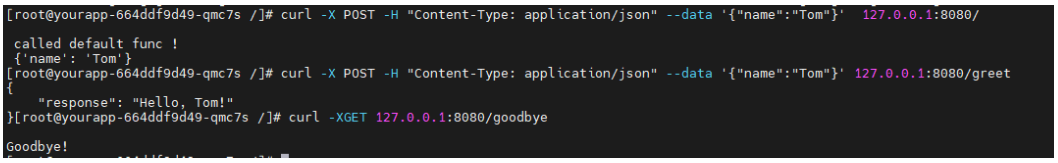
Figure 4-14 Deploying a real-time service


```
(MindSpore) [root@yourapp-664ddf9d49-qmc7s /]# python a.py
 * Serving Flask app 'a' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://172.16.0.45:8080
Press CTRL+C to quit
```

Step 8 Open a terminal in XShell and access the container (client) by referring to steps 5 to 7. Run the following commands to test the functions of the three APIs of the custom image. If the following information is displayed, the service is successfully invoked.

```
curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}'  127.0.0.1:8080/
curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet
curl -X GET 127.0.0.1:8080/goodbye
```

Figure 4-15 Accessing a real-time service


```
[root@yourapp-664ddf9d49-qmc7s /]# curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}'  127.0.0.1:8080/

 called default func !
 {'name': 'Tom'}
[root@yourapp-664ddf9d49-qmc7s /]# curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet
{
    "response": "Hello, Tom!"
}[root@yourapp-664ddf9d49-qmc7s /]# curl -XGET 127.0.0.1:8080/goodbye

Goodbye!
```

NOTE

Set **limit** and **request** to proper values to restrict the number of CPUs and memory size. A single Snt9B node is equipped with eight Snt9B cards and 192u1536g. Properly plan the CPU and memory allocations to avoid task failures due to insufficient CPU and memory limits.

----End

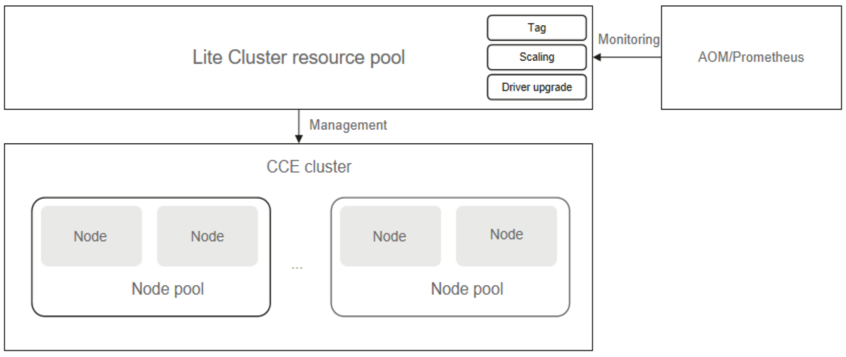
5 Managing Lite Server Resources

5.1 Managing Lite Cluster Resources

On the ModelArts console, you can manage created resources. You can click a resource pool name to access its details page and perform the following operations:

- **Managing Lite Cluster Resource Pools**: ModelArts allows you to manage resource pools, including renewing subscriptions, enabling or modifying auto-renewal, scaling resources, and upgrading drivers.
- **Managing Lite Cluster Node Pools**: To help you better manage nodes in a Kubernetes cluster, ModelArts provides node pools. A node pool is a group of nodes with the same configuration in a cluster. You can create, update, or delete node pools.
- **Managing Lite Cluster Nodes**: A node is a fundamental component of a container cluster. You can replace, delete, or reset a node within a resource pool. You can also delete, unsubscribe from, or renew nodes in batches.
- **Resizing a Lite Cluster Resource Pool**: The demand for resources in a Cluster resource pool may change due to the changes of AI development services. In this case, you can resize your resource pool in ModelArts.
- **Upgrading the Lite Cluster Resource Pool Driver**: If GPU or Ascend resources are used in a resource pool, you may need to customize GPU or Ascend drivers. ModelArts allows you to upgrade GPU or Ascend drivers of your dedicated resource pools.
- **Monitoring Lite Cluster Resources**: ModelArts leverages AOM and Prometheus to monitor resources, providing insights into resource usage.
- **Releasing Lite Cluster Resources**: You can release Lite Cluster resources that are no longer used.

Figure 5-1 Lite Cluster resource management



5.2 Managing Lite Cluster Resource Pools

Renewal Management of Lite Cluster Resource Pools

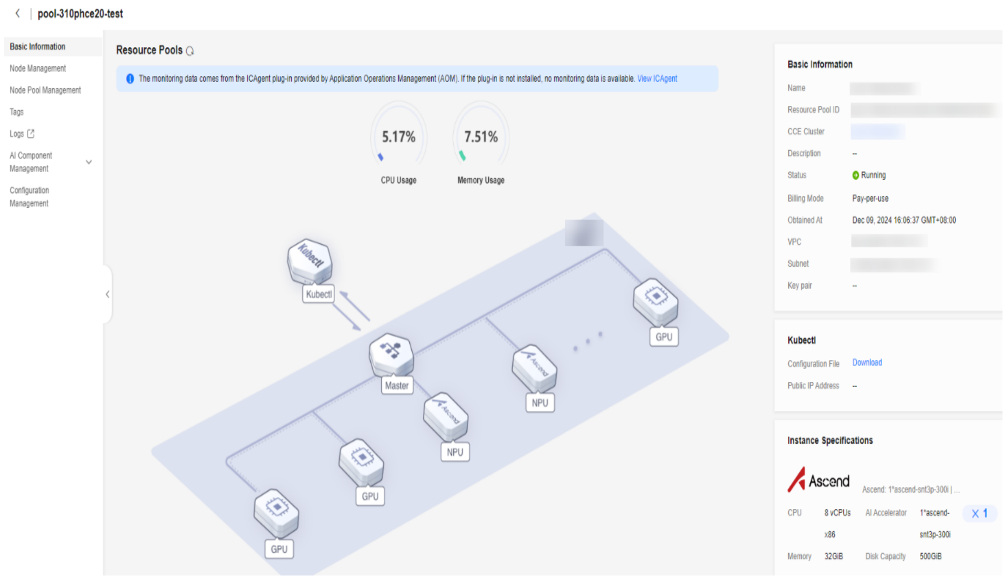
For yearly/monthly Lite Cluster resource pools, you can renew them, enable auto-renewal, and modify auto-renewal.

In the navigation pane of the ModelArts console, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. On the displayed page, perform the desired operations.

Viewing Basic Information About a Lite Cluster Resource Pool

In the navigation pane of the ModelArts console, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. On the displayed page, click the target resource pool to view more information.

Figure 5-2 Viewing basic information about a Lite Cluster resource pool



The monitoring data comes from the ICAgent plug-in provided by Application Operations Management (AOM). If the plug-in is not installed, no monitoring data is available. View ICAgent

Basic Information

Name	[redacted]
Resource Pool ID	[redacted]
CCE Cluster	[redacted]
Description	--
Status	Running
Billing Mode	Pay-per-use
Obtained At	Dec 09, 2024 16:06:37 GMT+08:00
VPC	[redacted]
Subnet	[redacted]
Key pair	--

Kubectl

Configuration File	Download
Public IP Address	--

Instance Specifications

Ascend   Ascend: 1*ascend-snt3p-300i | ...

CPU	8 vCPUs x86	AI Accelerator	1*ascend-snt3p-300i	X 1
Memory	32GiB	Disk Capacity	500GiB	

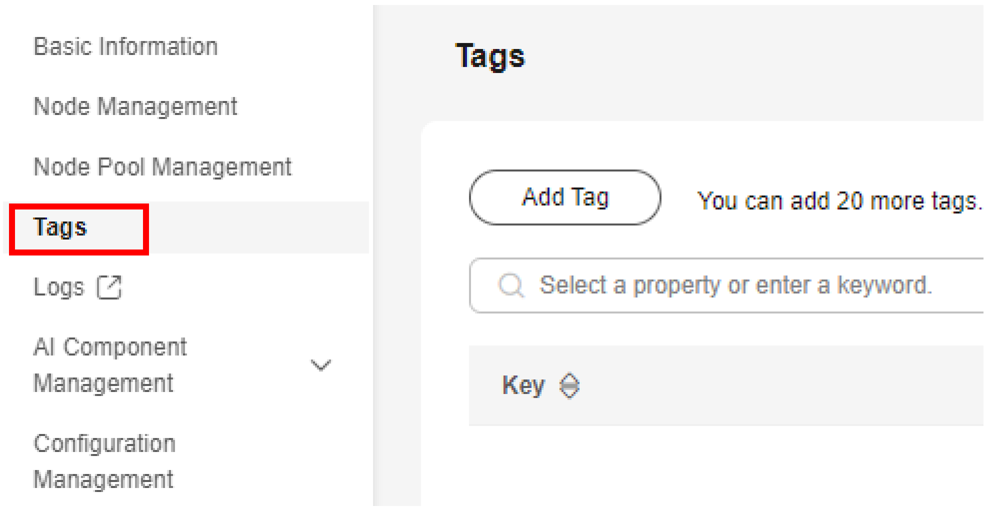
Managing Lite Cluster Resource Pool Tags

You can add tags to a resource pool for quick search.

1. Log in to the ModelArts console. In the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**.
2. In the Lite resource pool list, click the name of the target resource pool to view its details.
3. On the resource pool details page, click the **Tags** tab to view the tag information.

   Tags can be added, modified, and deleted. For details about how to use tags, see **Using TMS Tags to Manage Resources by Group**.

Figure 5-3 Tags



NOTE

You can add up to 20 tags.

Configuration Management of Lite Cluster Resource Pools

On the resource pool details page, click **Configuration Management**. From there, you can modify the namespace to be monitored, cluster configuration, and image pre-provisioning information.

- Click next to monitoring to enable or disable monitoring and set the namespace to be monitored. For details about how to use monitoring, see **Viewing Lite Cluster Metrics Using Prometheus**.
- Click next to cluster configuration to set core binding, dropcache, and hugepage memory parameters. If no value is set, the default value from the resource pool image will be used.
  – **Core Pinning**: If CPU pinning is enabled, workload pods exclusively use CPUs to improve performance (such as training and inference performance) and reduce the scheduling delay. This function is ideal for scenarios that are sensitive to CPU caching and scheduling delay. If CPU binding is disabled, exclusive CPUs will not be allocated to workload pods. Disable this function if you want a large pool of shareable CPUs.

You can also disable core binding and use **taskset** to flexibly bind cores in service containers.

– **Dropcache**: After this function is enabled, Linux cache clearing is enabled. This function can improve application performance in most scenarios. Clearing the cache can potentially lead to container startup failure or a degradation in system performance, as the system will need to reload data from the disk into memory. If this function is disabled, cache clearing is disabled.

– **Hugepage Memory**: When enabled, Transparent Huge Page (THP) is used. This memory management technique boosts system performance by increasing the memory page size. THP dynamically allocates huge page memory, simplifying its management. Enabling huge page memory can enhance application performance in most cases. However, it may trigger node restarts due to the soft lockup mechanism. If disabled, huge page memory is not used.

- Click for image pre-provisioning to set the image source, add an image key, and configure image pre-provisioning. For details, see **(Optional) Configuring Image Pre-provisioning**.

More Operations

For more operations, see the following:

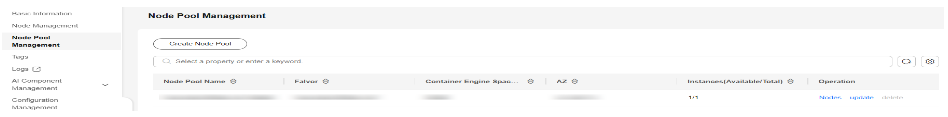
- Managing node pools: **Managing Lite Cluster Node Pools**
- Managing nodes: **Managing Lite Cluster Nodes**
- Resizing Lite Cluster resource pools: **Resizing a Lite Cluster Resource Pool**
- Upgrading the driver of a Lite Cluster resource pool: **Upgrading the Lite Cluster Resource Pool Driver**
- Upgrading the driver of a Lite Cluster resource pool node: **Upgrading the Driver of a Lite Cluster Resource Pool Node**

5.3 Managing Lite Cluster Node Pools

To help you better manage nodes in a Kubernetes cluster, ModelArts provides node pools. A node pool consists of one or more nodes, allowing you to set up a group of nodes with specific configurations.

On the resource pool details page, click the **Node Pools** tab to create, update, and delete node pools.

Figure 5-4 Node pool management



Node Pool Name	Falvor	Container Engine Spac...	AZ	Instances(Available/Total)	Operation
[redacted]	[redacted]	[redacted]	[redacted]	1/1	Nodes update delete

- Creating a node pool

  If you need more node pools, click **Create Node Pool** to create them. For details about the parameters, see **Step 6 Buying Lite Cluster Resources**.

- Viewing the node list

  To view information about nodes in a node pool, click **Nodes** in the **Operation** column to view the node name, specifications, and AZ.

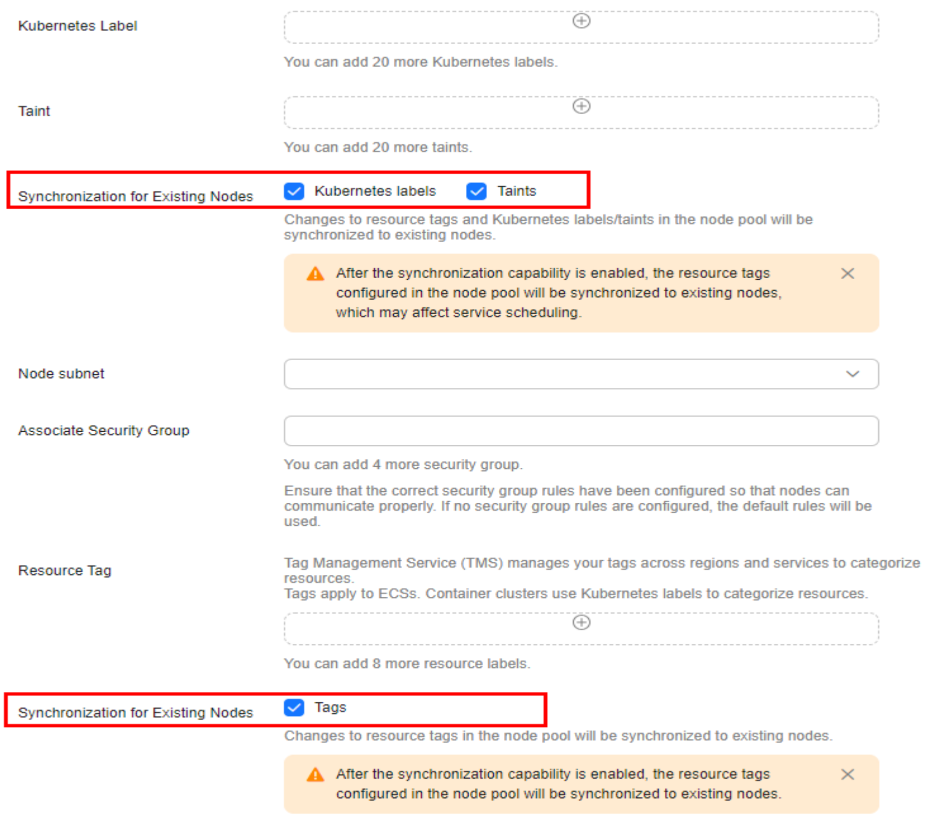
- Updating a node pool

  To update the configuration of a node pool, click **Update** in the **Operation** column. For details about the parameters, see **Step 6 Buying Lite Cluster Resources**.

  Note that when you update the node pool configuration, the advanced configuration takes effect only for new nodes. **Synchronization for Existing Nodes** (labels and taints) and **Synchronization for Existing Nodes** (labels) can be modified synchronously for existing nodes (by selecting the check boxes).

  The updated resource tag information in the node pool is synchronized to its nodes.

Figure 5-5 Updating a node pool



Kubernetes Label
You can add 20 more Kubernetes labels.
Taint
You can add 20 more taints.
Synchronization for Existing Nodes ✓ Kubernetes labels ✓ Taints
Changes to resource tags and Kubernetes labels/taints in the node pool will be synchronized to existing nodes.
After the synchronization capability is enabled, the resource tags configured in the node pool will be synchronized to existing nodes, which may affect service scheduling.
Node subnet
Associate Security Group
You can add 4 more security group.
Ensure that the correct security group rules have been configured so that nodes can communicate properly. If no security group rules are configured, the default rules will be used.
Resource Tag
Tag Management Service (TMS) manages your tags across regions and services to categorize resources.
Tags apply to ECSs. Container clusters use Kubernetes labels to categorize resources.
You can add 8 more resource labels.
Synchronization for Existing Nodes ✓ Tags
Changes to resource tags in the node pool will be synchronized to existing nodes.
After the synchronization capability is enabled, the resource tags configured in the node pool will be synchronized to existing nodes.

- Deleting a node pool

  If there are multiple node pools within a resource pool, you can delete them. To do so, click **Delete** in the **Operation** column, enter **DELETE**, and click **OK**.

  Each resource pool must have at least one node pool. If there is only one node pool in a resource pool, it cannot be deleted.

- Viewing the storage configuration of a node pool

  On the node pool update page, you can see details like disk type, size, quantity, write mode, container engine space size, and mount path for system, container, or data disks.

Additionally, on the Lite resource pool scaling page, you can view the storage configuration of its node pools.

- Searching for a node pool

  In the search box on the node pool management page, you can search for node pools by keyword, such as the node pool name, specifications, container engine space size, or AZ.

- Specifying node pool information to display

  On the node pool management page, click in the upper right corner to customize the information to display in the node pool list.

5.4 Managing Lite Cluster Nodes

Nodes are fundamental components of a container cluster. On the resource pool details page, click the **Nodes** tab to replace, delete, reset, or renew nodes. When you hover over a node name, the resource ID is displayed. You can use the resource ID to query bills or billing information of yearly/monthly resources in the Billing Center.

Deleting, Unsubscribing from, or Releasing a Node

- For a pay-per-use resource pool, click **Delete** in the **Operation** column.

  To delete nodes in batches, select the check boxes next to the node names, and click **Delete**.

- For a yearly/monthly resource pool whose resources are not expired, click **Unsubscribe** in the **Operation** column. You can unsubscribe from nodes in batches.
- For a yearly/monthly resource pool whose resources are expired (in the grace period), click **Release** in the **Operation** column. Nodes in the grace period cannot be released in batches.

  If the delete button is available for a yearly/monthly node, click the button to delete the node.

NOTE

- Before deleting, unsubscribing from, or releasing a node, ensure that there are no running jobs on this node. Otherwise, the jobs will be interrupted.
- If there are abnormal nodes in a resource pool, delete, unsubscribe from, or release these nodes and add new ones for substitution.
- If there is only one node, it cannot be deleted, unsubscribed from, or released.

Renewing a Subscription, Enabling Auto-Renewal, or Modifying Auto-Renewal

For yearly/monthly nodes, you can renew them, enable auto-renewal, and modify auto-renewal in the **Nodes** tab. You can also perform batch operations on nodes.

Replacing a Node

In the **Nodes** tab, locate the node to be replaced. Click **Replace** in the **Operation** column. No fee is charged for this operation.

Check the node replacement records on the **Records** page. **Running** indicates that the node is being replaced. After the replacement, you can check the new node in the node list.

The replacement can last no longer than 24 hours. If no suitable resource is found after the replacement times out, the status changes to **Failed**. Hover over ? to check the failure cause.

NOTE

- The number of replacements per day cannot exceed 20% of the total nodes in the resource pool. The number of nodes to replace cannot exceed 5% of the total nodes in the resource pool.
- Ensure that there are idle node resources. Otherwise, the replacement may fail.
- If there are any nodes in the **Resetting** state in the operation records, nodes in the resource pool cannot be replaced.

Resetting a Node

In the **Nodes** tab, locate the node you want to reset. Click **Reset** in the **Operation** column. You can also select the check boxes of multiple nodes and choose **More** > **Reset** above the node list to reset multiple nodes.

Configure the parameters described in the table below.

Table 5-1 Parameters

Parameter	Description
Operating system	Select an OS from the drop-down list box.
Configuration Mode	Select a configuration mode for resetting the node. • **By node percentage**: the maximum percentage of nodes that can be reset at a time • **By instance quantity**: the maximum number of nodes that can be reset at a time

Check the node reset records on the **Records** page. If a node is being reset, its status is **Resetting**. After the reset is complete, the node status changes to **Available**. Resetting a node will not be charged.

NOTE

- Resetting a node will impact the operation of related services. During the reset process, local disks and Kubernetes tags in the node will be cleared.
- Only nodes in the **Available** state can be reset.
- A single node can be in only one reset task at a time. Multiple reset tasks cannot be delivered for the same node at a time.
- If there are any nodes in the **Replacing** state in the operation records, nodes in the resource pool cannot be reset.
- When the driver of a resource pool is being upgraded, nodes in this resource pool cannot be reset.
- After resetting a node, there is a possibility that the driver for the GPU and NPU specifications will be upgraded.

Repairing a Node

NOTE

This is a whitelist function. Submit a service ticket for a trial as you need.

If a hardware fault occurs on a resource pool node, the repair button becomes available. You can click it to repair the node. After the node is repaired, the node status changes to **Available**.

Replacement and redeployment are supported.

Replacement: Replace the hardware to implement in-place repair, which takes a long time. Use this for non-local disk faults, so the local disk data is retained.

Redeployment: Replace the server to rectify the fault, which takes a short time. However, the local disk data will be lost.

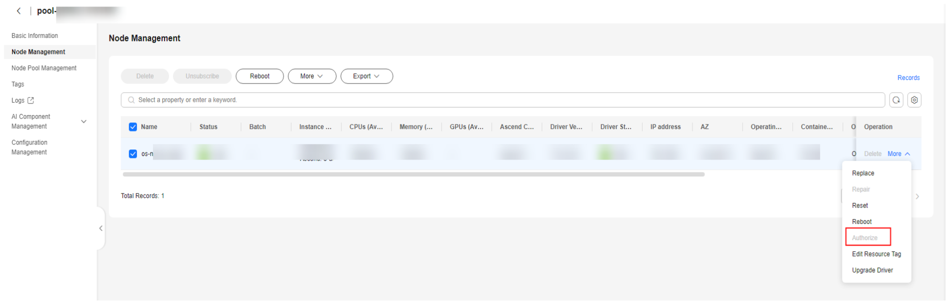
NOTE

- When a node is being repaired, it does not work. Ensure that there are no running services on the node. If the services cannot be stopped, do not repair the node. Contact technical support.
- If redeployment is selected, the instance will be stopped immediately and migrated to a new server, and the local disk data will be cleared. Migrate the service and back up the data beforehand.

Authorizing O&M Operations

During fault locating and performance diagnosis, you need to authorize certain O&M operations. To do so, go to the resource pool details page, click the **Nodes** tab, locate the target node, and choose **More** > **Authorize** in the **Operation** column. In the displayed dialog box, click **OK**.

Figure 5-6 Authorization



NOTE

The **Authorize** button is usually not available, but it will become available once Huawei technical support applies for O&M operations.

After the O&M process, Huawei technical support will disable the authorization, so you do not need to do anything else.

Restarting a Node

Locate the target node. Choose **More** > **Reboot** in the **Operation** column. You can also select node names and click **Reboot** above the node list to restart nodes in batches. Restarting a node will affect running services.

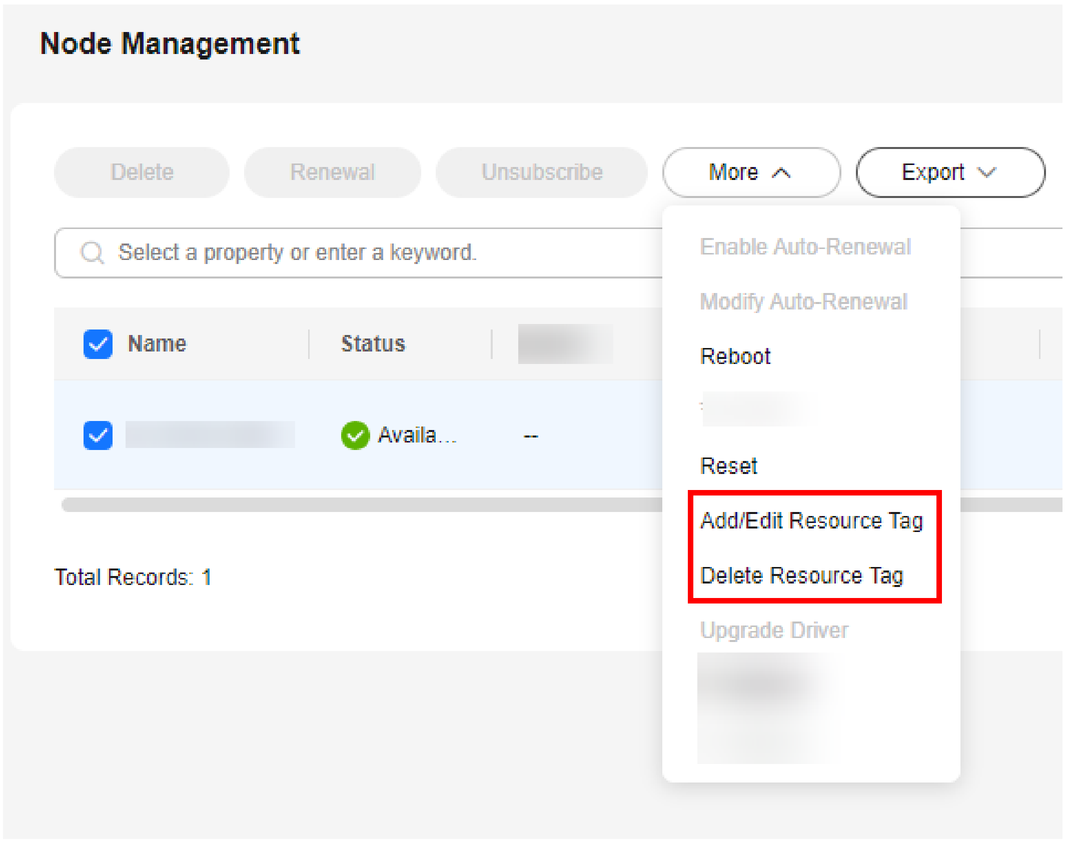
Adding, Editing, or Deleting Resource Tags

Use resource tags for easy billing management.

In the **Operation** column of the target node, choose **More** > **Edit Resource Tag**.

You can also select node names and choose **More** > **Add/Edit Resource Tag** or **Delete Resource Tag** above the node list to manage tags in batches.

Figure 5-7 Adding, editing, or deleting resource tags



Exporting Node Data

You can export the node information of a Lite resource pool as an Excel file.

Select the target nodes, choose **Export** > **Export All Data to XLSX** or **Export** > **Export Part Data to XLSX** above the node list, and click ⬇ in the browser to view the exported Excel file.

Upgrading a Driver

You can upgrade the driver version of a single node in a Lite resource pool or upgrade the driver versions of multiple nodes in batches. For details, see **Upgrading the Driver of a Lite Cluster Resource Pool Node**.

Searching for a Node

In the search box on the node management page, you can search for nodes by node name, status, batch, driver version, driver status, IP address, node pool, or resource tag.

Specifying Node Information to Display

On the node management page, click ⚙ in the upper right corner to customize the information to display in the node list.

5.5 Resizing a Lite Cluster Resource Pool

Description

The demand for resources in a Lite Cluster resource pool may change due to the changes of services. In this case, you can resize your resource pool as needed.

NOTE

Before scaling in a resource pool, ensure that there are no services running in the pool. Alternatively, switch to the details page of the target resource pool, delete the nodes where no services are running to scale in the pool. Otherwise, the services will be interrupted.

Notes and Constraints

- Only Lite Cluster resource pools in the **Running** state can be resized.
- When scaling in a Lite Cluster resource pool, the number of instances cannot be decreased to 0.
- Yearly/Monthly resources support only scale-out.

Procedure

1. Log in to the ModelArts console. In the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. Click the **ModelArts Lite** tab to view the resource pool list.
2. Click **Adjust Capacity** in the **Operation** column of the target resource pool. For a yearly/monthly resource pool, only **Scale Out** is displayed. To scale in the resource pool, go to its details page and unsubscribe from the resources.
3. On the dedicated resource pool scaling page, set **Target Instances** as needed. Scale-out is the process of increasing the number of target instances, while scale-in is the process of decreasing the number of target instances.

   If you purchase resource pool nodes by rack (supported by certain specifications), you will need to scale resources by rack too. The number of target instances equals the number of nodes times the number of racks. You can choose to purchase nodes by rack when creating a resource pool, but this cannot be changed when resizing the pool. Adjust the rack quantity to change the number of target instances.
4. In the **Resource Configurations** area, set **AZ** to **Automatic** or **Manual**.
   - If you select **Automatic**, nodes are randomly allocated to AZs after the scaling.
   - If you select **Manual**, you can allocate nodes to different AZs.
5. Resize the container engine space.

   When scaling out a resource pool, you can set the container engine space size of new nodes. This operation will cause inconsistencies in **dockerBaseSize** of nodes within the resource pool. As a result, some tasks may run differently on different nodes. The container engine space size cannot be changed for existing nodes.
6. Change the container engine type.

Container engine, one of the most important components of Kubernetes, manages the lifecycle of images and containers. kubelet interacts with a container engine through the Container Runtime Interface (CRI) to manage images and containers. Containerd has a shorter call chain, fewer components, and lower resource requirements, making it more stable. For details about the differences between Containerd and Docker, see **Container Engines**.

The CCE cluster version determines the available container engines. If it is earlier than 1.23, only Docker is supported. If it is 1.27 or later, only containerd is supported. For all other versions, both containerd and Docker are options.

7. Change the OS by selecting an OS version from the OS drop-down list.
8. Change the driver version by selecting a driver version from the driver version drop-down list.
9. Configure the node billing mode. When adding nodes, you can enable **Node Billing Mode** to change the billing mode, set the required duration, and enable auto-renewal. For example, you can create pay-per-use nodes in a yearly/monthly resource pool. If the billing mode is not specified, the new nodes shares the same billing mode with the resource pool.
10. Click **Submit** and then **OK** to complete the scaling.

5.6 Upgrading the Lite Cluster Resource Pool Driver

Description

If GPUs or Ascend resources are used in a dedicated resource pool, you may need to customize GPU or Ascend drivers. ModelArts allows you to upgrade GPU or Ascend drivers of your dedicated resource pools.

There are two driver upgrade modes: secure upgrade and forcible upgrade.

NOTE

- Secure upgrade: Running services are not affected. After the upgrade starts, the nodes are isolated (new jobs cannot be delivered). After the existing jobs on the nodes are complete, the upgrade is performed. The secure upgrade may take a long time because the jobs must be completed first.
- Forcible upgrade: The drivers are directly upgraded, regardless of whether there are running jobs.

Notes and Constraints

The target Lite Cluster resource pool must be running, and the resource pool contains GPU or Ascend resources.

Procedure

1. Log in to the ModelArts console. In the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. Click the **ModelArts Lite** tab to view the resource pool list.
2. In the resource pool list, locate the target resource pool, and choose ··· > **Upgrade Driver** in the **Operation** column.

3. In the displayed dialog box, you can view the driver type, number of instances, current version, target version, upgrade mode, upgrade scope, and rolling switch of the dedicated resource pool.
   - **Target Version**: Select a target driver version from the drop-down list.
   - Upgrade mode: You can select secure upgrade or forcible upgrade.
     - Secure upgrade: Perform the upgrade when no job is running on the node. The upgrade may take a long time.
     - Forcible upgrade: Ignore the running jobs and perform the upgrade directly. This may cause the running jobs to fail.
   - **Rolling Mode**: Once enabled, you can upgrade the driver in rolling mode. Currently, **By node percentage** and **By instance quantity** are supported.
     - **By node percentage**: The number of instances to be upgraded is the percentage multiplied by the total number of instances in the resource pool.
     - **By instance quantity**: You can set the number of instances to be upgraded in each batch.

     Different upgrade instances have different policies for selecting nodes.
     - Secure upgrade chooses nodes that are not running any services.
     - Forcible upgrade chooses nodes randomly.

     NOTE

     - To check if a node has any service, go to the resource pool details page. In the **Nodes** tab, ensure all GPUs and Ascend cards are available. If they are, the node has no services.
     - Nodes with abnormal drivers will be upgraded during a rolling upgrade, just like other nodes.
4. Click **OK** to start the driver upgrade.

5.7 Upgrading the Driver of a Lite Cluster Resource Pool Node

Description

If GPUs or Ascend resources are used in a Lite Cluster resource pool, you may need to customize GPU or Ascend drivers. ModelArts allows you to upgrade GPU or Ascend drivers of your Lite Cluster nodes.

Notes and Constraints

The target node driver must be running, and the resource pool contains GPU or Ascend resources.

Procedure

1. Log in to the ModelArts console. In the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. Click the **ModelArts Lite** tab to view the resource pool list.
2. Go to the resource pool details page. On the **Node Management** page, locate the node whose driver needs to be upgraded and choose **More** > **Upgrade Driver** in the **Operation** column.
3. In the displayed dialog box, select the target version.
4. Click **OK** to upgrade the node driver.

5.8 Managing Free Nodes in a Lite Cluster Resource Pool

Nodes that are not managed by the resource pool are considered as free nodes. To view the information about free nodes, log in to the ModelArts console, choose **AI Dedicated Resource Pools** > **Elastic Clusters**, and click the **Nodes** tab.

You can renew, unsubscribe from, enable or modify auto-renewal, add or edit resource tags, delete resource tags, and search for free nodes.

Renewing a Subscription, Enabling Auto-Renewal, or Modifying Auto-Renewal

For yearly/monthly nodes, you can renew them, enable auto-renewal, and modify auto-renewal in the **Nodes** tab. You can also perform batch operations on nodes.

Adding, Editing, or Deleting Resource Tags

Use resource tags for easy billing management.

Select the target nodes and click **Add/Edit Resource Label** or **Delete Resource Label** above the node list to manage resource tags.

Searching for a Node

In the search box on the node management page, you can search for nodes by node name, IP address, or resource tag.

Specifying Node Information to Display

On the node list page, click ⚙ in the upper right corner to customize the information to display.

Deleting, Unsubscribing from, or Releasing a Node

For details, see **Releasing a Free Node**.

5.9 Monitoring Lite Cluster Resources

5.9.1 Viewing Lite Cluster Metrics on AOM

ModelArts Lite Cluster regularly collects data on key resource usage for each node in a resource pool, including GPUs, NPUs, CPUs, and memory, and sends this information to AOM. You can view standard metrics on AOM or create custom metrics and send them to AOM for reporting.

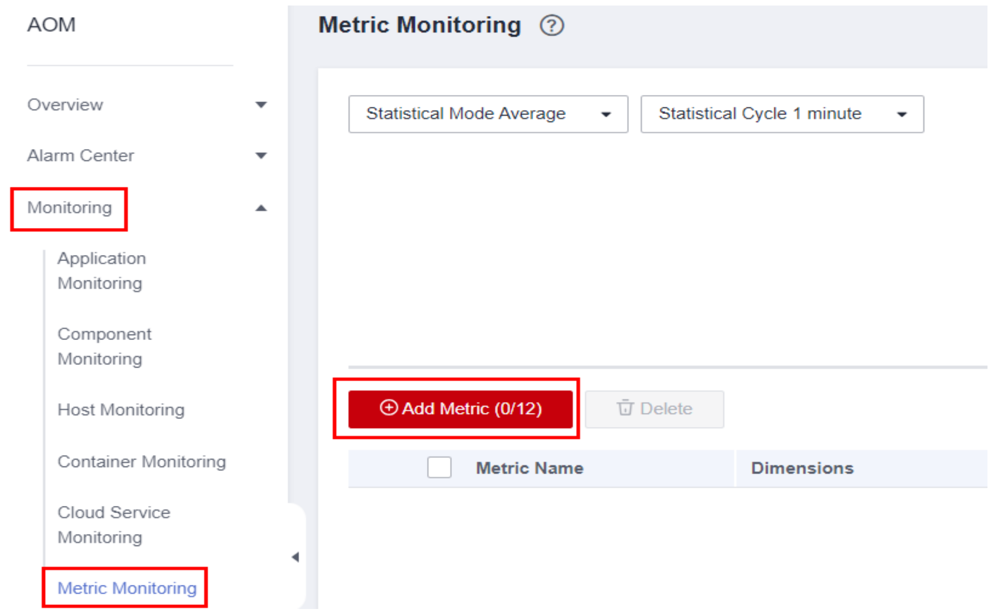
Additionally, you can install Prometheus on ModelArts Lite Cluster to collect metrics. For details, see **Viewing Lite Cluster Metrics Using Prometheus**.

This section describes how to view Lite Cluster metrics on AOM.

Viewing Existing Metrics on AOM

1. Log in to the console and search for **AOM** to go to the AOM console.
2. Choose **Monitoring** > **Metric Monitoring**. On the **Metric Monitoring** page that is displayed, click **Add Metric**.

Figure 5-8 Example



3. Add a metric for query.
   – **Add By**: Select **Dimension**.
   – **Metric Name**: Click **Custom Metrics** and select the desired ones for query. For details, see **Table 5-2** and **Table 5-3**.
   – Dimension: Enter the tag of the metric.
4. Click **Confirm**. The metric information is displayed.

Reporting Custom Metrics to AOM

ModelArts allows you to run commands to save custom metrics to AOM.

Constraints

- ModelArts invokes the commands or HTTP APIs specified in the custom configuration every 10 seconds to retrieve metric data.
- The size of the metric data text returned by these commands or HTTP APIs must not exceed 8 KB.

Collecting Custom Metric Data Using Commands

The following is an example of the YAML file for creating a pod for collecting custom metrics:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-task
  annotations:
ei.huaweicloud.com/metrics: '{"customMetrics":[{"containerName":"my-task","exec":{"command":["cat","/
metrics/task.prom"]}}]}'  # Replace the containerName and command parameters based on the container
from which metric data is obtained and the command used to obtain metric data.
spec:
  containers:
  - name: my-task
image: my-task-image:latest   # Replace it with the actual image.
```

To collect custom metrics, you can either run them alongside your service workload in the same container or use a separate sidecar container for this purpose. This keeps the service workload's resources unchanged.

Data Format of Custom Metrics

The format of custom metrics data must comply with the open metrics specifications. That is, the format of each metric must be:

```
<Metric name>{<Tag name>=<Tag value>, ...} <Sampled value>[Millisecond timestamp]
```

The following is an example (the comment starts with #, which is optional):

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total gauge
html_http_requests_total{method="post",code="200"} 1656 1686660980680
html_http_requests_total{method="post",code="400"} 2 1686660980681
```

Container-Level Metrics

Table 5-2 Container metrics

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
CPU	CPU Usage	ma_container_cpu_util	CPU usage of a measured object	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	Used CPU Cores	ma_container_cpu_used_core	Number of CPU cores used by a measured object	Core	≥ 0	N/A	N/A	N/A
	Total CPU Cores	ma_container_cpu_limit_core	Total number of CPU cores that have been applied for a measured object	Core	≥1	N/A	N/A	N/A
	CPU Memory Usage	ma_container_gpu_mem_util	Percentage of the used GPU memory to the total GPU memory	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
Memory	Total Physical Memory	ma_container_memory_capacity_megabytes	Total physical memory that has been applied for a measured object	MB	≥ 0	N/A	N/A	N/A
	Physical Memory Usage	ma_container_memory_util	Percentage of the used physical memory to the total physical memory	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	Used Physical Memory	ma_container_memory_used_megabytes	Physical memory that has been used by a measured object (**container_memory_working_set_bytes** in the current working set) (Memory usage in a working set = Active anonymous AND cache, and file-baked page ≤ **container_memory_usage_bytes**)	MB	≥ 0	N/A	N/A	N/A
Storage	Disk Read Rate	ma_container_disk_read_kilobytes	Volume of data read from a disk per second	KB/s	≥ 0	N/A	N/A	N/A
	Disk Write Rate	ma_container_disk_write_kilobytes	Volume of data written into a disk per second	KB/s	≥ 0	N/A	N/A	N/A
GPU memory	Total GPU Memory	ma_container_gpu_mem_total_megabytes	Total GPU memory of a training job	MB	>0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	GPU Memory Usage	ma_container_gpu_mem_util	Percentage of the used GPU memory to the total GPU memory	%	0%–100%	N/A	N/A	N/A
	Used GPU Memory	ma_container_gpu_mem_used_megabytes	GPU memory used by a measured object	MB	≥ 0	N/A	N/A	N/A
	Idle GPU Memory	ma_container_gpu_mem_free_megabytes	Idle GPU memory of a measured object	MB	≥ 0	N/A	N/A	N/A
GPU	GPU Usage	ma_container_gpu_util	GPU usage of a measured object	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	GPU Memory Bandwidth Usage	ma_container_gpu_mem_copy_util	GPU memory bandwidth usage of a measured object For example, the maximum memory bandwidth of NVIDIA GP Vnt1 is 900 GB/s. If the current memory bandwidth is 450 GB/s, the memory bandwidth usage is 50%.	%	0%–100%	N/A	N/A	N/A
	GPU Encoder Usage	ma_container_gpu_enc_util	GPU encoder usage of a measured object	%	%	N/A	N/A	N/A
	GPU Decoder Usage	ma_container_gpu_dec_util	GPU decoder usage of a measured object	%	%	N/A	N/A	N/A
	GPU Temperature	DCGM_FI_DEV_GPU_TEMP	GPU temperature	°C	Natural number	N/A	N/A	N/A
	GPU Power	DCGM_FI_DEV_POWER_USAGE	GPU power	Watt (W)	>0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	GPU Memory Temperature	DCGM_FI_DEV_MEMORY_TEMP	GPU memory temperature	°C	Natural number	N/A	N/A	N/A
Network I/O	Downlink rate	ma_container_network_receive_bytes	Inbound traffic rate of a measured object	Bytes/s	≥ 0	N/A	N/A	N/A
	Packet receive rate	ma_container_network_receive_packets	Number of data packets received by a NIC per second	Packets/s	≥ 0	N/A	N/A	N/A
	Downlink Error Rate	ma_container_network_receive_error_packets	Number of error packets received by a NIC per second	Packets/s	≥ 0	Raw data > 1 for two consecutive periods	Critical	Packet loss on the network. Submit a service ticket and contact the O&M support to locate the fault.
	Uplink rate	ma_container_network_transmit_bytes	Outbound traffic rate of a measured object	Bytes/s	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	Uplink Error Rate	ma_container_network_transmit_error_packets	Number of error packets sent by a NIC per second	Packets/s	≥ 0	Raw data > 1 for two consecutive periods	Critical	Packet loss on the network. Submit a service ticket and contact the O&M support to locate the fault.
	Packet send rate	ma_container_network_transmit_packets	Number of data packets sent by a NIC per second	Packets/s	≥ 0	N/A	N/A	N/A
NPU	NPU Usage	ma_container_npu_util	NPU usage of a measured object (To be replaced by **ma_container_npu_ai_core_util**)	%	0% – 100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NPU Memory Usage	ma_container_npu_memory_util	Percentage of the used NPU memory to the total NPU memory (To be replaced by **ma_container_npu_ddr_memory_util** for snt3 series, and **ma_container_npu_hbm_util** for snt9 series)	%	0%–100%	Raw data > 98% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	Used NPU Memory	ma_container_npu_memory_used_megabytes	NPU memory used by a measured object (To be replaced by **ma_container_npu_ddr_memory_usage_bytes** for snt3 series, and **ma_container_npu_hbm_usage_bytes** for snt9 series)	≥ 0	MB	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	Total NPU Memory	ma_container_npu_memory_total_megabytes	Total NPU memory of a measured object (To be replaced by **ma_container_npu_ddr_memory_bytes** for snt3 series, and **ma_container_npu_hbm_bytes** for snt9 series)	>0	MB	N/A	N/A	N/A
	Overall NPU Usage	ma_container_npu_general_util	NPU usage of Ascend AI processors (supported by driver version 24.1.RC2 and later)	%	0%–100%	N/A	N/A	N/A
AI Processor	AI Processor Error Codes	ma_container_npu_ai_core_error_code	Error codes of Ascend AI processors	N/A	N/A	Raw data > 0 for three consecutive periods	Critical	Abnormal card. Submit a service ticket and contact the O&M support.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor Health Status	ma_container_npu_ai_core_health_status	Health status of Ascend AI processors	N/A	• **1**: healthy • **0**: unhealthy	Raw data > 0 for two consecutive periods	Critical	Abnormal card. Submit a service ticket and contact the O&M support.
	AI Processor Power Consumption	ma_container_npu_ai_core_power_usage_watts	Power consumption of Ascend AI processors (processor power consumption for snt9 and snt3, and card power consumption for snt3P)	Watt (W)	>0	N/A	N/A	N/A
	AI Processor Temperature	ma_container_npu_ai_core_temperature_celsius	Temperature of Ascend AI processors	°C	Natural number	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Core Usage	ma_container_npu_ai_core_util	AI core usage of Ascend AI processors	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	AI Core Clock Frequency	ma_container_npu_ai_core_frequency_hertz	AI core clock frequency of Ascend AI processors	Hertz (Hz)	>0	N/A	N/A	N/A
	AI Processor Voltage	ma_container_npu_ai_core_voltage_volts	Voltage of Ascend AI processors	Volt (V)	Natural number	N/A	N/A	N/A
	AI Processor DDR Memory	ma_container_npu_ddr_memory_bytes	Total DDR memory capacity of Ascend AI processors	Byte	>0	N/A	N/A	N/A
	AI Processor DDR Usage	ma_container_npu_ddr_memory_usage_bytes	DDR memory usage of Ascend AI processors	Byte	>0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor DDR Memory Utilization	ma_container_npu_ddr_memory_util	DDR memory utilization of Ascend AI processors	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	AI Processor HBM Memory	ma_container_npu_hbm_bytes	Total HBM memory of Ascend AI processors (dedicated for Ascend snt9 processors)	Byte	>0	N/A	N/A	N/A
	AI Processor HBM Memory Usage	ma_container_npu_hbm_usage_bytes	HBM memory usage of Ascend AI processors (dedicated for Ascend snt9 processors)	Byte	>0	N/A	N/A	N/A
	AI Processor HBM Memory Utilization	ma_container_npu_hbm_util	HBM memory utilization of Ascend AI processors (dedicated for Ascend snt9 processors)	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor HBM Memory Bandwidth Utilization	ma_container_npu_hbm_bandwidth_util	HBM memory bandwidth utilization of Ascend AI processors (dedicated for Ascend snt9 AI processors)	%	0%–100%	Raw data > 95% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	AI Processor HBM Memory Clock Frequency	ma_container_npu_hbm_frequency_hertz	HBM memory clock frequency of Ascend AI processors (dedicated for Ascend snt9 processors)	Hertz (Hz)	>0	N/A	N/A	N/A
	AI Processor HBM Memory Temperature	ma_container_npu_hbm_temperature_celsius	HBM memory temperature of Ascend AI processors (dedicated for Ascend snt9 processors)	°C	Natural number	N/A	N/A	N/A
	AI CPU Utilization	ma_container_npu_ai_cpu_util	AI CPU utilization of Ascend AI processors	%	0%–100%	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor Control CPU Utilization	ma_container_npu_ctrl_cpu_util	Control CPU utilization of Ascend AI processors	%	0%–100%	N/A	N/A	N/A

Node-Level Metrics

Table 5-3 Node metric

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
CPU	Total CPU Cores	ma_node_cpu_limit_core	Total number of CPU cores that have been applied for a measured object	Core	≥1	N/A	N/A	N/A
	Used CPU Cores	ma_node_cpu_used_core	Number of CPU cores used by a measured object	Core	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	CPU Usage	ma_node_cpu_util	CPU usage of a measured object	%	0%–100%	Raw data > 95% for two consecutive periods	Major	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	CPU I/O Wait Time	ma_node_cpu_iowait_counter	Disk I/O wait time accumulated since system startup	jiffies	≥ 0	N/A	N/A	N/A
Memory	Physical Memory Usage	ma_node_memory_util	Percentage of the used physical memory to the total physical memory	%	0%–100%	Raw data > 95% for two consecutive periods	Major	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	Total Physical Memory	ma_node_memory_total_megabytes	Total physical memory that has been applied for a measured object	MB	≥ 0	N/A	N/A	N/A
Network I/O	Downlink Rate (BPS)	ma_node_network_receive_rate_bytes_seconds	Inbound traffic rate of a measured object	Bytes/s	≥ 0	N/A	N/A	N/A
	Uplink Rate (BPS)	ma_node_network_transmit_rate_bytes_seconds	Outbound traffic rate of a measured object	Bytes/s	≥ 0	N/A	N/A	N/A
Storage	Disk Read Rate	ma_node_disk_read_rate_kilobytes_seconds	Volume of data read from a disk per second (Only data disks used by containers are collected.)	KB/s	≥ 0	N/A	N/A	N/A
	Disk Write Rate	ma_node_disk_write_rate_kilobytes_seconds	Volume of data written into a disk per second (Only data disks used by containers are collected.)	KB/s	≥ 0	N/A	N/A	N/A
	Total Cache	ma_node_cache_space_capacity_megabytes	Total cache of the Kubernetes space	MB	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	Used Cache	ma_node_cache_space_used_capacity_megabytes	Used cache of the Kubernetes space	MB	≥ 0	N/A	N/A	N/A
	Cache Usage	ma_node_cache_space_used_percent	Cache usage of the Kubernetes space	%	≥ 0	Raw data > 90% for two consecutive periods	Critical	Check the disk in a timely manner to avoid affecting services. Clear invalid data on compute nodes.
	Total Container Space	ma_node_container_space_capacity_megabytes	Total container space	MB	≥ 0	N/A	N/A	N/A
	Used Container Space	ma_node_container_space_used_capacity_megabytes	Used container space	MB	≥ 0	N/A	N/A	N/A
	Container Space Usage	ma_node_container_space_used_percent	Space usage of a container	%	≥ 0	Raw data > 90% for two consecutive periods	Critical	Check the disk in a timely manner to avoid affecting services. Clear invalid data on compute nodes.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
GPU	GPU Usage	ma_node_gpu_util	GPU usage of a measured object	%	0%–100%	N/A	N/A	N/A
	Total GPU Memory	ma_node_gpu_mem_total_megabytes	Total GPU memory of a measured object	MB	>0	N/A	N/A	N/A
	GPU Memory Usage	ma_node_gpu_mem_util	Percentage of the used GPU memory to the total GPU memory	%	0%–100%	Raw data > 97% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.
	Used GPU Memory	ma_node_gpu_mem_used_megabytes	GPU memory used by a measured object	MB	≥ 0	N/A	N/A	N/A
	Tasks on a Shared GPU	node_gpu_share_job_count	Number of tasks running on a shared GPU	Number	≥ 0	N/A	N/A	N/A
	GPU Temperature	DCGM_FI_DEV_GPU_TEMP	GPU temperature	°C	Natural number	N/A	N/A	N/A
	GPU Power	DCGM_FI_DEV_POWER_USAGE	GPU power	Watt (W)	>0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	GPU Memory Temperature	DCGM_FI_DEV_MEMORY_TEMP	GPU memory temperature	°C	Natural number	N/A	N/A	N/A
NPU	NPU Usage	ma_node_npu_util	NPU usage of a measured object (To be replaced by **ma_node_npu_ai_core_util**)	%	0%–100%	N/A	N/A	N/A
	NPU Memory Usage	ma_node_npu_memory_util	Percentage of the used NPU memory to the total NPU memory (To be replaced by **ma_node_npu_ddr_memory_util** for snt3 series, and **ma_node_npu_hbm_util** for snt9 series)	%	0%–100%	Raw data > 97% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	Used NPU Memory	ma_node_npu_memory_used_megabytes	NPU memory used by a measured object (To be replaced by **ma_node_npu_ddr_memory_usage_bytes** for snt3 series, and **ma_node_npu_hbm_usage_bytes** for snt9 series)	MB	≥ 0	N/A	N/A	N/A
	Total NPU Memory	ma_node_npu_memory_total_megabytes	Total NPU memory of a measured object (To be replaced by **ma_node_npu_ddr_memory_bytes** for snt3 series, and **ma_node_npu_hbm_bytes** for snt9 series)	MB	>0	N/A	N/A	N/A
	AI Processor Error Codes	ma_node_npu_ai_core_error_code	Error codes of Ascend AI processors	N/A	N/A	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor Health Status	ma_node_npu_ai_core_health_status	Health status of Ascend AI processors	N/A	• **1**: healthy • **0**: unhealthy	The value is **0** for two consecutive periods.	Critical	Submit a service ticket.
	AI Processor Power Consumption	ma_node_npu_ai_core_power_usage_watts	Power consumption of Ascend AI processors (processor power consumption for snt9 and snt3, and card power consumption for snt3P)	Watt (W)	>0	N/A	N/A	N/A
	AI Processor Temperature	ma_node_npu_ai_core_temperature_celsius	Temperature of Ascend AI processors	°C	Natural number	N/A	N/A	N/A
	AI Processor Fan Speed	ma_node_npu_fan_speed_rpm	Fan speed of the Ascend series AI processors	RPM	Natural number	N/A	N/A	N/A
	AI Core Usage	ma_node_npu_ai_core_util	AI core usage of Ascend AI processors	%	0%–100%	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Core Clock Frequency	ma_node_npu_ai_core_frequency_hertz	AI core clock frequency of Ascend AI processors	Hertz (Hz)	>0	N/A	N/A	N/A
	AI Processor Voltage	ma_node_npu_ai_core_voltage_volts	Voltage of Ascend AI processors	Volt (V)	Natural number	N/A	N/A	N/A
	AI Processor DDR Memory	ma_node_npu_ddr_memory_bytes	Total DDR memory capacity of Ascend AI processors	Byte	>0	N/A	N/A	N/A
	AI Processor DDR Usage	ma_node_npu_ddr_memory_usage_bytes	DDR memory usage of Ascend AI processors	Byte	>0	N/A	N/A	N/A
	AI Processor DDR Memory Utilization	ma_node_npu_ddr_memory_util	DDR memory utilization of Ascend AI processors	%	0%–100%	Raw data > 90 % for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor HBM Memory	ma_node_npu_hbm_bytes	Total HBM memory of Ascend AI processors (dedicated for Ascend snt9 processors)	Byte	>0	N/A	N/A	N/A
	AI Processor HBM Memory Usage	ma_node_npu_hbm_usage_bytes	HBM memory usage of Ascend AI processors (dedicated for Ascend snt9 processors)	Byte	>0	N/A	N/A	N/A
	AI Processor HBM Memory Utilization	ma_node_npu_hbm_util	HBM memory utilization of Ascend AI processors (dedicated for Ascend snt9 processors)	%	0%–100%	Raw data > 97% for two consecutive periods	Suggestion	Check if the service resource usage meets the expectation. If the service is normal, no action is required.

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor HBM Memory Bandwidth Utilization	ma_node_npu_hbm_bandwidth_util	HBM memory bandwidth utilization of Ascend AI processors (dedicated for Ascend snt9 processors)	%	0%–100%	N/A	N/A	N/A
	AI Processor HBM Memory Clock Frequency	ma_node_npu_hbm_frequency_hertz	HBM memory clock frequency of Ascend AI processors (dedicated for Ascend snt9 processors)	Hertz (Hz)	>0	N/A	N/A	N/A
	AI Processor HBM Memory Temperature	ma_node_npu_hbm_temperature_celsius	HBM memory temperature of Ascend AI processors (dedicated for Ascend snt9 processors)	°C	Natural number	N/A	N/A	N/A
	AI CPU Utilization	ma_node_npu_ai_cpu_util	AI CPU utilization of Ascend AI processors	%	0%–100%	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	AI Processor Control CPU Utilization	ma_node_npu_ctrl_cpu_util	Control CPU utilization of Ascend AI processors	%	0%–100%	N/A	N/A	N/A
InfiniBand or RoCE network	Total Amount of Data Received by a NIC	ma_node_infiniband_port_received_data_bytes_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), received on all VLs from the port.	counting in double words, 32 bits	≥ 0	N/A	N/A	N/A
	Total Amount of Data Sent by a NIC	ma_node_infiniband_port_transmitted_data_bytes_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), transmitted on all VLs from the port.	counting in double words, 32 bits	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
NFS mounting status	NFS Getattr Congestion Time	ma_node_mountstats_getattr_backlog_wait	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permissions, owner, etc. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Getattr Round Trip Time	ma_node_mountstats_getattr_rtt	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permissions, owner, etc. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Access Congestion Time	ma_node_mountstats_access_backlog_wait	Access is an NFS operation that checks the access permissions of a file or directory for a given user. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Access Round Trip Time	ma_node_mountstats_access_rtt	Access is an NFS operation that checks the access permissions of a file or directory for a given user. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Lookup Congestion Time	ma_node_mountstats_lookup_backlog_wait	Lookup is an NFS operation that resolves a file name in a directory to a file handle. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Lookup Round Trip Time	ma_node_mountstats_lookup_rtt	Lookup is an NFS operation that resolves a file name in a directory to a file handle. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Read Congestion Time	ma_node_mountstats_read_backlog_wait	Read is an NFS operation that reads data from a file. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Read Round Trip Time	ma_node_mountstats_read_rtt	Read is an NFS operation that reads data from a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Write Congestion Time	ma_node_mountstats_write_backlog_wait	Write is an NFS operation that writes data to a file. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥ 0	N/A	N/A	N/A

Classification	Name	Metric	Description	Unit	Value Range	Alarm Threshold	Alarm Severity	Solution
	NFS Write Round Trip Time	ma_node_mountstats_write_rtt	Write is an NFS operation that writes data to a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0	N/A	N/A	N/A

Label Metrics

Table 5-4 Metric names

Classification	Metric	Description
Container metrics	pod_name	Name of the pod to which the container belongs

Classification	Metric	Description
	pod_id	ID of the pod to which the container belongs
	node_ip	IP address of the node to which the container belongs
	container_id	Container ID
	cluster_id	Cluster ID
	cluster_name	Cluster name
	container_name	Name of the container
	namespace	Namespace where the POD created by the user is located.
	app_kind	The value is obtained from the **kind** field in the first **ownerReferences**.
	app_id	The value is obtained from the **uid** field in the first **ownerReferences**.
	app_name	The value is obtained from the **name** field in the first **ownerReferences**.
	npu_id	Ascend card ID, for example, **davinci0** (to be discarded)
	device_id	Physical ID of Ascend AI processors
	device_type	Type of Ascend AI processors
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	pool_name	Name of a resource pool corresponding to a physical dedicated resource pool
	gpu_uuid	UUID of the GPU used by the container
	gpu_index	Index of the GPU used by the container
	gpu_type	Type of the GPU used by the container
Node metrics	cluster_id	ID of the CCE cluster to which the node belongs
	node_ip	IP address of the node
	host_name	Hostname of a node
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	project_id	Project ID of the user in a physical dedicated resource pool

Classification	Metric	Description
	npu_id	Ascend card ID, for example, **davinci0** (to be discarded)
	device_id	Physical ID of Ascend AI processors
	device_type	Type of Ascend AI processors
	gpu_uuid	UUID of a node GPU
	gpu_index	Index of a node GPU
	gpu_type	Type of a node GPU
	device_name	Device name of an InfiniBand or RoCE network NIC
	port	Port number of the IB NIC
	physical_state	Status of each port on the IB NIC
	firmware_version	Firmware version of the InfiniBand NIC
	filesystem	NFS-mounted file system
	mount_point	NFS mount point
Diagnos	cluster_id	ID of the CCE cluster to which the node with the GPU equipped belongs
	node_ip	IP address of the node where the GPU resides
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	project_id	Project ID of the user in a physical dedicated resource pool
	gpu_uuid	GPU UUID
	gpu_index	Index of a node GPU
	gpu_type	Type of a node GPU
	device_name	Device name of an InfiniBand or RoCE network NIC
	port	Port number of the IB NIC
	physical_state	Status of each port on the IB NIC
	firmware_version	Firmware version of the InfiniBand NIC

5.9.2 Viewing Lite Cluster Metrics Using Prometheus

Prometheus is an open-source monitoring tool. ModelArts supports the Exporter function, enabling you to use third-party monitoring systems like Prometheus to obtain metric data collected by ModelArts.

This section describes how to view Lite Cluster metrics using Prometheus.

Notes and Constraints

- You must enable the monitoring function on the configuration management page of the ModelArts Lite cluster resource pool details page.
- After this function is enabled, third-party components compatible with the Prometheus metric format can obtain the metric data collected by ModelArts through API **http://***<Node IP address>***:***<Port number>***/metrics**.
- Before enabling this function, you need to confirm the port number. It can be any number within the range of 10120 to 10139. Ensure that the selected port number is not being used by any other applications on each node.

Interconnecting Prometheus with ModelArts in Kubernetes

1. Use kubectl to connect to the target cluster. For details, see **Connecting to a Cluster Using kubectl**.
2. Configure Kubernetes access authorization.

   Use any text editor to create the **prometheus-rbac-setup.yml** file. The content of the YAML file is as follows:

   NOTE

   This YAML file defines the role (ClusterRole) for Prometheus and assigns the necessary access permissions. Additionally, it creates the account (ServiceAccount) for Prometheus and binds this account to the role (ClusterRoleBinding).

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus
rules:
- apiGroups: [""]
  resources:
  - pods
  verbs: ["get", "list", "watch"]
- nonResourceURLs: ["/metrics"]
  verbs: ["get"]
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: prometheus
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus
subjects:
- kind: ServiceAccount
```

```
  name: prometheus
  namespace: default
```

3. Create RBAC resources:

```
$ kubectl create -f prometheus-rbac-setup.yml
clusterrole "prometheus" created
serviceaccount "prometheus" created
clusterrolebinding "prometheus" created
```

4. Use any text editor to create the **prometheus-config.yml** file with the following content. This YAML file manages Prometheus configurations. When Prometheus is deployed, these configurations can be used by containers through file system mounting.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
data:
  prometheus.yml: |
    global:
      scrape_interval: 10s
    scrape_configs:
    - job_name: 'modelarts'
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      kubernetes_sd_configs:
      - role: pod
      relabel_configs:
        - source_labels: [__meta_kubernetes_pod_name]     # Specifies that metric data is collected from the pod whose name starts with maos-node-agent-.
          action: keep
          regex: ^maos-node-agent-.+
        - source_labels: [__address__]    # Specifies the IP address and port number for obtaining metric data. __address__:9390 specifies the IP address of the POD, which is also the node IP address.
          action: replace
          regex: '(.*)'
          target_label: __address__
          replacement: "${1}:10120"
```

5. Create ConfigMap resources:

```
$ kubectl create -f prometheus-config.yml
configmap "prometheus-config" created
```

6. Use any text editor to create the **prometheus-deployment.yml** file. The content is as follows:

NOTE

This YAML file is used to deploy Prometheus. It grants the permissions of the created account (ServiceAccount) to Prometheus and mounts the created ConfigMap resource to the **/etc/prometheus** directory of the Prometheus container as a file system. The **--config.file=/etc/prometheus/prometheus.yml** parameter specifies the configuration file used by **/bin/prometheus**.

```
apiVersion: v1
kind: "Service"
metadata:
  name: prometheus
  labels:
    name: prometheus
spec:
  ports:
  - name: prometheus
    protocol: TCP
    port: 9090
    targetPort: 9090
  selector:
    app: prometheus
  type: NodePort
```

```
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    name: prometheus
  name: prometheus
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: prometheus
    spec:
      hostNetwork: true
      serviceAccountName: prometheus
      serviceAccount: prometheus
      containers:
      - name: prometheus
        image: prom/prometheus:latest
        imagePullPolicy: IfNotPresent
        command:
        - "/bin/prometheus"
        args:
        - "--config.file=/etc/prometheus/prometheus.yml"
        ports:
        - containerPort: 9090
          protocol: TCP
        volumeMounts:
        - mountPath: "/etc/prometheus"
          name: prometheus-config
      volumes:
      - name: prometheus-config
        configMap:
          name: prometheus-config
```

7. Create a Prometheus instance and check the creation result:

```
$ kubectl create -f prometheus-deployment.yml
service "prometheus" created
deployment "prometheus" created

$ kubectl get pods
NAME                               READY     STATUS        RESTARTS   AGE
prometheus-55f655696d-wjqcl        1/1       Running       0          5s

$ kubectl get svc
NAME           TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes     ClusterIP   10.96.0.1        <none>        443/TCP          131d
prometheus     NodePort    10.101.255.236   <none>        9090:32584/TCP   42s
```

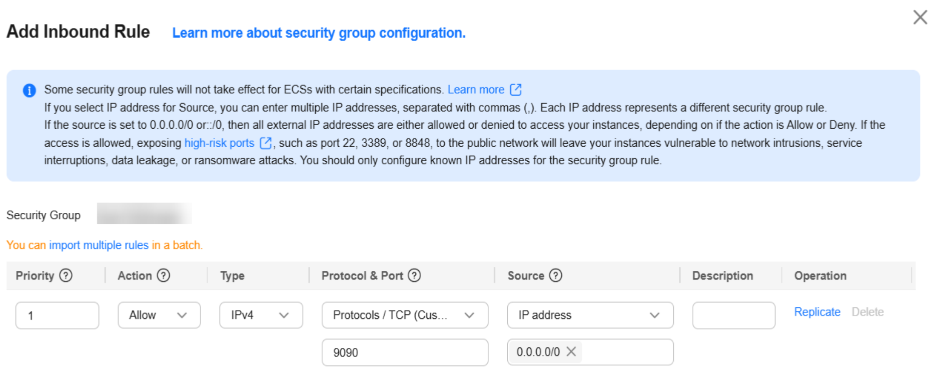
Viewing Metric Data Collected by Prometheus

1. On the CCE console, bind an EIP to the node where Prometheus is deployed. Enable the security group configuration for the node and add an inbound rule to allow external access to port 9090.

   NOTE

   If you use Grafana to interconnect with Prometheus for report creation, you can deploy Grafana within the cluster. In this scenario, there is no need to bind a public IP address to Prometheus or configure a security group for it. Instead, you only need to bind a public IP address to Grafana and configure its security group.

Figure 5-9 Adding an inbound rule



Add Inbound Rule Learn more about security group configuration.

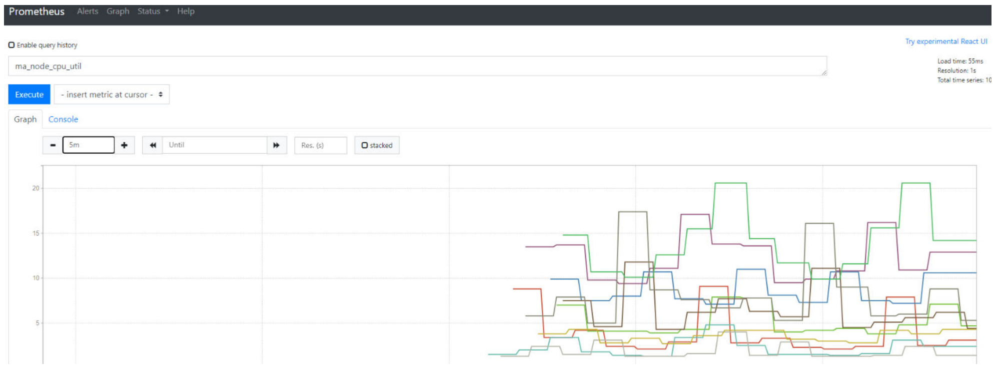
Some security group rules will not take effect for ECSs with certain specifications. Learn more
If you select IP address for Source, you can enter multiple IP addresses, separated with commas (,). Each IP address represents a different security group rule.
If the source is set to 0.0.0.0/0 or::/0, then all external IP addresses are either allowed or denied to access your instances, depending on if the action is Allow or Deny. If the access is allowed, exposing high-risk ports , such as port 22, 3389, or 8848, to the public network will leave your instances vulnerable to network intrusions, service interruptions, data leakage, or ransomware attacks. You should only configure known IP addresses for the security group rule.

Security Group [redacted]

You can import multiple rules in a batch.

Priority	Action	Type	Protocol & Port	Source	Description	Operation
1	Allow	IPv4	Protocols / TCP (Cus...	IP address		Replicate Delete
			9090	0.0.0.0/0		

2. Enter **http://<*EIP*>:9090** in the address box of the browser. The Prometheus monitoring page is displayed. Click **Graph** and enter a metric name in the text box to view the metric data collected by Prometheus.



5.10 Releasing Lite Cluster Resources

You can release Lite Cluster resources that are no longer used. For details about how to stop billing, see **Stopping Billing**.

NOTE

Released Lite Cluster resource pools cannot be restored.

Unsubscribing from a Yearly/Monthly Lite Cluster Resource Pool

Step 1 Log in to the ModelArts console. In the navigation pane, choose **AI Dedicated Resource Pools** > **Elastic Clusters**. Click the **ModelArts Lite** tab to view the resource pool list.

Step 2 In the resource pool list, choose ··· > **Unsubscribe** in the **Operation** column.

Step 3 Confirm the target resources and select the reason for unsubscription.

Step 4 Confirm the information and select **After being unsubscribed from, the resource not in the recycle bin will be deleted immediately and cannot be restored. I have backed up data or no longer need the data**.

Step 5 Click **Unsubscribe** and confirm the resources.

Step 6 Click **Unsubscribe** again to finish the subscription.

----End

Releasing a Free Node

Nodes that are not managed by the resource pool are considered as free nodes. To view the information about free nodes, log in to the ModelArts management console, choose **AI Dedicated Resource Pools** > **Elastic Clusters**, and click the **Nodes** tab.

Release the free nodes resources according to the following content:

- For a yearly/monthly node whose resources are not expired, click **Unsubscribe** in the **Operation** column. You can unsubscribe from node in batches.
- For a yearly/monthly node whose resources are expired (in the grace period), click **Release** in the **Operation** column. Nodes in the grace period cannot be released in batches.

NOTE

Unsubscription and release operations cannot be undone. Exercise caution when performing this operation.