ModelArts

Resource Management

Issue 01

Date 2024-06-07

Huawei Technologies Co., Ltd.

Address:	Huawei Industrial Base Bantian, Longgang Shenzhen 518129 People's Republic of China
Website:	https://www.huawei.com
Email:	support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Resource Pool

ModelArts Resource Pools

When using ModelArts for AI development, you can use either of the following resource pools:

- **Dedicated resource pool**: It delivers more controllable resources and cannot be shared with other users. Create a dedicated resource pool and select it during AI development. The dedicated resource pool can be an elastic cluster or an elastic BMS.

    - Elastic cluster: It can be Standard or Lite.

        - In a Standard elastic cluster, exclusive computing resources are provided, with which you can deliver instances during job training, model deployment, and environment development on ModelArts.

        - A Lite elastic cluster provides hosted Kubernetes clusters with mainstream AI development plug-ins and Huawei-developed acceleration plug-ins for Kubernetes resource users. You can operate the nodes and Kubernetes clusters in the resource pool with provided AI Native resources and tasks.

    - Elastic BMS: It provides different models of xPU BMSs. You can access an elastic BMS through an EIP and install GPU- and NPU-related drivers and software on a specified OS image. To meet the routine training requirements of algorithm engineers, SFS and OBS can be used to store and read data.

- **Public resource pool**: It provides large-scale public computing clusters, which are allocated based on job parameter settings. Resources are isolated by job. You can use ModelArts public resource pools to deliver training jobs, deploy models, or run DevEnviron instances. You will be billed on a pay-per-use basis.

Differences Between Dedicated Resource Pools and Public Resource Pools

- Dedicated resource pools provide dedicated computing clusters and network resources for users. The dedicated resource pools of different users are physically isolated, while public resource pools are only logically isolated. Compared with public resource pools, dedicated resource pools feature better performance in isolation and security.

- When a dedicated resource pool is used for creating jobs and the resources are sufficient, the jobs will not be queued. When a public resource pool is used for creating jobs, there is a high probability that the jobs will be queued.

- A dedicated resource pool is accessible to your network. All running jobs in the pool can access storage and resources in your network. For example, if you select a dedicated resource pool with an accessible network when creating a training job, you can access SFS data after the training job is created.

- Dedicated resource pools allow you to customize the runtime environment of physical nodes, for example, you can upgrade GPU or Ascend drivers. This function is not supported by public resource pools.

2 Elastic Cluster

2.1 Comprehensive Upgrades to ModelArts Resource Pool Management Functions

ModelArts dedicated resource pools have been upgraded. In the new system, there are only unified ModelArts dedicated resource pools, which are no longer classified as the pools dedicated for development/training and the pools dedicated for service deployment. The new-version dedicated resource pools support flexible configuration of job types, and allow you to manage networks and interconnect VPCs with the networks.

The new dedicated resource pool management page provides more comprehensive functions and displays more information about the resource pools. More details about how to use and manage dedicated resource pools are provided in subsequent sections of this document. If you are new to ModelArts dedicated resource pools, try new-version dedicated resource pools. If you have used ModelArts dedicated resource pools, the old-version pools will be smoothly switched to new-version pools.

Read the following contents to learn about new-version dedicated resource pools.

Features of New-Version Dedicated Resource Pools

The new-version dedicated resource pool management is a comprehensive technology and product improvement. The main improvements are as follows:

- **Single dedicated resource pool type for diverse purposes**: Dedicated resource pools are no longer classified into those for development/training and those for service deployment. You can run both training and inference workloads in a dedicated resource pool. You can also set the job types supported by a dedicated resource pool based on your needs.

- **Dedicated resource pool network connection**: You can create and manage dedicated resource pool networks on the ModelArts management console. If you need to access resources in your VPC for jobs running in a dedicated resource pool, interconnect the VPC with the dedicated resource pool network.

- **More cluster details**: The new-version dedicated resource pool details page provides more cluster details, such as jobs, nodes, and resource monitoring, helping you learn about the cluster status and better plan and use resources.

- **Cluster GPU/NPU driver management**: On the new-version dedicated resource pool details page, you can select an accelerator card driver and perform change upon submission or smooth upgrade of the driver based on service requirements.

- **Fine-grained resource allocation (coming soon)**: You can divide your dedicated resource pool into multiple small pools and assign different quotas and permissions to each small pool for flexible and refined resource allocation and management.

More features will be provided in later versions for a better user experience.

Can I Continue to Use the Existing Dedicated Resource Pools After the Upgrade Takes Effect?

If you have created dedicated resource pools, you can still access the old-version dedicated resource pool (elastic cluster) management page on the ModelArts management console and use the created resource pools, but you cannot create dedicated resource pool on that page. ModelArts allows you to migrate existing dedicated resource pools to the new management page. You will be contacted to complete the migration and this does not require you to perform any operations. In addition, the migration does not affect the workloads running in the dedicated resource pools. Pay attention to the easy-to-use new management functions of dedicated resource pools. There is no change in creating training jobs or inference services.

Differences Between New and Old Dedicated Resource Pools

- In the old version, the dedicated resource pools dedicated for development/training are separated from those dedicated for service deployment. In addition, the pools of the two types offer different functions and their user experience varies. In the new version, the dedicated resource pools of the two types are unified. You only need to configure one or multiple job types. Then, the dedicated resource pool automatically supports the configured job type.

- New dedicated resource pools inherit all functions of the old ones and have greatly improved user experience in key functions such as purchasing and resizing a resource pool. Use new dedicated resource pools for smooth, transparent experience.

- Additionally, the new dedicated resource pools offer enhanced functions, for example, allowing you to upgrade GPU or Ascend drivers, view details about job queuing, and use one network for multiple pools. More new functions of the new dedicated resource pools are coming soon.

How Can I Get Help or Provide Feedback if I Encounter Problems During Use?

Similar to other ModelArts functions, you can report problems or obtain help in the sidebar of the console. In addition, you are advised to read the subsequent sections of this document to further understand how to use ModelArts dedicated resource pools.

Instructions of Dedicated Resource Pools

- If you use dedicated resource pools for the first time, get started by reading **Resource Pool**.

- Create a dedicated resource pool by referring to **Creating a Resource Pool**.

- View the details about a created dedicated resource pool by referring to **Viewing Details About a Resource Pool**.

- If the specifications of a dedicated resource pool do not meet your service requirements, adjust the specifications by referring to **Resizing a Resource Pool**.

- Set or change job types supported by a dedicated resource pool by referring to **Changing Job Types Supported by a Resource Pool**.

- Upgrade the GPU/Ascend driver of your dedicated resource pools by referring to **Upgrading a Resource Pool Driver**.

- If a dedicated resource pool is no longer needed, delete it by referring to **Deleting a Resource Pool**.

- If any exception occurs when you use a dedicated resource pool, handle the exception by referring to **Abnormal Status of a Dedicated Resource Pool**.

- Manage dedicated resource pool networks or interconnect VPCs with the networks by referring to **ModelArts Network**.

2.2 Creating a Resource Pool

This section describes how to create a dedicated resource pool.

Creating a Dedicated Resource Pool

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. On the **Resource Pools** tab page, click **Create** and configure parameters.

Table 2-1 Dedicated resource pool parameters

Parameter	Sub-Parameter	Description
Name	N/A	Name of a dedicated resource pool. Only lowercase letters, digits, and hyphens (-) are allowed. The value must start with a lowercase letter and cannot end with a hyphen (-).
Description	N/A	Brief description of a dedicated resource pool

Para meter	Sub-Para meter	Description
Billi ng Mod e	N/A	You can select **Pay-per-use**.
Job Type	N/A	Select job types supported by the resource pool based on service requirements.
Net wor k	N/A	Network in which the target service instance is deployed. The instance can exchange data with other cloud service resources in the same network. Select a network from the drop-down list box. If no network is available, click **Create** on the right to create a network. For details about how to create a network, see **Creating a Network**.
Spec ifica tion Man age men t	Specif icati ons	Select required specifications. Due to system loss, the actual available resources are less than those specified in the specifications. After a dedicated resource pool is created, you can view the actual available resources on the **Nodes** tab page of the dedicated resource pool details page.
	AZ	You can select **Automatically allocated** or **Specifies AZ**. An AZ is a physical region where resources use independent power supplies and networks. AZs are physically isolated but interconnected over an intranet. • **Automatically allocated**: AZs are automatically allocated. • **Specifies AZ**: Specify AZs for resource pool nodes. To ensure system disaster recovery, deploy all nodes in the same AZ. You can set the number of nodes in an AZ.
	Nod es	Select the number of nodes in a dedicated resource pool. More nodes mean higher computing performance. If **AZ** is set to **Specifies AZ**, you do not need to configure **Nodes**. **NOTE** It is a good practice to create no more than 30 nodes at a time. Otherwise, the creation may fail due to traffic limiting.
Cust om Driv er	N/A	This parameter is available only when a GPU flavor is selected. Enable this function and select a GPU driver.
GPU Driv er	N/A	This parameter is available only when custom driver is enabled. Select a GPU accelerator driver. **NOTE** You should choose NVIDIA driver 535.129.03 or later for hnt8 series specifications.

| Parameter	Sub-Parameter	Description
Advanced Configuration	N/A	Select **Configure Now** to set the tag information, CIDR block, cluster specifications, and controller node distribution.
CIDR Block	N/A	You can select **Default** or **Custom**. ● **Default**: The system randomly allocates an available CIDR block to you, which cannot be modified after the resource pool is created. For commercial use, customize your CIDR block. ● **Custom**: You need to customize K8S container and K8S service CIDR blocks. 　– **K8S Container Network**: used by the container in a cluster, which determines how many containers there can be in a cluster. The value cannot be changed after the resource pool is created. 　– **K8S Service Network**: used when the containers in the same cluster access each other, which determines how many Services there can be. The value cannot be changed after the resource pool is created.
Cluster Specifications	N/A	**Cluster Scale**: maximum number of nodes that can be managed by the cluster.After the creation, the cluster can be scaled out but cannot be scaled in. You can select **Default** or **Custom**.
Master Distribution	N/A	Distribution locations of controller nodes. You can select **Random** or **Custom**. ● **Random**: Use the AZs randomly allocated by the system. ● **Custom**: Select AZs for controller nodes. Distribute controller nodes in different AZs for disaster recovery.

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

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

Figure 2-1 Viewing a resource pool

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

Figure 2-2 Resource pool list
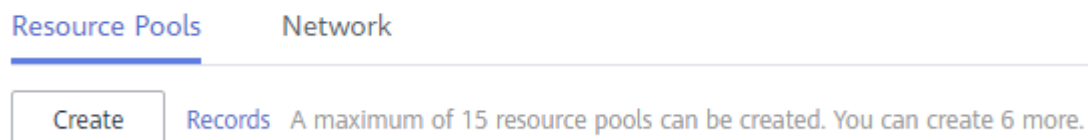
FAQs

What if I choose a flavor for a dedicated resource pool, but get an error message saying no resource is available?

The flavors of dedicated resources change based on real-time availability. Sometimes, you might choose a flavor on the purchase page, but it is sold out before you pay and create the resource pool. This causes the resource pool creation to fail.

You can try a different flavor on the creation page and create the resource pool again.

Q: Why cannot I use all the CPU resources on a node in a resource pool?

Resource pool nodes have systems and plug-ins installed on them. These take up some CPU resources. For example, if a node has 8 vCPUs, but some of them are used by system components, the available resources will be fewer than 8 vCPUs.

You can check the available CPU resources by clicking the **Nodes** tab on the resource pool details page, before you start a task.

2.3 Viewing Details About a Resource Pool

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. In the resource pool list, click a resource pool to go to its details page and view its information.

   – If there are multiple resource pools, click ▼ in the upper left corner of the details page of one resource pool to switch between resource pools. Click **More** in the upper right corner to perform operations such as adjust capacity, delete the resource pool, or set the job type. The available operations vary depending on resource pools.

   – In the **Network** area of **Basic Information**, you can click the number of resource pools associated to view associated resource pools.

   – In the extended information area, you can view the monitoring information, jobs, nodes, specifications, events, and subpools. For details, see the following section.

Viewing Resource Pool Jobs

On the resource pool details page, click **Jobs**. You can view all jobs running in the resource pool. If a job is queuing, you can view its queuing position.

▭ NOTE

Only training jobs can be viewed.

Viewing Resource Pool Monitoring Information

On the resource pool details page, click **Monitoring**. The resource usage including used CPUs, memory usage, and available disk capacity of the resource pool is displayed. If AI accelerators are used in the resource pool, the GPU and NPU monitoring information is also displayed.

Figure 2-3 Viewing resource views
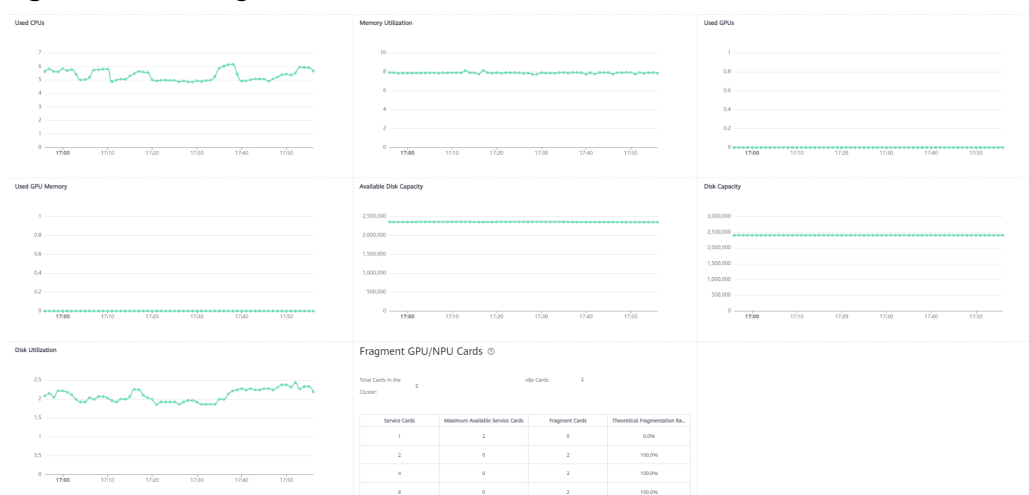
Figure 2-4 Jobs

Viewing Resource Pool Nodes

On the resource pool details page, click **Nodes**. You can view all nodes in the resource pool and the resource usage of each node.

Some resources are reserved for cluster components. Therefore, **CPUs (Available/Total)** does not indicate the number of physical resources on the node. It only displays the number of resources that can be used by services. CPU cores are metered in milicores, and 1000 milicores equal 1 physical core.

● Deleting nodes:

On the **Nodes** tab, locate the node to be deleted. In the **Operation** column, click **Delete**.

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

Delete abnormal nodes in a resource pool one by one or in batches and add new ones for substitution.

> **NOTE**
>
> ● Before deleting a node, ensure that there are no running jobs on this node. Otherwise, the jobs will be interrupted.
>
> ● If there is only one node, it cannot be deleted.

● Replacing nodes:

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

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

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

> **NOTE**
>
> ● The number of replacements per day cannot exceed 20% of the total nodes in the resource pool. The number of nodes to be replaced cannot exceed 5% of the total nodes in the resource pool.
>
> ● Ensure that there are idle node resources. Otherwise, the replacement may fail.

Figure 2-5 Nodes

Figure 2-6 Operation records

Viewing Resource Pool Specifications

On the resource pool details page, click **Specifications**. You can view the specifications used by the resource pool and the number of each specification.

Figure 2-7 Specifications

Specifications	Metering ID	CPU Cores	CPU Architecture	Memory	AI Accelerator	Disk Capacity	Quantity
GPU: 8*nvidia-v100-nv32 | CPU: 67 vC...	pool.dc24c362	67	x86	480GB	8*nvidia-v100-nv32	--	4

Viewing Resource Pool Events

On the resource pool details page, click **Events**. You can view all events of the resource pool. The cause of an event is **PoolStatusChange** or **PoolResourcesStatusChange**.

On the trace list, click ▽ on the right of **Event Type** to filter events.

- When a resource pool starts to be created or becomes abnormal, the resource pool status changes and the change will be recorded as an event.
- When the number of nodes that are available or abnormal or in the process of being created or deleted changes, the resource pool node status changes and the change will be recorded as an event.

Figure 2-8 Events

Event Type ▽	Cause	Details	Occurred At
Normal	PoolResourcesStatusChange	Pool resources status changed, available/abnormal/creating/deleting count from 0/1/0/0 to 1/0/0/0, timestamp: 1675179100.	Jan 31, 2023 23:31:40 GMT+08:00
Abnormal	PoolResourcesStatusChange	Pool resources status changed, available/abnormal/creating/deleting count from 1/0/0/0 to 0/1/0/0, timestamp: 1675178800.	Jan 31, 2023 23:26:40 GMT+08:00
Normal	PoolResourcesStatusChange	Pool resources status changed, available/abnormal/creating/deleting count from 0/0/1/0 to 1/0/0/0, timestamp: 1675178753.	Jan 31, 2023 23:25:53 GMT+08:00
Normal	PoolResourcesStatusChange	Pool resources status changed, available/abnormal/creating/deleting count from 0/0/0/0 to 0/0/1/0, timestamp: 1675178510.	Jan 31, 2023 23:21:50 GMT+08:00
Normal	PoolStatusChange	Pool status changed, from Creating to Running.	Jan 31, 2023 23:21:50 GMT+08:00
Normal	PoolStatusChange	Start creating pool.	Jan 31, 2023 23:15:44 GMT+08:00

2.4 Resizing a Resource Pool

Description

The demand for resources in a dedicated resource pool may change due to the changes of AI development services. In this case, you can resize your dedicated resource pool in ModelArts.

- You can add nodes for existing flavors in the resource pool.
- You can delete nodes for existing flavors in the resource pool.

📖 **NOTE**

Before scaling in a resource pool, ensure that there are no services running in the pool. Alternatively, go to the resource pool details page, delete the nodes where no services are running to scale in the pool.

Constraints

- Only dedicated resource pools in the **Running** status can be resized.
- When scaling in a dedicated resource pool, the number of flavors or nodes of a flavor cannot be decreased to 0.
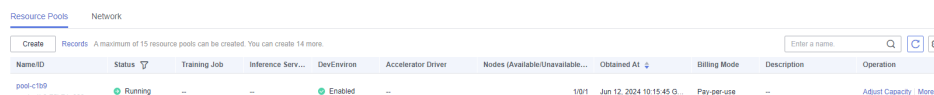
Resizing a Dedicated Resource Pool

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

📖 **NOTE**

> A resource pool is suspended when it is migrated from the old version to the new version. You cannot adjust the capacity of such a resource pool or unsubscribe from it.

Figure 2-9 Resource Pools

2. Add or delete nodes.

   Click **Adjust Capacity** in the **Operation** column of the target resource pool.

   In the **Resource Configurations** area, set **AZ** to **Automatically allocated** or **Specifies AZ**. Click **Submit** and then **OK** to save the changes.

   – If **AZ** is set to **Automatically allocated**, you can increase or decrease the number of nodes to scale out or in the resource pool. After the scaling, nodes are automatically allocated to AZs.

   – If you select **Specifies AZ**, you can allocate nodes to different AZs.

Figure 2-10 Resource Configurations
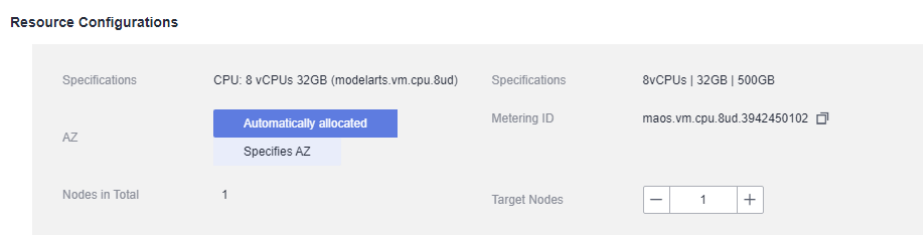
2.5 Migrating the Workspace

Context

The workspace of a dedicated resource pool is associated with an enterprise project, which involves bill collection. ModelArts provides workspaces to isolate resource operation permissions of different IAM users. Workspace migration includes resource pool migration and network migration. For details, see the following sections.

Migrating the Workspace for a Resource Pool

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. In the resource pool list, choose **More** > **Migrate Workspace** in the **Operation** column of the target resource pool.

3. In the **Migrate Dedicated Resource Pool** dialog box that appears, select the target workspace and click **OK**.

Figure 2-11 Migrating the workspace

Migrating the Workspace for a Network

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**. Then, click the **Network** tab.

2. In the network list, choose **More** > **Migrate Workspace** in the **Operation** column of the target network.

3. In the dialog box that appears, select the target workspace and click **OK**.

2.6 Changing Job Types Supported by a Resource Pool

Description

ModelArts supports many types of jobs. Some of them can run in dedicated resource pools, including training jobs, inference services, and notebook development environments.

You can change job types supported by a dedicated resource pool. Available options for **Job Type** are **Training Job**, **Inference Service**, and **DevEnviron**.

Only selected types of jobs can be delivered to the corresponding dedicated resource pool.

⚠️ **CAUTION**

To support different job types, different operations are performed in the backend, such as installing plug-ins and setting the network environment. Some operations use resources in the resource pool. As a result, available resources for you decrease. Therefore, select only the job types you need to avoid resource waste.

Constraints

The target dedicated resource pool must be running.

Procedure

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. In the **Operation** column of a resource pool, choose **More** > **Set Job Type**.

3. In the **Set Job Type** dialog box, select job types.

Figure 2-12 Setting the job type
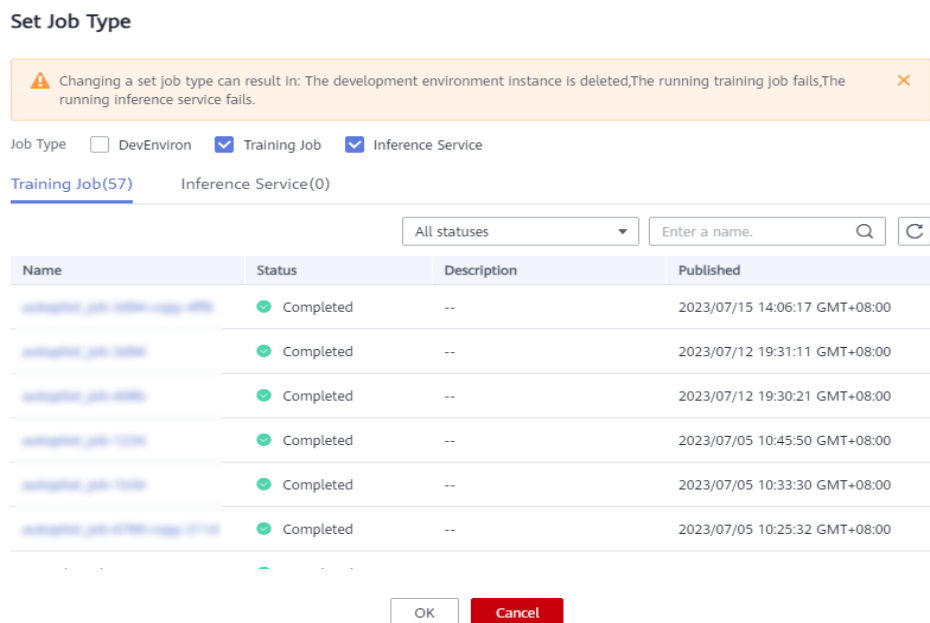
4. Click **OK**.

2.7 Upgrading a Resource Pool Driver

Description

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

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

📖 **NOTE**

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

Constraints

The target dedicated resource pool is in running, and the resource pool contains GPU or Ascend resources.

Upgrading the Driver

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. In the **Operation** column of the target resource pool, choose **More** > **Upgrade Driver**.

3. In the **Upgrade Driver** dialog box, the driver type, number of nodes, current version, target version, and upgrade mode of the dedicated resource pool are displayed.

   – **Target Version**: Select a target driver version from the drop-down list.

   – **Upgrade Mode**: Select **Secure upgrade** or **Forcible upgrade**.

Figure 2-13 Upgrade Driver
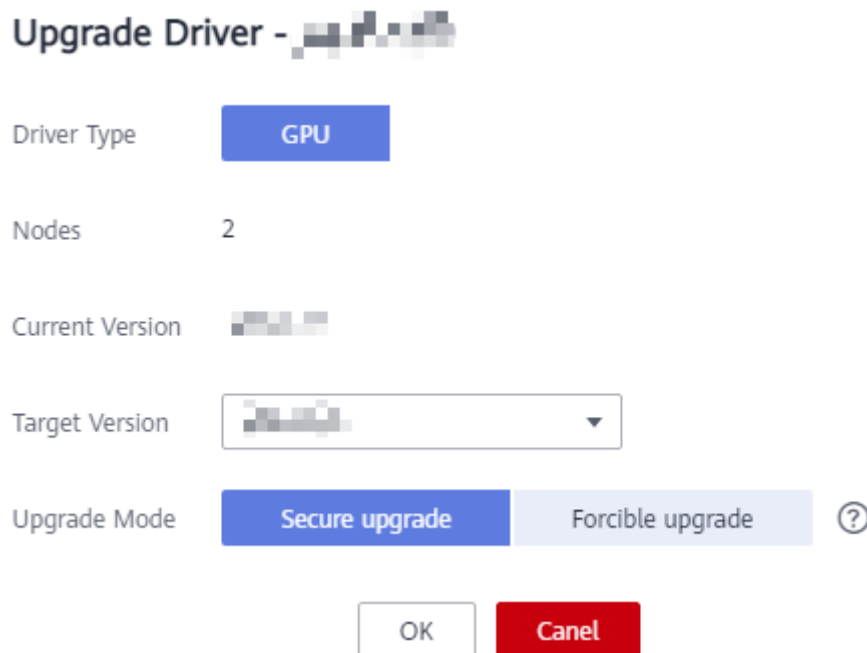
4. Click **OK** to start the driver upgrade.

2.8 Deleting a Resource Pool

If a dedicated resource pool is no longer needed for AI service development, you can delete the resource pool to release resources.

📖 **NOTE**

After a dedicated resource pool is deleted, the development environments, training jobs, and inference services that depend on the resource pool are unavailable. A dedicated resource pool cannot be restored after being deleted.

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. Locate the row that contains the target resource pool, choose **More** > **Delete** in the **Operation** column.

3. In the **Delete Dedicated Resource Pool** dialog box, enter **DELETE** in the text box and click **OK**.

   You can switch between tabs on the details page to view the training jobs,inference services, notebook instances, and subpools created using the resource pool.

2.9 Abnormal Status of a Dedicated Resource Pool

Resource Quota Limit

When you use a dedicated resource pool (for example, scaling resources, creating a VPC, creating a VPC and subnet, or interconnecting a VPC), if the system displays a message indicating that the resource quota is limited, submit a service ticket.

Creation Failed/Change Failed

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.

2. Click **Records** on the right of **Create**. On the **Records** dialog box, view failed task records.

   Figure 2-14 Creating a resource pool failed

3. Hover the cursor over ⑦, view the cause of task failures.

   ☐ NOTE

   By default, failed task records are sorted by application time. A maximum of 500 failed task records can be displayed and retained for three days.

Locating Faulty Node

ModelArts will add a taint on a detected K8S faulty node so that jobs will not be affected or scheduled to the tainted node. The following table lists the faults can be detected. You can locate the fault by referring to the isolation code and detection method.

Table 2-2 Isolation code

Isolation Code	Category	Sub-Category	Description	Detection Method
A05 0101	GPU	GPU memory	GPU ECC error exists.	Run the **nvidia-smi -a** command and check whether **Pending Page Blacklist** is **Yes** or the value of **multi-bit Register File** is greater than 0. For Ampere GPUs, check whether the following content exists: 1. Uncorrectable SRAM error 2. Remapping Failure or Pending records 3. Xid 95 events in **dmesg**
A05 0102	GPU	Other	The **nvidia-smi** output contains ERR.	Run **nvidia-smi -a** and check whether the output contains ERR. Normally, the hardware, such as the power supply or the fan, is faulty.
A05 0103	GPU	Other	The execution of **nvidia-smi** times out or does not exist.	Check that exit code of **nvidia-smi** is not **0**.
A05 0104	GPU	GPU Memory	ECC error occurred 64 times.	Run the **nvidia-smi -a** command, locate **Retired Pages**, and check whether the sum of **Single Bit** and **Double Bit** is greater than 64.
A05 0148	GPU	Other	An infoROM alarm occurs.	Run the **nvidia-smi** command and check whether the output contains the alarm "infoROM is corrupted".
A05 0109	GPU	Other	Other GPU errors	Check whether other GPU error exists. Normally, there is a faulty hardware. Contact the technical engineer.
A05 0147	IB	Link	The IB NIC is abnormal.	Run the **ibstat** command and check whether the NIC is not in active state.
A05 0121	NPU	Other	A driver exception is detected by NPU DCMI.	The NPU driver environment is abnormal.

Isolation Code	Category	Sub-Category	Description	Detection Method
A05 0122	NPU	Other	The NPU DCMI device is abnormal.	The NPU device is abnormal. The Ascend DCMI interface returns a major or urgent alarm.
A05 0123	NPU	Link	The NPU DCMI net is abnormal.	The NPU network connection is abnormal.
A05 0129	NPU	Other	Other NPU errors	Check whether other NPU error exists. You cannot rectify the fault. Contact the technical engineer.
A05 0149	NPU	Link	Check whether the network port of the hccn tool is intermittently disconnected.	The NPU network is unstable and intermittently disconnected. Run the **hccn_tool-i ${device_id} -link_stat -g** command and the network is disconnected more than five times within 24 hours.
A05 0951	NPU	GPU memory	The number of NPU ECCs reaches the maintenance threshold.	The NPU's HBM Double Bit Isolated Pages Count value is greater than or equal to 64.
A05 0146	Runtime	Other	The NTP is abnormal.	The ntpd or chronyd service is abnormal.
A05 0202	Runtime	Other	The node is not ready.	The node is unavailable. The K8S node contains one of the following taints: ● node.kubernetes.io/unreachable ● node.kubernetes.io/not-ready
A05 0203	Runtime	Disconnection	The number of normal AI cards does not match the actual capacity.	The GPU or NPU is disconnected.
A05 0206	Runtime	Other	The Kubelet hard disk is read-only.	The **/mnt/paas/kubernetes/kubelet** directory is read-only.
A05 0801	Node management	Node O&M	Resource is reserved.	The node is marked as the standby node and contains a taint.

Isolation Code	Category	Sub-Category	Description	Detection Method
A05 0802	Node management	Node O&M	An unknown error occurs.	The node is marked with an unknown taint.
A20 0001	Node management	Driver upgrade	The GPU is being upgraded.	The GPU is being upgraded.
A20 0002	Node management	Driver upgrade	The NPU is being upgraded.	The NPU is being upgraded.
A20 0008	Node management	Node admission	The admission is being examined.	The admission is being examined, including basic node configuration check and simple service verification.
A05 0933	Node management	Fault tolerance Failover	The Failover service on the tainted node will be migrated.	The Failover service on the tainted node will be migrated.
A05 0931	Training toolkit	Pre-check container	A GPU error is detected in the pre-check container.	A GPU error is detected in the pre-check container.
A05 0932	Training toolkit	Pre-check container	An IB error is detected in the pre-check container.	An IB error is detected in the pre-check container.

2.10 ModelArts Network

ModelArts Network and VPC

ModelArts networks are used for interconnecting nodes in a ModelArts resource pool. You can only configure the name and CIDR block for a network. To ensure that there is no IP address segment in the CIDR block overlapped with that of the VPC to be accessed, multiple CIDR blocks are available for you to select.

A VPC provides a logically isolated virtual network for your instances. You can configure and manage the network as required. VPC provides logically isolated,

configurable, and manageable virtual networks for cloud servers, cloud containers, and cloud databases. It helps you improve cloud service security and simplify network deployment.

Prerequisites

- A VPC is available.
- A subnet is available.

Creating a Network

1. Log in to the ModelArts management console. In the navigation pane, choose **Dedicated Resource Pools** > **Elastic Cluster**.
2. Click **Network** and then **Create**.

   Figure 2-15 Network list

3. In the **Create Network** dialog box, set parameters.
   - **Network Name**: customizable name
   - **CIDR Block**: You can select **Preset** or **Custom**.

     ☐ **NOTE**

     - Each user can create a maximum of 15 networks.
     - Ensure there is no IP address segment in the CIDR block overlaps that of the VPC to be accessed. The CIDR block cannot be changed after the network is created. Possible conflict CIDR blocks are as follows:
       - Your VPC CIDR block
       - Container CIDR block (consistently to be 172.16.0.0/16)
       - Service CIDR block (consistently to be 10.247.0.0/16)
4. Confirm the settings and click **OK**.

(Optional) Interconnecting a VPC with a ModelArts Network

VPC interconnection allows you to use resources across VPCs, improving resource utilization.

1. On the **Network** page, click **Interconnect VPC** in the **Operation** column of the target network.

   Figure 2-16 Interconnect VPC
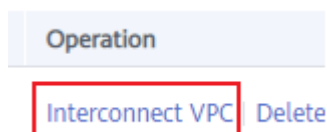
2. In the displayed dialog box, click the button on the right of **Interconnect VPC**, and select an available VPC and subnet from the drop-down lists.

📖 NOTE

The peer network to be interconnected cannot overlap with the current CIDR block.

Figure 2-17 Parameters for interconnecting a VPC with a network

- – If no VPC is available, click **Create VPC** on the right to create a VPC.
- – If no subnet is available, click **Create Subnet** on the right to create a subnet.

Deleting a Network

If a network is no longer needed for AI service development, you can delete the network.

1. Go to the **Network** tab page and click **Delete** in the **Operation** column of a network.
2. Confirm the information and click **OK**.

3 Monitoring Resources

3.1 Viewing All ModelArts Monitoring Metrics on the AOM Console

ModelArts periodically collects the usage of key metrics (such as GPUs, NPUs, CPUs, and memory) of each node in a resource pool as well as the usage of key metrics of the development environment, training jobs, and inference services, and reports the data to AOM. You can view the information on AOM.

1. Log in to the console and search for **AOM** to go to the AOM console.

2. Choose **Metric Monitoring**. On the **Metric Monitoring** page that is displayed, click **Add Metric**.
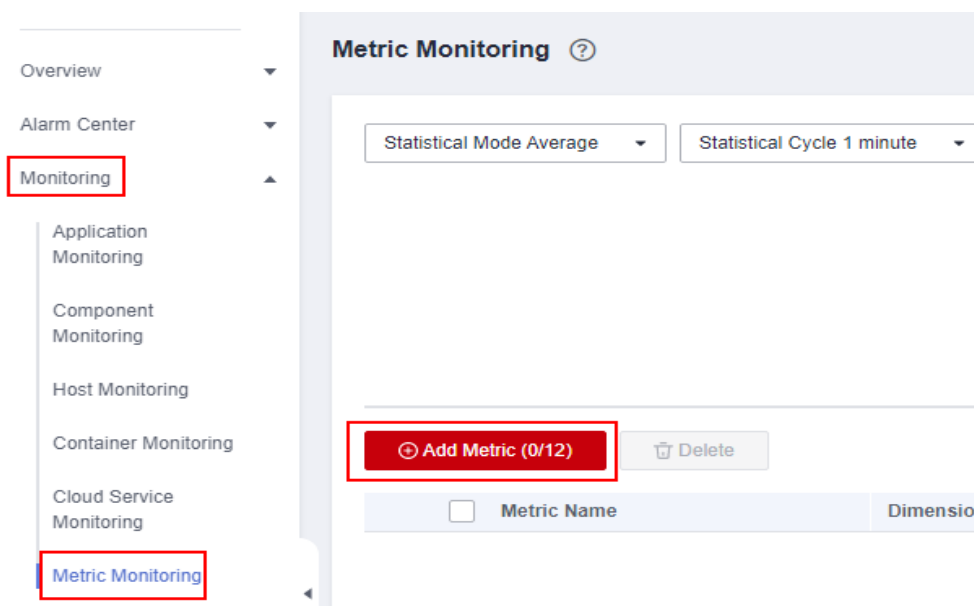
3. Add metrics and click Add to Metric List.

   – **Add By**: Select All Metrics.

   – **Metric Name**: Select the desired ones for query. For details, see **Table 3-1**, **Table 3-2**, and **Table 3-3**.

– Scope: Enter the tag for filtering the metric. For details, see **Table 3-4**. The following shows an example.

4. View the metrics.
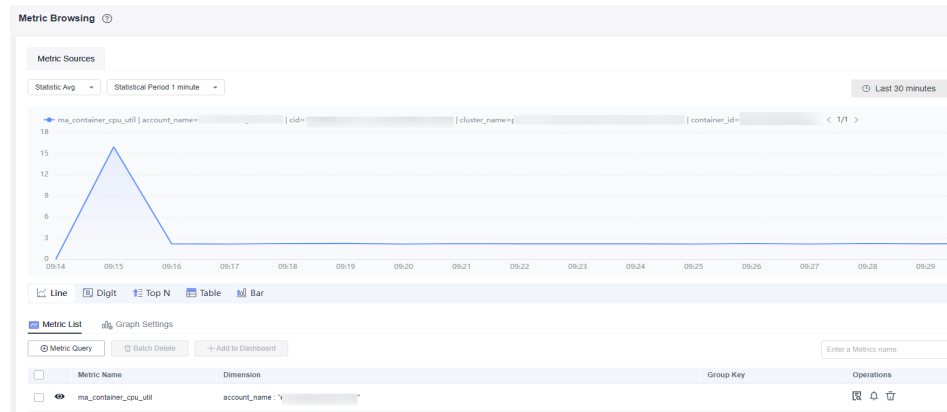
Table 3-1 Container metrics

Classification	Name	Metric	Description	Unit	Value Range
CPU	CPU Usage	ma_container_cpu_util	CPU usage of a measured object	%	0%–100%
	Used CPU Cores	ma_container_cpu_used_core	Number of CPU cores used by a measured object	Cores	≥ 0
	Total CPU Cores	ma_container_cpu_limit_core	Total number of CPU cores that have been applied for a measured object	Cores	≥ 1
Memory	Total Physical Memory	ma_container_memory_capacity_megabytes	Total physical memory that has been applied for a measured object	MB	≥ 0

Classification	Name	Metric	Description	Unit	Value Range
	Physical Memory Usage	ma_container_memory_util	Percentage of the used physical memory to the total physical memory	%	0%–100%
	Used Physical Memory	ma_container_memory_used_megabytes	Physical memory that has been used by a measured object (**container_memory_working_set_bytes** in the current working set) (Memory usage in a working set = Active anonymous page and cache, and file-baked page ≤ **container_memory_usage_bytes**)	MB	≥ 0
Storage	Disk Read Rate	ma_container_disk_read_kilobytes	Volume of data read from a disk per second	KB/s	≥ 0
	Disk Write Rate	ma_container_disk_write_kilobytes	Volume of data written into a disk per second	KB/s	≥ 0

Classification	Name	Metric	Description	Unit	Value Range
GPU memory	Total GPU Memory	ma_container_ gpu_mem_total _megabytes	Total GPU memory of a training job	MB	> 0
	GPU Memory Usage	ma_container_ gpu_mem_util	Percentage of the used GPU memory to the total GPU memory	%	0%–100%
	Used GPU Memory	ma_container_ gpu_mem_used _megabytes	GPU memory used by a measured object	MB	≥ 0
GPU	GPU Usage	ma_container_ gpu_util	GPU usage of a measured object	%	0%–100%
	GPU Memory Bandwidth Usage	ma_container_ gpu_mem_copy _util	GPU memory bandwidth usage of a measured object For example, the maximum memory bandwidth of NVIDIA GPU V100 is 900 GB/s. If the current memory bandwidth is 450 GB/s, the memory bandwidth usage is 50%.	%	0%–100%

Classification	Name	Metric	Description	Unit	Value Range
	GPU Encoder Usage	ma_container_gpu_enc_util	GPU encoder usage of a measured object	%	%
	GPU Decoder Usage	ma_container_gpu_dec_util	GPU decoder usage of a measured object	%	%
	GPU Temperature	DCGM_FI_DEV_GPU_TEMP	GPU temperature	°C	Natural number
	GPU Power	DCGM_FI_DEV_POWER_USAGE	GPU power	Watt (W)	> 0
	GPU Memory Temperature	DCGM_FI_DEV_MEMORY_TEMP	GPU memory temperature	°C	Natural number
Network I/O	Downlink Rate (BPS)	ma_container_network_receive_bytes	Inbound traffic rate of a measured object	Bytes/s	≥ 0
	Downlink Rate (PPS)	ma_container_network_receive_packets	Number of data packets received by an NIC per second	Packets/s	≥ 0
	Downlink Error Rate	ma_container_network_receive_error_packets	Number of error packets received by an NIC per second	Packets/s	≥ 0
	Uplink Rate (BPS)	ma_container_network_transmit_bytes	Outbound traffic rate of a measured object	Bytes/s	≥ 0

Classif ication	Name	Metric	Descriptio n	Unit	Value Range
	Uplink Error Rate	ma_container_ network_trans mit_error_pack ets	Number of error packets sent by an NIC per second	Packets/s	≥ 0
	Uplink Rate (PPS)	ma_container_ network_trans mit_packets	Number of data packets sent by an NIC per second	Packets/s	≥ 0
Noteb ook servic e metric s	Notebook Cache Directory Size	ma_container_ notebook_cach e_dir_size_byte s	A high-speed local disk is attached to the **/cache** directory for GPU notebook instances. This metric indicates the total size of the directory.	Bytes	≥ 0
	Notebook Cache Directory Utilization	ma_container_ notebook_cach e_dir_util	A high-speed local disk is attached to the **/cache** directory for GPU notebook instances. This metric indicates the utilization of the directory.	%	0%–100%

Table 3-2 Node metrics (collected only in dedicated resource pools)

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
CPU	Total CPU Cores	ma_node_c pu_limit_co re	Total number of CPU cores that have been applied for a measured object	Cores	≥ 1
	Used CPU Cores	ma_node_c pu_used_co re	Number of CPU cores used by a measured object	Cores	≥ 0
	CPU Usage	ma_node_c pu_util	CPU usage of a measured object	%	0%–100%
	CPU I/O Wait Time	ma_node_c pu_iowait_ counter	Disk I/O wait time accumulate d since system startup	jiffies	≥ 0
Memory	Physical Memory Usage	ma_node_ memory_ut il	Percentage of the used physical memory to the total physical memory	%	0%–100%
	Total Physical Memory	ma_node_ memory_to tal_megab ytes	Total physical memory that has been applied for a measured object	MB	≥ 0

Classification	Name	Metric	Description	Unit	Value Range
Network I/O	Downlink Rate (BPS)	ma_node_network_receive_rate_bytes_seconds	Inbound traffic rate of a measured object	Bytes/s	≥ 0
	Uplink Rate (BPS)	ma_node_network_transmit_rate_bytes_seconds	Outbound traffic rate of a measured object	Bytes/s	≥ 0
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
	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
	Total Cache	ma_node_cache_space_capacity_megabytes	Total cache of the Kubernetes space	MB	≥ 0
	Used Cache	ma_node_cache_space_used_capacity_megabytes	Used cache of the Kubernetes space	MB	≥ 0
	Total Container Space	ma_node_container_space_capacity_megabytes	Total container space	MB	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	Used Container Space	ma_node_c ontainer_sp ace_used_c apacity_me gabytes	Used container space	MB	≥ 0
	Disk Informatio n	ma_node_d isk_info	Basic disk informatio n	N/A	≥ 0
	Total Reads	ma_node_d isk_reads_c ompleted_t otal	Total number of successful reads	N/A	≥ 0
	Merged Reads	ma_node_d isk_reads_ merged_tot al	Number of merged reads	N/A	≥ 0
	Bytes Read	ma_node_d isk_read_by tes_total	Total number of bytes that are successfully read	Bytes	≥ 0
	Read Time Spent	ma_node_d isk_read_ti me_second s_total	Time spent on all reads	Seconds	≥ 0
	Total Writes	ma_node_d isk_writes_ completed_ total	Total number of successful writes	N/A	≥ 0
	Merged Writes	ma_node_d isk_writes_ merged_tot al	Number of merged writes	N/A	≥ 0
	Written Bytes	ma_node_d isk_written _bytes_tota l	Total number of bytes that are successfully written	Bytes	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	Write Time Spent	ma_node_d isk_write_ti me_second s_total	Time spent on all write operations	Seconds	≥ 0
	Ongoing I/Os	ma_node_d isk_io_now	Number of ongoing I/Os	N/A	≥ 0
	I/O Execution Duration	ma_node_d isk_io_time _seconds_t otal	Time spent on executing I/Os	Seconds	≥ 0
	I/O Execution Weighted Time	ma_node_d isk_io_time _weighted_ seconds_to ta	Weighted time spent on executing I/Os	Seconds	≥ 0
GPU	GPU Usage	ma_node_g pu_util	GPU usage of a measured object	%	0%–100%
	Total GPU Memory	ma_node_g pu_mem_t otal_mega bytes	Total GPU memory of a measured object	MB	> 0
	GPU Memory Usage	ma_node_g pu_mem_u til	Percentage of the used GPU memory to the total GPU memory	%	0%–100%
	Used GPU Memory	ma_node_g pu_mem_u sed_megab ytes	GPU memory used by a measured object	MB	≥ 0
	Tasks on a Shared GPU	node_gpu_ share_job_c ount	Number of tasks running on a shared GPU	Number	≥ 0

Classification	Name	Metric	Description	Unit	Value Range
	GPU Temperature	DCGM_FI_DEV_GPU_TEMP	GPU temperature	°C	Natural number
	GPU Power	DCGM_FI_DEV_POWER_USAGE	GPU power	Watt (W)	> 0
	GPU Memory Temperature	DCGM_FI_DEV_MEMORY_TEMP	GPU memory temperature	°C	Natural number
InfiniBand or RoCE network	Total Amount of Data Received by an 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.	Double words (32 bits)	≥ 0
	Total Amount of Data Sent by an 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.	Double words (32 bits)	≥ 0

Classification	Name	Metric	Description	Unit	Value Range
NFS mounting status	NFS Getattr Congestion Time	ma_node_ mountstats _getattr_ba cklog_wait	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, 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

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Getattr Round Trip Time	ma_node_ mountstats _getattr_rtt	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permission s, owner, etc. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Access Congestion Time	ma_node_ mountstats _access_ba cklog_wait	Access is an NFS operation that checks the access permission s of a file or directory for a given user. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performanc e and slow system response times.	ms	≥ 0

Classification	Name	Metric	Description	Unit	Value Range
	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

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

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

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

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

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

Classificati on	Name	Metric	Descriptio n	Unit	Value Range
	NFS Write Round Trip Time	ma_node_ mountstats _write_rtt	Write is an NFS operation that writes data to a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply34. RTT includes network transit time and server execution time. RTT is a good measurem ent for NFS latency. A high RTT can indicate network or server issues.	ms	≥ 0

Table 3-3 Diagnosis (IB, collected only in dedicated resource pools)

Classification	Name	Metric	Description	Unit	Value Range
InfiniBand or RoCE network	PortXmitData	infiniband_port_xmit_data_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), transmitted on all VLs from the port.	Total count	Natural number
	PortRcvData	infiniband_port_rcv_data_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), received on all VLs from the port.	Total count	Natural number
	SymbolErrorCounter	infiniband_symbol_error_counter_total	Total number of minor link errors detected on one or more physical lanes.	Total count	Natural number
	LinkErrorRecoveryCounter	infiniband_link_error_recovery_counter_total	Total number of times the Port Training state machine has successfully completed the link error recovery process.	Total count	Natural number
	PortRcvErrors	infiniband_port_rcv_errors_total	Total number of packets containing errors that were received on the port including: Local physical errors (ICRC, VCRC, LPCRC, and all physical errors that cause entry into the BAD PACKET or BAD PACKET DISCARD states of the packet receiver state machine) Malformed data packet errors (LVer, length, VL) Malformed link packet errors (operand, length, VL) Packets discarded due to buffer overrun (overflow)	Total count	Natural number

Classification	Name	Metric	Description	Unit	Value Range
	LocalLinkIntegrityErrors	infiniband_local_link_integrity_errors_total	This counter indicates the number of retries initiated by a link transfer layer receiver.	Total count	Natural number
	PortRcvRemotePhysicalErrors	infiniband_port_rcv_remote_physical_errors_total	Total number of packets marked with the EBP delimiter received on the port.	Total count	Natural number
	PortRcvSwitchRelayErrors	infiniband_port_rcv_switch_relay_errors_total	Total number of packets received on the port that were discarded when they could not be forwarded by the switch relay for the following reasons: DLID mapping VL mapping Looping (output port = input port)	Total count	Natural number
	PortXmitWait	infiniband_port_transmit_wait_total	The number of ticks during which the port had data to transmit but no data was sent during the entire tick (either because of insufficient credits or because of lack of arbitration).	Total count	Natural number
	PortXmitDiscards	infiniband_port_xmit_discards_total	Total number of outbound packets discarded by the port because the port is down or congested.	Total count	Natural number

Table 3-4 Metric names

Classification	Metric	Description
Container metrics	modelarts_service	Service to which a container belongs, which can be **notebook**, **train**, or **infer**
	instance_name	Name of the pod to which the container belongs

Classification	Metric	Description
	service_id	Instance or job ID displayed on the page, for example, **cf55829e-9bd3-48fa-8071-7ae870dae93a** for a development environment **9f322d5a-b1d2-4370-94df-5a87de27d36e** for a training job
	node_ip	IP address of the node to which the container belongs
	container_id	Container ID
	cid	Cluster ID
	container_name	Name of the container
	project_id	Project ID of the account to which the user belongs
	user_id	User ID of the account to which the user who submits the job belongs
	npu_id	Ascend card ID, for example, **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
	logical_pool_id	ID of a logical subpool
	logical_pool_name	Name of a logical subpool
	gpu_uuid	UUID of the GPU used by the container
	gpu_index	Index of the GPU used by the container
	gpu_type	Type of the GPU used by the container
	account_name	Account name of the creator of a training, inference, or development environment task
	user_name	Username of the creator of a training, inference, or development environment task

Classification	Metric	Description
	task_creation_time	Time when a training, inference, or development environment task is created
	task_name	Name of a training, inference, or development environment task
	task_spec_code	Specifications of a training, inference, or development environment task
	cluster_name	CCE cluster name
Node metrics	cid	ID of the CCE cluster to which the node belongs
	node_ip	IP address of the node
	host_name	Hostname of a node
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	project_id	Project ID of the user in a physical dedicated resource pool
	npu_id	Ascend card ID, for example, **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 IB NIC
	filesystem	NFS-mounted file system
	mount_point	NFS mount point
Diagnos	cid	ID of the CCE cluster to which the node where the GPU resides belongs
	node_ip	IP address of the node where the GPU resides

Classification	Metric	Description
	pool_id	ID of a resource pool corresponding to a physical dedicated resource pool
	project_id	Project ID of the user in a physical dedicated resource pool
	gpu_uuid	GPU UUID
	gpu_index	Index of a node GPU
	gpu_type	Type of a node GPU
	device_name	Name of a network device or disk device
	port	Port number of the IB NIC
	physical_state	Status of each port on the IB NIC
	firmware_version	Firmware version of the IB NIC