# Cloud Eye

# FAQs

**Issue** 05

**Date** 2023-09-15

**HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.**

### Trademarks and Permissions


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

### Notice

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

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

## 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 Product Usage



## 1.1 Server Monitoring

### 1.1.1 How Do I Configure DNS and Security Groups?

This topic describes how to add DNS server addresses and security groups to a Linux ECS to ensure successful Agent downloading and monitoring data collection. Here, ECSs are used as an example. The operations for other types of hosts are similar.

You can modify DNS configurations of an ECS in either of the following ways: command lines and management console. You can choose one as needed.

NOTE

DNS and security group configurations are intended for the primary NIC.

#### DNS

- **Modifying a DNS Server Address (Command Lines)**

  The following describes how to add a DNS server address to the **resolv.conf** file using command lines.

  To use the management console, see **Modifying a DNS Server Address (Management Console)**.

  a. Log in to an ECS as user **root**.

  b. Run the **vi /etc/resolv.conf** command to open the **resolv.conf** file.

  c. Add **nameserver 100.125.1.250** and **nameserver 100.125.21.250** to the file. Enter **:wq**, and press **Enter** to save the settings and exit.

**Figure 1-1** Adding DNS server addresses (Linux)

```
# Generated by NetworkManager
search openstacklocal
nameserver 100.125.1.250
nameserver 100.125.21.250
options single-request-reopen
```

NOTE

The **nameserver** value varies depending on the region. For details, see **What Are Huawei Cloud Private DNS Server Addresses?**

- **Modifying a DNS Server Address (Management Console)**

  The following describes how to modify a DNS server address of an ECS on the management console. Here, ECSs are used as an example. The operations for BMSs are similar.

  a. Log in to the management console.

  b. In the upper left corner, select a region and project.

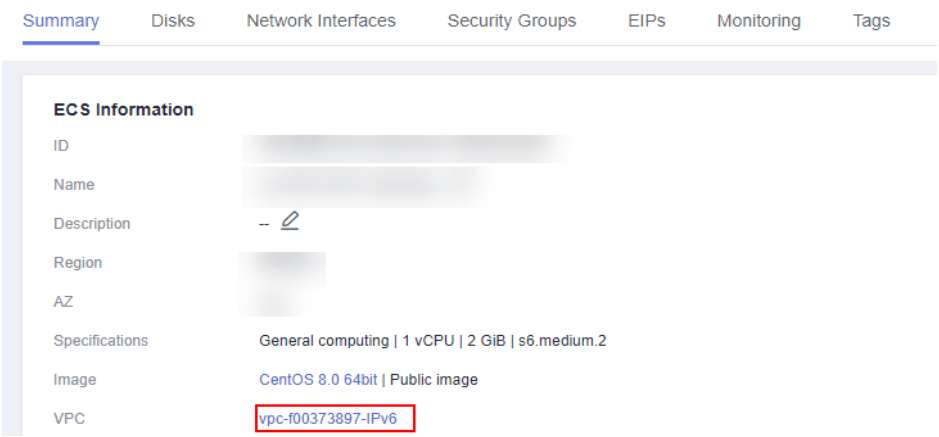
  c. Click **Service List** in the upper left corner. Under **Compute**, select **Elastic Cloud Server**.

     In the ECS list, click an ECS name to view its details.

  d. In the **ECS Information** area of the **Summary** tab, click the VPC name as is shown in **Figure 1-2**.

     The **Virtual Private Cloud** page is displayed.

     **Figure 1-2** VPC in ECS basic information

  e. Click the VPC name.

  f. In the **Networking Components** area, click the number next to **Subnets**.

     The **Subnets** page is displayed.

  g. In the subnet list, click the subnet name.

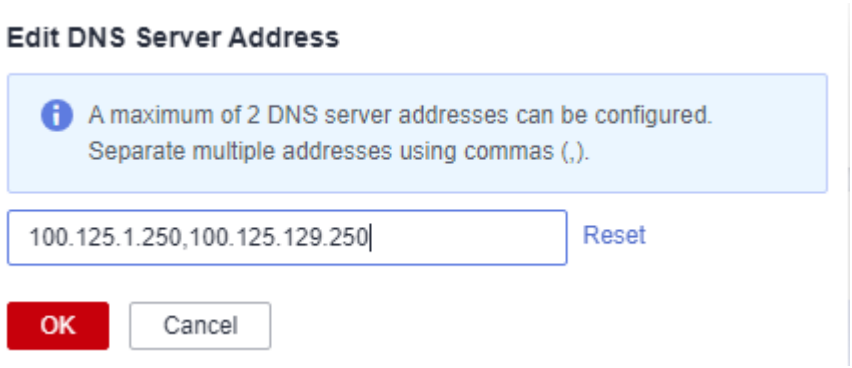
  h. In the **Gateway and DNS Information** area, click  after the **DNS Server Address**.

NOTE

Set the DNS server address to the value of **nameserver** in 3.

**Figure 1-3** Changing DNS server addresses




i. Click **OK**.

NOTE

The new DNS server address is applied after the ECS or BMS is restarted.

## Security Groups

- **Modifying the ECS Security Group Rules (Management Console)**

The following describes how to modify security group rules for an ECS on the management console. ECSs are used as an example. The operations for BMSs are similar.

1. On the ECS details page, select the **Security Groups** tab.
   The security group list is displayed.
2. Click a security group name.
3. Click **Modify Security Group Rule**.
   The security group details page is displayed.

   NOTE

   Procedure for BMS:
   1. Click the security group ID on the upper left corner of the list.
   2. Click **Manage Rule** in the **Operation** column of the security group.
4. In the **Outbound Rules** tab, click **Add Rule**.
5. Add rules based on **Table 1-1**.

**Table 1-1** Security group rules

| Protocol | Port | Type | Destination IP Address | Description |
|---|---|---|---|---|
| TCP | 80 | IPv4 | 100.125.0.0/16 | Used to download the Agent installation package from an OBS bucket to an ECS or BMS and obtain the ECS or BMS metadata and authentication information. |
| TCP | 53 | IPv4 | 100.125.0.0/16 | Used by DNS to resolve domain names, for example, the OBS domain name for downloading the Agent installation package, and the Cloud Eye endpoint for sending monitoring data to Cloud Eye. |
| UDP | 53 | IPv4 | 100.125.0.0/16 | Used by DNS to resolve domain names, for example, the OBS domain name for downloading the Agent installation package, and the Cloud Eye endpoint for sending monitoring data to Cloud Eye. |
| TCP | 443 | IPv4 | 100.125.0.0/16 | Used to collect monitoring data to Cloud Eye. |

## 1.1.2 How Do I Configure an Agency?

To enable you to monitor servers more securely and efficiently, Cloud Eye provides the latest Agent permission-granting method. That is, before installing Agents, you only need to click **Configure** on the **Server Monitoring** page of the Cloud Eye console, or select **cesgency** for **Agency** in **Advanced Options** when buying an ECS, the system automatically performs temporary AK/SK authorization for the Agents installed on all ECSs or BMSs in the region. And in the future, newly created ECSs or BMSs in this region will automatically get this authorization. This section describes the authorization as follows:

- Authorization object

  On the Cloud Eye console, if you choose **Server Monitoring** > **Elastic Cloud Server** (or **Bare Metal Server**), selecting an ECS (or BMS), and click **One-Click Restore**, the system automatically creates an agency named **cesagency** on IAM. The agency permissions are automatically granted to Cloud Eye internal account **op_svc_ces**.

> NOTE
>
> If the system displays a message indicating that you do not have the required permissions, see **2.1.3 What Can I Do If the System Displays a Message Indicating Insufficient Permissions When I Click Configure on the Server Monitoring Page?**

- Authorization scope

  Add the **CES AgentAccess** permissions to internal account **op_svc_ces** in the region.

- Authorization reason

  The Cloud Eye Agent runs on ECSs or BMSs and reports the collected monitoring data to Cloud Eye. After being authorized, the Agent automatically obtains a temporary AK/SK. This way, you can query the ECS or BMS monitoring data on the Cloud Eye console or by calling the Cloud Eye APIs.

  a. Security: The AK/SK used by the Agent is only the temporary AK/SK that has the **CES AgentAccess** permissions. That is, the temporary AK/SK can only be used to operate Cloud Eye resources.

  b. Convenient: You only need to configure the Cloud Eye Agent once in each region instead of manually configuring each Agent.

## 1.1.3 How Does the Cloud Eye Agent Obtain a Temporary AK/SK?

To enable you to monitor servers more securely and efficiently, Cloud Eye provides the latest Agent permission-granting method. That is, before installing Agents, you only need to click **Configure** on the **Server Monitoring** page of the Cloud Eye console, or select **cesgency** for **Agency** in **Advanced Options** when creating an ECS. The system automatically authorizes the Agent and provides a temporary AK/SK for the Agent. New ECSs or BMSs in this region will automatically get this authorization, which is detailed as follows:

- Authorization object

  On the Cloud Eye console, if you choose **Server Monitoring** > **Elastic Cloud Server** (or **Bare Metal Server**), selecting an ECS (or BMS), and click **One-Click Restore**, the system automatically creates an agency named **cesagency** on IAM. The agency permissions are automatically granted to Cloud Eye internal account **op_svc_ces**.

  > NOTE
  >
  > If the system displays a message indicating that you do not have the required permissions, see **2.1.3 What Can I Do If the System Displays a Message Indicating Insufficient Permissions When I Click Configure on the Server Monitoring Page?**

- Authorization scope

  Add the **CES Administrator** permission to internal account **op_svc_ces** in the region.

- Authorization reason

  The Cloud Eye Agent runs on ECSs or BMSs and reports the collected monitoring data to Cloud Eye. After being authorized, the Agent automatically obtains a temporary AK/SK. This way, you can query the ECS or BMS monitoring data on the Cloud Eye console or by calling the Cloud Eye APIs.

- Secure: The AK/SK used by the Agent is temporary and has only the **CES Administrator** permissions to allow you to operate Cloud Eye resources.
- Convenient: You only need to configure the Cloud Eye Agent once in each region instead of manually configuring each Agent.

If **cesagency** cannot be found on the IAM **Agencies** page after authorization, you can manually create it on the IAM console. For details, see **Creating an Agency (by a Delegating Party)**.

NOTE

- The name of the agency to be created must be **cesagency**.
- If **Agency Type** is set to **Common account**, **Delegated Account** must be **op_svc_ces**.

## 1.1.4 What OSs Does the Agent Support?

The following table lists OSs that are proven to be compatible with the Agent. More OSs will be supported soon.

NOTICE

The following systems are created using public images or those provided by Huawei Cloud Image Management Service (IMS). Using an unverified external system may lead to dependency issues or introduce other instability factors.

| Operating System | Version | Agent Installation (ECS) | One-Click Agent Installation (ECS) | Agent Installation (BMS) |
|---|---|---|---|---|
| Windows | Windows 2012 | √ | × | √ |
| | Windows 2016 | √ | × | √ |
| | Windows 2019 | √ | × | √ |
| | Windows 2022 | √ | × | √ |
| CentOS | CentOS 6.9 64bit | √ | × | × |
| | CentOS 6.10 64bit | √ | × | × |
| | CentOS 7.2 64bit | √ | √ | √ |
| | CentOS 7.3 64bit | √ | √ | √ |
| | CentOS 7.4 64bit | √ | √ | √ |

| Operating System | Version | Agent Installation (ECS) | One-Click Agent Installation (ECS) | Agent Installation (BMS) |
|---|---|---|---|---|
| | CentOS 7.5 64bit | √ | √ | × |
| | CentOS 7.6 64bit | √ | √ | √ |
| | CentOS 7.6 64bit(ARM) | × | × | √ |
| | CentOS 7.7 64bit | √ | √ | × |
| | CentOS 7.8 64bit | √ | √ | × |
| | CentOS 7.9 64bit | √ | √ | √ |
| | CentOS 8.0 64bit | √ | √ | × |
| | CentOS 8.1 64bit | √ | √ | × |
| | CentOS 8.2 64bit | √ | √ | × |
| | CentOS Stream 8/x86 | √ | × | × |
| | CentOS Stream 8/Arm | √ | × | × |
| | CentOS Stream 9/x86 | √ | × | × |
| Alma Linux | AlmaLinux 8.3 64bit | √ | √ | × |
| | AlmaLinux 8.4 64bit | √ | √ | × |
| | AlmaLinux 8.6 64bit | √ | × | × |
| | AlmaLinux 8.7 | √ | × | × |
| | AlmaLinux 9.1 | √ | × | × |
| | AlmaLinux 9.0 64bit | √ | √ | × |

| Operating System | Version | Agent Installation (ECS) | One-Click Agent Installation (ECS) | Agent Installation (BMS) |
|---|---|---|---|---|
| Debian | Debian 9.0.0 64bit | √ | × | × |
| | Debian 8.8.0 64bit | √ | × | × |
| | Debian 8.2.0 64bit | √ | × | × |
| | Debian 10.0.0 64bit | √ | × | × |
| | Debian 10.2.0 64bit(ARM) | √ | × | × |
| | Debain10.5 64bit | √ | × | × |
| | Debain10.6 64bit | √ | × | × |
| | Debain11.10 64bit | √ | √ | × |
| | Debian 11.4 64bit | √ | × | × |
| | Debian 11.5 64bit | √ | × | × |
| | Debain12.0.0 64bit | √ | √ | × |
| EulerOS | EulerOS 2.8 64bit | × | × | √ |
| | EulerOS 2.5 64bit | √ | √ | × |
| | EulerOS 2.3 64bit | × | × | √ |
| | EulerOS 2.2 64bit | √ | × | × |
| | EulerOS 2.8 64bit(ARM) | √ | × | √ |
| | EulerOS 2.9 64bit | √ | √ | √ |
| | EulerOS 2.9 64bit(ARM) | √ | × | × |

| Operating System | Version | Agent Installation (ECS) | One-Click Agent Installation (ECS) | Agent Installation (BMS) |
|---|---|---|---|---|
| | EulerOS 2.10 64bit | √ | √ | √ |
| Fedora | Fedora 30 64bit | √ | × | × |
| | Fedora 31 64bit | √ | × | × |
| | Fedora 36 64bit | √ | × | × |
| | Fedora 37 64bit | √ | × | × |
| | Fedora 38 64bit | √ | × | × |
| Huawei Cloud EulerOS | Huawei Cloud EulerOS 1.0 64bit | √ | × | × |
| | Huawei Cloud EulerOS 1.1 64bit | √ | √ | × |
| | Huawei Cloud EulerOS 2.0 64bit | √ | √ | √ |
| | Huawei Cloud EulerOS 2.0 ARM 64bit | √ | √ | √ |
| KylinOS | Kylin Linux Advanced Server for Kunpeng V1 | √ | × | × |
| | Kylin-Server-10-SP2-20210524-x86.iso | √ | × | × |
| | Kylin-Server-10-SP2-20210524-arm.iso | √ | × | × |
| openEuler | openEuler 20.03 64bit | √ | × | × |
| | openEuler 20.03 LTS SP3 64bit | √ | × | × |
| | openEuler 22.03 LTS(ARM) | × | × | √ |

| Operating System | Version | Agent Installation (ECS) | One-Click Agent Installation (ECS) | Agent Installation (BMS) |
|---|---|---|---|---|
| | openEuler 22.03 LTS 64bit | √ | × | × |
| OpenSUSE | OpenSUSE 15.0 64bit | √ | × | × |
| | SUSE 15 SP5 X86 | √ | × | × |
| | SUSE 15 SP6 X86 | √ | × | × |
| Redhat | Redhat Linux Enterprise 6.9 64bit | × | × | √ |
| | Redhat Linux Enterprise 7.4 64bit | × | × | √ |
| Rocky Linux | Rocky Linux 8.4 64bit | √ | √ | × |
| | Rocky Linux 8.5 64bit | √ | √ | × |
| | Rocky Linux 8.6 64 bit | √ | × | × |
| | Rocky Linux 9.0 64bit | √ | √ | × |
| | Rocky Linux 9.1 | √ | × | × |
| | Rocky Linux 8.7-X86 | √ | × | × |
| | Rocky Linux 9.3-X86 | √ | × | × |
| | Rocky Linux 8.7-ARM | √ | × | × |
| Ubuntu | Ubuntu 22.04 server 64bit | √ | √ | × |
| | Ubuntu 20.04 server 64bit | √ | √ | √ |
| | Ubuntu 18.04 server 64bit | √ | √ | √ |

| Operating System | Version | Agent Installation (ECS) | One-Click Agent Installation (ECS) | Agent Installation (BMS) |
|---|---|---|---|---|
| | Ubuntu 18.04 server 64bit(ARM) | × | × | √ |
| | Ubuntu 16.04 server 64bit | √ | √ | √ |
| | Ubuntu 14.04 server 64bit | × | × | √ |
| | Ubuntu 18.04.6 server 64bit | √ | × | × |
| UnionTechOS | UnionTech OS Server 20 Euler (1000) 64bit(ARM) | √ | × | × |
| | UnionTech OS-Server-20-1050e-amd64-UFU.iso | √ | × | × |

# 1.1.5 What Are Resource Usage and Circuit Breaker Pattern of Agent?

## Resource Usage

The Agent uses very few system resources. The Agent will use 10% of a CPU core at most. Its memory usage will not exceed 200 MB. Generally, the CPU usage for a single core is less than 5% and the memory usage is less than 100 MB.

## Circuit Breaker Pattern

When the CPU usage of a single core is greater than 10%, or the memory usage exceeds 200 MB three times in a row, the Agent will implement the circuit breaker pattern, and server metrics will not be collected. The Agent will restart it later.

# 1.1.6 Will the Agent Affect the Server Performance?

The Agent uses a small portion of system resources and basically it will not affect server performance.

- Agent resource usage for an ECS is as follows:

  No more than 10% of a CPU core and no more than 200 MB of memory. Generally, the CPU usage of a single core is less than 5%, and the memory is less than 100 MB.

- Agent resource usage for a BMS is as follows:

  No more than 10% of a CPU core and no more than 200 MB of memory. Generally, the CPU usage of a single core is less than 5%, and the memory is less than 100 MB.

# 1.1.7 What Metrics Are Supported by the Agent?

## OS metric: CPU

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage | (Agent) CPU Usage | Used to monitor CPU usage<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s)** value.<br>• Collection method (Windows): Obtain the metric value using the API **GetSystemTimes**. | 0-100 | % | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_idle | (Agent) Idle CPU Usage | Percentage of the time that CPU is idle<br>Unit: Percent<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period.<br>• Collection method (Windows): Obtain the metric value using the API **GetSystemTimes**. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_other | (Agent) Other Process CPU Usage | Other CPU usage of the monitored object<br>• Collection method (Linux): **Other Process CPU Usage** = 1– **Idle CPU Usage** – **Kernel Space CPU Usage** – **User Space CPU Usage**<br>• Collection method (Windows): **Other Process CPU Usage** = 1– **Idle CPU Usage** – **Kernel Space CPU Usage** – **User Space CPU Usage** | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_system | (Agent) Kernel Space CPU Usage | Percentage of time that the CPU is used by kernel space<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s) sy** value.<br>• Collection method (Windows): Obtain the metric value using the API **GetSystemTimes**. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_user | (Agent) User Space CPU Usage | Percentage of time that the CPU is used by user space<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s) us** value.<br>• Collection method (Windows): Obtain the metric value using the API **GetSystemTimes**. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_nice | (Agent) Nice Process CPU Usage | Percentage of the time that the CPU is in user mode with low-priority processes which can easily be interrupted by higher-priority processes<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s) ni** value.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_iowait | (Agent) iowait Process CPU Usage | Percentage of time that the CPU is waiting for I/O operations to complete<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s) wa** value.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_irq | (Agent) CPU Interrupt Time | Percentage of time that the CPU is servicing interrupts<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s) hi** value.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| cpu_usage_softirq | (Agent) CPU Software Interrupt Time | Percentage of time that the CPU is servicing software interrupts<br>• Collection method (Linux): Check the metric value changes in file **/proc/stat** in a collection period. You can run the **top** command to check the **%Cpu(s) si** value.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |

## OS Metric: CPU Load

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| load_average1 | (Agent) 1-Minute Load Average | CPU load averaged from the last 1 minute<br>• Collection method (Linux): Obtain the metric value from the number of logic CPUs in **load1/** in file **/proc/loadavg**. You can run the **top** command to check the **load1** value. | ≥0 | None | N/A | 2.4.1 | 1 minute |
| load_average5 | (Agent) 5-Minute Load Average | CPU load averaged from the last 5 minutes<br>• Collection method (Linux): Obtain the metric value from the number of logic CPUs in **load5/** in file **/proc/loadavg**. You can run the **top** command to check the **load5** value. | ≥0 | None | N/A | 2.4.1 | 1 minute |
| load_average15 | (Agent) 15-Minute Load Average | CPU load averaged from the last 15 minutes<br>• Collection method (Linux): Obtain the metric value from the number of logic CPUs in **load15/** in file **/proc/loadavg**. You can run the **top** command to check the **load15** value. | ≥0 | None | N/A | 2.4.1 | 1 minute |

## OS Metric: Memory

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| mem_available | (Agent) Available Memory | Amount of memory that is available and can be given instantly to processes<br>• Collection method (Linux): Obtain the metric value from **/proc/meminfo**. If **MemAvailable** is displayed in **/proc/meminfo**, obtain the value.<br>If **MemAvailable** is not displayed in **/proc/meminfo**, **MemAvailable** = **MemFree** + **Buffers** **+Cached**<br>• Collection method (Windows): formula (Available memory – Used memory) The value is obtained by calling the Windows API GlobalMemoryStatusEx. | ≥0 | GB | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| mem_usedPercent | (Agent) Memory Usage | Memory usage of the instance<br>• Collection method (Linux): Obtain the metric value from the **/proc/meminfo** file (**MemTotal**-**MemAvailable**)/**MemTotal**.<br>If **MemAvailable** is displayed in **/proc/meminfo**, **MemUsedPercent** = (**MemTotal**-**MemAvailable**)/**MemTotal**<br>If **MemAvailable** is not displayed in **/proc/meminfo**, **MemUsedPercent** = (**MemTotal** – **MemFree** – **Buffers** – **Cached**)/**MemTotal**<br>• Collection method (Windows): formula (Used memory size/Total memory size x 100%) | 0-100 | % | N/A | 2.4.1 | 1 minute |
| mem_free | (Agent) Idle Memory | Amount of memory that is not being used<br>• Linux: Obtain the metric value from **/proc/meminfo**.<br>• Windows does not support this metric. | ≥0 | GB | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| mem_buffers | (Agent) Buffer | Amount of memory that is being used for buffers<br>• Collection method (Linux): Obtain the metric value from **/proc/meminfo**. You can run the **top** command to check the **KiB Mem:buffers** value.<br>• Windows does not support this metric. | ≥0 | GB | N/A | 2.4.5 | 1 minute |
| mem_cached | (Agent) Cache | Amount of memory that is being used for file caches<br>• Collection method (Linux): Obtain the metric value from **/proc/meminfo**. You can run the **top** command to check the **KiB Swap:cached Mem** value.<br>• Windows does not support this metric. | ≥0 | GB | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| total_open_files | (Agent) Total File Handles | Total handles used by all processes<br>• Collection method (Linux): Use the **/proc/{pid}/fd** file to summarize the handles used by all processes.<br>• Windows does not support this metric. | ≥0 | None | N/A | 2.4.5 | 1 minute |

## OS Metric: Disk

NOTE

Currently, CES Agent can collect only physical disk metrics and does not support disks mounted using the network file system protocol.

By default, CES Agent will not monitor Docker-related mount points. The prefix of the mount point is as follows:

```
/var/lib/docker;/mnt/paas/kubernetes;/var/lib/mesos
```

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_free | (Agent) Available Disk Space | Free space on the disks<br>• Collection method (Linux): Run the **df -h** command to check the value in the **Avail** column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use the Windows Management Instrumentation (WMI) API GetDiskFreeSpaceExW to obtain disk space data. The path of | ≥0 | GB | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). | | | | | |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_total | (Agent) Disk Storage Capacity | Total disk capacity<br>• Collection method (Linux): Run the **df -h** command to check the value in the **Size** column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use the WMI API GetDiskFreeSpaceExW to obtain disk space data. The path of the mount point prefix cannot exceed | ≥0 | GB | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). | | | | | |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_used | (Agent) Used Disk Space | Disk's used space<br>• Collection method (Linux): Run the **df -h** command to check the value in the **Used** column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use the WMI API GetDiskFreeSpaceExW to obtain disk space data. The path of the mount point prefix cannot exceed 64 characters. | ≥0 | GB | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). | | | | | |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_usedPercent | (Agent) Disk Usage | Percentage of used disk space. It is calculated as follows: **Disk Usage** = **Used Disk Space**/**Disk Storage Capacity**.<br>• Collection method (Linux): It is calculated as follows: Used/Size. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use the WMI API GetDiskFreeSpaceExW to obtain disk space data. The path of | 0-100 | % | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~). | | | | | |

## OS Metric: Disk I/O

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_agt_read_bytes_rate | (Agent) Disks Read Rate | Volume of data read from the instance per second<br>• Collection method (Linux): Calculate the data changes in the sixth column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain disk I/O data.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), | ≥ 0 | byte/s | 1024(IEC) | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | periods (.), and swung dashes (~).<br>When the CPU usage is high, monitoring data obtaining timeout may occur and monitoring data cannot be obtained. | | | | | |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_agt_read_requests_rate | (Agent) Disks Read Requests | Number of read requests sent to the monitored disk per second<br>• Collection method (Linux): The disk read requests are calculated by calculating the data changes in the fourth column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain disk I/O data.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), | ≥ 0 | Request/s | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | periods (.), and swung dashes (~).<br>When the CPU usage is high, monitoring data obtaining timeout may occur and monitoring data cannot be obtained. | | | | | |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_agt_write_bytes_rate | (Agent) Disks Write Rate | Volume of data written to the instance per second<br>• Collection method (Linux): The disk write rate is calculated by calculating the data changes in the tenth column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain disk I/O data.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), | ≥ 0 | byte/s | 1024(IEC) | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | periods (.), and swung dashes (~).<br>When the CPU usage is high, monitoring data obtaining timeout may occur and monitoring data cannot be obtained. | | | | | |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_agt_write_requests_rate | (Agent) Disks Write Requests | Number of write requests sent to the monitored disk per second<br>• Collection method (Linux): The disk write requests are calculated by calculating the data changes in the eighth column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Collection method (Windows): Use Win32_PerfFormattedData_PerfDisk_LogicalDisk object in WMI to obtain disk I/O data.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and | ≥ 0 | Request/s | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| | | contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>When the CPU usage is high, monitoring data obtaining timeout may occur and monitoring data cannot be obtained. | | | | | |
| disk_readTime | (Agent) Average Read Request Time | The average time taken for disk read operations<br>• Collection method (Linux):<br>The average read request time is calculated by calculating the data changes in the seventh column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | ms/count | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_writeTime | (Agent) Average Write Request Time | The average time taken for disk write operations<br>• Collection method (Linux): The average write request time is calculated by calculating the data changes in the eleventh column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | ms/count | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_ioUtils | (Agent) Disk I/O Usage | Percentage of the time that the disk has had I/O requests queued to the total disk operation time<br>• Collection method (Linux): The disk I/O usage is calculated by calculating the data changes in the thirteenth column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_queue_length | (Agent) Disk Queue Length | Average number of read or write requests queued up for completion for the monitored disk in the monitoring period<br>• Collection method (Linux): The average disk queue length is calculated by calculating the data changes in the fourteenth column of the corresponding device in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_write_bytes_per_operation | (Agent) Average Disk Write Size | Average number of bytes in an I/O write for the monitored disk in the monitoring period<br>• Collection method (Linux): The average disk write size is calculated by calculating the data changes in the tenth column of the corresponding device to divide that of the eighth column in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | Byte/op | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_read_bytes_per_operation | (Agent) Average Disk Read Size | Average number of bytes in an I/O read for the monitored disk in the monitoring period<br>• Collection method (Linux): The average disk read size is calculated by using the data changes in the sixth column of the corresponding device to divide that of the fourth column in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | Byte/op | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_io_svctm | (Agent) Disk I/O Service Time | Average time in an I/O read or write for the monitored disk in the monitoring period<br>• Collection method (Linux): The average disk I/O service time is calculated by using the data changes in the thirteenth column of the corresponding device to divide the sum of data changes in the fourth and eighth columns in file **/proc/diskstats** in a collection period.<br>The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | ms/op | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_device_used_percent | Block Device Usage | Percentage of total disk space that is used. The calculation formula is as follows: Used storage space of all mounted disk partitions/Total disk storage space.<br>• Collection method (Linux): Summarize the disk usage of each mount point, calculate the total disk size based on the disk sector size and number of sectors, and calculate the overall disk usage.<br>• Currently, Windows does not support this metric. | 0-100 | % | N/A | 2.5.6 | 1 minute |

## OS Metric: File System

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_fs_rwstate | (Agent) File System Read/ Write Status | Read and write status of the mounted file system of the monitored object Possible statuses are **0** (read and write) and **1** (read only).<br>• Collection method (Linux): Check file system information in the fourth column in file **/proc/mounts**.<br>• Windows does not support this metric. | • **0**: readable and writable<br>• **1**: read-only | None | N/A | 2.4.5 | 1 minute |
| disk_inodesTotal | (Agent) Disk inode Total | Total number of index nodes on the disk<br>• Collection method (Linux): Run the **df -i** command to check the value in the **Inodes** column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | None | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| disk_inodesUsed | (Agent) Total inode Used | Number of used index nodes on the disk<br>• Collection method (Linux): Run the **df -i** command to check the value in the **IUsed** column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | ≥ 0 | None | N/A | 2.4.5 | 1 minute |
| disk_inodesUsedPercent | (Agent) Percentage of Total inode Used | Number of used index nodes on the disk<br>• Collection method (Linux): Run the **df -i** command to check the value in the **IUse %** column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), periods (.), and swung dashes (~).<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.1 | 1 minute |

## OS Metric: TCP

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_total | (Agent) Total Number of TCP Connections | Total number of TCP connections<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.1 | 1 minute |
| net_tcp_established | (Agent) Number of connections in the ESTABLISHED state | Number of TCP connections in the ESTABLISHED state<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.1 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_sys_sent | (Agent) Number of connections in the TCP SYS_SENT state. | Number of TCP connections that are being requested by the client<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |
| net_tcp_sys_recv | (Agent) Number of connections in the TCP SYS_RECV state. | Number of pending TCP connections received by the server<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_fin_wait1 | (Agent) Number of TCP connections in the FIN_WAIT1 state. | Number of TCP connections waiting for ACK packets when the connections are being actively closed by the client<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_fin_wait2 | (Agent) Number of TCP connections in the FIN_WAIT2 state. | Number of TCP connections in the FIN_WAIT2 state<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |
| net_tcp_time_wait | (Agent) Number of TCP connections in the TIME_WAIT state. | Number of TCP connections in the TIME_WAIT state<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_close | (Agent) Number of TCP connections in the CLOSE state. | Number of closed TCP connections<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |
| net_tcp_close_wait | (Agent) Number of TCP connections in the CLOSE_WAIT state. | Number of TCP connections in the CLOSE_WAIT state<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_last_ack | (Agent) Number of TCP connections in the LAST_ACK state. | Number of TCP connections waiting for ACK packets when the connections are being passively closed by the client<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_listen | (Agent) Number of TCP connections in the LISTEN state. | Number of TCP connections in the LISTEN state<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_closing | (Agent) Number of TCP connections in the CLOSING state. | Number of TCP connections to be automatically closed by the server and the client at the same time<br>• Collection method (Linux): Obtain TCP connections in all states from the **/proc/net/tcp** file, and then collect the number of connections in each state.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpTable2** API. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Metric | Metric | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_tcp_retrans | (Agent) TCP Retransmission Rate | Percentage of packets that are resent<br>• Collection method (Linux): Obtain the metric value from the **/proc/net/snmp** file. The value is the ratio of the number of sent packets to the number of retransmitted packages in a collection period.<br>• Collection method (Windows): Obtain the metric value using the **GetTcpStatistics** API. | 0-100 | % | N/A | 2.4.5 | 1 minute |

## OS Metric: NIC

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_bitRecv | (Agent) Outbound Bandwidth | Number of bits sent by this NIC per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Collection method (Windows): Use the MibIfRow object in WMI to obtain network metric data. | ≥ 0 | bit/s | 1024(IEC) | 2.4.1 | 1 minute |
| net_bitSent | (Agent) Inbound Bandwidth | Number of bits received by this NIC per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Collection method (Windows): Use the MibIfRow object in WMI to obtain network metric data. | ≥ 0 | bit/s | 1024(IEC) | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_packetRecv | (Agent) NIC Packet Receive Rate | Number of packets received by this NIC per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Collection method (Windows): Use the MibIfRow object in WMI to obtain network metric data. | ≥ 0 | Count/s | N/A | 2.4.1 | 1 minute |
| net_packetSent | (Agent) NIC Packet Send Rate | Number of packets sent by this NIC per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Collection method (Windows): Use the MibIfRow object in WMI to obtain network metric data. | ≥ 0 | Count/s | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_errin | (Agent) Receive Error Rate | Percentage of receive errors detected by this NIC per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |
| net_errout | (Agent) Transmit Error Rate | Percentage of transmit errors detected by this NIC per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |
| net_dropin | (Agent) Received Packet Drop Rate | Percentage of packets received by this NIC which were dropped per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| net_dropout | (Agent) Transmitted Packet Drop Rate | Percentage of packets transmitted by this NIC which were dropped per second<br>• Collection method (Linux): Check metric value changes in file **/proc/net/dev** in a collection period.<br>• Windows does not support this metric. | 0-100 | % | N/A | 2.4.5 | 1 minute |

## Process Monitoring Metrics

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_pHashId_cpu | (Agent) CPU Usage | CPU consumed by a process. **pHashId** (process name and process ID) is the value of **md5**.<br>• Collection method (Linux): Check the metric value changes in file **/proc/pid/stat**.<br>• Collection method (Windows): Call the Windows API GetProcessTimes to obtain the CPU usage of the process. | 0–1 x Number of vCPUs | % | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_pHashId_mem | (Agent) Memory Usage | Memory consumed by a process. **pHashId** (process name and process ID) is the value of **md5**.<br>• Collection method (Linux): RSS*PAGESIZE/ MemTotal<br>Obtain the **RSS** value by checking the second column of file **/proc/pid/statm**.<br>Obtain the **PAGESIZE** value by running the **getconf PAGESIZE** command.<br>Obtain the **MemTotal** value by checking file **/proc/meminfo**.<br>• Collection method (Windows): Call the Windows API **procGlobalMemoryStatusEx** to obtain the total memory size. Call **GetProcessMemoryInfo** to obtain the used memory size. Use the used memory size to divide the total memory size to get the memory usage. | 0-100 | % | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_pHashId_file | (Agent) Number of opened files | Number of files opened by a process. **pHashId** (process name and process ID) is the value of **md5**.<br>• Collection method (Linux): Run the **ls -l /proc/pid/fd** command to view the number of opened files.<br>• Windows does not support this metric. | ≥0 | Count | N/A | 2.4.1 | 1 minute |
| proc_running_count | (Agent) Number of running processes | Number of processes that are running<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Windows does not support this metric. | ≥0 | None | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_idle_count | (Agent) Idle Processes | Number of processes that are idle<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Windows does not support this metric. | ≥0 | None | N/A | 2.4.1 | 1 minute |
| proc_zombie_count | (Agent) Zombie Processes | Number of zombie processes<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Windows does not support this metric. | ≥0 | None | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_blocked_count | (Agent) Blocked Processes | Number of processes that are blocked<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Windows does not support this metric. | ≥0 | None | N/A | 2.4.1 | 1 minute |
| proc_sleeping_count | (Agent) Sleeping Processes | Number of processes that are sleeping<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Windows does not support this metric. | ≥0 | None | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_total_count | (Agent) Total Processes | Total number of processes on the monitored object<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Collection method (Windows): Obtain the total number of processes by using the system process status support module **psapi.dll**. | ≥0 | None | N/A | 2.4.1 | 1 minute |

| Metric | Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Monitoring Period (Raw Data) |
|---|---|---|---|---|---|---|---|
| proc_specified_count | (Agent) Specified Processes | Number of specified processes<br>• Collection method (Linux): You can obtain the state of each process by checking the **Status** value in the **/proc/pid/status** file, and then collect the total number of processes in each state.<br>• Collection method (Windows): Obtain the total number of processes by using the system process status support module **psapi.dll**. | ≥0 | None | N/A | 2.4.1 | 1 minute |

## GPU Specifications

NOTE

If a GPU server has eight GPU cards and the PM mode is disabled, data may fail to be collected. You can enable the PM mode and restart the monitoring process.

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| GPU Specifications | gpu_status | GPU health status of the VM. This metric is a composite metric.<br>• Possible causes: 1. The ECC exceeded the threshold. 2. The GPU memory address failed to be remapped. 3. The GPU card is in the **rev ff** state. 4. infoROM error. 5. There are pages to be isolated. 6. The remapped rows are incorrect. (For details, see the following detailed metrics.)<br>• Collection method (Linux): Call APIs from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU status.<br>• Collection method (Windows): Call APIs from the GPU driver library file **nvml.dll** to obtain the GPU status. | • **0**: healthy<br>• **1**: subhealthy<br>• **2**: faulty | None | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_performance_state | Performance status of the GPU<br>• P0-P15, P32<br>• **P0** indicates the maximum performance status. **P15** indicates the minimum performance status. **P32** indicates the unknown status.<br>• Collection mode (Linux): Call the **NvmlDeviceGetPerformanceState** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU performance level.<br>• Collection method (Windows): Call the **NvmlDeviceGetPerformanceState** API from the GPU driver library file **nvml.dll** to obtain the GPU performance level. | • P0-P15: **P0** indicates the maximum performance status, and **P15** indicates the minimum performance status.<br>• **P32** indicates the unknown status. | None | N/A | 2.4.1 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_power_draw | Power of the GPU.<br>• If the power exceeds the maximum power or is an incorrect value, the GPU hardware may be faulty.<br>• Collection method (Linux): Call the **NvmlDeviceGetPowerUsage** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU power.<br>• Collection method (Windows): Call the **NvmlDeviceGetPowerUsage** API from the GPU driver library file **nvml.dll** to obtain the GPU power. | ≥ 0 | W | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_temperature | Temperature of the GPU.<br>● If the temperature exceeds the maximum operating temperature threshold or is an incorrect value, the GPU hardware may be faulty.<br>● Collection method (Linux): Call the **NvmlDeviceGetTemperature** API from the GPU driver library file file **libnvidia-ml.so.1** to obtain the GPU temperature.<br>● Collection method (Windows): Call the **NvmlDeviceGetTemperature** API from the GPU driver library file **nvml.dll** to obtain the GPU temperature. | ≥ 0 | °C | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_usage_gpu | GPU computing power usage.<br>● The GPU computing power usage is displayed in percentage. The value is an instantaneous value at the sampling point.<br>● Collection method (Linux): Call the **NvmlDeviceGetUtilizationRates** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU computing power usage.<br>● Collection method (Windows): Call the **NvmlDeviceGetUtilizationRates** API from **nvml.dll** to obtain the GPU computing power usage. | 0-100 | % | N/A | 2.4.1 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_usage_mem | GPU memory usage.<br>● The GPU memory usage is displayed in percentage. The value is an instantaneous value at the sampling point.<br>● Collection method (Linux): Call the **NvmlDeviceGetUtilizationRates** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU memory usage.<br>● Collection method (Windows): Call the **NvmlDeviceGetUtilizationRates** API from **nvml.dll** to obtain the GPU memory usage. | 0-100 | % | N/A | 2.4.1 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_used_mem | GPU memory usage.<br>● The GPU memory usage is displayed in percentage. The value is an instantaneous value at the sampling point.<br>● Collection method (Linux): Call the **NvmlDeviceGetMemoryInfo** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU memory usage.<br>● Collection method (Windows): Call the **NvmlDeviceGetMemoryInfo** API from the GPU driver library file **nvml.dll** to obtain the GPU memory usage. | ≥ 0 | MB | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_free_mem | Remaining GPU memory.<br>• The idle GPU memory data is displayed.<br>• Collection method (Linux): Call the **NvmlDeviceGetMemoryInfo** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the remaining GPU memory.<br>• Collection method (Windows): Call the **NvmlDeviceGetMemoryInfo** API from **nvml.dll** to obtain the remaining GPU memory. | ≥ 0 | MB | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_usage_encoder | GPU encoder usage.<br>• The GPU encoder usage is displayed in percentage. The value is an instantaneous value at the sampling point.<br>• Collection method (Linux): Call the **NvmlDeviceGetEncoderUtilization** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU encoding capability usage.<br>• Collection method (Windows): Call the **NvmlDeviceGetEncoderUtilization** API from **nvml.dll** to obtain the GPU encoding capability usage. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_usage_decoder | GPU decoder usage.<br>• The GPU decoder usage is displayed in percentage. The value is an instantaneous value at the sampling point.<br>• Collection method (Linux): Call the **NvmlDeviceGetDecoderUtilization** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU decoding capability usage.<br>• Collection method (Windows): Call the **NvmlDeviceGetDecoderUtilization** API from **nvml.dll** to obtain the GPU decoding capability usage. | 0-100 | % | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_graphics_clocks | GPU graphics (shader) clock frequency.<br>• Displays the GPU clock frequencies related to graphics performance. If no graphics capability is used, you can ignore it.<br>• Collection method (Linux): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU graphics clock frequency.<br>• Collection method (Windows): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file **nvml.dll** to obtain the GPU graphics clock frequency. | ≥ 0 | MHz | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_sm_clocks | Streaming processor clock frequency of the GPU.<br>• Clock frequency for controlling the GPU memory running speed.<br>• Collection method (Linux): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file file **libnvidia-ml.so.1** to obtain the streaming processor clock frequency of the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file file **nvml.dll** to obtain the streaming processor clock frequency of the GPU. | ≥ 0 | MHz | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_mem_clocks | Memory clock frequency of the GPU.<br>• Displays the clock frequency closely related to CUDA core computing of the GPU.<br>• Collection method (Linux): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the GPU memory clock frequency.<br>• Collection method (Windows): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file **nvml.dll** to obtain the GPU memory clock frequency. | ≥ 0 | MHz | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_video_clocks | Video (including codec) clock frequency of the GPU.<br>• Displays the codec clock frequency of the current GPU.<br>• Collection method (Linux): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the video clock frequency of the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetClockInfo** API from the GPU driver library file **nvml.dll** to obtain the GPU video clock frequency. | ≥ 0 | MHz | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_tx_throughput_pci | Outbound bandwidth of the GPU.<br>• Displays the amount of data sent by the GPU to the host via PCIe.<br>• Collection method (Linux): Call the **NvmlDeviceGetPcieThroughput** API from **libnvidia-ml.so.1** to obtain the outbound bandwidth of the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetPcieThroughput** API from **nvml.dll** to obtain the outbound bandwidth of the GPU. | ≥ 0 | Mbyte/s | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_rx_throughput_pci | Inbound bandwidth of the GPU.<br>• Displays the amount of data sent by the host to the GPU via PCIe.<br>• Collection method (Linux): Call the **NvmlDeviceGetPcieThroughput** API from **libnvidia-ml.so.1** to obtain the inbound bandwidth of the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetPcieThroughput** API from **nvml.dll** to obtain the inbound bandwidth of the GPU. | ≥ 0 | Mbyte/s | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_volatile_correctable | Number of correctable ECC errors since the GPU is reset. The value is reset to 0 each time the GPU is reset.<br>• Collection method (Linux): Call the **NvmlDeviceGetPcieThroughput** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of correctable ECC errors since the GPU is reset.<br>• Collection method (Windows): Call the **NvmlDeviceGetPcieThroughput** API from the GPU driver library file **nvml.dll** to obtain the number of correctable ECC errors since the GPU is reset. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_volatile_uncorrectable | Number of uncorrectable ECC errors since the GPU is reset. The value is reset to 0 each time the GPU is reset.<br>• Collection method (Linux): Call the **NvmlDeviceGetTotalEccErrors** and **NvmlDeviceGetMemoryErrorCounter** APIs from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of uncorrectable ECC errors since the GPU is reset.<br>• Collection method (Windows): Call the **NvmlDeviceGetTotalEccErrors** and **NvmlDeviceGetMemoryErrorCounter** APIs from the GPU driver library file **nvml.dll** to obtain the number of uncorrectable ECC errors since the GPU is reset. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_aggregate_correctable | Number of correctable ECC errors on the GPU.<br>• Collection method (Linux): Call the **NvmlDeviceGetTotalEccErrors** and **NvmlDeviceGetMemoryErrorCounter** APIs from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of correctable ECC errors on the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetTotalEccErrors** and **NvmlDeviceGetMemoryErrorCounter** APIs from the GPU driver library file **nvml.dll** to obtain the number of correctable ECC errors on the GPU. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_aggregate_uncorrectable | Number of uncorrectable ECC Errors on the GPU.<br>• Collection method (Linux): Call the **NvmlDeviceGetTotalEccErrors** and **NvmlDeviceGetMemoryErrorCounter** APIs from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of uncorrectable ECC errors on the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetTotalEccErrors** and **NvmlDeviceGetMemoryErrorCounter** APIs from the GPU driver library file **nvml.dll** to obtain the number of uncorrectable ECC errors on the GPU. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_retired_page_single_bit | Number of retired page single bit errors, which indicates the number of single-bit pages isolated by the GPU.<br>• Collection method (Linux): Call the **NvmlDeviceGetRetiredPages** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of single-bit pages isolated by the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetRetiredPages** API from the GPU driver library file **nvml.dll** to obtain the number of single-bit pages isolated by the GPU. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_retired_page_double_bit | Number of retired page double bit errors, which indicates the number of double-bit pages isolated by the GPU.<br>• Collection method (Linux): Call the **NvmlDeviceGetRetiredPages** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of double-bit pages isolated by the GPU.<br>• Collection method (Windows): Call the **NvmlDeviceGetRetiredPages** API from the GPU driver library file **nvml.dll** to obtain the number of double-bit pages isolated by the GPU. | ≥ 0 | count | N/A | 2.4.5 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_lnkcap_speed | Maximum speed supported by the PCIe link of the GPU.<br>● Maximum data throughput capability of the GPU on the PCIe bus.<br>● Collection method (Linux): Use **lspci -d 10de: -vv \| grep -i lnkcap** to query the maximum speed supported by the PCIe link of the GPU.<br>● Collection method (Windows): Use **gwmi Win32_Bus -Filter 'DeviceID like "PCI %"').GetRelated('Win32_PnPEntity')** to query the maximum speed supported by the PCIe link of the GPU. | ≥ 0 | GT/s | N/A | 2.6.7 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_lnkcap_width | Link width of the PCIe link.<br>● Maximum number of PCIe lanes supported by the GPU.<br>● Collection method (Linux): Use **lspci -d 10de: -vv \| grep -i lnksta** to query the maximum speed supported by the PCIe link of the GPU.<br>● Collection method (Windows): Use **gwmi Win32_Bus -Filter 'DeviceID like "PCI %"').GetRelated('Win32_PnPEntity')** to query the maximum speed supported by the PCIe link of the GPU. | ≥ 0 | count | N/A | 2.6.7 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_lnksta_speed | PCIe connection speed of the GPU.<br>● Maximum PCIe link speed supported by the GPU.<br>● Collection method (Linux): Use **lspci -d 10de: -vv \| grep -i lnkcap** to query the PCIe connection speed of the GPU.<br>● Collection method (Windows): Not supported. | ≥ 0 | GT/s | N/A | 2.6.7 | 1 minute |
| | gpu_lnksta_width | PCIe link width of the GPU.<br>● Maximum number of lanes in the PCIe link supported by the GPU.<br>● Collection method (Linux): Use **lspci -d 10de: -vv \| grep -i lnksta** to query the PCIe link bandwidth of the GPU.<br>● Collection method (Windows): Not supported. | ≥ 0 | count | N/A | 2.6.7 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_nvlink_number | Number of NVLink links of the GPU.<br>• Number of NVLink links supported by the GPU. For example, A100 supports 12 NVLink links.<br>• Collection method (Linux): Call the **nvmlDeviceGetFieldValue** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the number of NVLink links of the GPU.<br>• Collection method (Windows): Not supported. | ≥ 0 | count | N/A | 2.6.7 | 1 minute |

| Category | Metric Name | Description | Value Range | Unit | Conversion Rule | Supported Version | Collection Interval |
|---|---|---|---|---|---|---|---|
| | gpu_nvlink_bandwidth | NVLink link width of the GPU.<br>• Indicates the total bandwidth for data transmission used by the GPU.<br>• Collection method (Linux): Call the **nvmlDeviceGetFieldValue** API from the GPU driver library file **libnvidia-ml.so.1** to obtain the NVLink link width of the GPU.<br>• Collection method (Windows): Not supported. | ≥ 0 | GB/s | N/A | 2.6.7 | 1 minute |

## 1.1.8 Environment Constraints for GPU Monitoring

1. Only Linux OSs are supported, and only some Linux public image versions support GPU monitoring. For details, see **1.1.4 What OSs Does the Agent Support?**
2. Supported flavors: G6v, G6, P2s, P2v, P2vs, G5, Pi2, Pi1, ECSs of P1 series, the BMSs of the P, Pi, G, and KP series.
3. You have installed the Agent of the enhanced edition. For details, see **Installing the Agent**. **Table 1-2** describes the differences between Agents of the basic edition and enhanced edition.

**Table 1-2** Basic edition and enhanced edition of the Agent

| Edition | Description |
| --- | --- |
| Basic | Provides basic OS monitoring metrics, such as CPU, memory, disk, and NIC metrics, helping you improve system performance.<br>Generally, the version number consists of three digits, for example, 2.7.5. |
| Enhanced | Provides GPU, NPU, and BMS hardware monitoring, in addition to the capabilities provided in the basic edition.<br>Generally, the version number consists of four digits, for example, 2.7.5.1.<br>**CAUTION**<br>Install the Agent of the enhanced edition if you indeed need it because it collects more metrics, which may occupy more server resources. |

4. GPU metric collection depends on the following driver files. Check whether there are corresponding driver files in the environment.

   a. Linux driver file

   ```
   nvmlUbuntuNvidiaLibraryPath = "/usr/lib/x86_64-linux-gnu/libnvidia-ml.so.1"
   nvmlCentosNvidiaLibraryPath = "/usr/lib64/libnvidia-ml.so.1"
   nvmlCceNvidiaLibraryPath    = "/opt/cloud/cce/nvidia/lib64/libnvidia-ml.so.1"
   ```

   b. Windows driver file

   ```
   DefaultNvmlDLLPath = "C:\\Program Files\\NVIDIA Corporation\\NVSMI\\nvml.dll"
   WHQLNvmlDLLPath    = "C:\\Windows\\System32\\nvml.dll"
   ```

## 1.1.9 BMS Hardware Metrics

The following table describes BMS hardware monitoring metrics and how the metrics are collected.

| Metrics | Description | Collected by |
| --- | --- | --- |
| Server information | Includes the server SN, product name, manufacturer. | Running the **dmidecode** command |
| Solid state drive (SSD) and hard disk drive (HDD) basic information and Self-Monitoring Analysis and Reporting Technology (SMART) information | Includes basic information (such as the SN, model, capacity, protocol type, and firmware version) and indicators (such as the health status, temperature, number of bad blocks, number of errors, and number of failures) in the SMART log of the SSD and HDD. | Running the **smartctl -a** *<Drive letter>* command |

| Metrics | Description | Collected by |
| --- | --- | --- |
| Basic information about the Non-Volatile Memory Express (NVMe) SSD | Includes SN, model, capacity, and firmware version. | Running the **nvme list** command |
| Standard SMART information of the NVMe SSD | Includes indicators in the SMART log of the NVMe SSD (such as the health status, temperature, service life, number of errors, and number of failures). | Running the **nvme smart-log** *<NVMe device name>* command |
| Additional SMART information of the Huawei NVMe SSD | Includes more detailed indicators and counts (such as power consumption, capacitor status, the number of bad blocks, and numbers of different errors). | Running the **hioadm info -d** *<NVMe device name>* **-a** and **hioadm info -d** *<NVMe device name>* **-e** commands |
| Additional SMART information of Intel NVMe SSDs | Includes more detailed error counts. | Run the **nvme intel smart-log-add** *<NVMe device name>* command |
| Network interface status information | Includes the MAC address, link status, and lost & wrong packets at the receiving and sending ends. | Running the **ifconfig** *<Network interface name>* command |
| Network port device information | Includes the port type, link status, and network rate. | Running the **ethtool** *<Network interface name>* command |
| Network interface driver information | Includes the firmware version, driver version, and bus number. | Running the **ethtool -i** *<Network interface name>* command |
| Optical module information | Includes the basic device information (such as the SN, manufacturer, production date, connection type, encoding mode, and bandwidth) and device status information (such as offset current, input power, output power, voltage, and temperature). | Running the **ethtool -m** *<Network interface name>* command |
| Number of Huawei Intelligent NIC (HiNIC) port errors | HiLink errors, Base encoding errors, and RS encoding errors | Running the **hinicadm hilink_port -i** *<dev_id>* **-p** *<port_id>* **-s** and **hinicadm hilink_count -i** *<dev_id>* **-p** *<port_id>* commands |

| Metrics | Description | Collected by |
| --- | --- | --- |
| HiNIC card working mode | Current working mode and configured working mode | Running the **hinicadm mode -i** *<dev_id>* command |
| HiNIC card core temperature | Temperature of the HiNIC card core | Running the **hinicadm temperature -i** *<dev_id>* command |
| HiNIC card event records | Includes HiNIC card heartbeat losses, PCIe exceptions, chip errors, and chip health status. | Running the **hinicadm event -i** *<dev_id>* command |
| PCIe errors of the HiNIC card | Different PCIe errors of the HiNIC card | Running the **hinicadm counter -i** *<dev_id>* **-t 4** command |
| Memory information | Includes the DIMM SN, manufacturer, Part Number (PN), bit width, capacity, and frequency. | Running the **dmidecode -t 17** command |
| CPU information | Includes the CPU ID, name, frequency, architecture, and model. | Running the **dmidecode -t 4** and **lscpu** commands |
| Memory error records | Memory CE/UCE error records, including the error type, fault code, error location information (chip, rank, bank, column, row), MCI ADDR register, MCI MISC register, MCG CAP register, MCG STATUS register, retry registers, and other registers. | Reading files such as **/dev/mem**, **/dev/cpu/***<core_id>***/msr**, and **/sys/firmware/acpi/tables/HEST** to collect memory error records and chip register information |

## 1.1.10 How Can I Quickly Restore Agent Configurations?

After the Agent is installed, you can configure the AK/SK, region ID, or project ID with a few clicks. This improves configuration efficiency.

- One-click configuration restoration is available in most regions. You can choose **Server Monitoring** > **Elastic Cloud Server** and click **Configure** on top of the page. After the configuration is completed, the Agent configurations of all servers in these regions are restored by default, and the **Configure** button is no longer displayed. If the system displays a message indicating that you do not have the required permissions, see **2.1.3 What Can I Do If the System Displays a Message Indicating Insufficient Permissions When I Click Configure on the Server Monitoring Page?** After the Agent permission is granted for a region, you do not need to perform the following steps.
- If you are in a region that does not support one-click configuration restoration of the Agent, on the **Server Monitoring** page, select the target ECS and click

**Restore Agent Configurations**. In the displayed **Restore Agent Configurations** dialog box, click **One-Click Restore**.

# 1.1.11 How Can I Enable OS Monitoring for a New ECS?

## Scenarios

This topic describes how to ensure that the newly purchased ECS comes with the OS monitoring function.

NOTE

A private image can only be used in the region where it is created. If it is used in other regions, no monitoring data will be generated for the ECSs created with this private image.

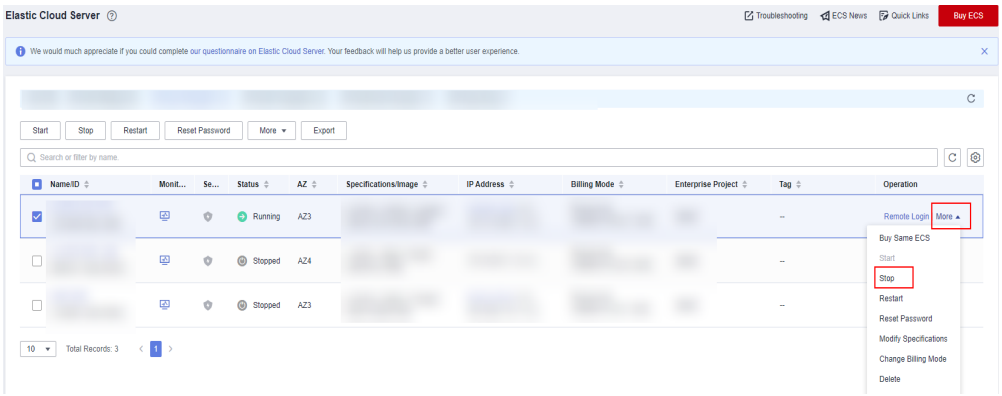
## Prerequisites

An ECS with the Agent installed is available.

## Procedure

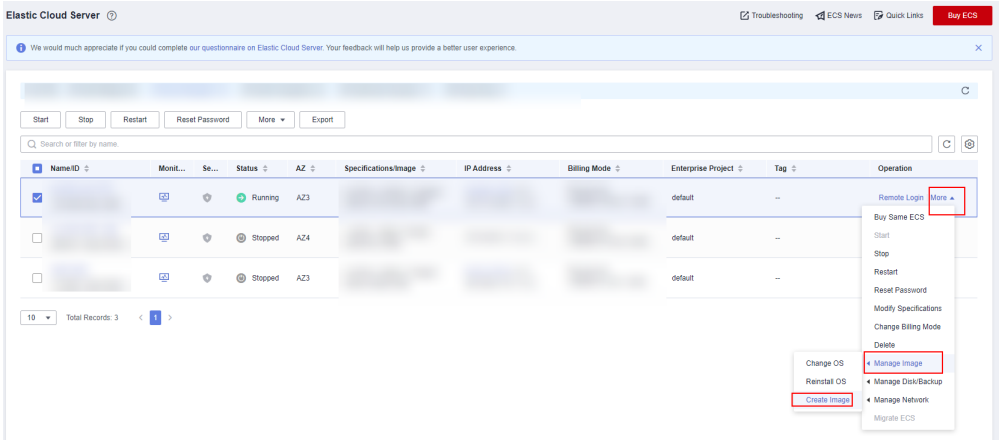
1. Log in to the ECS console. In the ECS list, locate a target ECS with the Agent installed, choose **More** > **Stop** in the **Operation** column, and click **OK**.

**Figure 1-4** Stopping an ECS




2. Choose **More** > **Manage Image/Disk** > **Create Image**.

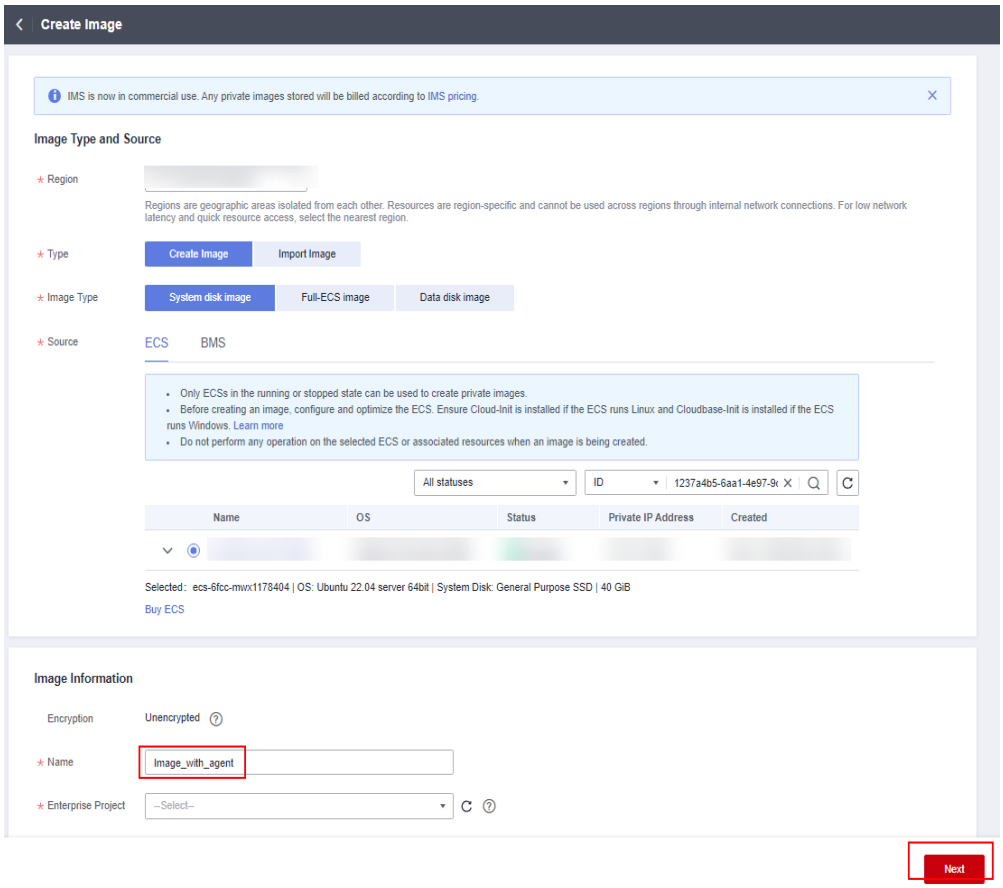
**Figure 1-5** Creating an image



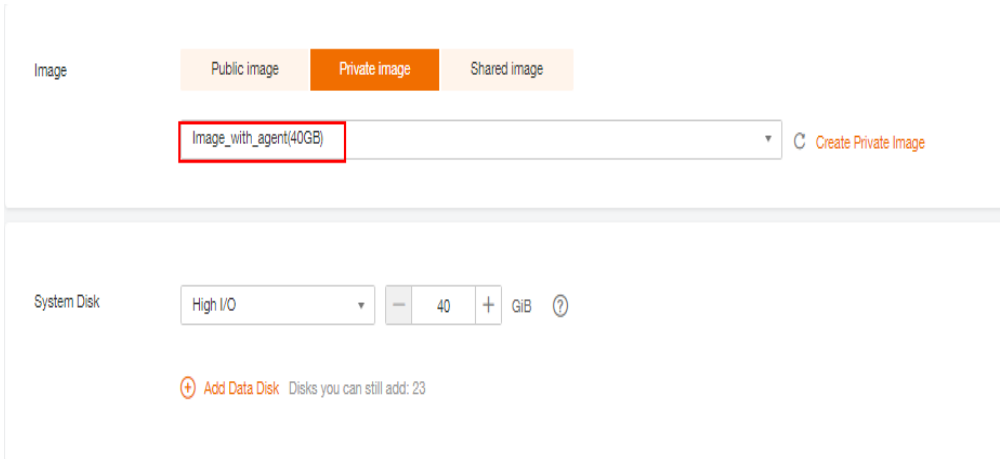
3. Set the private image name to **Image_with_agent** and click **Next**.

**Figure 1-6 Image_with_agent**




4. Purchase a new ECS and select the newly created private image **Image_with_agent (40GB)**.

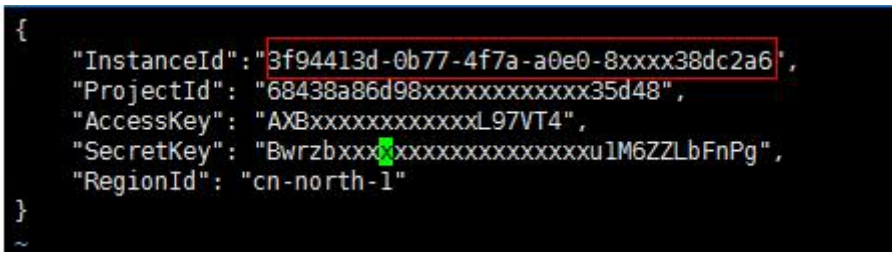
**Figure 1-7** Image_with_agent




5. Log in to the ECS. In the **/usr/local/telescope/bin/conf.json** file, set **InstanceId** to the ECS ID.

**Figure 1-8** Modifying the Agent configuration file




## 1.1.12 What Are Agent Statuses and Troubleshooting Methods?

The Agent can be in any of the following states:

- **Running**: The Agent is running properly with monitoring data properly reported.
- **Not installed**:
  - The Agent has not been installed. For details about how to install the Agent, see section of agent installation in the *Cloud Eye User Guide*.
  - If the Agent has been installed, but the agency has not been configured, configure the agency based on **1.1.2 How Do I Configure an Agency?**
  - If the Agent has been installed, but the network configuration is abnormal, rectify the fault by referring to **Modifying the DNS Server Address and Adding Security Group Rules**.
- **Stopped**:
  - The Agent is manually stopped. For details about how to start the Agent, see **Managing the Agent**.
- **Faulty**: The Agent failed to send a heartbeat message to Cloud Eye for three consecutive minutes. In this case:
  - If the Agent domain name cannot be resolved, rectify the fault by referring to **Modifying the DNS Server Address and Adding Security Group Rules**.
  - The account is in arrears.
  - If the Agent process is faulty, restart the Agent. For details about how to restart the Agent, see **Managing the Agent**. If the fault persists after the restart, the Agent files may be damaged. In this case, reinstall the Agent. For details, see **Agent Installation and Configuration**.
  - The server time is inconsistent with the local standard time.
  - If the DNS server is not a Huawei Cloud DNS server, run a command in the pattern: **dig** plus domain name, to obtain the resolved IP address of **agent.ces.myhuaweicloud.com**, which is resolved by the Huawei Cloud DNS server over the intranet. Then, add the IP address into the corresponding **hosts** file. For details about the private DNS addresses provided by Huawei Cloud, see **What Are Huawei Cloud Private DNS Server Addresses?**
  - Upgrade the Agent to the latest version.

# 1.1.13 Why Is Basic Monitoring Data Inconsistent with OS Monitoring Data?

## Symptoms

**CPU Usage** under **Basic Monitoring** is close to 100%, which is different from the CPU usage monitored by the OS (50%).

## Possible Causes

1. Setting **idle** to **poll** in the guest operating system (guest OS) causes it to enter the **polling** state while idling. This results in the guest OS consuming compute resources and not proactively releasing CPU resources, leading to abnormal CPU usage.
2. If you set **idle** to **mwait** in the guest OS for an SAP HANA ECS, the guest OS will enter the **mwait** state when idling. This results in the guest OS using fewer compute resources than it does when **idle** is set to **poll**. However, it still does not release CPU resources proactively, leading to abnormal CPU usage.

   NOTE

   - You can run the **cat /proc/cmdline** command to check whether **idle** is set to **poll** for your guest OS.
   - If you want to check whether **idle** is set to **mwait** for your guest OS, contact technical support.
   - SAP High-Performance Analytic Appliance (HANA) is a high-performance real-time data computing platform based on memory computing technologies. The cloud platform offers high-performance IaaS services that meet SAP HANA requirements. These services help you rapidly request for SAP HANA resources (such as applying for HANA ECSs and public IP addresses) and install and configure SAP HANA. This improves operation efficiency, reduces costs, and enhances your experience.

     HANA ECSs are dedicated for SAP HANA. If you have deployed SAP HANA on cloud servers, you can purchase HANA ECSs.

## Solution

**Install and configure the Agent** to view OS monitoring data.

# 1.1.14 Why Are the Network Traffic Values on Cloud Eye Different from Those in ECS?

Because the sampling period in Cloud Eye is different from that of the metric monitoring tool in ECS.

Cloud Eye collects ECS and EVS disk data every 4 minutes (5 minutes for KVM ECSs). In contrast, the data is collected in ECS every second.

The larger the sampling period, the greater the data distortion in the short term. Cloud Eye is more suitable for long-term monitoring for websites and applications running on ECSs.

To improve reliability, you can configure alarm thresholds to enable Cloud Eye to generate alarms where there are resource exceptions or shortages.

## 1.1.15 What Are the Impacts on ECS Metrics If UVP VMTools Is Not Installed on ECSs?

If UVP VMTools is not installed on your ECSs, Cloud Eye can still monitor the outband incoming rate and outband outgoing rate. However, it cannot monitor memory usage, disk usage, inband incoming rate, or inband outgoing rate, which lowers the CPU monitoring accuracy.

To learn more about ECS metrics supported by Cloud Eye, see **Basic ECS Metrics**.

## 1.1.16 Why Are Memory Usage, Disk Usage, Inband Incoming Rate, and Inband Outgoing Rate Not Displayed for an ECS?

Linux ECSs do not support the four metrics, but Windows ECSscan.

To learn more about basic monitoring metrics supported by different OSs, see **Basic ECS Metrics**.

To monitor the memory usage, disk usage, inband incoming rate, and inband outgoing rate, see **Installing the Agent on a Linux Server**.

## 1.1.17 Why Cannot I Install or Upgrade Agents in Batch?

Conditions for batch Agent installation or upgrade:

- The host is running, and the Agent status is **Running**.
- The host is running, the Agent status is **Not installed**, and the host OS supports one-click installation. For details about supported OSs, see **What OSs Does the Agent Support?**

Possible causes for Agent installation or upgrade failures:

- The host is not in the running state.
- The Agent status is faulty. For details about troubleshooting methods, see **What Should I Do If the Agent Status Is Faulty?**
- The Agent is not installed, but the host OS does not support one-click installation. You need to log in to the cloud server and run the installation command to install or upgrade the Agent. For details, see **Installing the Agent**.

## 1.1.18 What Should I Do If There Are No GPU Monitoring Records?

If no GPU monitoring records are displayed on the OS monitoring page when you view server monitoring metrics, check whether your server supports GPUs. If your server supports GPUs and the driver is running properly, perform the following steps to upgrade the Agent to the Enhanced Edition:

**Step 1** Uninstall the Agent Basic Edition.

- Linux: Log in to a server and run the **bash /usr/local/uniagent/script/uninstall.sh** command.

- Windows: In the **C:\Program Files\uniagent\script** directory where the Agent installation package is stored, double-click the **uninstall.bat** script.

**Step 2** Install the Agent Enhanced Edition: Add **.1** to the **-t** value in the installation command. For example, change **-t a.b.c** to **-t a.b.c.1**.

**----End**

**Table 1-3** Basic edition and enhanced edition of Cloud Eye Agent

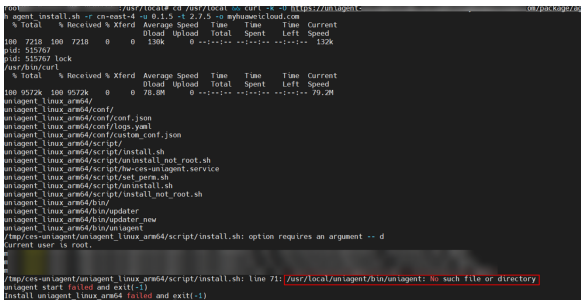
| Edition | Description |
| --- | --- |
| Basic Edition | Provides basic OS monitoring metrics, such as CPU, memory, disk, and NIC metrics, helping you improve system performance.<br>Generally, the version number consists of three digits, for example, 2.7.5. |
| Enhanced Edition | Provides GPU, NPU, and BMS hardware monitoring, in addition to the capabilities provided in the Basic Edition.<br>Generally, the version number consists of four digits, for example, 2.7.5.1.<br>**CAUTION**<br>Install the Agent Enhanced Edition if you indeed need it because it collects more metrics, which may occupy more server resources. |

# 1.1.19 What Should I Do If an Error Is Reported When I Run the Agent Installation Command?

## Symptom

When you run the Cloud Eye Agent installation command, the error message "/usr/local/uniagent/bin/uniagent: No such file or directory" is displayed, as shown in **Figure 1-9**.

**Figure 1-9** Agent installation failed




## Possible Causes

The earlier version of Cloud Eye Agent is incompatible with some Arm images.

## Solution

**Step 1** Uninstall the Agent.

```
bash /usr/local/uniagent/script/uninstall.sh
```

**Step 2** Modify the **-u** and **-t** parameters in the Agent installation command. Change the value of **-u** to **0.2.1** and that of **-t** to **2.7.5**.

For CN East 2:

Command before modification

```
cd /usr/local && curl -k -O https://uniagent-cn-east-4.obs.cn-east-4.myhuaweicloud.com/package/agent_install.sh && bash agent_install.sh -r cn-east-4 -u 0.1.5 -t 2.5.6 -o myhuaweicloud.com
```

Command after modification

```
cd /usr/local && curl -k -O https://uniagent-cn-east-4.obs.cn-east-4.myhuaweicloud.com/package/agent_install.sh && bash agent_install.sh -r cn-east-4 -u 0.2.1 -t 2.7.5 -o myhuaweicloud.com
```

**----End**

# 1.1.20 How Do I Enable or Disable Metric Collection by Modifying the Configuration File?

This following describes how to enable or disable metric collection by modifying the configuration file.

## Modifying the Configuration File to Enable Metric Collection

**Step 1** Log in to a server as user **root**.

**Step 2** Modify the configuration file.

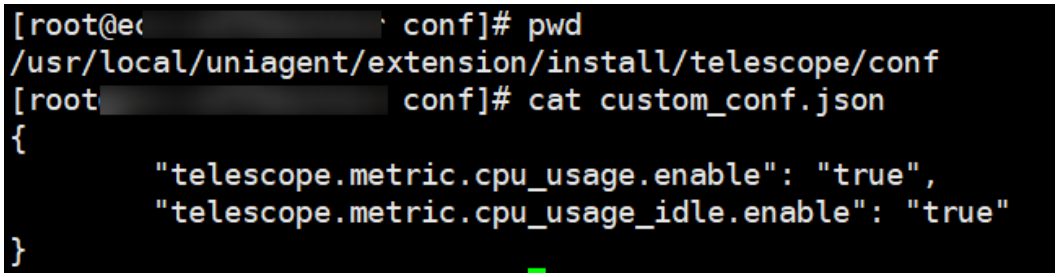
```
cd /usr/local/uniagent/extension/install/telescope/conf && vi custom_conf.json
```

Enter the following configuration content in {}. For details about contents in italic, see the metric values in **OS Monitoring Metrics Supported by ECSs with the Agent Installed**.

```
"telescope.metric.metric_name1.enable": "true",
"telescope.metric.metric_name2.enable": "true"
```

Example: **Figure 1-10** shows the configuration content for enabling metric collection for cpu_usage and cpu_usage_idle.

**Figure 1-10** Enabling metric collection for cpu_usage and cpu_usage_idle




**Step 3** Restart the Agent.

```
cd /usr/local/uniagent/extension/install/telescope && ./telescoped restart
```

**----End**

## Modifying the Configuration File to Disable Metric Collection

**Step 1** Log in to a server as user **root**.

**Step 2** Modify the configuration file.

```
cd /usr/local/uniagent/extension/install/telescope/conf && vi custom_conf.json
```

Enter the following configuration content in {}. For details about contents in italic, see the metric values in **OS Monitoring Metrics Supported by ECSs with the Agent Installed**.

```
"telescope.metric.metric_name1.enable": "false",
"telescope.metric.metric_name2.enable": "false"
```

Example: **Figure 1-11** shows the configuration content for disabling metric collection for cpu_usage and cpu_usage_idle.

**Figure 1-11** Disabling metric collection for cpu_usage and cpu_usage_idle

```
[root@e                    r conf]# pwd
/usr/local/uniagent/extension/install/telescope/conf
[root@e                    conf]# cat custom_conf.json
{
        "telescope.metric.cpu_usage.enable": "false",
        "telescope.metric.cpu_usage_idle.enable": "false"
}
```

**Step 3** Restart the Agent.

```
cd /usr/local/uniagent/extension/install/telescope && ./telescoped restart
```

**----End**

# 1.1.21 How Do I Change the Agent Resource Usage Threshold by Modifying the Configuration File?

This following describes how to modify the configuration file to change the Agent resource usage threshold.

1. Use the **root** account to log in to the ECS or BMS for which the Agent does not report data.
2. Modify the **conf.json** configuration file.
   a. Go to the Agent installation path **bin** by running the following command:

      Windows:

      ```
      cd C:\Program Files\uniagent\extension\install\telescope
      ```

      Linux:

      ```
      cd /usr/local/uniagent/extension/install/telescope/bin
      ```

   b. Open **conf.json**.

      ```
      vi conf.json
      ```

   c. Add the following parameters to the **conf.json** file.

      ```
      {
        "cpu_first_pct_threshold": xx,
      ```

```
    "memory_first_threshold": xxx,
    "cpu_second_pct_threshold": xx,
    "memory_second_threshold": xxx
}
```

**Table 1-4** Parameters in the **conf.json** file

| Parameter | Description | How to Query Parameter Values |
| --- | --- | --- |
| cpu_first_pct_threshold | Tier-1 threshold of CPU usage. The default value is 10 (%). | To query the CPU usage and memory usage of the Agent process, use either of the following methods:<br>• Linux:<br>**top -p** *telescope PID*<br>• Windows:<br>View the details of the Agent process in **Task Manager**. |
| memory_first_threshold | Tier-1 threshold of memory usage. The default value is 209715200 (200 MB). The unit is byte. | |
| cpu_second_pct_threshold | Tier-2 threshold of CPU usage. The default value is 30 (%). | |
| memory_second_threshold | Tier-2 threshold of memory usage. The default value is 734003200 (700 MB). The unit is byte. | |

d. Save the **conf.json** file and exit.

```
:wq
```

3. Modify the **manifest.json** configuration file.

a. Switch to the **manifest.json** file directory.

Windows:

```
cd C:\Program Files\uniagent\extension\holder\telescope
```

Linux:

```
cd /usr/local/uniagent/extension/holder/telescope
```

b. Open **manifest.json**.

```
vi manifest.json
```

c. Modify the parameters **mem** and **cpuUsage** in **resourceLimit**, three parameter sets in total. Retain the default values of other parameters.

```
[
  {
    "arch": "amd64",
    ...
    "resourceLimit": {
      "mem": 200,
      "cpuUsage": 10
    }
  },
  {
    "arch": "arm64",
    ...
    "resourceLimit": {
      "mem": 200,
      "cpuUsage": 10
```

```
    }
  },
  {
    "arch": "amd64",
    "os": "linux",
    ...
    "resourceLimit": {
      "mem": 200,
      "cpuUsage": 10
    }
  }
]
```

**Table 1-5** Parameters in the **manifest.json** file

| Parameter | Description |
| --- | --- |
| "resourceLimit": {<br>"mem": 200,<br>"cpuUsage": 10<br>} | **mem**: memory threshold. The default value is **200**, in MB.<br>**cpuUsage**: CPU usage threshold (%). The default value is **10**. |

d. Save the **manifest.json** file and exit.

```
:wq
```

4. Restart the Agent.
   - Windows:
     i. In the **C:\Program Files\uniagent\extension\install\telescope** directory, double-click **shutdown.bat** to stop the Agent, and then run **start.bat** to start the Agent.
     ii. In the **C:\Program Files\uniagent\script** directory, double-click **shutdown.bat** to stop the Agent, and then run **start.bat** to start the Agent.
   - Linux:

```
/usr/local/uniagent/bin/uniagent stop
/usr/local/uniagent/bin/uniagent start
/usr/local/uniagent/extension/install/telescope/telescoped restart
```

# 1.1.22 How Do I Change the Process Collection Frequency by Modifying the Configuration File?

The following describes how to modify the configuration file to change the process collection frequency.

## Linux

**Step 1** Log in to a server as user **root**.

**Step 2** Modify the configuration file.

```
cd /usr/local/uniagent/extension/install/telescope/conf && vi custom_conf.json
```

Enter the following configuration content in {} to change the number of processes updated per minute. The calculation method is as follows: (*${telescope.metric.proc.flush_metric_batch_size}* x *${telescope.metric.proc.flush_metric_period}*)/60

```
"telescope.metric.proc.flush_metric_batch_size":"num1",
"telescope.metric.proc.flush_metric_period":"num2"
```

Configurations for updating 50 processes per minute

```
[root@ conf]# pwd
/usr/local/uniagent/extension/install/telescope/conf
[root@e nf]# cat custom_conf.json
{
        "telescope.metric.proc.flush_metric_batch_size":"50",
        "telescope.metric.proc.flush_metric_period":"60"
}
```

**Step 3** Restart the Agent.

```
cd /usr/local/uniagent/extension/install/telescope && ./telescoped restart
```

**----End**

### Windows

**Step 1** Log in to a server.

**Step 2** Modify the **custom_conf.json** configuration file in **C:\Program Files\uniagent\extension\install\telescope/conf** to change the number of processes updated per minute. The calculation method is as follows: (*${telescope.metric.proc.flush_metric_batch_size}* x *${telescope.metric.proc.flush_metric_period}*)/60

Example: If you want to update 50 processes every minute, enter the following content in {}:

```
"telescope.metric.proc.flush_metric_batch_size":"50",
"telescope.metric.proc.flush_metric_period":"60"
```

**Step 3** In the **C:\Program Files\uniagent\extension\install\telescope** directory, restart the Agent.

1. Double-click **shutdown.bat** to stop the Agent process.
2. Double-click **start.bat** to start the Agent process.

**----End**

# 1.2 Cloud Service Monitoring

## 1.2.1 What Is Aggregation?

Aggregation is a process where Cloud Eye aggregates the maximum, minimum, average, sum, or variance value of raw data sampled for different periods, and this process repeats for each subsequent period. Each period is called an aggregation period. A calculation period is called an aggregation period.

During the aggregation, data sets are smoothed. A longer aggregation period means more smoothing and precise data, enabling you to predict trends more precisely. On the contrary, a shorter aggregation period means more accurate alarms.

The aggregation period can be 5 minutes, 20 minutes, 1 hour, 4 hours, or 1 day.

During the aggregation, Cloud Eye processes the sampled data based on the data type.

- If the data sampled is integers, Cloud Eye rounds off the aggregation results.
- If the data includes decimal values (floating point number), Cloud Eye truncates the data after the second decimal place.

For example, the instance quantity in Auto Scaling is an integer. If the aggregation period is 5 minutes, and the current time is 10:35, Cloud Eye will aggregate the raw data generated between 10:30 and 10:35 to the time point of 10:30. If the raw data 1 and 4, after aggregation, the maximum value is 4, the minimum value is 1, and the average value is [(1 + 4)/2] = 2, instead of 2.5.

Choose whichever aggregation method best meets your service requirements.

## 1.2.2 How Long Is Metric Data Retained?

Metric data includes raw data and rolled-up data.

- Raw data is retained for 2 days.
- Rolled-up data is data aggregated based on raw data. The retention period for rolled-up data depends on the rollup period.

**Table 1-6** Retention periods for rolled-up data

| Rollup Period | Retention Period |
| --- | --- |
| 1 minute | 2 days |
| 5 minutes | 10 days |
| 20 minutes | 20 days |
| 1 hour | 155 days |
| 4 hours | 155 days |
| 24 hours | 155 days |

NOTE

- For metric data in the AP-Bangkok region, the maximum retention period is one year, and the rollup period is 24 hours.

If an instance is disabled, stopped, or deleted, its metrics will be deleted one hour after the raw data reporting of those metrics stops. When the instance is enabled or restarted, raw data reporting will resume. If the instance has been disabled or stopped for less than two days or for less time than the previous rolled-up data retention period, you can view the historical metric data generated before these metrics were deleted.

## 1.2.3 What Aggregation Methods Does Cloud Eye Support?

Cloud Eye supports the following aggregation methods:

- Average

  The arithmetic average of metrics collected during an aggregation period. It helps identify long-term trends in cloud product performance, serves as a baseline reference for anomaly detection, and supports capacity planning based on average load. However, average values can obscure extreme fluctuations. When data is unevenly distributed, the average may not accurately reflect reality. You need to analyze the average in conjunction with the maximum and minimum values.

- Maximum

  The highest value of metrics collected during an aggregation period. This metric is crucial for identifying spikes and possible issues in the monitoring system.

- Minimum

  The minimum value of metrics collected during an aggregation period. This metric is crucial for identifying abnormally low values (for example, traffic dropping to zero) and potential issues in the monitoring system.

- Sum

  The sum of metric data collected during an aggregation period. This is a key statistical value used for analyzing resource consumption metrics, such as traffic volume and storage usage.

  **NOTE**

  During an aggregation process, data generated within a specified time range are consolidated to the start point of the aggregation period using the relevant aggregation algorithm. Take a 5-minute period as an example. If the current time is 10:35, the raw data generated between 10:30 and 10:35 will be aggregated to 10:30.

## 1.2.4 How Can I Export Collected Data?

You can export monitoring data from Cloud Eye. The procedure is as follows:

1. On the Cloud Eye console, choose **Cloud Service Monitoring** or **Server Monitoring**.
2. Click **Export Data**.
3. Configure the time range, period, resource type, dimension, monitored object, and metric.
4. Click **Export**.

**NOTE**

You can export data for multiple metrics at a time to a **CSV** file.

- The first row in the exported file displays the username, region, service, instance name, instance ID, metric name, metric data, time, and timestamp. You can view historical monitoring data.
- To convert the time using a Unix timestamp to the time of the target time zone, perform the following steps:

a. Use Excel to open a CSV file.

b. Use the following formula to convert the time:

   Target time = [Unix timestamp/1000 + (Target time zone) x 3600]/86400 + 70 x 365 + 19

c. Set the cell format to **Date**.

Assume that you need to convert a Unix timestamp of 1475918112000 to Shanghai time (UTC+8). Calculate the Shanghai time as follows: [1475918112000/1000 + (+8) x 3600]/86400 + 70 x 365 + 19. Then, select a presentation format such as 2016/3/14 13:30, and the target time calculated will be presented as 2016/10/8 17:15.

# 1.2.5 What Are Outband Incoming Rate and Outband Outgoing Rate?

## Concept Explanation

You need to understand the meaning of outband and inband:

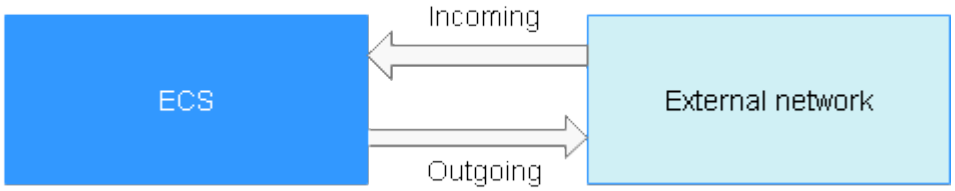
**Outband**

- Outband is the opposite to inband. Inband indicates that the monitored object is an ECS. Outband indicates that the monitored object is the physical server at the virtualization layer.

**Incoming and Outgoing**

- Incoming indicates traffic comes to an ECS per second.
- Outgoing indicates traffic sent from an ECS to an external network or client per second.

The following figure shows the traffic directions.




## Metric Description

**Table 1-7** Outband incoming/outgoing rate

| Item | Description |
| --- | --- |
| Outband incoming rate | Traffic coming into an ECS per second<br>For example, traffic generated when you download resources to an ECS from an external network or upload files to the ECS.<br>Unit: byte/s |

| Item | Description |
| --- | --- |
| Outband outgoing rate | Traffic going out of an ECS per second<br>For example, traffic generated when users access an ECS via the internet or when the ECS functions as an FTP server for users to download resources.<br>Unit: byte/s |

**Table 1-8** Outband incoming/outgoing rate

| Item | Description |
| --- | --- |
| Outband incoming rate | Traffic coming to an ECS per second at the virtualization layer. Generally, the outband incoming rate is slightly larger than the traffic coming to the ECS because the virtualization layer will filter some unnecessary packets.<br>Unit: byte/s |
| Outband outgoing rate | Traffic going out of an ECS per second at the virtualization layer. Generally, the outband outgoing rate is slightly larger than the traffic sent from the ECS because the virtualization layer will filter some unnecessary packets.<br>Unit: byte/s |

# 1.3 Alarm Management

## 1.3.1 Why Can't a User of an Enterprise Project View One-Click Monitoring?

One-click monitoring is only available to the enterprise project account or its IAM users with the **Tenant Administrator** permissions.

For details about how to assign the Tenant Administrator permission to a user, see **Creating a User Group and Assigning Permissions**.

## 1.3.2 Why Can't a User of an Enterprise Project Select All Resources When Configuring Alarm Rules?

When configuring alarm rules, only Huawei Cloud accounts or IAM users with the **Tenant Administrator** permissions can select all resources.

For details about how to assign the **Tenant Administrator** permissions to an IAM user, see **Creating a User Group and Assigning Permissions**.

## 1.3.3 What Are Alarm Notifications? How Many Types of Alarm Notifications Are There?

Alarm notifications are email or SMS messages that are sent out when an alarm status is **Alarm**, **OK**, or both.

You can configure Cloud Eye to send or not send alarm notifications when you create or modify an alarm rule.

Cloud Eye can:

- Send you email, or send HTTP/HTTPS messages to servers.
- Work with Auto Scaling to trigger the system to automatically add or remove servers.

## 1.3.4 What Alarm Status Does Cloud Eye Support?

**Alarm**, **Resolved**, **Insufficient data**, **Triggered**, **Expired**, and **Resolved (forcible clear)** are supported.

- **Alarm**: The metric value reached the alarm threshold, and an alarm has been triggered but not cleared for the resource.
- **Resolved**: The metric value went back to the normal range, and the resource alarm was cleared.
- **Insufficient data**: No monitoring data has been reported for three consecutive hours, and this is generally because the instance has been deleted or is abnormal.
- **Triggered**: An event configured in the alarm policy triggered an alarm.
- **Expired**: The monitored resources or alarm policies in the alarm rule were adjusted, so the original alarm record status expired.
- **Resolved (forcible clear)**: You can forcibly clear alarms in the **Triggered**, **Alarm**, or **Insufficient data** state.

## 1.3.5 What Alarm Severities Are Available on Cloud Eye?

There are four levels of alarm severity: critical, major, minor, and informational.

- **Critical**: An emergency fault has occurred and services are affected.
- **Major**: A relatively serious problem has occurred and may hinder the use of resources.
- **Minor**: A less serious problem has occurred but will not hinder the use of resources.
- **Informational**: A potential error exists and may affect services.

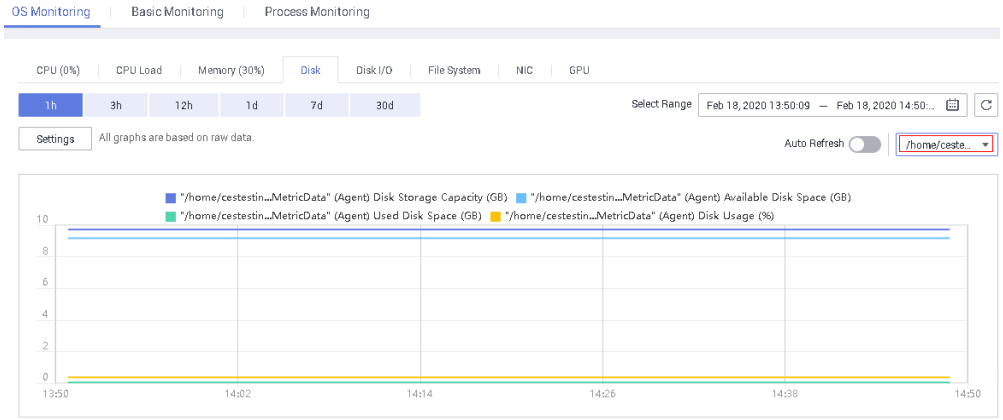
## 1.3.6 How Do I Monitor and View the Disk Usage?

To monitor the disk usage, install the Agent and create an alarm rule for the disk usage. In the alarm rule, set the metric to **(Agent) Disk Usage (Recommended)** and select a mount point. Enable and configure **Alarm Notification**. For details, see **Creating an Alarm Rule to Monitor a Server**.

After you install the Agent, you can view the data disk usage on the Cloud Eye console. On the **OS Monitoring** page, click the **Disk** tab and select a mount point on the right of the **Auto Refresh** button.

**Figure 1-12** Viewing the data disk usage on the **OS Monitoring** page




# 1.3.7 How Can I Change the Phone Number and Email Address for Receiving Alarm Notifications?

Alarm notifications can be sent to the account contact or SMN topic subscribers configured in alarm rules.

You can change phone numbers and email addresses of the account contact or SMN topic subscribers.

## Account Contact

If you set **Recipient** to **Account contact**, alarm notifications will be sent to the phone number and email address registered for your account.

You can update them on the **My Account** page by performing the following steps:

1. Log in to the management console.
2. Hover your mouse over the username in the upper right corner and select **Basic Information**.

   The **My Account** page is displayed.
3. Click **Edit** next to the phone number or email address.
4. Change the phone number or email address as prompted.

## SMN Topic Subscribers

If you set **Recipient** to an SMN topic, perform the following steps to change the phone numbers:

1. Log in to the management console.
2. In the service list, select **Simple Message Notification**.
3. In the navigation pane, choose **Topic Management** >**Topics**.

4. Locate the topic and click its name.
5. Add subscription endpoints to or delete subscription endpoints from the topic.

## 1.3.8 How Can an IAM User Receive Alarm Notifications?

To enable an IAM user to receive alarm notifications, subscribe the email address or phone number of the user to an SMN topic and select the topic when you create alarm rules. For details, see **Creating a Topic** and **Adding Subscriptions**.

# 2 Troubleshooting



## 2.1 Permissions Management

### 2.1.1 What Should I Do If the IAM Account Permissions Are Abnormal?

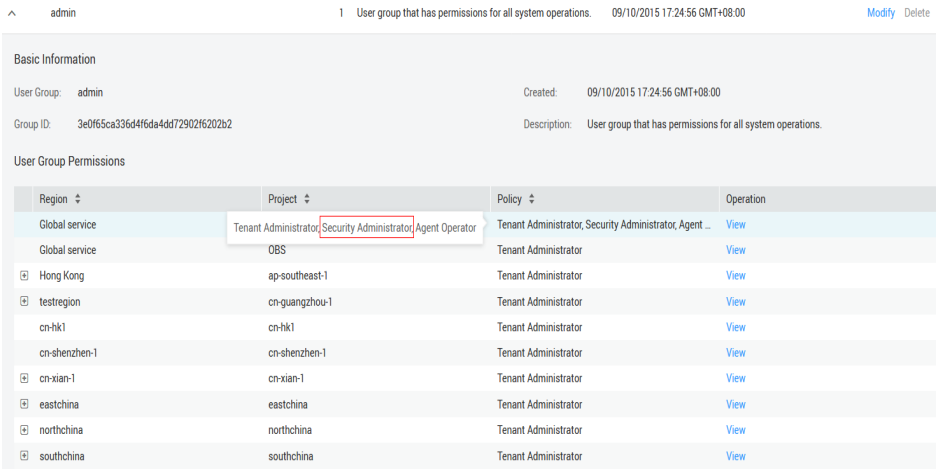
To use server monitoring, IAM users in a user group must have the **Security Administrator** permissions. If they do not have the permissions, a message indicating abnormal permissions is displayed. Contact the account administrator to grant the permissions.

NOTE

Cloud Eye provides a list of system policies, operations, and policy permissions. For details, see **Permissions Management**.

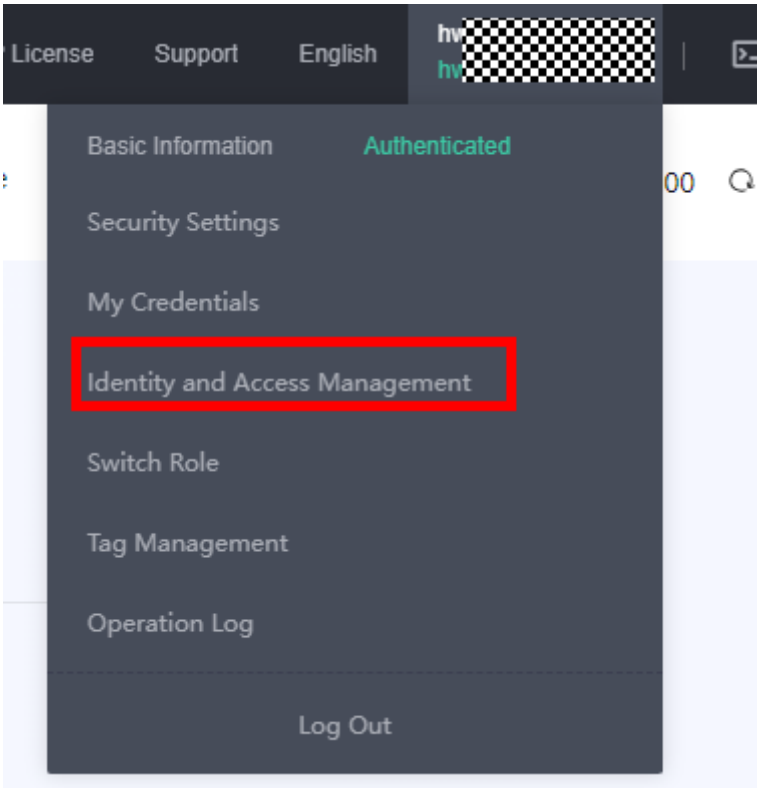
**Figure 2-1** Checking the permissions




## 2.1.2 What Can I Do If the System Displays a Message Indicating Insufficient Permissions When I Access Cloud Eye?

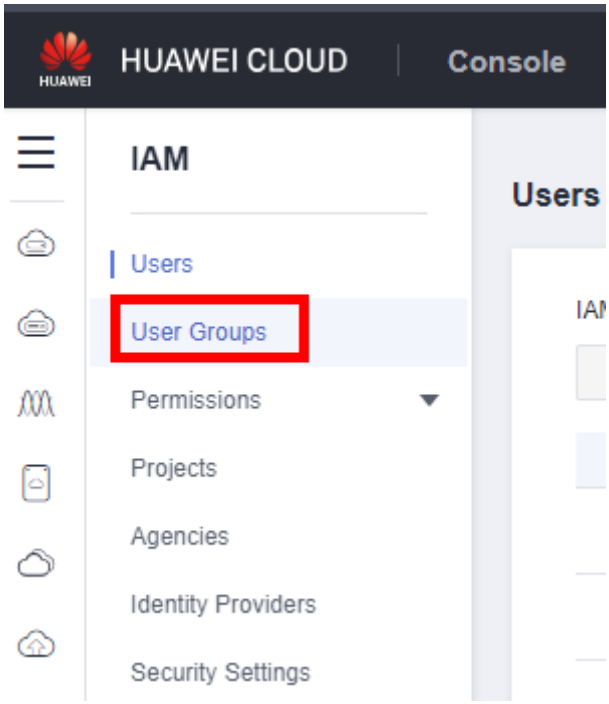
This is because that you use an IAM user account, which does not have sufficient permissions. Check your permissions configured on IAM.

1. Use the Huawei Cloud account to log in to the Huawei Cloud management console.
2. On the management console, in the upper right corner, hover your mouse over the username, and choose **Identity and Access Management** from the drop-down list.



3. In the navigation pane, choose **User Groups**.




4. Expand details of the user group the user belongs to.


5. Grant permissions to the user group that the IAM user belongs to. For details, see **Creating a User Group and Assigning Permissions**.

NOTE

Cloud Eye provides a list of system policies, operations, and policy permissions. For details, see **Permissions**.

# 2.1.3 What Can I Do If the System Displays a Message Indicating Insufficient Permissions When I Click Configure on the Server Monitoring Page?

## Symptoms

When an IAM user clicked **Configure** on the **Server Monitoring** page, a message indicating insufficient permissions was displayed.

## Probable Causes

The IAM agency permissions are not configured for the IAM user.

## Procedure

**Step 1** Add a policy for querying the agencies.

1. Log in to the Huawei Cloud management console using an account or IAM user that has the permissions to create custom policies and grant permissions to other IAM users.
2. Ensure that the account has been granted the Agent permissions for the region by performing the following operations: On the Cloud Eye console, choose **Server Monitoring** > **Elastic Cloud Server**. Check whether **Configure** is displayed above the ECS list.
   - If no, the Agent permissions have been granted for the region.
   - If yes, click **Configure** to enable the Agent permissions for the region.
3. Hover your mouse over the username in the upper right corner and choose **Identity and Access Management** from the drop-down list.
4. In the navigation pane, choose **Permissions** > **Policies/Roles**. Search for the CES agency policy **CES AgencyCheck Access**. If the policy can be found, grant permissions by referring to **Step 2**. If the policy does not exist, perform the following steps to create a custom policy:
   a. Click **Create Custom Policy** in the upper right corner.
      i. Configure the following parameters:
         - **Policy Name**: Specify a custom policy name.
         - **Policy View**: Select **JSON**.
         - **Policy Content**: Copy the following code and paste it to the text box.

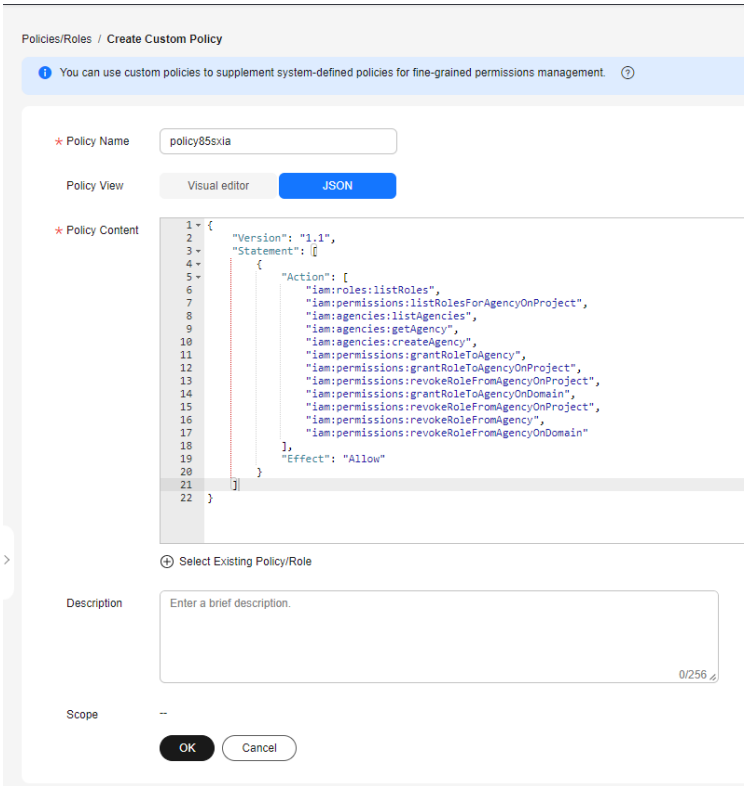
```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "iam:agencies:createAgency",
        "iam:agencies:getAgency",
        "iam:agencies:listAgencies",
        "iam:permissions:grantRoleToAgency",
        "iam:permissions:grantRoleToAgencyOnDomain",
        "iam:permissions:grantRoleToAgencyOnProject",
        "iam:permissions:listRolesForAgency",
        "iam:permissions:listRolesForAgencyOnDomain",
        "iam:permissions:listRolesForAgencyOnProject",
        "iam:permissions:revokeRoleFromAgency",
        "iam:permissions:revokeRoleFromAgencyOnDomain",
        "iam:permissions:revokeRoleFromAgencyOnProject",
        "iam:roles:createRole",
        "iam:roles:listRoles",
        "iam:roles:updateRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

         - **Description**: (Optional) Provide supplementary information about the policy.

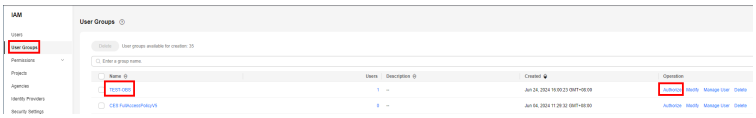
ii. Confirm the policy content(see **Figure 2-2**) and click **OK** to save the policy.

**Figure 2-2** Create Custom Policy

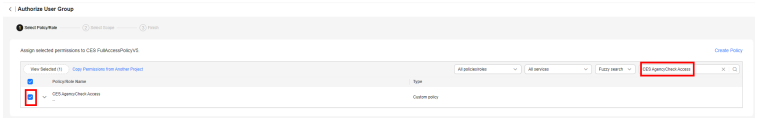



**Step 2** Assign permissions to the IAM user.

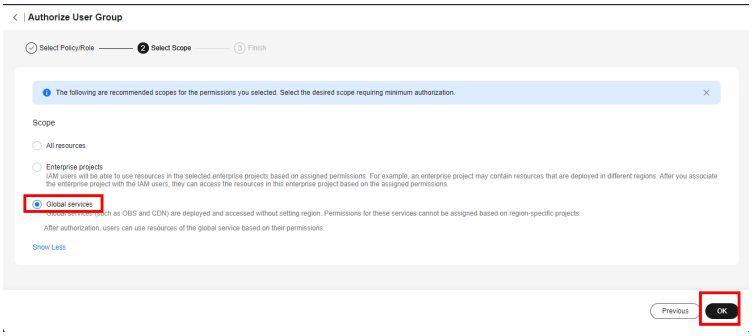
1. In the navigation pane, choose **User Groups**, locate the user group that contains the IAM user, and choose **Authorize** in the **Operation** column.



2. Search for the added custom policy name or **CES AgencyCheck Access**, select the policy, and click **Next**.

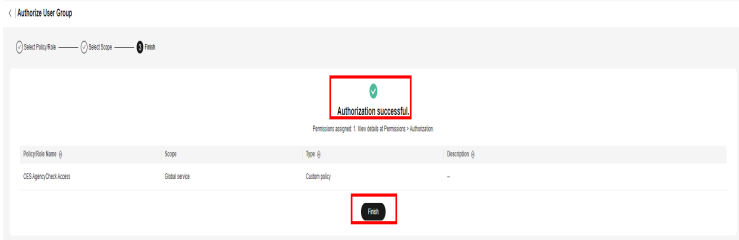


3. Select **Global services** and click **OK**.



4. If **Authorization successful** is displayed, click **Finish**.




**----End**

# 2.2 Server Monitoring

## 2.2.1 What Should I Do If the Monitoring Is Periodically Interrupted or the Agent Status Keeps Changing?

### Symptom

Monitoring interruptions and unstable Agent status may be caused by Agent overload. The Agent is overloaded if you see either of the following symptoms:

- On the **Server Monitoring** page, the Agent status frequently changes between **Running** and **Faulty**.
- The period in the metric dashboard is discontinuous.

### Constraints

The restoration method in this section only supports new Agent version. If your Agent is of an earlier version, you are advised to upgrade it to the new version.

Run the following command to check the current Agent version:

```
if [[ -f /usr/local/uniagent/extension/install/telescope/bin/telescope ]]; then /usr/local/uniagent/extension/install/telescope/bin/telescope -v; elif [[ -f /usr/local/telescope/bin/telescope ]]; then echo "old agent"; else echo 0; fi
```

- If **old agent** is displayed, the early version of the Agent is used.
- If a version ID is returned, the new version of the Agent is used.
- If **0** is returned, the Agent is not installed.

### Possible Causes

The circuit patter is implemented by the Agent when the CPU and memory usage is too high to prevent other services from being affected. The circuit breaker pattern will be implemented automatically when the Agent is overloaded, and no monitoring data will not be reported.

### Circuit Breaker Principles

By default, the Agent detection system is as follows:

The system checks the CPU usage and memory usage of the Agent process every minute. If the CPU usage exceeds 30% or the memory usage exceeds 700 MB, that is, the second threshold, the Agent process will exit. If both the CPU usage and memory usage do not exceed the second threshold, the system checks whether they exceed the first threshold (10% of CPU usage or 200 MB of memory usage). If any of them exceeds the first threshold for three consecutive times, the Agent process will exit and the information will be recorded.

After the Agent exits, the daemon process automatically starts the Agent process and checks the exit records. If there are three consecutive exit records, the Agent will hibernate for 20 minutes, during which monitoring data will not be collected.

When too many disks are attached to a server, the CPU or memory usage of the Agent process will become high. You can configure the tier-1 and tier-2 thresholds based on **Procedure** to trigger the circuit-breaker pattern according to the actual resource usages.

## Procedure

1. Use the **root** account to log in to the ECS or BMS for which the Agent does not report data.
2. **Optional:** Go to the Agent installation path.

   For Windows, the path is **C:\Program Files\uniagent\extension\install\telescope\bin**.

   For Linux, the path is **/usr/local/uniagent/extension/install/telescope/bin**.
3. Modify configuration file **conf.json**.
   a. Run the following command to open **conf.json**:

      **vi conf.json**

   b. Add the following parameters to the **conf.json** file. For details about the parameters, see **Table 2-1**.

**Table 2-1** Parameters

| Parameter | Description |
| --- | --- |
| cpu_first_pct_threshold | Tier-1 threshold of CPU usage. The default value is 10 (%). |
| memory_first_threshold | Tier-1 threshold of memory usage. The default value is 209715200 (200 MB). The unit is byte. |
| cpu_second_pct_threshold | Tier-2 threshold of CPU usage. The default value is 30 (%). |
| memory_second_threshold | Tier-2 threshold of memory usage. The default value is 734003200 (700 MB). The unit is byte. |

| Parameter | Description |
| --- | --- |
| [a] To query the CPU usage and memory usage of the Agent, use the following method:<br>• Linux:<br>**top -p** *telescope PID*<br>• Windows:<br>View the details about the Agent process in **Task Manager**. | |

```
{
   "cpu_first_pct_threshold": xx,
   "memory_first_threshold": xxx,
   "cpu_second_pct_threshold": xx,
   "memory_second_threshold": xxx
}
```

c. Save the **conf.json** file and exit.

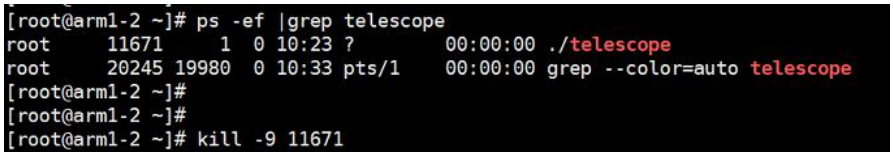
**:wq**

4. Restart the Agent.
   - Windows:
     - In the directory where the Agent installation package is stored, double-click the **shutdown.bat** script to stop the Agent, and then execute the **start.bat** script to start the Agent.
   - Linux:
     - Check the Agent PID.

       **ps -ef |grep telescope**
     - Run **kill -9** *PID* to stop the process and then wait for 3 to 5 minutes for the Agent to automatically restart.

       **kill -9** *PID*

       **Figure 2-3** Restarting the Agent




## 2.2.2 What Should I Do If a Service Port Is Used by the Agent?

Cloud Eye Agent uses a port from path **/proc/sys/net/ipv4/ip_local_port_range** to send HTTP requests. Any port in the range obtained may be occupied. If the port used by the Agent is the same as the service port, you can modify path **/proc/sys/net/ipv4/ip_local_port_range** and restart the Agent to solve the problem.

## Constraints

The restoration method in this section only supports new Agent version. If your Agent is of an earlier version, you are advised to upgrade it to the new version.

Run the following command to check the current Agent version:

```
if [[ -f /usr/local/uniagent/extension/install/telescope/bin/telescope ]]; then /usr/local/uniagent/extension/install/telescope/bin/telescope -v; elif [[ -f /usr/local/telescope/bin/telescope ]]; then echo "old agent"; else echo 0; fi
```

- If **old agent** is displayed, the early version of the Agent is used.
- If a version is returned, the new version of the Agent is used.
- If **0** is returned, the Agent is not installed.

## Procedure

1. Log in to the primary node as a root user.
2. Open the **sysctl.conf** file.

   **vim /etc/sysctl.conf**
3. (Permanent change) Add new ports to the **sysctl.conf** file:

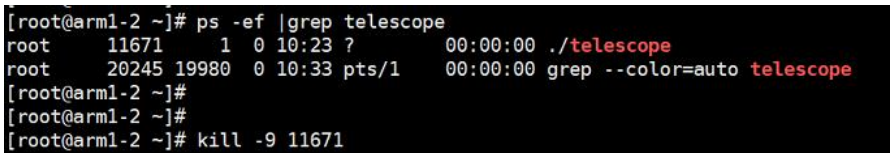
   **net.ipv4.ip_local_port_range=49152 65536**
4. Apply the changes.

   **sysctl -p /etc/sysctl.conf**

   NOTE

   - The modification is permanent and still takes effect after the host is restarted.
   - To make a temporary modification (the password becomes invalid after the host is restarted), run the **# echo 49152 65536 > /proc/sys/net/ipv4/ip_local_port_range** command.
5. Restart the Agent:
   - Windows:
     - In the directory where the Agent installation package is stored, double-click the **shutdown.bat** script to stop the Agent, and then execute the **start.bat** script to start the Agent.
   - Linux:
     - Run the following command to check the PID of telescope:
     - **ps -ef |grep telescope**
     - Run the following command to stop the process and then wait for 3 to 5 minutes for the Agent to restart.
     - **kill -9** *PID*

     **Figure 2-4** Restarting the Agent

# 2.2.3 Troubleshooting Agent One-Click Restoration Failures

## Symptom

After you click **Restore Agent Configurations**, the Agent status is still **Configuration error**.

## Constraints

The restoration method in this section only supports new Agent version. If your Agent is of an earlier version, you are advised to upgrade it to the new version.

Run the following command to check the current Agent version:

```
if [[ -f /usr/local/uniagent/extension/install/telescope/bin/telescope ]]; then /usr/local/uniagent/extension/install/telescope/bin/telescope -v; elif [[ -f /usr/local/telescope/bin/telescope ]]; then echo "old agent"; else echo 0; fi
```

- If **old agent** is displayed, the early version of the Agent is used.
- If a version is returned, the new version of the Agent is used.
- If **0** is returned, the Agent is not installed.

## Possible Causes

The following may cause this issue:

1. DNS configurations
2. IAM agency quota
3. User permissions

## Procedure

**Step 1** Check DNS configurations.

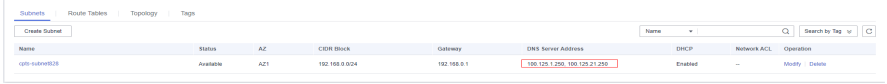
1. Log in to the management console.
2. Under **Compute**, select **Elastic Cloud Server**.
3. Click the name of the ECS.

   The ECS details page is displayed.
4. Click the VPC name.

   The VPC console is displayed.
5. In the VPC list, click the VPC name.
6. On the **Subnets** page, check whether the DNS server addresses of the ECS are correct.

   For details about how to configure DNS server addresses for different regions, see **Modifying the DNS Server Address and Adding Security Group Rules**.

   **Figure 2-5** DNS server address



**Step 2** Check IAM agency quota.

1. Log in to the management console.
2. In the service list, select **Identity and Access Management**.
3. On the IAM console, choose **Agencies**.
4. Check the agency quota.

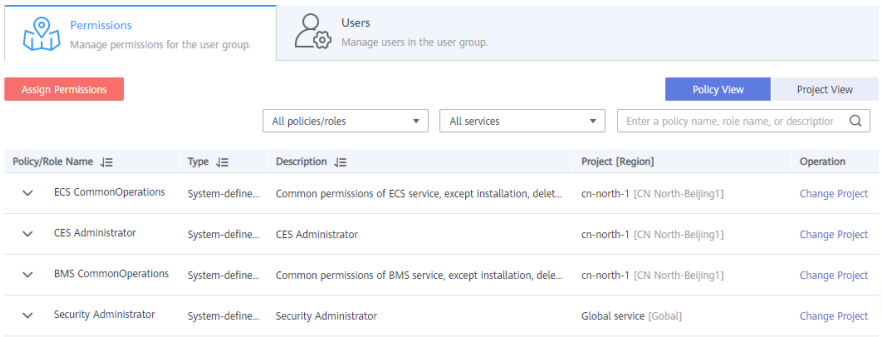
   Check whether there is the agency: **CESAgentAutoConfigAgency**.

   If there is no such an agency and the quota has been used up, delete unnecessary agencies and then perform one-click Agent restoration.

**Step 3** Check user permissions.

1. Log in to the management console.
2. In the service list, select **Identity and Access Management**.
3. In the navigation pane, click **User Groups**.
4. Locate your user group and click **Assign Permissions** in the **Operation** column.
5. To install the Agent, you must have the following permissions:
   - Global: Security Administrator
   - Region: ECS CommonOperationsr, or BMS CommonOperations and CES Administrator, or CES FullAccess

   **Figure 2-6** Permissions required for installing the Agent




**----End**

# 2.2.4 Why Is No Monitoring Data Displayed After Performing a One-Click Restoration for the Agent?

## Symptom

The Agent is running normally after being restored, but no monitoring data is generated.

## Constraints

The restoration method in this section only supports new Agent version. If your Agent is of an earlier version, you are advised to upgrade it to the new version.

Run the following command to check the current Agent version:

```
if [[ -f /usr/local/uniagent/extension/install/telescope/bin/telescope ]]; then /usr/local/uniagent/extension/install/telescope/bin/telescope -v; elif [[ -f /usr/local/telescope/bin/telescope ]]; then echo "old agent"; else echo 0; fi
```

- If **old agent** is displayed, the early version of the Agent is used.
- If a version is returned, the new version of the Agent is used.
- If **0** is returned, the Agent is not installed.

## Possible Causes

If no OS monitoring data is available for an ECS or BMS with the Agent installed, the possible causes are as follows:

- There is a problem with the Agent process.
- There is a problem with agency configurations.
- The network is not well connected.

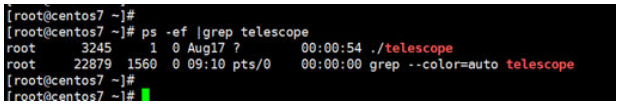
## Procedure (Linux)

1. Log in to the ECS or BMS as user **root**.
2. Run the following command to check whether the **telescope** process is running:

   **ps -ef |grep telescope**

   If following information is displayed, the telescope process is normal.

   **Figure 2-7** Viewing the telescope process

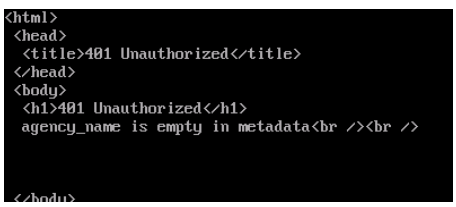
   - If the telescope process is normal, go to **4**.
   - If the telescope process is abnormal, go to **3**.
3. Run the following command to start the Agent:

   **service uniagent restart**
4. Run the following command to check whether the required agency has been created:

   **curl -ivk https://agent.ces.myhuaweicloud.com/v1.0/agencies/cesagency/securitykey**

   - If data is returned, the agency is normal and AK/SK can be obtained. No further action is required.
   - If the request fails or the following information is displayed, go to **5**.

   **Figure 2-8** Failing to obtain the AK/SK

5. On the IAM console, in the left navigation pane, choose **Agencies**, and search for **cesagency**. Expand the **cesagency** details, check whether the current region is included in **Project [Region]**. If no, in the **Operation** column, click **More**, and choose **Manage Permissions**. Click **Assign Permissions**, search for **CES Administrator**, click the drop-down list box, and select the current region.

   **Figure 2-9** Searching for **cesagency**

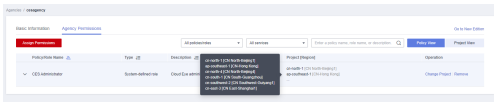
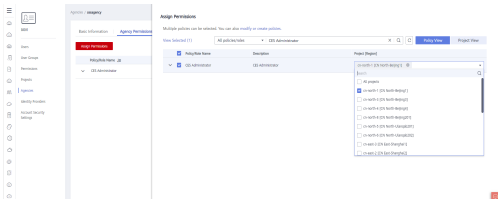
   

   **Figure 2-10** Assigning permissions

   - If the problem is resolved, no further action is required.
   - Otherwise, go to **6**.

6. Run the following command to check whether the DNS settings are normal:

   **ping *agent.ces.myhuaweicloud.com***

   - If yes, no further action is required.
   - If no, modify the **DNS server address** or the Cloud Eye endpoint.

     NOTE

     For details about Cloud Eye endpoints for each region, see **Regions and Endpoints**.

## Procedure (Windows)

1. Log in to the ECS or BMS as an administrator.
2. Open the **Task Manager** and check whether the telescope process is running.

   If there are **Figure 2-11** and **Figure 2-12**, the telescope process is running.

**Figure 2-11** Agent process (Windows)

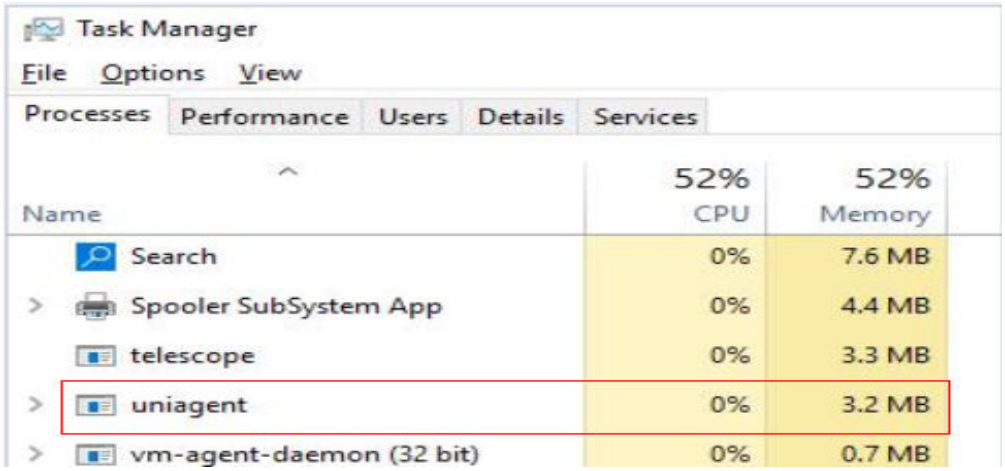
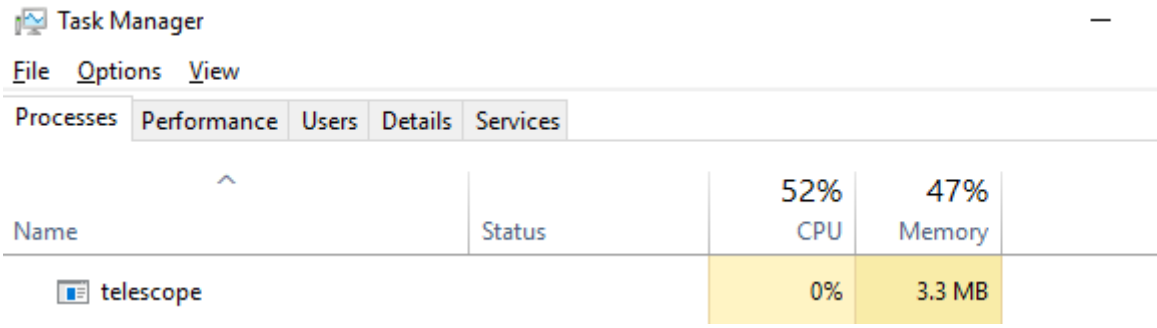



**Figure 2-12** telescope process (Windows)




- If the process is normal, go to **4**.
- If the process is abnormal, go to **3**.

3. Double-click **start.bat** in **C:\Program Files\uniagent\script** to start the Agent.
4. On the IAM console, in the left navigation pane, choose **Agencies**, and search for **cesagency**. Expand the **cesagency** details, check whether the current region is in **Project [Region]**. If no, in the **Operation** column, click **More**, and choose **Manage Permissions**. Click **Assign Permissions**, search for **CES Administrator**, click the drop-down list box, and select the current region.

**Figure 2-13** Searching for **cesagency**

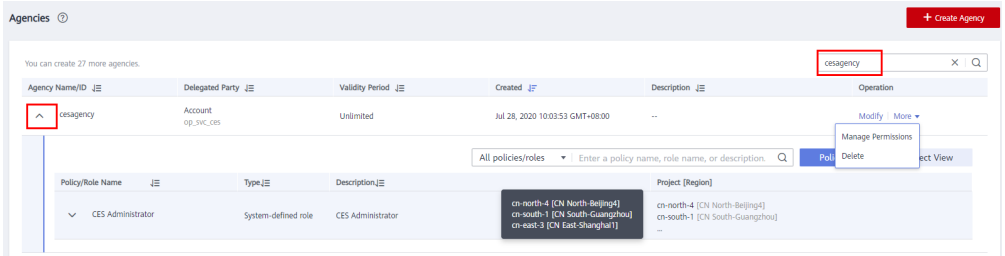
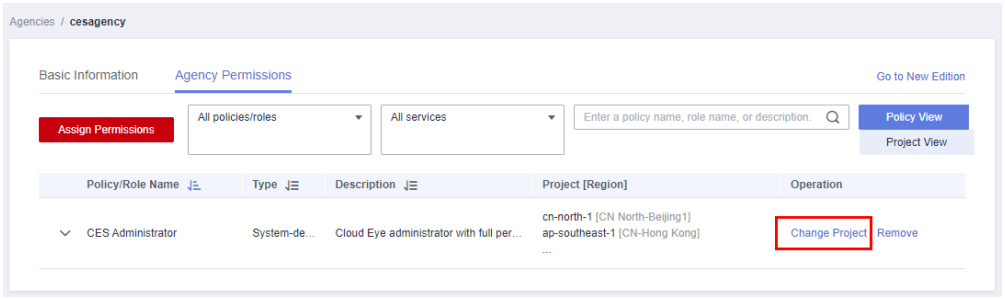


**Figure 2-14** Assigning permissions




- If the problem is resolved, no further action is required.
- Otherwise, go to **6**.

5. Run the following command to check whether the DNS settings are normal:

   **ping *agent.ces.myhuaweicloud.com***

   - If yes, no further action is required.
   - If no, modify the **DNS server address** or the Cloud Eye endpoint.

   NOTE

   For details about Cloud Eye endpoints for each region, see **Regions and Endpoints**.

# 2.2.5 How Can I Troubleshoot the Issue of Reported Metrics Being Discarded?

## Symptom

The plug-in status is normal, but the monitoring data for some metric is not continuous.

## Analysis

Possible causes are as follows:

- When there is a large gap between the Linux time and the actual time, the metrics collected by the Agent are considered invalid when being reported to the server. As a result, the reported metrics are discarded.

## Procedure (Linux)

Log in to the host as a root user, ensure that the ntp service is normal, and the run the following command:

```
ntpdate -u ntp.myhuaweicloud.com
```

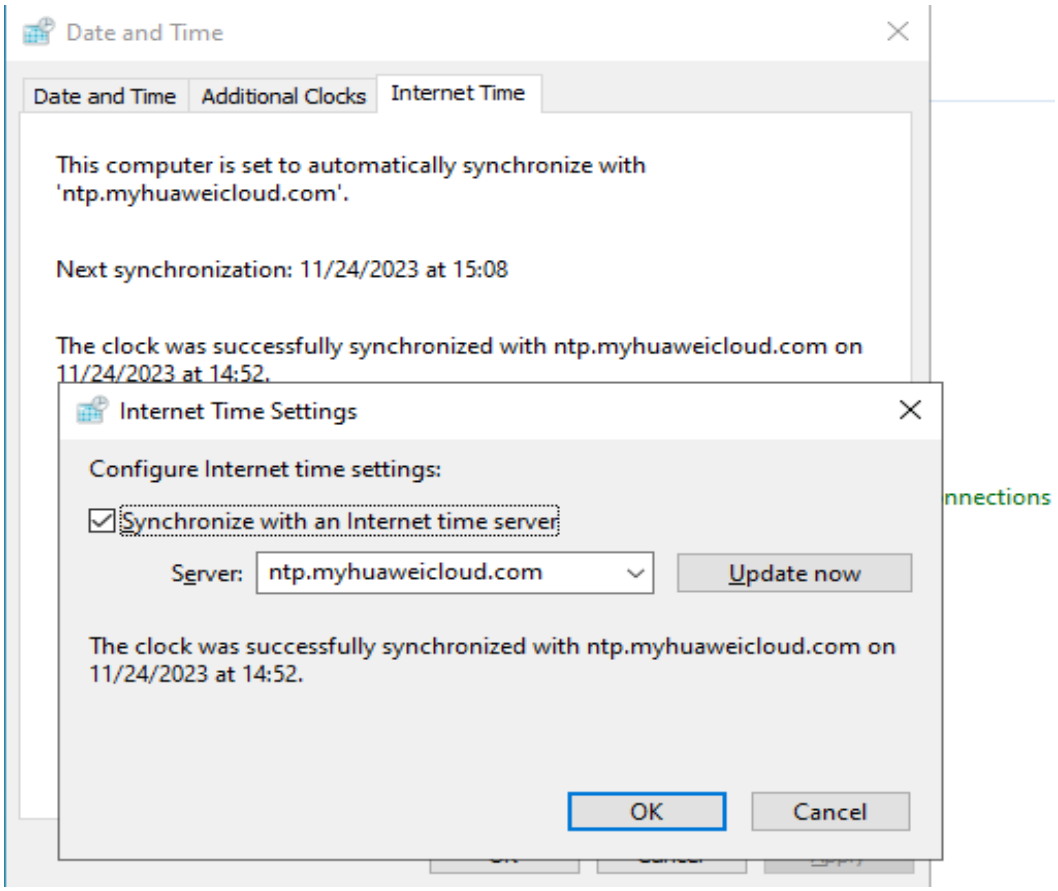
Or use another ntp address.

## Procedure (Windows)

Log in to the host as an administrator and ensure that the NTP service is normal. Choose **Control Panel** > **Date and Time** > **Internal Time** > **Change Settings**.

Enter the ntp address, for example, ntp.myhuaweicloud.com.




## 2.2.6 What Should I Do If the Agent Status Is Faulty?

The OS monitoring Agent sends a heartbeat message to Cloud Eye every minute. If Cloud Eye does not receive any heartbeat messages for 3 minutes, **Agent Status** is **Faulty**.

The possible causes are:

- The domain name of the Agent cannot be resolved. Check whether the DNS server address is correct by referring to **Modifying the DNS Server Address and Adding Security Group Rules (Linux)**. If yes, check whether the Agent is correctly configured by referring to **(Optional) Manually Configuring the Agent (Linux)**.
- Your account is in arrears.
- If the Agent process is faulty, restart the Agent by referring to **Managing the Agent**. If the Agent cannot be restarted, related files are deleted by mistake. In this case, reinstall the Agent.
- The server time is inconsistent with the local standard time.
- The log path varies depending on the Agent version.

  The log paths are as follows:

  – Linux:

  New Agent version: **/usr/local/uniagent/extension/install/telescope/log/ces.log**

  Early Agent version: **/usr/local/telescope/log/ces.log**

  - Windows:

    New version: **C:\Program Files\uniagent\extension\install\telescope\log\ces.log**

    Earlier version: **C:\Program Files\telescope\log\ces.log**

- If the DNS server is not a Huawei Cloud DNS server, run the **dig agent.ces.myhuaweicloud.com** command to obtain the IP address of the Huawei Cloud private DNS server and then add the IP address to the **hosts** file. For details, see **What Are the Private DNS Server Addresses Provided by Huawei Cloud?**

## 2.2.7 What Should I Do If the Agent Is Stopped?

### Viewing the Agent Version

1. Log in to an ECS as user **root**.
2. Run the following command to check the Agent version:

   **if [[ -f /usr/local/uniagent/extension/install/telescope/bin/telescope ]]; then /usr/local/uniagent/extension/install/telescope/bin/telescope -v; elif [[ -f /usr/local/telescope/bin/telescope ]]; then echo "old agent"; else echo 0; fi**

   - If **old agent** is displayed, the early version of the Agent is used.
   - If a version ID is returned, the new version of the Agent is used.
   - If **0** is returned, the Agent is not installed.

### Starting the Agent (New Version)

Run the following command to start the Agent:

**/usr/local/uniagent/extension/install/telescope/telescoped start**

If a fault is reported, the Agent has been uninstalled or related files have been deleted. In this case, reinstall the Agent.

### Starting the Agent (Early Version)

Run the following command to start the Agent:

**service telescoped start**

If a fault is reported, the Agent has been uninstalled or related files have been deleted. In this case, reinstall the Agent.

## 2.2.8 What Should I Do If the Agent Status Is Running But There Is No Monitoring Data?

If there is no monitoring data 10 minutes after the Agent is installed, **InstanceId** in the **conf** file may be incorrectly configured.

Correct the configuration by performing operations described in **(Optional) Manually Configuring the Agent (Linux)**.

# 2.2.9 What Can I Do If No Monitoring Data Is Displayed After One-Click Agent Restoration? (Old Agent)

## Symptom

The Agent is running normally after being restored, but no monitoring data is generated.

## Possible Causes

If no OS monitoring data is available for an ECS or BMS with the Agent installed, the possible causes are as follows:

- There is a problem with the Agent process.
- There is a problem with agency configurations.
- Temporary AK/SK cannot be obtained due to incorrect route configurations.
- The network is not well connected.

Check the Agent version.

1. Log in to an ECS as user **root**.
2. Check the Agent version.

   **if [[ -f /usr/local/uniagent/extension/install/telescope/bin/telescope ]]; then /usr/local/uniagent/extension/install/telescope/bin/telescope -v; elif [[ -f /usr/local/telescope/bin/telescope ]]; then echo "old agent"; else echo 0; fi**

   - If **old agent** is displayed, the early version of the Agent is used.
   - If a version ID is returned, the new version of the Agent is used.
   - If **0** is returned, the Agent is not installed.

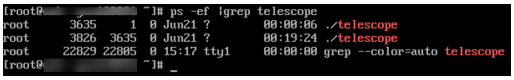
## Procedure (Linux)

1. Log in to the ECS or BMS as user **root**.
2. Check whether the Agent process is running.

   **ps -ef |grep telescope**

   The following information indicates that the Agent process is normal.

   **Figure 2-15** Viewing the telescope processes

   - If the telescope process is normal, go to **4**.
   - If the telescope process is abnormal, go to **3**.
3. Start the Agent.

   **/usr/local/telescope/telescoped start**
4. Check whether an agency has been created for the server.

**curl http://169.254.169.254/openstack/latest/securitykey**

- If data is returned, the agency is normal and AK/SK can be obtained. No further action is required.
- If the request fails or the following information is displayed, go to **5**.

**Figure 2-16** Failing to obtain the AK/SK

```
<html>
 <head>
  <title>401 Unauthorized</title>
 </head>
 <body>
  <h1>401 Unauthorized</h1>
  agency_name is empty in metadata<br /><br />


 </body>
```

5. On the Cloud Eye console, choose **Server Monitoring** > **Elastic Cloud Server**, select the target ECS, and click **Restore Agent Configurations**.
   - If the problem is resolved, no further action is required.
   - Otherwise, go to **6**.
6. Check the route.

   **route -n**

   The following information indicates that the route is normal.

   **Figure 2-17** Normal route configuration-Linux

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 eth0
169.254.169.254 192.168.0.1     255.255.255.255 UGH   100    0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 eth0
```

   - If the route is normal, no further action is required.
   - Otherwise, go to **7**.
7. If the route does not exist, run the following command to add a route:

   **route add -host 169.254.169.254 gw *192.168.0.1***

   NOTE

   Replace *192.168.0.1* in the example command with the gateway of the server.

   Check whether monitoring data can be reported normally.
   - If yes, no further action is required.
   - If no, go to **8**.
8. Open the Agent configuration file.

   **cat /usr/local/telescope/bin/conf_ces.json**
9. Obtain the endpoint from the Agent configuration file.

   **Figure 2-18** Querying the Agent endpoint

```
[root@hss log]# cat /usr/local/telescope/bin/conf_ces.json
{
"Endpoint": "https://ces.cn-south-1.myhuaweicloud.com"
}[root@hss log]#
```

10. Check whether the DNS settings are normal.

    **ping *ces.cn-south-1.myhuaweicloud.com***

- If yes, no further action is required.
- If no, modify the **DNS server address** or the Cloud Eye endpoint.

  NOTE

  For details about Cloud Eye endpoints for each region, see **Regions and Endpoints**.

## Procedure (Windows)

1. Log in to the ECS or BMS as an administrator.
2. Open the **Task Manager** and check whether the Agent process is running.

   If there are **Figure 2-19** and **Figure 2-20**, the Agent process is running.

   **Figure 2-19** agent process (Windows)

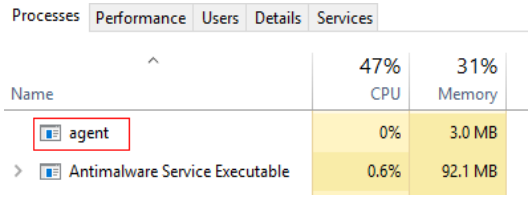
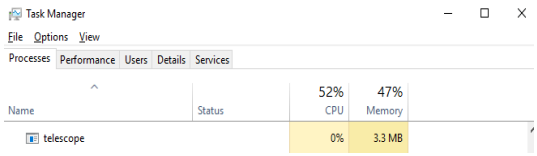
   
   

   **Figure 2-20** telescope process (Windows)

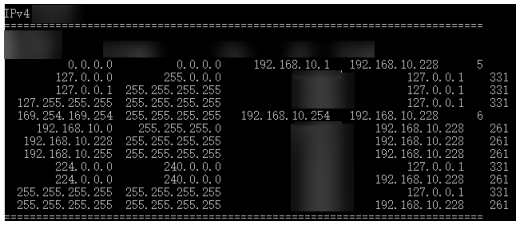
   
   

   - If the telescope process is normal, go to **4**.
   - If the telescope process is abnormal, go to **3**.
3. Double-click **start.bat** to start the Agent.
4. Access **http://169.254.169.254/openstack/latest/meta_data.json** and check whether the agency has been created.
   - If the website is accessible, the agency is normal. No further action is required.
   - Otherwise, go to **6**.
5. Check the route.

   **route print**

   The following information indicates that the route is normal.

   **Figure 2-21** Normal route configuration-Windows

- If the route is normal, no further action is required.
- Otherwise, go to **7**.

6. If the route does not exist, run the following command to add a route:

   **route add -host 169.254.169.254 gw *192.168.0.1***

   NOTE

   Replace *192.168.0.1* in the example command with the gateway of the server.

   Check whether monitoring data can be reported normally.

   - If yes, no further action is required.
   - If no, go to **7**.

7. Open the configuration file in **bin/conf_ces.json** in the directory where the Agent installation package is stored.
8. Obtain the endpoint from the Agent configuration file.

   {"Endpoint":"https://ces.cn-north-4.myhuaweicloud.com"}

9. Check whether the DNS settings are normal.

   **ping *ces.cn-north-4.myhuaweicloud.com***

   - If yes, no further action is required.
   - If no, modify the **DNS server address** or the Cloud Eye endpoint.

     NOTE

     For details about Cloud Eye endpoints for each region, see **Regions and Endpoints**.

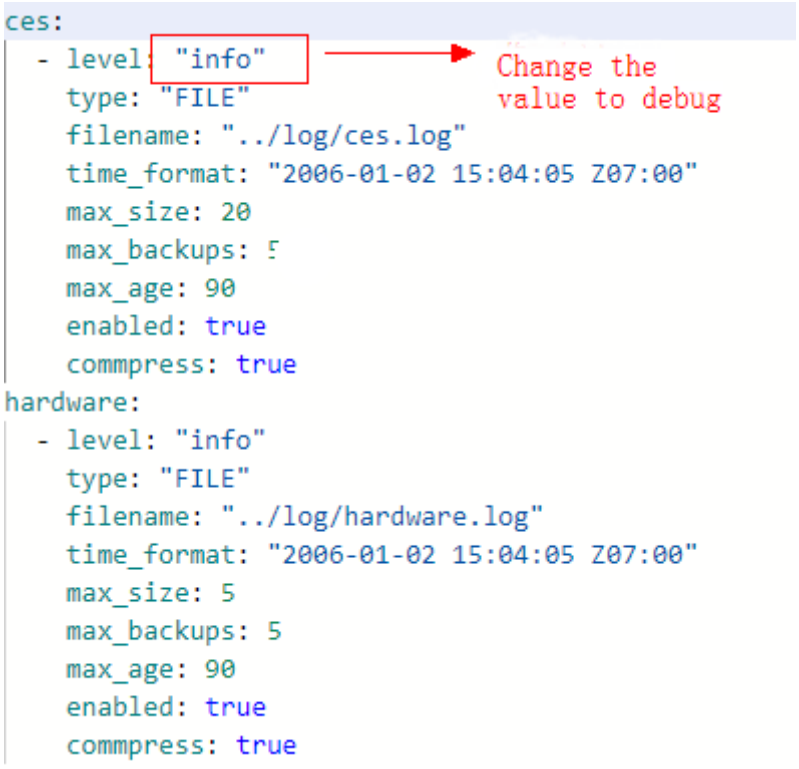
## 2.2.10 How Do I Obtain Debug Logs of the Agent?

### Procedure

1. Locate and modify the Agent log configuration file. Change **info** to **debug** in the **<ces>** and **<ces_new>** sections. If there is only one of the **<ces>** or the **<ces_new>** sections, you only need to modify one section.
   - Linux: /usr/local/uniagent/extension/install/telescope/bin/logs_config.xml
   - Windows: C:\Program Files\uniagent\extension\install\telescope\bin\logs_config.xml


2. If the configuration file in **1** is not found, modify the other configuration file.
   - Linux: /usr/local/uniagent/extension/install/telescope/conf/logs.yaml
   - Windows: C:\Program Files\uniagent\extension\install\telescope\conf\logs.yaml




3. Restart the Agent based on **Managing the Agent**.
4. After obtaining the debug logs, restore the modified configurations and restart the Agent based on **Managing the Agent**.

## 2.2.11 Why Is OS Monitoring Data Not Displayed Immediately After the Agent Is Installed and Configured or Not Displayed at All?

After you install the Agent successfully, choose **Server Monitoring**, wait for 2 minutes before you can see the monitoring data on the Cloud Eye console.

If **Agent Status** is **Running**, you have waited for 5 minutes, but there is still no OS monitoring data displayed, check whether the ECS or BMS time and the console client time are consistent.

When the Agent reports data, it uses the ECS or BMS local time. When the console delivers requests, it uses the browser time of the user client. If the two times are different, no OS monitoring data will be displayed on the Cloud Eye console.

NOTE

Run the **timedatectl set-timezone 'Asia/Shanghai'** command to change the BMS time to the browser time of the user client.

## 2.2.12 Why Is the Metric Collection Point Lost During Certain Periods of Time?

There may be no monitoring data for that period, which can be perfectly normal. The Agent collects metrics based on the server OS time, and sometimes time synchronization leads to server time changes, which can result in the appearance of periods when no data was collected.

## 2.2.13 Why Are the Inbound Bandwidth and Outbound Bandwidth Negative?

If Docker is installed, the early version of the Agent cannot collect statistics on the inbound and outbound bandwidth of virtual NICs when the container is restarted. As a result, a negative value is generated because the difference is calculated.

To update the Agent, see **Managing the Agent**.

## 2.2.14 Why Is There No Block Device Usage Metric for One of the Two Disks on a Server?

### Symptom

When you view the disk I/O metrics on the Cloud Eye console, two disks **vda** and **vdb** are displayed. The block device usage metric is available for **vda** but not for **vdb**.


## Possible Causes

If a disk is not attached to a mount point, the block device usage of the disk will not be collected. In the following figure, the **vdb** disk is not attached.

```
[root@ecs-block-0001 conf]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
└─vda1 253:1    0  40G  0 part /
vdb    253:16   0  10G  0 disk
```

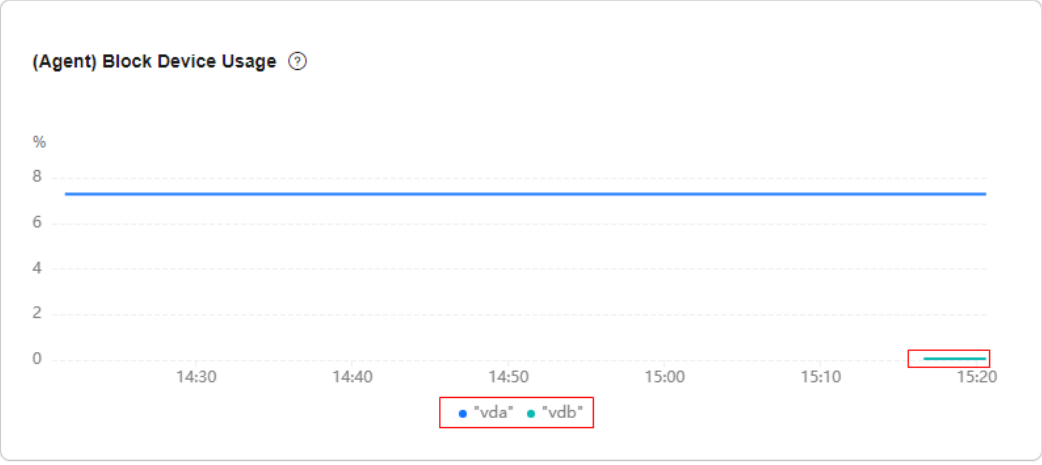
## Procedure

1. Attach the disk to a mount point.

```
[root@ecs-block-0001 mnt]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
└─vda1 253:1    0  40G  0 part /
vdb    253:16   0  10G  0 disk
└─vdb1 253:17   0   2G  0 part /mnt/sdc
```

2. Verify that the block device usage can be collected.




## 2.2.15 Why Is the Agent Status Abnormal on the Cloud Eye Server Monitoring Page While OS Monitoring Metrics Are Displayed as Normal?

### Symptom

On the Cloud Eye server monitoring page, the Agent status was abnormal, but the OS monitoring metrics were displayed as normal.

### Possible Causes

The Agent of an earlier version had a problem with status management. As a result, the Agent was displayed as abnormal on the console, but the Agent was functioning properly and OS monitoring metrics cloud be collected and reported.

### Procedure

**Step 1** Uninstall the Agent. For details, see **Managing the Agent**.

**Step 2** Install an Agent of the latest version. For details, see **Installing the Agent**.

**----End**

# 2.3 Cloud Service Monitoring

## 2.3.1 What Should I Do If I See Garbled Chinese Characters in an Exported CSV File?

You can export the Cloud Eye monitoring data to a CSV file, but when you open this file with Excel, there may be garbled Chinese characters. This happens when the exported CSV file is encoded in UTF-8, but the Excel is opened in ANSI format. To solve this problem, use either of the following solutions:

- Use a text editor such as Notepad or use WPS to open the CSV file you exported.

- Open the CSV file with Excel, but in the following manner:
  a. Create an EXCEL file.
  b. Choose **Data** > **From Text**.
  c. Select the exported CSV file and click **Import**.
     The **Text Import Wizard** dialog box is displayed.
  d. Select **Delimited** and click **Next**.
  e. Deselect **Tab**, select **Comma**, and click **Next**.
  f. Click **Finish**.
  g. In the **Import Data** dialog box, click **OK**.

## 2.3.2 Why Is the Monitoring Data Not Displayed on the Cloud Eye Console?

Possible causes are as follows:

- The cloud service is not interconnected with Cloud Eye. To check whether a service has been interconnected with Cloud Eye, see **Services Interconnected with Cloud Eye**.
- The collection and monitoring frequency for each service that has been interconnected with Cloud Eye is not the same. The data may have just not been collected yet.
- The ECS or BMS has been stopped for more than 1 hour.
- The EVS disk has not been attached to an ECS or BMS.
- No backend server is bound to the elastic load balancer or all of the backend servers are stopped.
- It has been less than 10 minutes since the resource was purchased.
- By default, if cloud services do not report data to Cloud Eye, they will not be displayed on the **Cloud Service Monitoring** page 1 to 3 hours later, or seven days later for some services, such as APIG, OBS, FunctionGraph, and APIG (dedicated).

## 2.3.3 Why I Cannot See the Monitoring Data on the Cloud Eye Console After Purchasing Cloud Service Resources?

The cloud platform is working to interconnect Cloud Eye with more cloud services. Before the interconnection is complete, you cannot view the resource monitoring data of the cloud services that have not been interconnected with Cloud Eye. If you want to check the resource monitoring data of the cloud services you purchased, you need to first check whether the cloud services have been interconnected with Cloud Eye.

If the services have been interconnected with Cloud Eye, wait for a period of time, because the frequencies of each service to collect and report data to Cloud Eye are different. You can view the resource monitoring graph after Cloud Eye collects the first piece of monitoring data.

# 2.4 Alarm Management

## 2.4.1 When Will an "Insufficient data" Alarm Be Triggered?

When monitoring data of a metric is not reported to Cloud Eye for three consecutive hours, the alarm rule status changes to **Insufficient data**.

In special cases, if the monitoring data reporting interval is longer than three hours and no monitoring data is reported for three consecutive intervals, the alarm rule status also changes to **Insufficient data**.

## 2.4.2 Why Did I Receive a Bandwidth Overflow Notification While There Is No Bandwidth Overflow Record in the Monitoring Data?

You may have configured Cloud Eye to trigger alarm notifications immediately when the bandwidth overflow occurs. However, if the average bandwidth for the last 5 minutes falls under the preset threshold, no alarm will be recorded in the system.

## 2.4.3 Why Can't an Alarm Be Triggered After a 5-Minute Aggregated Metric Alarm Rule Is Configured?

### Symptom

A 5-minute aggregated metric alarm rule has been configured for services such as CBR. No alarms were reported even if the triggering conditions had been met three times in a row.

### Possible Causes

A metric data record is reported for the CBR service every 15 minutes. After the region of the CloudSense engine is switched, such metrics cannot reach the threshold in two consecutive 5-minute time windows. As a result, the alarm cannot be triggered.

## 2.4.4 Why Is an Alarm Triggered Contrary to the Alarm Rule That Both the Disk Read and Write Metrics Must Reach the Thresholds

### Symptom

The alarm rule is that both the disk read and write metrics must reach the thresholds, but an alarm was triggered when only one of them reached the threshold.

### Possible Causes

The ECS reporting the alarm has more than one disk. The read metric of one disk and the write metric of another disk reached the thresholds.

Currently, cloud product alarms are determined by instance. An alarm can be triggered for an instance as long as the metric of any resource of this instance

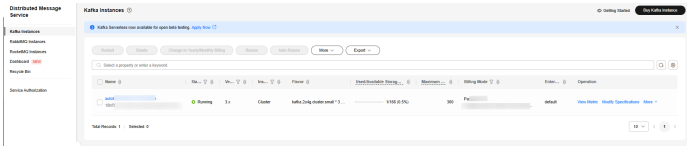
reaches the threshold, instead of the metric of a specific resource. To configure an alarm rule for a specific resource, you are advised to select **Specific dimension** for **Resource Level**.

# 2.5 Data Dump

## 2.5.1 What Should I Do When There Is an Abnormal Dump Destination?

On the **Data Dump** page, if **The resource is abnormal** is displayed in the **Destination** column, the possible causes are as follows:

1. **Resource not found**: If no dump destination is found, data dump will fail on Cloud Eye. You can log in to the Distributed Message Service console and check if the destination Kafka cluster has been deleted from the resource list. If it is, you cannot configure it as a dump destination. You need to modify the dump task or delete it and then create a new one.



2. **Insufficient agency permissions**: If the destination is **Other account**, and the agency does not have sufficient permissions, data dump will fail on Cloud Eye. In this case, you need to confirm that the agency has **all necessary permissions**. If not, add them again.
3. **Other**: Click **Modify** in the **Operation** column to reconfigure a dump destination. Then, restart the task, and check if the dump task can be restored.

# 2.6 API

## 2.6.1 How Can I Query Monitoring Data of Multiple Metrics?

### FAQ Before API Calling

1. Issue 1

   **Description**

   How can I call the POST /V1.0/{*project_id*}/batch-query-metric-data API?

   **Solution**

   Rectify the fault by following the instructions provided in **Querying Monitoring Data of Multiple Metrics**.

2. Issue 2

   **Description**

   What is the maximum query time range supported when I call the POST /V1.0/{project_id}/batch-query-metric-data API?

**Solution**

The maximum query range is 155 days.

3. Issue 3

**Description**

What are the domain names for calling this API in different regions?

**Solution**

For details, see **Regions and Endpoints**.

## FAQ During API Calling

1. Issue 1

**Description**

The response status code is 200 and no metric data is returned. Example response:

```
{
   "metrics": [{
      "namespace": "SYS.RDS",
      "metric_name": "rds039_disk_util",
      "dimensions": [{
         "name": "instance_id",
         "value": "5e319882ffa04c968e469035a116b2d1in04"
      }],
      "datapoints": [],##No metric data is displayed in the data points.
      "unit": "unknown"
   }]
}
```

**Possible causes**

- Cause 1: The namespace of the cloud service is incorrect. For details, see Case 1.
- Cause 2: The requested resource cannot be found. For details, see Case 2.
- Cause 2: The dimension is not applicable to the cloud service. For details, see Case 3.
- Cause 3: The metric ID is not applicable to the cloud service. For details, see Case 4.

**Solution**

For details about the namespaces, dimensions, and metrics, see **Services Interconnected with Cloud Eye**.

**Case 1** : **The namespace of the cloud service is incorrect.**

The namespace corresponding to the metric **mem_usedPercent** is **AGT.ECS**.

Request parameters

```
{
   "from": 1724311893283,
   "to": 1724315493283,
   "period": "1",
   "filter": "average",
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "129718f5-833d-4f78-b685-6b1c3091ea6"
      }],
      "metric_name": "mem_usedPercent",
      "namespace": "SYS.ECS"  ##Incorrect namespace
   }]
}
```

Response parameters

```
{
   "metrics": [{
      "namespace": "SYS.ECS",
      "metric_name": "mem_usedPercent",
      "dimensions": [{
         "name": "instance_id",
         "value": "129718f5-833d-4f78-b685-6b1c3091ea6"
      }],
      "datapoints": [],
      "unit": "unknown"
   }]
}
```

**Case 2 : The requested resource cannot be found.**

The instance **129718f5-833d-4f78-b685-6b1c3091ea7** is not in the ECS list.

Request parameters

```
{
   "from": 1724311893283,
   "to": 1724315493283,
   "period": "1",
   "filter": "average",
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "129718f5-833d-4f78-b685-6b1c3091ea7"  ##129718f5-833d-4f78-b685-6b1c3091ea7 is not in the ECS list.
      }],
      "metric_name": "mem_usedPercent",
      "namespace": "AGT.ECS"
   }]
}
```

Response parameters

```
{
   "metrics": [{
      "namespace": "AGT.ECS",
      "metric_name": "mem_usedPercent",
      "dimensions": [{
         "name": "instance_id",
         "value": "129718f5-833d-4f78-b685-6b1c3091ea6"
      }],
      "datapoints": [],
      "unit": "unknown"
   }]
}
```

**Case 3: The dimension is not applicable to the cloud service.**

**instance_id** is not a dimension of RDS.

Request parameters

```
{
   "metrics": [{
      "dimensions": [{
"name": "instance_id," ##The dimension is not a dimension of the cloud serviceinstance_id is not a dimension of RDS.
         "value": "5e319882ffa04c968e469035a116b2d1in04"
      }],
      "metric_name": "rds039_disk_util",
      "namespace": "SYS.RDS"
   }],
   "filter": "average",
   "period": "1",
   "from": 1724312777938,
   "to": 1724316377938
}
```

Response parameters

```
{
   "metrics": [{
      "namespace": "SYS.RDS",
      "metric_name": "rds039_disk_util",
      "dimensions": [{
         "name": "instance_id",
         "value": "5e319882ffa04c968e469035a116b2d1in04"
      }],
      "datapoints": [],
      "unit": "unknown"
   }]
}
```

**Case 4: The metric ID is not applicable to the cloud service.**

The **rds958_disk_util** metric ID is not applicable to RDS.

Request parameters

```
{
   "metrics": [{
      "dimensions": [{
         "name": "rds_cluster_sqlserver_id",
         "value": "5e319882ffa04c968e469035a116b2d1in04"
      }],
      "metric_name": "rds958_disk_util",   ##The rds958_disk_util metric ID is not applicable to RDS.
      "namespace": "SYS.RDS"
   }],
   "filter": "average",
   "period": "1",
   "from": 1724312777938,
   "to": 1724316377938
}
```

Response parameters

```
{
   "metrics": [{
      "namespace": "SYS.RDS",
      "metric_name": "rds958_disk_util",
      "dimensions": [{
         "name": "rds_cluster_sqlserver_id",
         "value": "5e319882ffa04c968e469035a116b2d1in04"
      }],
      "datapoints": [],
      "unit": "unknown"
   }]
}
```

2. Issue 2

**Description**

The response status code is 200, and no data of the ECS disk usage is returned.

Example response

```
{
   "metrics": [{
      "namespace": "AGT.ECS",
      "metric_name": "disk_usedPercent",
      "dimensions": [{
         "name": "disk",
         "value": "012bec14bc176310c19f40e384fd629b"
      }, {
         "name": "instance_id",
         "value": "07d878a9-2243-4e84-aeef-c47747d18024"
      }],
      "datapoints": [],  ##No metric data is displayed in the metric data list.
      "unit": "unknown"
   }]
}
```

**Possible causes**

Cause 1: The namespace is incorrect. For details, see Case 1.

Cause 2: The metric dimension is incorrect. For details, see Case 2.

Cause 3: The Agent is not installed on the ECS. For details, see Case 3.

Cause 4: The Agent installed on the ECS does not report the disk usage. For details, see Case 4.

**Case 1: incorrect namespace**

If the OS monitoring metrics of an ECS are queried, the namespace must be **AGT.ECS**.

Request parameters

```
{
   "from": 1724118017498,
   "to": 1724121617498,
   "period": "1",
   "filter": "average",
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "07d878a9-2243-4e84-aeef-c47747d18024"
      }, {
         "name": "mount_point",
         "value": "012bec14bc176310c19f40e384fd629b"
      }],
      "metric_name": "disk_usedPercent",
      "namespace": "SYS.ECS"  ##Incorrect namespace
   }]
}
```

Response parameters

```
{
   "metrics": [{
      "namespace": "SYS.ECS",
      "metric_name": "disk_usedPercent",
      "dimensions": [{
         "name": "mount_point",
         "value": "012bec14bc176310c19f40e384fd629b"
      }, {
         "name": "instance_id",
         "value": "07d878a9-2243-4e84-aeef-c47747d18024"
      }],
      "datapoints": [],
      "unit": "unknown"
   }]
}
```

**Case 2: incorrect metric dimension**

The disk usage is queried by mount point. Two dimensions: **instance_id** and **mount_point** are required in the request parameters for querying the disk usage.

Request parameters

```
{
   "from": 1724118017498,
   "to": 1724121617498,
   "period": "1",
   "filter": "average",
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "07d878a9-2243-4e84-aeef-c47747d18024"
      }, {
         "name": "disk," ## Incorrect metric dimension
```

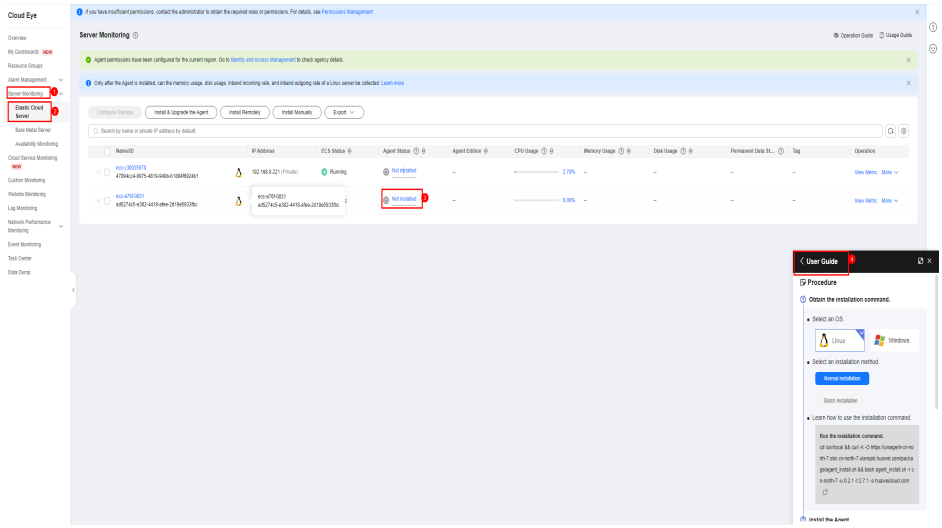
```
            "value": "012bec14bc176310c19f40e384fd629b"
        }],
        "metric_name": "disk_usedPercent",
        "namespace": "AGT.ECS"
    }]
}
```

Response parameters

```
{
    "metrics": [{
        "namespace": "AGT.ECS",
        "metric_name": "disk_usedPercent",
        "dimensions": [{
            "name": "disk",
            "value": "012bec14bc176310c19f40e384fd629b"
        }, {
            "name": "instance_id",
            "value": "07d878a9-2243-4e84-aeef-c47747d18024"
        }],
        "datapoints": [],
        "unit": "unknown"
    }]
}
```

**Case 3: The Agent is not installed on an ECS.**

On the Cloud Eye console, choose **Server Monitoring** > **Elastic Cloud Server**, locate the ECS, and click **Not installed** in the **Agent Status** column. Install the Agent as instructed.




**Case 4: The Agent installed on the ECS does not report the disk usage.**

The Agent is faulty, so there is no metric data reported. For details, see **Troubleshooting Common Agent Issues**.

3. Issue 2

**Description**

The number of metric data records reported within a specified time range is greater than 3,000. However, the number of returned metric data records is less than 3,000 when the current API is called.

**Possible causes**

The default maximum query interval (**to**–**from**) varies depending on **period** and the number of metrics to be queried. The rule is as follows: The number of metrics × (**to** - **from**)/Monitoring interval ≤ 3000.

NOTE

- The number of metrics: the number of elements in the **metrics** attribute, a request parameter
- Monitoring period: value of the **period** attribute in the request. The unit is minute by default and needs to be converted into milliseconds.
- **3000**: the total number of all data points (the metric data list) in the response body

**Cause 1**: If the metric reporting period is shorter than the monitoring period for querying monitoring data of multiple metrics, multiple metrics reported during the monitoring period are aggregated into a point based on the aggregation rule. As a result, the queried metric data volume is less than the reported data volume.

**Cause 2**: If the metric reporting period is the same as the monitoring period for querying monitoring data of multiple metrics, a maximum of 3,000 metric data records can be returned according to the preceding calculation rule.

**Solution**

a. Set the monitoring period in the request parameters to a smaller value based on the enumerated values provided in the API reference.

b. Use the API for querying monitoring data of a single metric. This API has no limit on the number of returned metric data records.

4. Issue 2

**Description**

When the API for querying monitoring data of multiple metrics is called, the time taken for returning the metric data points is much greater than the value of the **from** parameter in the request.

**Possible causes**

The default maximum query interval (**to**–**from**) varies depending on **period** and the number of metrics to be queried. The rule is as follows: The number of metrics × (**to** - **from**)/Monitoring interval ≤ 3000.

NOTE

- The number of metrics: the number of elements in the **metrics** attribute, a request parameter
- Monitoring period: value of the **period** attribute in the request. The unit is minute by default and needs to be converted into milliseconds.
- **3000**: the total number of all data points (the metric data list) in the response body

For example, if 300 metrics are queried in batches and the monitoring interval is 60,000 ms, the maximum value of (**to**-**from**) is **600000**. If (**to-from**) exceeds 600,000, **from** is automatically changed to **to-600000**.

**Cause 1**: According to the preceding formula, the number of metrics is too large. For details, see Case 1.

**Cause 2**: According to the preceding formula, the monitoring period is too short. For details, see Case 1.

**Solution**

**Cause 1**: There are too many metrics.

**Solution**: Reduce the number of metrics.

Use the API for querying monitoring data to query a single metric.

**Cause 2**: A small enumerated value for the monitoring period is selected.

**Solution**: Set the monitoring period in the request parameter to a large enumerated value provided in the API reference.

**Case 1: too many metrics and short monitoring period**

If the number of requested metrics is 300 and the monitoring period is 1 minute, the maximum value of (**to** - **from**) is 600000. If the value of the request parameter (**to** – **from**) is 3600000 (1724742027556 – 1724738427556), which exceeds the value of 600000, the value of **from** is automatically changed to "**to** – 600000", that is, 1724742027556 – 600000 = 1724741427556.

The earliest metric data point returned by the API within the specified time range is **2024-08-27 14:51:27**, but the query start time is **2024-08-27 14:00:27**.

Request parameters


```
{
 "metrics": [
  {
   "dimensions": [
    {
     "name": "disk_name",
     "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vda"
    }
   ],
   "metric_name": "disk_device_read_bytes_rate",
   "namespace": "SYS.EVS"
  },
  {
   "dimensions": [
    {
     "name": "disk_name",
     "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vdc"
    }
   ],
   "metric_name": "disk_device_read_bytes_rate",
   "namespace": "SYS.EVS"
  },
  {
   "dimensions": [
    {
     "name": "disk_name",
     "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vda"
    }
   ],
   "metric_name": "disk_device_write_bytes_rate",
   "namespace": "SYS.EVS"
  },
  {
   "dimensions": [
    {
     "name": "disk_name",
     "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vdc"
    }
   ],
   "metric_name": "disk_device_write_bytes_rate",
   "namespace": "SYS.EVS"
  },
  ......
  {
   "dimensions": [
    {
     "name": "nat_gateway_id",
```

```
        "value": "3c55363f-6416-45ca-8512-cf1f6f2533e7"
      }
     ],
     "metric_name": "inbound_pps",
     "namespace": "SYS.NAT"
   }
 ],
 "filter": "max",
 "period": "1",
 "from": 1724738427556, ##2024-08-27 14:00:27
 "to": 1724742027556 ##2024-08-27 15:00:27
}
```

Response parameters

```
{
   "metrics": [
      {
         "namespace": "SYS.EVS",
         "metric_name": "disk_device_read_bytes_rate",
         "dimensions": [
            {
               "name": "disk_name",
               "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vda"
            }
         ],
         "datapoints": [
            {
               "max": 0,
               "timestamp": 1724741487000  ##2024-08-27 14:51:27
            },
            {
               "max": 0,
               "timestamp": 1724741547000  ##2024-08-27 14:52:27
            },
            {
               "max": 0,
               "timestamp": 1724741607000
            },
            {
               "max": 0,
               "timestamp": 1724741667000
            },
            {
               "max": 0,
               "timestamp": 1724741727000
            },
            ......
         ],
         "unit": "B/s"
      },
      {
         "namespace": "SYS.EVS",
         "metric_name": "disk_device_read_bytes_rate",
         "dimensions": [
            {
               "name": "disk_name",
               "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vdc"
            }
         ],
         "datapoints": [
            {
               "max": 0,
               "timestamp": 1724741487000
            },
            {
               "max": 0,
               "timestamp": 1724741547000
            },
            {
               "max": 0,
```

```
                "timestamp": 1724741607000
            },
            {
                "max": 0,
                "timestamp": 1724741667000
            },
            {
                "max": 0,
                "timestamp": 1724741727000
            },
             ......
        ],
        "unit": "B/s"
    },
    {
        "namespace": "SYS.EVS",
        "metric_name": "disk_device_write_bytes_rate",
        "dimensions": [
            {
                "name": "disk_name",
                "value": "6a2bf14a-e3be-4fc9-8522-ba6fe7f0b503-vda"
            }
        ],
        "datapoints": [
            {
                "max": 3055.1,
                "timestamp": 1724741487000
            },
            {
                "max": 3195.78,
                "timestamp": 1724741547000
            },
            {
                "max": 2973.39,
                "timestamp": 1724741607000
            },
            {
                "max": 3533.52,
                "timestamp": 1724741667000
            },
            {
                "max": 2636.8,
                "timestamp": 1724741727000
            },
             .......
        ],
        "unit": "B/s"
    },
    ......
  ]
}
```


## Troubleshooting Common 4*XX* Issues

1. **HTTP status code**

   400

   **Error code**

   ces.0014

   **Possible causes**

   Cause 1: The request parameter format is incorrect. For details, see Case 1.

   Cause 2: Mandatory fields are not transferred. For details, see Case 2.

   **Case 1: incorrect request parameter format**

a. The values of the **from** and **to** attributes must be converted into millisecond.

b. The **period** attribute supports the following enumerated values: **1300**, **1200**, **3600**, **14400**, and **86400**.

c. The **filter** attribute supports the following enumerated values: **average**, **max**, **min**, **sum**, and **variance**.

Request parameters

```
{
   "from": 1724331974,   ## The input parameter is in seconds instead of milliseconds.
   "to": 1724315493,     ##The input parameter is in seconds instead of milliseconds.
   "period": "10086",    ##Invalid period value
   "filter": "standard", ##Invalid filter value
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "129718f5-833d-4f78-b685-6b1c3091ea69"
      }],
      "metric_name": "mem_usedPercent",
      "namespace": "AGT.ECS"
   }]
}
```

Response parameters

```
{
   "http_code": 400,
   "message": {
      "details": "Some content in message body is not correct, error message: [from,
to]",##from and to issues
      "code": "ces.0014"
   },
   "encoded_authorization_message": null
}
```

**Case 2: Mandatory fields are not transferred.**

The mandatory field **filter** is not transferred. For more mandatory fields, see the API reference.

Request parameters

```
{
   "from": 1724119607020,
   "to": 1724123207020,
   "period": "1",
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "238764d4-c4e1-4274-88a1-5956b057766b"
      }],
      "metric_name": "mem_usedPercent",
      "namespace": "AGT.ECS"
   }]
}
```

Response parameters

```
{
   "http_code": 400,
   "message": {
      "details": "Some content in message body is not correct, error message: [filter]",  ##filter
issues
      "code": "ces.0014"
   },
   "encoded_authorization_message": null
}
```

2. **HTTP status code**

   404

**Error code**

APIGW.0101

**Possible causes**

Cause 1: The path URI is inconsistent with that in the API reference. For details, see Case 1.

**Case 1: The URI of the path is inconsistent with that in the API reference.**

The version in the URI in the request path is incorrect. The correct version is V1.0 instead of V1. The correct URI is /V1.0/{project_id}/batch-query-metric-data.

Request path

```
POST  /V1/04f9aca88c00d3202fd4c01ed679daf0/batch-query-metric-data
```

Response parameters

```
{
   "error_code": "APIGW.0101",
   "error_msg": "The API does not exist or has not been published in the environment",
   "request_id": "7d7a8258354300ac158c7b14a158d6ec"
}
```

3. **HTTP status code**

   401

   **Error code**

   ces.0015

   **Possible causes**

   Cause 1: The project ID in the request for obtaining the token from IAM is different from the project ID used for calling the API for querying monitoring data of multiple metrics.

   Cause 2: The token has expired.

   Cause 3: The token content is copied less or more.

   Cause 4: The AK and SK do not match.

   **Troubleshooting**

   Rectify the fault based on the possible causes.

   **Solution**

   Cause 1: The project ID in the request for obtaining the token from IAM is different from the project ID used for calling the API for querying monitoring data of multiple metrics.

   Solution: Ensure that the two project IDs are the same.

   Cause 2: The token has expired.

   Solution: Generate a new token.

   Cause 3: The token content is copied less or more.

   Solution: Obtain the correct token.

   Cause 4: The AK and SK do not match.

   Solution: Obtain the AK and SK of the tenant.

   **Case**

   The authentication fails due to a token exception.

   Request header

   ```
   X-Auth-Token: MIIqDgYJKoZIhvcNAQcCoIIp-zCC.....+6CIyAFrbHVxQZJ2Jq    ##Token is abnormal.
   ```

Request parameters

```
{
   "from": 1724311893283,
   "to": 1724315493283,
   "period": "1",
   "filter": "average",
   "metrics": [{
      "dimensions": [{
         "name": "instance_id",
         "value": "129718f5-833d-4f78-b685-6b1c3091ea69"
      }],
      "metric_name": "mem_usedPercent",
      "namespace": "AGT.ECS"
   }]
}
```

Response parameters

```
{
   "http_code": 401,
   "message": {
      "details": "Authenticate failed.",
      "code": "ces.0015"
   },
   "encoded_authorization_message": null
}
```

4. **HTTP status code**

   403

   **Error code**

   ces.0050

   **Possible causes**

   Possible cause 1: The user policy does not contain the **ces:metricData:list** permissions. For details, see Case 1.

   **Case 1: The user policy does not contain the ces:metricData:list fine-grained permissions.**

   Add the **ces:metricData:list** action to the policy to which the user belongs.

   Request header

   ```
   X-Auth-Token:  MIIqDgYJKoZIhvcNAQcCoIIp-zCC.....+6CIyAFrbHVxQZJ2Jq
   ```

   Request parameters

   ```
   {
      "from": 1724311893283,
      "to": 1724315493283,
      "period": "1",
      "filter": "average",
      "metrics": [{
         "dimensions": [{
            "name": "instance_id",
            "value": "129718f5-833d-4f78-b685-6b1c3091ea69"
         }],
         "metric_name": "mem_usedPercent",
         "namespace": "AGT.ECS"
      }]
   }
   ```

   Response parameters

   ```
   {
      "http_code": 403,
      "message": {
         "details": "Policy doesn't allow [ces:metricData:list] to be performed.",   ## The user policy does not contain the ces:metricData:list fine-grained permissions.
         "code": "ces.0050"
   ```

```
    },
    "encoded_authorization_message": null
}
```

5. **HTTP status code**

   429

   **Error code**

   ces.0429

   **Possible causes**

   Cause 1: The API is throttled. For details, see Case 1.

   **Case 1: The API is throttled.**

   The request API is throttled. If the API is throttled, contact O&M personnel to configure a new process policy immediately.

   Request parameters

```
{
    "from": 1724311893283,
    "to": 1724315493283,
    "period": "1",
    "filter": "average",
    "metrics": [{
        "dimensions": [{
            "name": "instance_id",
            "value": "129718f5-833d-4f78-b685-6b1c3091ea69"
        }],
        "metric_name": "mem_usedPercent",
        "namespace": "AGT.ECS"
    }]
}
```

   Response parameters

```
{
    "http_code": 429,
    "message": {
        "details": "Too Many Requests.",
        "code": "ces.0429"
    },
    "encoded_authorization_message": null
}
```

# 2.6.2 How Can I Query Monitoring Data of a Metric?

## FAQ Before API Calling

1. Issue 1

   **Description**

   How can I call the GET /V1.0/{*project_id*}/metric-data API?

   **Solution**

   For details, see **Querying Monitoring Data of a Metric**.

2. Issue 2

   **Description**

   What are the domain names for calling this API in different regions?

   **Solution**

   For details, see **Regions and Endpoints**.

## Troubleshooting Common 4*XX* Issues

1. **HTTP status code**

   429

   **Error code**

   ces.0429

   **Possible causes**

   Cause 1: The API is throttled. For details, see Case 1.

   **Solution**

   If the API is throttled, contact O&M personnel to configure a new process policy immediately.

   **Case 1: The API is throttled.**

   View the CPU usage of the ECS whose ID is **6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d** from 20:00:00 to 22:00:00 on April 30, 2019. The monitoring interval is 20 minutes.

   ```
   GET https://{Cloud Eye endpoint}/V1.0/{project_id}/metric-data?
   namespace=SYS.ECS&metric_name=cpu_util&dim.0=instance_id,6f3c6f91-4b24-4e1b-b7d1-
   a94ac1cb011d&from=1556625600000&to=1556632800000&period=1200&filter=min
   ```

   Response parameters

   ```
   {
       "http_code": 429,
       "message": {
           "details": "Too Many Requests.",
           "code": "ces.0429"
       },
       "encoded_authorization_message": null
   }
   ```