Bare Metal Server

Best Practices

Issue 03
Date 2020-01-13

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

Notice

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

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

Contents

1 Monitoring

1.1 Overview

Solution Introduction

After purchasing a BMS, you want to know its running status. Bare Metal Server (BMS) works with the Cloud Eye service to automatically collect monitoring metrics, such as the CPU, memory, disk, and network usage of a BMS. These metrics help you learn about the running status and performance of your BMS in time.

This document is prepared based on the BMS and Cloud Eye practices and provides guidance for you to configure server monitoring for BMSs.

Constraints

- Agent can be installed only on BMSs running a 64-bit Linux OS.

- An agency must be configured for monitoring BMSs. For details, see **How Do I Create an Agency for Server Monitoring of the BMS?**.

- Only CN North-Beijing1 (**cn-north-1**), CN South-Guangzhou (**cn-south-1**), AP-Bangkok (**ap-southeast-2**), and CN-Hong Kong (**ap-southeast-1**) are supported now.

- Private images do not support this function.

  Table 1-1 lists the Linux images that support server monitoring.

  Table 1-1 Linux images that support server monitoring

OS Type (64-bit)	Version
SUSE	Enterprise11 SP4
CentOS	6.9, 7.2, and 7.3

1.2 Installing and Configuring the Agent for an Existing BMS

1.2.1 Installing the Agent

This section describes how to install the Agent for an existing BMS. The procedure is as follows:

1. **Adding the Resolved Domain Names**: Add the resolved domain names of regions to the **/etc/resolv.conf** file on the BMS.

2. **Configuring the Security Group**: Download the Telescope package, send metrics, and collect logs.

3. **Installing the Agent**: Manually install the Agent on the BMS.

Adding the Resolved Domain Names

1. Log in to the BMS as user **root**.

2. Enter **vi /etc/resolv.conf** to open the **/etc/resolv.conf** file.

3. Add **nameserver 100.125.1.250** and **nameserver 100.125.21.250** to the file, as shown in **Figure 1-1**.

   Figure 1-1 Adding the resolved domain names


   ```
   # Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
   #     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
   nameserver 100.125.1.250
   nameserver 114.114.114.114
   nameserver 114.114.115.115
   search openstacklocal
   ~
   ```

   📖 **NOTE**

   The values of **nameserver** vary depending on the region.

   CN North-Beijing1: 100.125.1.250 and 100.125.21.250

   CN North-Beijing4: 100.125.1.250 and 100.125.129.250

   CN East-Shanghai1: 100.125.1.250,100.125.64.250

   CN South-Guangzhou: 100.125.1.250 and 100.125.136.29

   CN-Hong Kong: 100.125.1.250, 100.125.3.250

   AP-Bangkok: 100.125.1.250,100.125.3.250

   LA-Santiago: 100.125.1.250

4. Press **Esc** and enter **:wq!** to save the configuration.

Configuring the Security Group

1. On the page showing the BMS details, click the **Security Groups** tab.

2. Click ⌄ to expand the security group details, showing the configured security group rules.

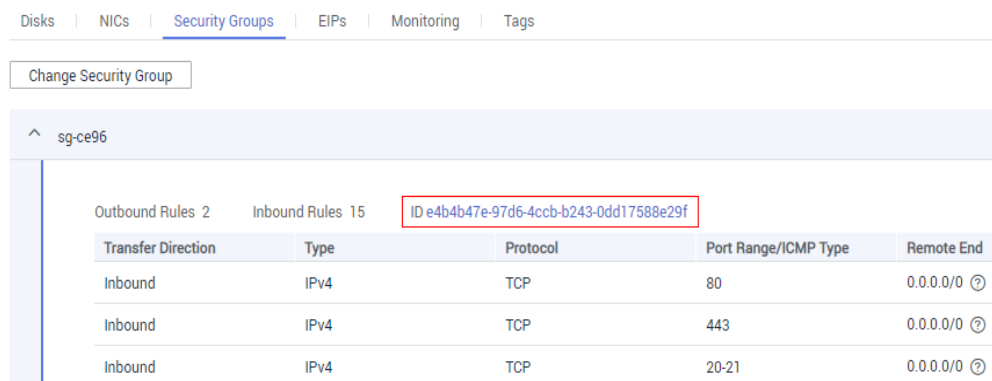3. In the upper right corner of the rule list, click the security group ID to go to the **Security Groups** page.

Figure 1-2 Security group rules

4. In the **Operation** column, click **Manage Rule**. On the **Outbound Rules** tab page, click **Add Rule** to add a rule based on **Table 1-2**.

Table 1-2 Security group rules

Direction	Protocol	Port	Destination IP address	Description
Outbound	TCP	80	100.125.0.0/16	Used to download the Agent installation package from the OBS bucket to the BMS and obtain the metadata and authentication information of the BMS.
Outbound	TCP and UDP	53	100.125.0.0/16	Used by DNS to resolve domain names, for example, resolve the OBS domain name when users are downloading the Agent installation package, and resolve the Cloud Eye endpoint when the Agent is sending monitoring data to Cloud Eye.
Outbound	TCP	443	100.125.0.0/16	Used to collect monitoring data that will be sent to Cloud Eye.

Installing the Agent

1. Run the following command to install the Agent:

CN North-Beijng1 (x86):

```
cd /usr/local && curl -k -O https://obs.cn-north-1.myhuaweicloud.com/uniagent-cn-north-1/script/uniagent_install_amd64.sh && bash uniagent_install_amd64.sh
```

CN North-Beijing1 (Kunpeng):

```
cd /usr/local && curl -k -O https://obs.cn-north-1.myhuaweicloud.com/uniagent-cn-north-1/script/uniagent_install_arm64.sh && bash uniagent_install_arm64.sh
```

CN North-Beijng4 (x86):

```
cd /usr/local && curl -k -O https://obs.cn-north-4.myhuaweicloud.com/uniagent-cn-north-4/script/
uniagent_install_amd64.sh && bash uniagent_install_amd64.sh
```

CN South-Guangzhou (x86):

```
cd /usr/local && curl -k -O https://obs.cn-south-1.myhuaweicloud.com/uniagent-cn-south-1/script/
uniagent_install_amd64.sh && bash uniagent_install_amd64.sh
```

CN South-Guangzhou (Kunpeng):

```
cd /usr/local && curl -k -O https://obs.cn-south-1.myhuaweicloud.com/uniagent-cn-south-1/script/
uniagent_install_arm64.sh && bash uniagent_install_arm64.sh
```

CN East-Shanghai1 (x86):

```
cd /usr/local && curl -k -O https://obs.cn-east-3.myhuaweicloud.com/uniagent-cn-east-3/script/
uniagent_install_amd64.sh && bash uniagent_install_amd64.sh
```

CN East-Shanghai1 (Kunpeng):

```
cd /usr/local && curl -k -O https://obs.cn-east-3.myhuaweicloud.com/uniagent-cn-east-3/script/
uniagent_install_arm64.sh && bash uniagent_install_arm64.sh
```

CN East-Shanghai2 (x86):

```
cd /usr/local && curl -k -O https://obs.cn-east-2.myhuaweicloud.com/uniagent-cn-east-2/script/
uniagent_install_amd64.sh && bash uniagent_install_amd64.sh
```

CN East-Shanghai2 (Kunpeng):

```
cd /usr/local && curl -k -O https://obs.cn-east-2.myhuaweicloud.com/uniagent-cn-east-2/script/
uniagent_install_arm64.sh && bash uniagent_install_arm64.sh
```

CN-Hong Kong

```
cd /usr/local && wget https://telescope-ap-southeast-1.obs.ap-southeast-1.myhuaweicloud.com/
scripts/agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

AP-Bangkok:

```
cd /usr/local && wget https://telescope-ap-southeast-2.obs.ap-southeast-2.myhuaweicloud.com/
scripts/agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

AP-Singapore:

```
cd /usr/local && wget https://telescope-ap-southeast-3.obs.ap-southeast-3.myhuaweicloud.com/
scripts/agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

AF-Johannesburg:

```
cd /usr/local && wget https://telescope-af-south-1.obs.af-south-1.myhuaweicloud.com/scripts/
agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

LA-Santiago:

```
cd /usr/local && wget https://telescope-la-south-2.obs.la-south-2.myhuaweicloud.com/scripts/
agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

LA-Sao Paulo1:

```
cd /usr/local && wget https://telescope-sa-brazil-1.obs.sa-brazil-1.myhuaweicloud.com/scripts/
agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

LA-Mexico City1:

```
cd /usr/local && wget http://telescope-na-mexico-1.obs.na-mexico-1.myhuaweicloud.com/scripts/
agentInstall.sh && chmod 755 agentInstall.sh && ./agentInstall.sh
```

The Agent is installed successfully if the command output similar to **Figure 1-3** is displayed.

Figure 1-3 Successful installation

2. After the installation is complete, configure the Agent as instructed in **Manually Configuring the Agent for Linux**.

1.2.2 (Optional) Managing the Agent

This section guides you to manage the Agent. You can view, start, stop, and uninstall the Agent as needed.

📖 **NOTE**

You need to view, start, stop, and uninstall the Agent as user **root**.

Checking the Agent Status

Log in to the BMS and run the following command to check the Agent status:

service telescoped status

The Agent is running properly if the system displays the following information:

"Telescope process is running well."

Starting the Agent

Run the following command to start the Agent:

/usr/local/telescope/telescoped start

Restarting the Agent

Run the following command to restart the Agent:

/usr/local/telescope/telescoped restart

Stopping the Agent

Run the following command to stop Agent:

service telescoped stop

> 📖 **NOTE**
>
> If the Telescope installation fails, you may fail to stop the Agent, and you can run the following command to stop the Agent again:
>
> **/usr/local/telescope/telescoped stop**

Uninstalling the Agent

You can manually uninstall the Agent. After the uninstallation, Cloud Eye does not collect the BMS monitoring data. If you need to use the Agent again, install it again. For details, see section **Installing the Agent**.
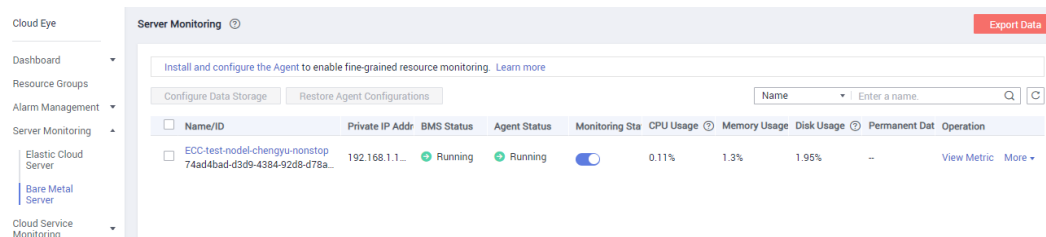
Run the following command to uninstall the Agent:

/usr/local/telescope/uninstall.sh

1.3 Monitoring Data

Log in to the management console. Under **Management & Deployment**, click **Cloud Eye**. In the navigation pane on the left, choose **Server Monitoring** > **Bare Metal Server**. In the right pane, **Name/ID**, **Status**, and **Agent Status** of the BMS are displayed.

Figure 1-4 Server monitoring

You can click **View Metric** in the **Operation** column to obtain the visualized monitoring graph of the BMS and view monitoring metrics of the BMS, such as the CPU usage, CPU load, and memory usage.

Figure 1-5 Visualized monitoring graph
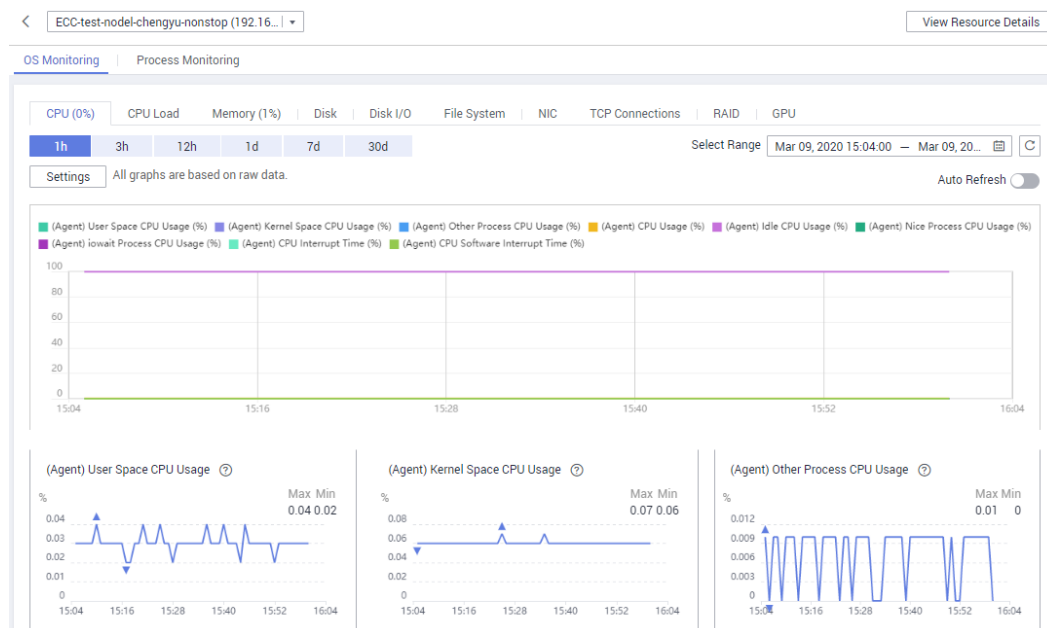
1.4 Supported Monitoring Metrics (with Agent Installed)

Description

This section describes monitoring metrics reported by BMS to Cloud Eye as well as their namespaces and dimensions. You can use the management console or APIs provided by Cloud Eye to query the metrics of the monitored objects and alarms generated for BMS.

📖 **NOTE**

After installing the Agent on a BMS, you can view its OS monitoring metrics. Monitoring data is collected at an interval of 1 minute.

Namespace

SERVICE.BMS

Metrics

Supported BMS **OS Monitoring** metrics include CPU metrics listed in **Table 1-3**, CPU load metrics listed in **Table 1-4**, memory metrics listed in **Table 1-5**, disk metrics listed in **Table 1-6**, disk I/O metrics listed in **Table 1-7**, file system metrics listed in **Table 1-8**, NIC metrics listed in **Table 1-9**, software RAID metrics listed in **Table 1-10**, and process metrics in **Table 1-11**.

📖 **NOTE**

To monitor software RAID metrics, Agent 1.0.5 or later is required.

Currently, BMSs running the Windows OS cannot be monitored.

Table 1-3 CPU metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
cpu_usage_idle	(Agent) Idle CPU Usage	Percentage of time that CPU is idle Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) id** value. Unit: percent	0-100%	BMS	1 minute
cpu_usage_other	(Agent) Other Process CPU Usage	Percentage of time that the CPU is used by other processes Formula: **Other Process CPU Usage** = 1- **Idle CPU Usage** - **Kernel Space CPU Usage** - **User Space CPU Usage** Unit: percent	0-100%	BMS	1 minute
cpu_usage_system	(Agent) Kernel Space CPU Usage	Percentage of time that the CPU is used by kernel space Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) sy** value. Unit: percent	0-100%	BMS	1 minute
cpu_usage_user	(Agent) User Space CPU Usage	Percentage of time that the CPU is used by user space Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) us** value. Unit: percent	0-100%	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
cpu_usage	(Agent) CPU Usage	CPU usage of the monitored object Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s)** value. Unit: percent	0-100%	BMS	1 minute
cpu_usage_nice	(Agent) Nice Process CPU Usage	Percentage of time that the CPU is used by the Nice process Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) ni** value. Unit: percent	0-100%	BMS	1 minute
cpu_usage_iowait	(Agent) iowait Process CPU Usage	Percentage of time during which the CPU is waiting for I/O operations to complete Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) wa** value. Unit: percent	0-100%	BMS	1 minute
cpu_usage_irq	(Agent) CPU Interrupt Time	Percentage of time that the CPU is servicing interrupts Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) hi** value. Unit: percent	0-100%	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
cpu_usage_softirq	(Agent) CPU Software Interrupt Time	Percentage of time that the CPU is servicing software interrupts Check the metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s) si** value. Unit: percent	0-100%	BMS	1 minute

Table 1-4 CPU load metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
load_average1	(Agent) 1-Minute Load Average	CPU load averaged from the last 1 minute Obtain its value by dividing the **load1/** value in **/proc/loadavg** by the number of logical CPUs. Run the **top** command to check the **load1** value.	≥ 0	BMS	1 minute
load_average5	(Agent) 5-Minute Load Average	CPU load averaged from the last 5 minutes Obtain its value by dividing the **load5/** value in **/proc/loadavg** by the number of logical CPUs. Run the **top** command to check the **load5** value in the **/proc/loadavg** file.	≥ 0	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
load_average15	(Agent) 15-Minute Load Average	CPU load averaged from the last 15 minutes Obtain its value by dividing the **load15/** value in **/proc/loadavg** by the number of logical CPUs. Run the **top** command to check the **load15** value in the **/proc/loadavg** file.	≥ 0	BMS	1 minute

Table 1-5 Memory metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mem_available	(Agent) Available Memory	Available memory size of the monitored object Obtain the **MemAvailable** value by checking the file **/proc/meminfo**. If it is not displayed in the file: **MemAvailable** = **MemFree** + **Buffers** + **Cached** Unit: GB	≥ 0 GB	BMS	1 minute
mem_usedPercent	(Agent) Memory Usage	Memory usage of the monitored object Obtain its value by checking the file **/proc/meminfo**. **Memory Usage** = (**MemTotal** - **MemAvailable**)/ **MemTotal** Unit: percent	0-100%	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mem_free	(Agent) Idle Memory	Amount of memory that is not being used Obtain its value by checking the file **/proc/meminfo**. Unit: GB	≥ 0 GB	BMS	1 minute
mem_buffers	(Agent) Buffer	Memory that is being used for buffers Obtain its value by checking the file **/proc/meminfo**. Run the **top** command to check the **KiB Mem:buffers** value. Unit: GB	≥ 0 GB	BMS	1 minute
mem_cached	(Agent) Cache	Memory that is being used for file caches Obtain its value by checking the file **/proc/meminfo**. Run the **top** command to check the **KiB Swap:cached Mem** value. Unit: GB	≥ 0 GB	BMS	1 minute

Table 1-6 Disk metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mount PointP refix_d isk_fre e	(Agent) Available Disk Space	Available disk space of the monitored object Run the **df -h** command to check the data 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 (-), dots (.), and swung dashes (~). Unit: GB	≥ 0 GB	BMS	1 minute
mount PointP refix_d isk_tot al	(Agent) Disk Storage Capacity	Disk storage capacity of the monitored object Run the **df -h** command to check the data 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 (-), dots (.), and swung dashes (~). Unit: GB	≥ 0 GB	BMS	1 minute
mount PointP refix_d isk_us ed	(Agent) Used Disk Space	Used disk space of the monitored object Run the **df -h** command to check the data 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 (-), dots (.), and swung dashes (~). Unit: GB	≥ 0 GB	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mount PointP refix_d isk_us edPerc ent	(Agent) Disk Usage	Disk usage of the monitored object. It is calculated as follows: **Disk Usage** = **Used Disk Space**/**Disk Storage Capacity**. **Disk Usage** = **Used Disk Space**/**Disk Storage Capacity** The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: percent	0-10 0%	BMS	1 minute

Table 1-7 Disk I/O metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
moun tPoint Prefix _disk_ agt_re ad_by tes_ra te	(Agent) Disks Read Rate	Volume of data read from the monitored object per second The disk read rate is calculated by checking data changes in the sixth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: byte/s	≥ 0 bytes /s	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mountPointPrefix_disk_agt_read_requests_rate	(Agent) Disks Read Requests	Number of read requests sent to the monitored object per second The disk read requests are calculated by checking data changes in the fourth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: request/s	≥ 0	BMS	1 minute
mountPointPrefix_disk_agt_write_bytes_rate	(Agent) Disks Write Rate	Volume of data written to the monitored object per second The disk write rate is calculated by checking data changes in the tenth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: byte/s	≥ 0 bytes/s	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mountPoint Prefix _disk_ agt_w rite_re quests _rate	(Agent) Disks Write Requests	Number of write requests sent to the monitored object per second The disk write requests are calculated by checking data changes in the eighth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: request/s	≥ 0	BMS	1 minute
disk_r eadTi me	(Agent) Average Read Request Time	Average amount of time that read requests have waited on the disks The average read request time is calculated by checking data changes in the seventh column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: ms/count	≥ 0 ms/ Coun t	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
disk_writeTime	(Agent) Average Write Request Time	Average amount of time that write requests have waited on the disks The average write request time is calculated by checking data changes in the eleventh column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: ms/count	≥ 0 ms/ Count	BMS	1 minute
disk_ioUtils	(Agent) Disk I/O Usage	Disk I/O usage of the monitored object Check the data changes in the thirteenth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: percent	0-100%	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
disk_queue_length	(Agent) Disk Queue Length	Average number of read or write requests to be processed for the monitored disk in the monitoring period The average disk queue length is calculated by checking data changes in the fourteenth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: count	≥ 0	BMS	1 minute
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 The average disk write size is calculated by dividing the data changes in the tenth column of the corresponding device by that of the eighth column in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: KB/op	≥ 0 KB/op	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (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 The average disk read size is calculated by dividing the data changes in the sixth column of the corresponding device by that of the fourth column in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: KB/op	≥ 0 KB/op	BMS	1 minute
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 The average disk I/O service time is calculated by dividing the data changes in the thirteenth column of the corresponding device by the sum of data changes in the fourth and eighth columns in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: ms/op	≥ 0 ms/op	BMS	1 minute

Table 1-8 File system metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (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 values are **0** (read and write) and **1** (read only). Check file system information in the fourth column in the **/proc/mounts** file.	0 and 1	BMS	1 minute
disk_inodes Total	(Agent) Disk inode Total	Total number of index nodes on the disk Run the **df -i** command to check information 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 (-), dots (.), and swung dashes (~).	≥ 0	BMS	1 minute
disk_inodes Used	(Agent) Total inode Used	Number of used index nodes on the disk Run the **df -i** command to check data 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 (-), dots (.), and swung dashes (~).	≥ 0	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
disk_inodesUsedPercent	(Agent) Percentage of Total inode Used	Percentage of used index nodes on the disk Run the **df -i** command to check data 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 (-), dots (.), and swung dashes (~). Unit: percent	0-100%	BMS	1 minute

Table 1-9 NIC metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
net_bitRecv	(Agent) Inbound Bandwidth	Number of bits received by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: bit/s	≥ 0 bit/s	BMS	1 minute
net_bitSent	(Agent) Outbound Bandwidth	Number of bits sent by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: bit/s	≥ 0 bit/s	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
net_packetRecv	(Agent) NIC Packet Receive Rate	Number of packets received by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: count/s	≥ 0 counts/s	BMS	1 minute
net_packetSent	(Agent) NIC Packet Send Rate	Number of packets sent by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: count/s	≥ 0 counts/s	BMS	1 minute
net_errin	(Agent) Receive Error Rate	Percentage of receive errors detected by this NIC per second Unit: percent	0-100 %	BMS	1 minute
net_errout	(Agent) Transmit Error Rate	Percentage of transmit errors detected by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: percent	0-100 %	BMS	1 minute
net_dropin	(Agent) Received Packet Drop Rate	Percentage of packets discarded by this NIC to the total number of packets received by the NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: percent	0-100 %	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
net_dropout	(Agent) Transmitted Packet Drop Rate	Percentage of packets transmitted by this NIC which were dropped per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: percent	0-100 %	BMS	1 minute

Table 1-10 Software RAID metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
md1_status _device:1	(Agent) Status	Software RAID status of the monitored object. Its value is **0** if the RAID is abnormal. Run the plug-in script **/usr/local/telescope/plugins/raid-monitor.sh** in a collection period. Obtain its value by checking data changes in the **/proc/mdstat** file and run **mdadm -D/dev/md0** (**md0** indicates the RAID name).	0 and 1	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
md1_active_device:2	(Agent) Active Disks	Number of active disks in software RAID of the monitored object. Its value is **-1** if the RAID is abnormal. Run the plug-in script **/usr/local/telescope/plugins/raid-monitor.sh** in a collection period. Obtain its value by checking data changes in the **/proc/mdstat** file and run **mdadm -D/dev/md0** (**md0** indicates the RAID name).	≥ 0, −1	BMS	1 minute
md1_working_device:2	(Agent) Working Disks	Number of working disks in software RAID of the monitored object. Its value is **-1** if the RAID is abnormal. Run the plug-in script **/usr/local/telescope/plugins/raid-monitor.sh** in a collection period. Obtain its value by checking data changes in the **/proc/mdstat** file and run **mdadm -D/dev/md0** (**md0** indicates the RAID name).	≥ 0, −1	BMS	1 minute
md1_failed_device:0	(Agent) Failed Disks	Number of failed disks in software RAID of the monitored object. Its value is **-1** if the RAID is abnormal. Run the plug-in script **/usr/local/telescope/plugins/raid-monitor.sh** in a collection period. Obtain its value by checking data changes in the **/proc/mdstat** file and run **mdadm -D/dev/md0** (**md0** indicates the RAID name).	≥ 0, −1	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
md1_spare_device:0	(Agent) Spare Disks	Number of spare disks in software RAID of the monitored object. Its value is **-1** if the RAID is abnormal. Run the plug-in script **/usr/local/telescope/plugins/raid-monitor.sh** in a collection period. Obtain its value by checking data changes in the **/proc/mdstat** file and run **mdadm -D/dev/md0** (**md0** indicates the RAID name).	≥ 0, –1	BMS	1 minute

Table 1-11 Process metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
proc_pHashId_cpu	CPU Usage	CPU consumed by a process. **pHashId** (process name and process ID) is the value of **md5**. Check the metric value changes in the **/proc/pid/stat** file. Unit: percent	0-100%	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
proc_pHashId_mem	Memory Usage	Memory consumed by a process. **pHashId** (process name and process ID) is the value of **md5**. **Memory Usage** = **RSS** x **PAGESIZE**/**MemTotal** • Obtain the **RSS** value by checking the second column of the file **/proc/pid/statm**. • Obtain the **PAGESIZE** value by running the **getconf PAGESIZE** command. • Obtain the **MemTotal** value by checking the file **/proc/meminfo**. Unit: percent	0-100 %	BMS	1 minute
proc_pHashId_file	Opened Files	Number of files opened by a process. **pHashId** (process name and process ID) is the value of **md5**. Run the **ls -l /proc/pid/fd** command to view the number of opened files.	≥0	BMS	1 minute
proc_running_count	(Agent) Running Processes	Number of running processes You can obtain the status 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.	≥0	BMS	1 minute
proc_idle_count	(Agent) Idle Processes	Number of idle processes You can obtain the status 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.	≥0	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
proc_zombie_count	(Agent) Zombie Processes	Number of zombie processes You can obtain the status 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.	≥0	BMS	1 minute
proc_blocked_count	(Agent) Blocked Processes	Number of blocked processes You can obtain the status 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.	≥0	BMS	1 minute
proc_sleeping_count	(Agent) Sleeping Processes	Number of sleeping processes You can obtain the status 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.	≥0	BMS	1 minute
proc_total_coun	(Agent) Total Processes	Total number of processes on the monitored object You can obtain the status 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.	≥0	BMS	1 minute

1.5 Supported Monitoring Metrics

Description

📖 **NOTE**

> After installing the Agent on a BMS, you can view its OS monitoring metrics. Monitoring data is collected at an interval of 1 minute.

Namespace

SERVICE.BMS

Metrics

Table 1-12 lists the metrics supported by BMS.

Table 1-12 Metrics

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
cpu_usage	(Agent) CPU Usage	CPU usage of the monitored object Obtain its value by checking metric value changes in the **/proc/stat** file in a collection period. Run the **top** command to check the **%Cpu(s)** value. Unit: percent	0-100 %	BMS	1 minute
load_average5	(Agent) 5-Minute Load Average	CPU load averaged from the last 5 minutes Obtain its value by dividing the **load5/** value in **/proc/loadavg** by the number of logical CPUs. Run the **top** command to check the **load5** value in the **/proc/loadavg** file.	≥ 0	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mem_ usedP ercent	(Agent) Memory Usage	Memory usage of the monitored object Obtain its value by checking the file **/proc/meminfo**. **Memory Usage** = (**MemTotal** - **MemAvailable**)/**MemTotal** Unit: percent	0-100 %	BMS	1 minute
moun tPoint Prefix _disk_ free	(Agent) Available Disk Space	Available disk space of the monitored object Run the **df -h** command to check the data 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 (-), dots (.), and swung dashes (~). Unit: GB	≥ 0 GB	BMS	1 minute
moun tPoint Prefix _disk_ usedP ercent	(Agent) Disk Usage	Disk usage of the monitored object. It is calculated as follows: **Disk Usage** = **Used Disk Space**/ **Disk Storage Capacity**. **Disk Usage** = **Used Disk Space**/**Disk Storage Capacity** The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: percent	0-100 %	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
mountPointPrefix_disk_ioUtils and volumePrefix_disk_ioUtils	(Agent) Disk I/O Usage	Disk I/O usage of the monitored object Obtain its value by checking data changes in the thirteenth column of the corresponding device in the **/proc/diskstats** file in a collection period. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). Unit: percent	0-100 %	BMS	1 minute
mountPointPrefix_disk_inodesUsedPercent	(Agent) Percentage of Total inode Used	Percentage of used index nodes on the disk Run the **df -i** command to check data 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 (-), dots (.), and swung dashes (~). Unit: percent	0-100 %	BMS	1 minute
net_bitRecv	(Agent) Inbound Bandwidth	Number of bits received by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: bit/s	≥ 0 bit/s	BMS	1 minute
net_bitSent	(Agent) Outbound Bandwidth	Number of bits sent by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: bit/s	≥ 0 bit/s	BMS	1 minute

Metric ID	Metric	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
net_packetRecv	(Agent) NIC Packet Receive Rate	Number of packets received by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: count/s	≥ 0 counts/s	BMS	1 minute
net_packetSent	(Agent) NIC Packet Send Rate	Number of packets sent by this NIC per second Check metric value changes in the **/proc/net/dev** file in a collection period. Unit: count/s	≥ 0 counts/s	BMS	1 minute
net_tcp_total	(Agent) TCP TOTAL	Total number of TCP connections of this NIC	≥0	BMS	1 minute
net_tcp_established	(Agent) TCP ESTABLISHED	Number of ESTABLISHED TCP connections of this NIC	≥0	BMS	1 minute

1.6 FAQs

1.6.1 Why Does Not the Cloud Eye Console Display Monitoring Data or Why Is There a Delay in Data Display After Agent Is Installed and Configured?

1. After the Agent is installed successfully, server monitoring data is displayed on the Cloud Eye console after two minutes. If **BMS** is not displayed on the **Monitoring Overview** page after five minutes, check whether the time of the BMS is the same as that of the client where you are using the management console.

   The time when the Agent reports data depends on the local time of the BMS. The time when the console delivers requests is related to the browser time of the client. If the two are inconsistent, no monitoring data is displayed on the Cloud Eye console.

2. Log in to the BMS and run the **service telescoped status** command to check the status of Agent. If the following information is displayed, Agent is running properly:

   Telescope process is running well.

If monitoring data is still not displayed, check the configuration as instructed in **Manually Configuring the Agent for Linux**.

1.6.2 How Do I Create an Agency for Server Monitoring of the BMS?

1. On the management console homepage, choose **Service List** > **Management & Deployment** > **Identity and Access Management**.

2. In the navigation pane on the left, choose **Agency** and then click **Create Agency** in the upper right corner.

   – **Agency Name**: Enter **bms_monitor_agency**.

   – **Agency Type**: Select **Cloud service**.

   – **Cloud Service**: This parameter is available if you select **Cloud service** for **Agency Type**. Click **Select**, select **ECS BMS** in the displayed **Select Cloud Service** dialog box, and click **OK**.

   – **Validity Period**: Select **Permanent**.

   – **Description**: This parameter is optional. You can enter **"Support BMS server monitoring"**.

   – **Permissions**: Locate the region where the BMS resides or the sub-project of the region and click **Modify** in the **Operation** column. In the displayed dialog box, enter **CES** in the **Available Policies** search box. Then select **CES (CES Administrator)** and click **OK**.

   📖 NOTE

   > If the BMS belongs to a sub-project, ensure that the sub-project has the CES Administrator permission.

3. Click **OK**.

   The operations to create an agency for server monitoring of the BMS are complete.

A Change History

Release On	Description
2018-11-30	This issue is the first official release.