# Elastic MapReduce

# EMR on CVM Operation Guide

# Product Documentation

# Contents

# EMR on CVM Operation Guide
# Planning Cluster
# Cluster Types

Last updated：2023-12-27 09:59:01

EMR supports six cluster types and their respective use cases and defines five node types. Different cluster types and their respective use cases support different node types, number of deployed nodes, and deployed services. You can select the most appropriate cluster type and use case based on your business needs when creating a cluster.

**Note**

ClickHouse and Doris cluster types are not available by default. To use them, submit a ticket for application.

## Cluster Type Description

**Hadoop cluster**

| Use Case | Description | Node Deployment |
|---|---|---|
| Default | Based on open-source Hadoop and the components that form a Hadoop ecosystem, it provides big data solutions for massive data storage, offline/real-time data analysis, streaming data computing, and machine learning. | Master node: It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode, ResourceManager, and HMaster are deployed here. The number of master nodes is 1 in non-HA mode and 2 in HA mode.Note: If Kudu is deployed, the cluster supports only the HA mode, and there are 3 master nodes. Core node: It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, core nodes cannot be scaled in once scaled out to ensure data security. Processes such as DataNode, NodeManager, and RegionServer are deployed here. The number of core nodes is ≥ 2 in non-HA mode and ≥ 3 in HA mode. Task node: It is a node for computing only and does not store any data. The computed data comes from a core node or COS. Therefore, task nodes are often elastic nodes and can be scaled in or out as needed. Processes such as NodeManager and PrestoWork are deployed here. The number of task nodes can be changed at any time to scale the cluster, with a minimum value of 0. |

| | | Common node: It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode and ≥ 3 in HA mode.<br><br>Router node: It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. Hadoop packages, including software programs and processes such as Hive, Hue, and Spark, are deployed here. The number of router nodes can be changed at any time, with a minimum value of 0. |
|---|---|---|
| ZooKeeper | It is suitable for creating a distributed, high-availability coordination service for large clusters. | Common node: Distributed coordinator components such as ZooKeeper are deployed here. The number of deployed common nodes must be odd and be at least three. Common nodes support only the HA mode. |
| HBase | It is suitable for storing massive amounts of unstructured or semi-structured data. It provides a high-reliability, high-performance, column-oriented, scalable distributed storage system that supports real-time data read/write. | Master node: It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode, ResourceManager, and HMaster are deployed here. The number of master nodes is 1 in non-HA mode and 2 in HA mode.<br><br>Core node: It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, core nodes cannot be scaled in once scaled out to ensure data security. Processes such as DataNode, NodeManager, and RegionServer are deployed here. The number of core nodes is ≥ 2 in non-HA mode and ≥ 3 in HA mode.<br><br>Task node: It is a node for computing only and does not store any data. The computed data comes from a core node or COS. Therefore, task nodes are often elastic nodes and can be scaled in or out as needed. Processes such as NodeManager are deployed here. The number of task nodes can be changed at any time to scale the cluster, with a minimum value of 0.<br><br>Common node: It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode and ≥ 3 in HA mode.<br><br>Router node: It is used to share the load of a master node or as the task submitter of the cluster. It can be |

| | | |
|---|---|---|
| | | scaled in or out at any time. The number of router nodes can be changed at any time, with a minimum value of 0. |
| Presto | It provides an open-source distributed SQL query engine for quick query and analysis of massive amounts of data. It is suitable for interactive analytical queries. | Master node: It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode and ResourceManager are deployed here. The number of master nodes is 1 in non-HA mode and 2 in HA mode. <br> Core node: It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, core nodes cannot be scaled in once scaled out to ensure data security. Processes such as DataNode and NodeManager are deployed here. The number of core nodes is ≥ 2 in non-HA mode and ≥ 3 in HA mode. <br> Task node: It is a node for computing only and does not store any data. The computed data comes from a core node or COS. Therefore, task nodes are often elastic nodes and can be scaled in or out as needed. Processes such as NodeManager and PrestoWork are deployed here. The number of task nodes can be changed at any time to scale the cluster, with a minimum value of 0. <br> Common node: It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode and ≥ 3 in HA mode. <br> Router node: It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time, with a minimum value of 0. |
| Kudu | It provides a distributed and scalable columnar storage manager and supports random reads/writes and OLAP analysis to process frequently updated data. | Master node: It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode and ResourceManager are deployed here. The number of master nodes is 1 in non-HA mode and 2 in HA mode. <br> Core node: It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, core nodes cannot be scaled in once scaled out to ensure data security. The number of core nodes is ≥ 2 in non-HA mode and ≥ 3 in HA mode. |

| | | Task node: It is a node for computing only and does not store any data. The computed data comes from a core node or COS. Therefore, task nodes are often elastic nodes and can be scaled in or out as needed. The number of task nodes can be changed at any time to scale the cluster, with a minimum value of 0. Common node: It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The number of common nodes is 0 in non-HA mode and ≥ 3 in HA mode. Router node: It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time, with a minimum value of 0. |
|---|---|---|

## Druid cluster

| Use Case | Description | Node Deployment |
|---|---|---|
| Default | It supports high-performance real-time analysis, big data queries in milliseconds, and multiple data ingestion methods. It is suitable for real-time big data query scenarios. | Master node: It is a management node that ensures the scheduling of the cluster works properly. Processes such as NameNode and ResourceManager are deployed here. The number of master nodes is 1 in non-HA mode and 2 in HA mode. Core node: It is a compute and storage node. All your data in HDFS is stored in core nodes. Therefore, core nodes cannot be scaled in once scaled out to ensure data security. Processes such as DataNode and NodeManager are deployed here. The number of core nodes is ≥ 2 in non-HA mode and ≥ 3 in HA mode. Task node: It is a node for computing only and does not store any data. The computed data comes from a core node or COS. Therefore, task nodes are often elastic nodes and can be scaled in or out as needed. Processes such as NodeManager are deployed here. The number of task nodes can be changed at any time to scale the cluster, with a minimum value of 0. Common node: It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper and JournalNode are deployed here. The |

| | | number of common nodes is 0 in non-HA mode and ≥ 3 in HA mode.<br>Router node: It is used to share the load of a master node or as the task submitter of the cluster. It can be scaled in or out at any time. The number of router nodes can be changed at any time, with a minimum value of 0. |
| --- | --- | --- |

## ClickHouse cluster

| Use Case | Description | Node Deployment |
| --- | --- | --- |
| Default | It provides a column-oriented database management system. It is suitable for data warehouse analysis scenarios such as real-time wide table analysis, real-time BI report analysis, and user behavior analysis. | Core node: It is a compute and storage node. ClickHouseServer is deployed here.<br>Common node: It provides data sharing and syncing and HA fault tolerance services for the master nodes in an HA cluster. Distributed coordinator components such as ZooKeeper are deployed here. The number of common nodes is 0 in non-HA mode and ≥ 3 in HA mode. |

## Doris cluster

| Use Case | Description | Node Deployment |
| --- | --- | --- |
| Default | It is an MPP analytical database product that supports sub-second queries on PB-scale, structured data. It is compatible with MySQL protocol and uses the standard SQL syntax. It is suitable for historical report analysis, real-time data analysis, interactive data analysis, etc. | Master node: It is a frontend module that provides the WebUI feature. Processes such as FE Follower and Broker are deployed here. The number of master nodes is ≥ 1 in non-HA mode and ≥ 3 in HA mode.<br>Core node: It is a backend module that provides the data storage feature. Processes such as BE and Broker are deployed here. The number of core nodes is ≥ 3.<br>Router node: It is a frontend module that helps achieve high read/write availability. Processes such as FE Observer and Broker are deployed here. Router nodes can be scaled out but not scaled in. |

## Kafka cluster

| Use Case | Description | Node Deployment |
|---|---|---|
| Default | It is a distributed, partitioned, multi-replica, and multi-subscriber message processing system based on ZooKeeper coordination. It is suitable for asynchronous processing, message communication, and streaming data receiving and distribution. | Core node: It is a backend module for data storage. Processes such as BE and Broker are deployed here. The number of core nodes is ≥ 1 in non-HA mode or ≥ 2 in HA mode. Common node: It provides data sharing and syncing and HA fault tolerance services for the core nodes in an HA cluster. The number of common nodes is 0 in non-HA mode or ≥ 3 in HA mode. |

## StarRocks cluster

| Use Case | Description | Node Deployment |
|---|---|---|
| Default | StarRocks adopts full vectorization technology. It supports extremely fast and unified OLAP databases. It is suitable for many data analysis scenarios, such as multidimensional, real-time, and high-concurrency analysis. | Master node: It is a frontend module that provides the WebUI feature. Processes such as FE Follower and Broker are deployed here. The number of master nodes is ≥ 1 in non-HA mode and ≥ 3 in HA mode. Core node: It is a backend module that provides the data storage feature. Processes such as BE and Broker are deployed here. The number of core nodes is ≥ 3. Task node: It is a node for computing only and does not store any data. The computed data comes from a core node or COS. Therefore, task nodes are often elastic nodes and can be scaled in or out as needed. Processes such as Compute Node are deployed here. The number of task nodes can be changed at any time to scale the cluster, with a minimum value of 0. Router node: It is a frontend module that helps achieve high read/write availability. Processes such as FE Observer and Broker are deployed here. Router nodes can be scaled out but not scaled in. |

# Configuring Cluster Administrative rights CAM Overview

Last updated：2023-12-27 09:59:20

## CAM Overview

When using Tencent Cloud EMR, different departments and roles need different permissions in order to avoid security risks such as leakages and maloperations. To this end, you can assign different permissions to different users through sub-accounts. By default, a sub-account does not have the permission to use EMR or related resources. Therefore, you need to create a policy to grant the required permission to the sub-account first.

Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access permissions, resources, and use permissions of your Tencent Cloud account. Using CAM, you can create, manage, and terminate users (groups), and control the Tencent Cloud resources that can be used by the specified user through identity and policy management.

When using EMR, you can associate a policy with a user or user group to allow or forbid them to use specified resources to complete specified tasks. For more information on CAM policies, see Element Reference. For more information on how to use CAM policies, see Policy.

## CAM Use Cases

### 1. Resource management permissions need to be granted to sub-users

You can create users or roles in CAM and assign them separate security credentials (console login passwords, TencentCloud API keys, etc.) or request temporary credentials for them to access Tencent Cloud resources. You can also manage permissions to control the operations users and roles can perform and the resources they can access.

### 2. Users with external Tencent Cloud roles can access Tencent Cloud resources

You can use your existing authentication system through CAM to grant your employees and services the access permissions for Tencent Cloud services and resources.

### 3. Strengthen your security with an additional layer of protection

Currently, three authentication methods are supported: WeChat QR code, hardware/virtual MFA, and mobile verification code.

# Supported CAM Granularities

EMR supports **resource-level authentication** and **API-level authentication**. For API-level authentication, API input parameters include the cluster string ID, and the TencentCloud API directly forwards the authentication request to CAM for authentication. Resource-level authentication involves fine-granularity authentication on the EMR backend. EMR supports three authorization methods: resource-level authorization, API-level authorization, and authorization by tag.

**Resource-level and API-level authorization**: You can use policy syntax to grant sub-accounts permissions to manage individual resources. For more information, see Authentication Granularity Scheme.

**Authorization by tag**: You can tag resources and grant sub-accounts permissions to manage resources with particular tags. For more information, see Authentication Granularity Scheme.

# Role Authorization

Last updated：2023-12-27 09:59:37

When using the EMR service, you need to authorize the default system role `EMR_QCSRole` for the service account. Only after the role is authorized successfully can EMR call relevant services such as CVM and COS to create clusters and store logs.

**Caution**

When you activate the EMR service for the first time, you must use the root account to complete role authorization; otherwise, neither sub-accounts nor the root account can use EMR.

## Role authorization process

1. When you create a cluster or a scheduled plan, if the `EMR_QCSRole` role fails to be authorized for the service account and the following prompt is displayed, you need to click **Go to CAM** to complete role authorization.
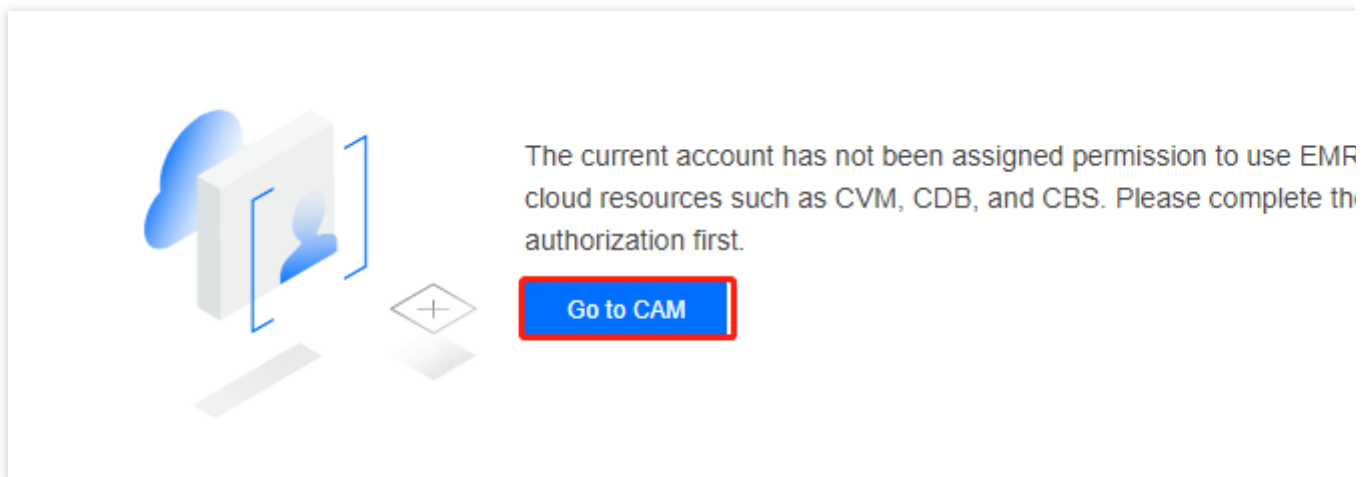


2. Click **Grant** to authorize the default role `EMR_QCSRole` for the EMR service account.
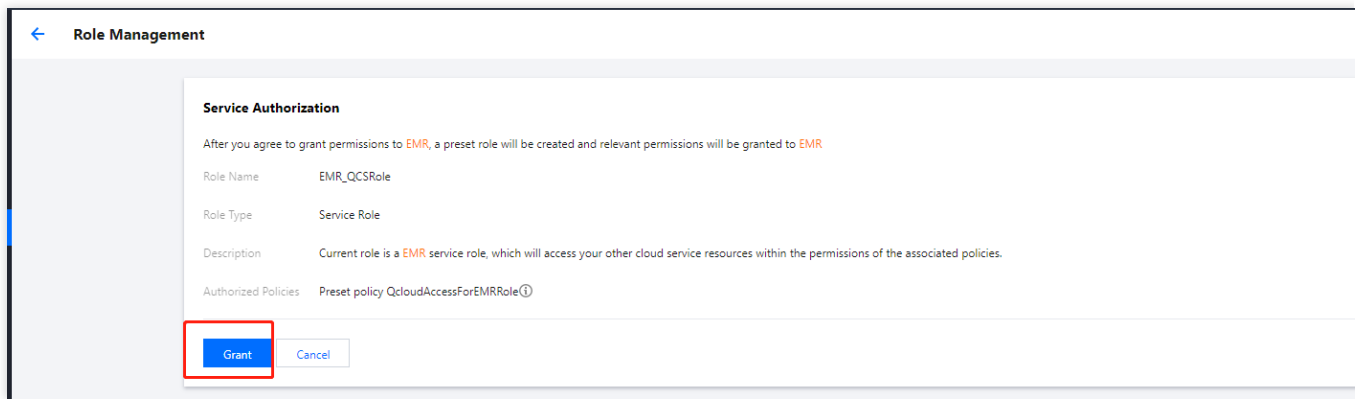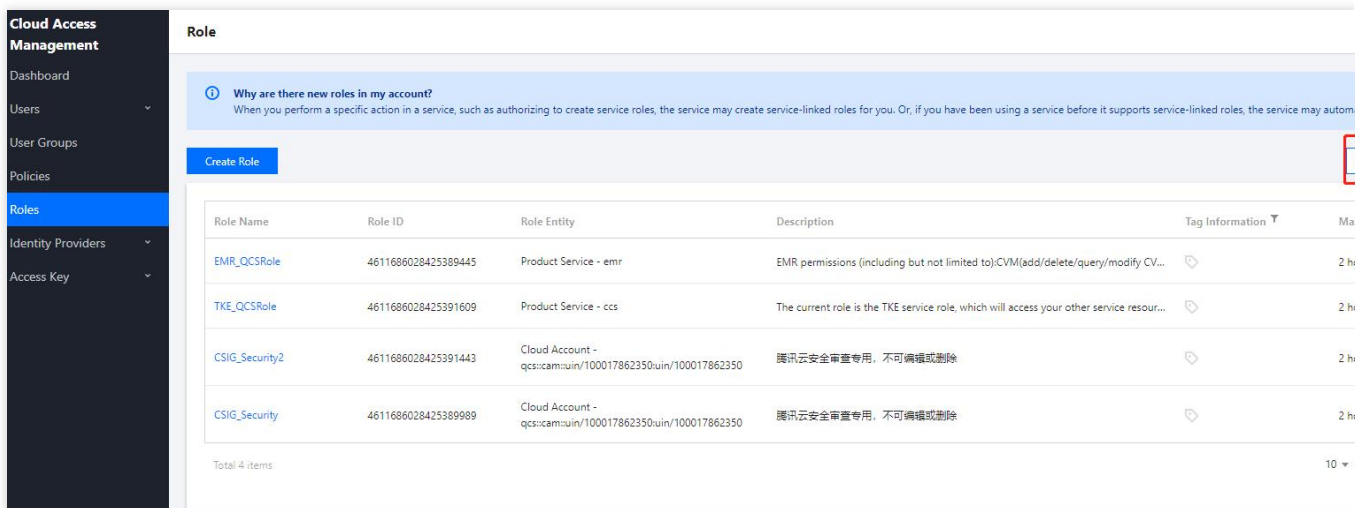
3. After the authorization, you need to refresh the EMR console or the purchase page. For more information on the policies related to `EMR_QCSRole` , please log in to the CAM console. For more information on the permissions included in `EMR_QCSRole` , please see Collaborator/Sub-account Permissions.

# Notes on rule authorization for container-based EMR

1. Before creating a container-based EMR cluster, you need to check whether the `CVM_QCSRole` role exists.

2. If the `CVM_QCSRole` role does not exist, you need to create it in advance. Log in to the CAM console, create a role, and select **Tencent Cloud Product Service**.



3. Select **Cloud Virtual Machine (cvm)** and enter the role name.

| | | | |
|---|---|---|---|
| ☐ cloudWaf (waf) | ☐ Cloud Monitor (monitor) | ☐ Cloud Media Engine (cme) | ☐ CODING DevOps (coding) |
| ☐ Cloud Storage Gateway (csg) | ☐ Cloud Training Platform (ctp) | ☐ TencentDB for CTSDB (ctsdb) | ☑ Cloud Virtual Machine (cvm) |
| ☐ Tencent Cloud Developer-TDP (devops) | ☐ DI (di) | ☐ Data Lake Compute (dlc) | ☐ Data Security Center (dsgc) |
| ☐ EventBridge (eb) | ☐ Elasticsearch MapReduce (emr) | ☐ faceid (faceid) | ☐ Game Sever Elastic-scaling (gse) |
| ☐ IDaaS (idaas) | ☐ IotHub (iothub) | ☐ IoT Suite (iotsuite) | ☐ Internet of Things Video (iotvideo) |
| ☐ Developer Laboratory (labs) | ☐ Cloud Streaming Services (live) | ☐ CDB for MariaDB (TDSQL) (mariadb) | ☐ StreamLive (mdl) |
| ☐ StreamPackage (mdp) | ☐ Message Center (message) | ☐ Mobile Game Online Battle Engine (mgobe) | ☐ Cloud MongoDB (mongodb) |
| ☐ Migration Service Platform (msp) | ☐ Media Transcoding Service (mts) | ☐ Network Assets Risk Monitor System (narms) | ☐ Publicly Accessible Instance-PAI (pai) |
| ☐ Stream Compute Service (scs) | ☐ Serverless Framework (sls) | ☐ Security Situation Awareness (ssa) | ☐ Secrets Manager (ssm) |
| ☐ Tencent Cloud Display (tcd) | ☐ Tencent Cloud Mesh (tcm) | ☐ Tencent Container Registry (tcr) | ☐ Tencent Container Security Service (tcss) |
| ☐ 4444444 (cams) | ☐ TI (ti) | ☐ TI Accelerator (tia) | ☐ Tencent Cloud Infrastructure as Code (tic) |
| ☐ TI Self-Learning (tis) | ☐ Tencent Interactive Whiteboard (tiw) | ☐ Tencent Cloud Service Engine (tse) | ☐ Tencent Service Framework (tsf) |
| ☐ Cloud Shield - Data Data Access Security Broker (dasb) | ☐ Video Moderation System (vm) | ☐ VOD (vod) | ☐ Vulnerability Scan Service (vss) |
| ☐ WeMall (wemall) | ☐ workorder (workorder) | ☐ YouMall (youmall) | ☐ Cloud Operations Console (zhiyun) |

s to choose

**Cloud Virtual Machine**
Allow Cloud Virtual Machine to access your tencent cloud resources

---

← **Create Custom Role**

✔ **Enter Role Entity Info** ＞ ② **Configure Role Policy** ＞ ③ Set Role Tag ＞ ④ Review

**Select Policies (17 Total)**

KMS ⊗ 🔍

| Policy Name | Policy type ⊤ |
|---|---|
| ☐ QcloudAccessForCARole<br>CloudAudit permissions (including but not limited to): COS (COSDataWriteOnly PostObject/PutObject*/Uplo... | Preset Policy |
| ☐ QcloudAccessForMariaDBRole<br>MariaDB permissions (including but not limited to): KMS(add/delete/query KMS instance，generate datake... | Preset Policy |
| ☐ QcloudAccessForSSMRole<br>SSM permissions (including but not limited to): KMS(add/query KMS CMK, decrypt, encrypt, create data key). | Preset Policy |
| ☐ QcloudAccessForTIONERoleInCodeRepository<br>TI-ONE permissions (including but not limited to): KMS(create key, encrypt, decrypt, gernerate data key, list ... | Preset Policy |
| ☐ QcloudKMSAccessForCASBRole | Preset Policy |

Support for holding shift key down for multiple selection.

**2 selected**

| Policy Name |
|---|
| QcloudCOSDataReadOnly<br>Grant READ-only access on COS (exclude LIST) |
| QcloudKMSAccessForCVMRole<br>Cross-service access of Cloud Virtual Machine (CVM) to Key Management Service (KMS) |

↔

Back    Next

---

# Collaborator/Sub-account Permissions

Last updated：2023-12-27 09:59:59

Elastic MapReduce (EMR) will need to access or operate other cloud products. To ensure that sub-users or collaborators can use and operate EMR normally, this document describe how to grant sub-users or collaborators related permissions.

## Permission Policy Overview

| Policy | Description | Required | Notes |
|---|---|---|---|
| QcloudCamSubaccountsAuthorizeRoleFullAccess | Permission required for CAM sub-users to obtain permissions granted by service roles | No | For more information, see Authorizing EMR to access other services. |
| QcloudCamRoleFullAccess | Full access to CAM roles | No | Permission to custom service roles to control access to data across services. For more information, see Custom Service Roles. |
| QcloudEMRFullAccess | Full access to EMR | No | Full permission to use all EMR features. For more information, see Purchasing and managing EMR clusters. |
| QcloudEMRReadOnlyAccess | Read-only access to EMR | No | Permission to view EMR features |
| QcloudEMRPurchaseAccess | EMR finance permission | No | For more information, see Purchasing and managing EMR clusters. This permission is not required if you don't |

| | | | need to purchase EMR clusters or adjust their configurations. |
| --- | --- | --- | --- |

**Caution**

The `QcloudEMRPurchaseAccess` preset policy allows you to manage all users' permission to purchase EMR instances. It grants users the finance permissions of CVM, TencentDB, and EMR at the same time. To restrict users from purchasing CVM or TencentDB instances, do not grant the permission to place orders for the corresponding product.

# Use Cases

## Authorizing EMR to access other cloud services

Tencent Cloud root accounts and sub-users and collaborators with the `QcloudCamSubaccountsAuthorizeRoleFullAccess` permission can access other cloud services after being authorized.

To use EMR to access CVM, CBS, TencentDB, and other services, you need to assign the `EMR_QCSRole` service role and grant the `QcloudAccessForEMRRole` permission (for EMR to read CVM, CBS, TencentDB, COS, and other services) to the first EMR instance you purchase.

To use EMR to access the data stored in COS, you need to assign the `EMR_QCSRole` service role and grant the `QcloudAccessForEMRRoleInApplicationDataAccess` permission (for EMR big data applications to access other data services, such as COS) to EMR.

The root account can grant the `QcloudCamSubaccountsAuthorizeRoleFullAccess` permission to sub-users or collaborators via the following steps:

1. Log in to the CAM console, click **Users** > **User List**, find the target sub-user or collaborator, and click **Authorize**.

2. Search for and select the `QcloudCamSubaccountsAuthorizeRoleFullAccess` policy, and then click **Confirm**.

You can associate the `QcloudAccessForEMRRoleInApplicationDataAccess` and `QcloudAccessForEMRRole` policies with the root account, sub-user, or collaborator. The process is the same as step 2.

## Purchasing and managing EMR clusters

To create a cluster, add a component, or scale out a cluster, a sub-user or collaborator must be associated with the `QcloudEMRFullAccess` and the custom TencentDB purchase policy. In cases not involving resource purchase, such as service configuration management, only the `QcloudEMRFullAccess` policy is required.

| Policy Type | Policy Name | Description |
|---|---|---|
| Preset EMR policy | QcloudEMRFullAccess | Full access to EMR |
| Preset EMR policy | QcloudEMRReadOnlyAccess | Read-only access to EMR |
| Preset EMR policy | QcloudEMRPurchaseAccess | EMR finance permission |

The root account can grant the above permissions to a sub-user or collaborator via the following steps:

1. Log in to the CAM console, click **Users** > **User List**, find the target sub-user or collaborator, and click **Authorize**.



2. Search for and select each policy listed in the above table in the **Associate Policy** dialog box, and then click

**Confirm**. The `QcloudEMRFullAccess` policy is used as an example in the following figure:

3. The process of associating the EMR finance policy `QcloudEMRPurchaseAccess` is the same as step 2.

## Custom Service Roles

Tencent Cloud root accounts and collaborators and sub-users with the `QcloudCamRoleFullAccess` permission can precisely control COS bucket permissions and other cloud resource permissions. For more information see Custom Service Roles.

A root account can grant the `QcloudCamRoleFullAccess` permission to a sub-user or collaborator via the following steps:

1. Log in to the CAM console, click **Users** > **User List**, find the target sub-user or collaborator, and click **Authorize**.

2. Search for and select the `QcloudCamRoleFullAccess` policy, and then click **Confirm**.

**Associate Policy**

Select Policies (2 Total)

1 selected

| qcloudcamr | | |
|---|---|---|

| Policy Name | Policy Type ▼ |
|---|---|
| ☐ QcloudCamReadOnlyAccess<br>Read-only access to Cloud Access Managem... | Preset Policy |
| ☑ QcloudCamRoleFullAccess<br>Full read-write access to Role | Preset Policy |

| Policy Name | Polic |
|---|---|
| QcloudCamRoleFullAccess<br>Full read-write access to Role | Prese |

↔

Press Shift to select multiple items

**Confirm**    Cancel

# CAM-Enabled EMR API Authorization Granularity Details

Last updated：2023-12-27 10:00:46

## List of APIs supporting resource-level authorization

EMR supports resource-level authorization. You can grant a specified sub-account the API permission of a specified resource.

**Caution**

A permission error may occur when new APIs are subsequently added. In case of such permission error, you can add the missing API permission in the policy based on the error message.

APIs supporting resource-level authorization include:

| API | Description | Resource Type/Six-Segment Example Resource |
|---|---|---|
| DescribeKeyTabFile | Exports keytab file (user management) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeShowUserManagerTab | Specifies whether to show the user management tab (user management) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeResourceSchedule | Gets data from the YARN resource scheduling page | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeCamUserList | Queries CAM user list | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeClusterServiceInfo | Queries service information | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeUserManagerUserList | Queries user list (user management) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyUserManagerPwd | Changes user | emr- |

| | password (user management) | instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
|---|---|---|
| ModifyResourcePools | Refreshes dynamic resource pool | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DeleteUserManagerUserList | Deletes user (user management) | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyResourceScheduler | Modifies the YARN resource scheduler (the change will take effect after you click **Apply**) | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyResourceScheduleSwitch | After this switch is turned on, the configuration file of resource scheduling needs to be synced to the resource scheduler first before the resource scheduler page can be viewed and operations can be performed on the page | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| AddUserManagerUserList | Adds user (user management) | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyUserManagerInfo | Modifies user information (user management) | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyResourceScheduleConfig | Modifies the resource configuration of YARN resource scheduling | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyResourceScheduleConfigForRollback | Cancels saving the resource configuration of YARN resource scheduling | emr-instance|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| InquirePriceRefundEmr | Queries the refund | emr- |

| | amount of terminated node | instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
|---|---|---|
| ModifySecurityGroup | Modifies cluster security group | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstanceRenewNodes | Queries renewed nodes of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| InquirePriceRenewEmr | Queries cluster price for renewal | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstancesList | Queries the list of EMR cluster instances | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyFlowStatus | Changes workflow status | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyMasterIp | Updates EMR cluster IP | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| CheckFlowCanBeCancelled | Checks whether the workflow can be canceled | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeModifySpec | Queries target specification | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| AddShellScriptTask | Generates cluster script task | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeShellScriptTaskList | Queries the list of cluster script tasks | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeShellScriptNodes | Queries the list of nodes in a single cluster script task | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeNodeList | Views node | emr- |

| | information | instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
|---|---|---|
| DescribeNodeList | Views node information | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeShellScriptNodeDetails | Queries the detailed execution result of cluster script task on a single node | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DeleteShellScriptList | Deletes the records of cluster script tasks | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeMasterIp | Gets EMR cluster instance IP | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeEKSInstances | Queries the API information of TKE cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| AddServiceConfFile | Adds custom configuration file | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DeleteServiceConfFile | Deletes custom configuration file | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyBootScript | Modifies bootstrap script | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstanceAlias | Gets alias | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeBootScript | Gets bootstrap script | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeSubJobFlowStatus | Describes EMR subtask workflow | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |

| ListConfLogs | Gets configuration delivery logs | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
|---|---|---|
| GenerateScaleoutGoodsDetail | Generates scale-out order | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| GenerateRenewGoodsDetail | Generates renewal order | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| GenerateModifyGoodsDetail | Generates configuration adjustment order | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyAutoScaleGlobalConf | Updates the global configuration of auto scaling | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeFlowStatusDetail | Queries EMR task execution details | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeFileIps | Queries the list of IPs of the specified configuration file | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeExportConfsList | Queries the list of exportable configuration files | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeFlowStatus | Queries the EMR instance workflow status | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeFlowNum | Queries the number of workflows in EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeClusterNodes | Queries the hardware node information of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ClearMetadata | Clears the metadata information of EMR | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |

| | | |
|---|---|---|
| | cluster and terminates cluster | |
| DescribeAutoScaleGlobalConf | Gets the global configuration of auto scaling | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeAutoScaleSpecs | Gets auto scaling specification | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyAutoScaleSpecs | Modifies auto scaling specification | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| AddMetricScaleStrategy | Adds metric load-based scaling rule | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DeleteAutoScaleSpec | Deletes auto scaling specification | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeAutoScaleRecords | Gets auto scaling history | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DeleteAutoScaleStrategy | Deletes auto scaling rule | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyStrategyPriority | Changes rule priority | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeAutoScaleMetaRange | Gets auto scaling metadata | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeAutoScaleStrategies | Gets auto scaling rules | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyAutoScaleStrategy | Modifies auto scaling rule | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| AddAutoScaleSpec | Adds auto scaling | emr- |

| | specification | instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
|---|---|---|
| TerminateAutoScaleNodes | Terminates all automatically scaled nodes | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeCbsEncrypt | Describes CBS encryption | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyConfigGroup | Modifies configuration group | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DeleteConfigGroup | Deletes configuration group | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| AddConfigGroup | Adds configuration group | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeConfigGroup | Describes configuration group | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| SynchronizeGroupConfCheck | Syncs configuration check | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| UnbindInstanceAndNodesTags | Unbinds cluster from tags | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeNodeResourceConfigFast | Quickly gets node specification configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| InstallCos | Enables COS upon installation | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeScaleoutableService | Describes services that can be scaled out | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeNodeResourceConfig | Gets node | emr- |

| | specification configuration | instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
|---|---|---|
| DeleteNodeResourceConfig | Deletes node specification configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| AddNodeResourceConfig | Adds node specification configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| SetNodeResourceConfigDefault | Sets the default attributes of node specification configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeHbaseTableMetricDataOverview | Gets HBase table-level monitoring data overview | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstanceNode | Pulls node resources from the Tag console | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| UpdateWebproxyPassword | Updates the password of proxy component | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| TerminateTasks | Terminates task nodes | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| TerminateNodes | Terminates nodes | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| TerminateInstance | Terminates EMR instance | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| StopService | Stops component service | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| StopMonitor | Stops component monitoring | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |

| StartService | Starts component service | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
|---|---|---|
| StartMonitor | Starts component monitoring | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ScaleOutRouter | Adds router nodes | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeDestroyInfo | Queries the termination information of EMR cluster or node | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ScaleOutInstance | Scales out instance | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| RollBackConf | Rolls back configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| RestartService | Restarts component service | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyServiceParams | Modifies service component parameters | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyResource | Adjusts instance configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| AssignInstancesProject | Assigns EMR cluster to specified project | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyOptionalSpecStatus | Changes the status of optional specification | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyOptionalSpec | Modifies optional specification | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |

| DescribeSelectedOptionalSpec | Queries selected optional specification | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
|---|---|---|
| ModifyInstanceBasic | Renames cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| InstallSoftware | Installs component | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| InquiryPriceAddRouter | Queries the price of router adding | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| InquiryPriceRenewInstance | Queries the price for renewal | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| InquiryPriceUpdateInstance | Queries the price for configuration adjustment | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| InquiryPriceScaleOutInstance | Queries the price for scale-out | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeClusterNodes | Queries the hardware node information of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceNodeInfos | Queries the service process information of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceGroups | Queries the service group information of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceConfs | Queries all configuration information of the EMR cluster service | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeScaleoutGoodsDetail | Queries scale-out order | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |

| | | instance/$emrInstanceId |
|---|---|---|
| DescribeRouterGoodsDetail | Queries router node order | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeRenewGoodsDetail | Queries renewal order | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeOptionalSpec | Queries the optional specifications of EMR instance node | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeModifyGoodsDetail | Queries configuration adjustment order | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeMetricsDimension | Queries the dimension values of monitoring at different levels | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeMetricMeta | Queries the monitoring metadata of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstanceOplog | Queries the operation log of EMR instance | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeExecCustomScript | Queries the information of custom script | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstanceAlerts | Queries the alarm information of EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstances | Queries the information of EMR instances | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeInstallSoftwareInfo | Queries the information of components installed and uninstalled for EMR cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |

| DescribeExportConfs | Queries export configuration | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| --- | --- | --- |
| CheckDiskInfo | Checks whether the disk metadata in the console is updated | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| SyncDiskInfo | Updates disk metadata in console | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceConfsNew | Gets the configuration information of component (configuration management page) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeConfFileList | Gets the list of configuration files (configuration management page) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceConfCategories | Gets the configuration type of component (configuration management page) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceConfDiff | Compares configurations (configuration management page) | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeConfigGroupList | Queries the configuration group information of node type | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeHbaseStatus | Displays the information of `DescribeHbaseStatus` | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyHbaseRit | Fixes HBase RIT issue | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| TerminateSparkApp | Terminates Spark | emr- |

| | job | instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
|---|---|---|
| DescribeAccessKey | Gets the `AccessKey` of cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| DescribeAttachableDisks | Queries the cloud disks that can be mounted to the node | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| InquirePriceRenewDisks | Queries the price for cloud disk renewal | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| AttachDisks | Mounts cloud disk | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| DescribeNodeDataDisks | Queries the data disk information of node | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| InquirePriceResizeDisks | Queries the price for cloud disk expansion | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| ResizeDataDisks | Expands cloud disk | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| ModifyAutoRenewFlag | Automatic Renewal | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| ScaleOutCluster | Adds cluster nodes | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| StartStopServiceOrMonitor | Starts or stops monitoring or service | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| DescribeClusterClients | Views the client information on the page | emr-instance\|qcs::emr:${region}:uin/${uin}:instance/$emrInstanceId |
| DescribeDelayedServiceConfig | Gets expired | emr- |

| | | |
|---|---|---|
| | configuration (configuration management page) | instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeFailedServiceConfig | Gets failed configuration (configuration management page) | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeServiceConfDimensionDiff | Compares the configurations at different levels (configuration management page) | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyEmrManagerAgent | Updates EMR Manager Agent | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyServiceConfDiff | Overwrites different configuration (configuration management) | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyYarnLabelState | Delivers an instruction to add, delete, or bind a label | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeYarnScheduleHistory | Views the YARN resource scheduling history | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeYarnDeployMessage | Gets the prompt message | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| ModifyYarnDeploy | Applies deployment | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeParentLabels | Gets the list of labels of the parent queue | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |
| DescribeYarnLastestLabels | Gets the latest label information | emr-instance\|qcs::emr:${region}:uin/${uin}:e instance/$emrInstanceId |

| ModifyYarnLabels | Syncs YARN node labels | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
|---|---|---|
| ModifyYarnQueue | Modifies the resource pool in resource scheduling | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyOldLabelConfig | Cancels saving the edited content in YARN label management | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeNodeLabelWebUrl | Gets the web URL of YARN node labels | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeNodeManagerHosts | Gets the list of NodeManager IPs | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyGlobalScaleConf | Modifies the scaling configuration of cluster, including whether to enable scaling and the scaling type | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeGlobalScaleConf | Gets the scaling configuration of cluster, including whether to enable scaling and the scaling type | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| ModifyManagedScaleGlobalConf | Updates the global configuration of managed scaling | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeManagedScaleGlobalConf | Gets the global configuration of managed scaling | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeServiceComponentInfos | Describes the role information of container cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DescribeRssClusterList | Gets the RSS clusters in the same EKS | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |

| | cluster as the current Spark cluster | instance/$emrInstanceId |
|---|---|---|
| CreateSparkLinkRss | Associates RSS cluster with Spark cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |
| DeleteSparkLinkRss | Disassociates RSS cluster from Spark cluster | emr-instance\|qcs::emr:${region}:uin/${uin}:emr-instance/$emrInstanceId |

# List of APIs supporting API-level authorization

| API | Description |
|---|---|
| RunJobFlow | Creates and runs job |
| DescribeJobFlow | Queries running jobs |
| DescribeK8sResourcePrice | Queries the price of K8s resource |
| DescribePodSpecs | Describes Pod specification |
| GenerateCreateGoodsDetail | Generates cluster creation order |
| DescribeLogSearchFileNames | Gets the list of log search files |
| DescribeInstanceServiceRoleNames | Gets service role names |
| DescribeCompareMetricsList | Gets the list of comparison metrics |
| DescribeHeatMapMetricList | Returns the list of metrics in the aggregated cluster server dimension |
| DescribeHBaseRegionList | Gets the list of HBase regions |
| DescribeClusterOverview | Queries cluster overview |
| DescribeEMRNodeOverview | Queries the deployment status of various processes on node |
| DescribeNodeHardwareInfo | Queries the basic configuration of the host |
| DescribeTopNMeta | Gets the top N metadata information items |

| DescribeMetricDataAutoGranularity | Gets monitoring data and automatically sets time granularity |
|---|---|
| DescribeLogSearchTabs | Gets log search tabs |
| DescribeEMRHostOverview | Queries the process status on the host details page |
| DescribeLogSearchRecords | Gets log search content |
| DescribeTopNByProcess | Gets the top N processes |
| DescribeDiskInfo | Returns specific disk information |
| DescribeTopNByHost | Queries the top N hosts on the overview page |
| DescribeHeatMapDistribution | Returns the heat map data of cluster |
| DescribeInstanceServiceRoleTable | Gets the data of service role table |
| DescribeNodeAlias | Gets the alias of EMR node |
| DescribeInstanceNodes | Gets the host information of cluster |
| DescribeKeyPairs | Queries the information of key pairs |
| DescribeHbaseTableMetricData | Queries HBase table-level monitoring data |
| DescribeKeyPairs | Queries the information of key pairs |
| DescribeEmrMetaDB | Gets the unified metadatabase information of Hive |
| ModifyEmrRole | Updates the information of EMR role |
| DescribeDisasterRecoverGroup | Gets the information of spread placement group |
| DescribeTags | Pulls all tags of cluster |
| InquiryPriceCreateInstance | Queries the price of created instance |
| DescribleAccountBalance | Queries account balance |
| DescribeSecurityGroup | Gets the information of EMR security group |
| DescribeCvmSpec | Queries CVM instance specification |
| DescribeCdbPrice | Queries the price of TencentDB instance |
| CreateInstance | Creates EMR instance |
| GetMetricDataForMcController | Gets the monitoring information on the details page in the console |

| DescribeVpcList | Queries the list of VPCs |
| --- | --- |
| DescribeSceneProductInfo | Gets the cluster use case, type, product, and component information from the purchase page |
| DescribeRegionAndZoneSaleInfo | Gets the region and AZ information from the console and the purchase page |
| DescribeCgwProjects | Gets the list of CGW projects |
| DescribeServiceUpgradeVersion | View the service minor version that can be upgraded |
| ModifyServiceVersion | Upgrades service minor version |
| DescribeServiceRoleInstanceConstraints | Gets the constraint information of role instances removed from the service |
| CheckSupportServiceRoleInstance | Checks whether a role instance can be added to the service |
| DescribeServiceRoleNames | Gets the drop-down list of role names (or role types) of the service |
| DescribeServiceRoleInstanceDeployableNodes | Queries the list of nodes where the service role can be deployed |
| DescribeServiceDeletableRoleInstances | Queries the list of role instances that can be deleted from the service |
| DeleteServiceRoleInstance | Removes role instance |
| AddServiceRoleInstance | Adds role instance |
| ModifyResourcesTags | Forcibly modifies tag |
| CreateCluster | Creates cluster |
| RepairDisk | Repairs the business after the customer's corrupted disk is replaced at the CVM side |
| CreateOutsideClient | Provides component packages and deployment scripts for servers outside the EMR cluster to support their access to EMR and job submission |
| ModifyCvmReplaceFlag | Modifies the EMR cluster to support abnormal node replacement |

For more information on resource-level and API-level authorization, see Authorization Granularity Scheme.

# Authentication Granularity Scheme

Last updated：2023-12-27 10:01:09

## Six-Segment Resource Description

qcs: Describes the abbreviation of `qcloud service` , indicating that a resource is a Tencent Cloud resource. This field is required.

project_id: Describes the project information, which is used to enable compatibility with legacy CAM logic and can be left empty.

service_type: Describes the product abbreviation, such as `CVM` .

region: Describes the region information, such as `bj` (see the CAM documentation).

account: Describes the root account of the resource owner, such as `uin/164256472` .

resource: Describes the detailed resource information of each product, such as `instance/instance_id1` or `instance/*` .

## Tag Authentication Operation Guide

This section describes how to authenticate EMR tags. Before using tags to manage the authentication of EMR resources, you need to plan tags based on your departments or organizations; for example, you can set usernames, clusters, and tags for departments A, B, and C respectively. For more information on tags, see Tag.
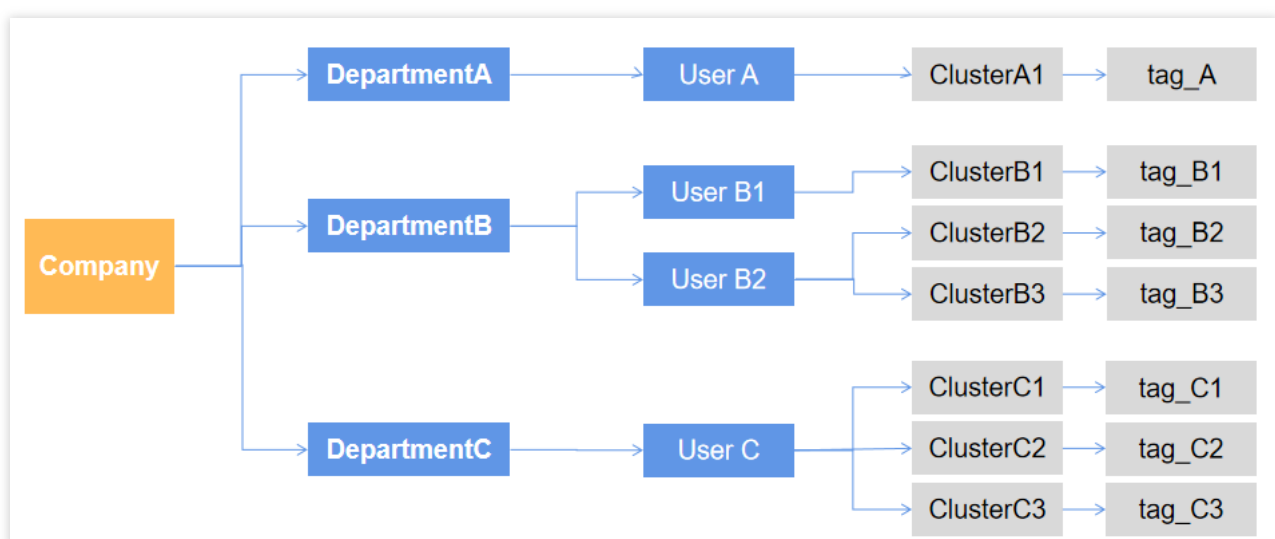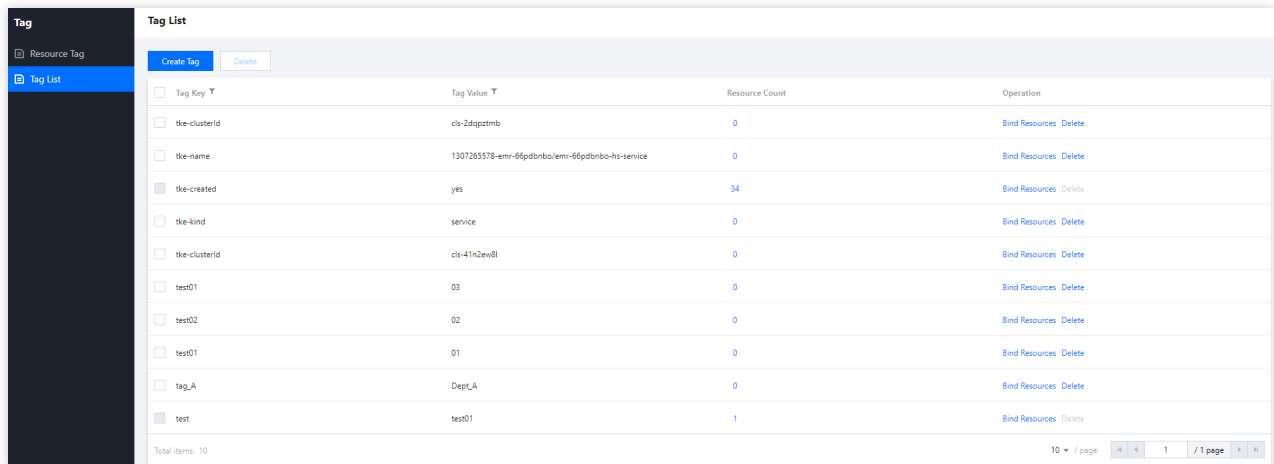
After completing the planning, to enable department A to manage cluster A1, you need to perform the following operations in sequence.

## 1. Create a tag and tag the cluster

Enter the Tag List page and click **Create Tag**.
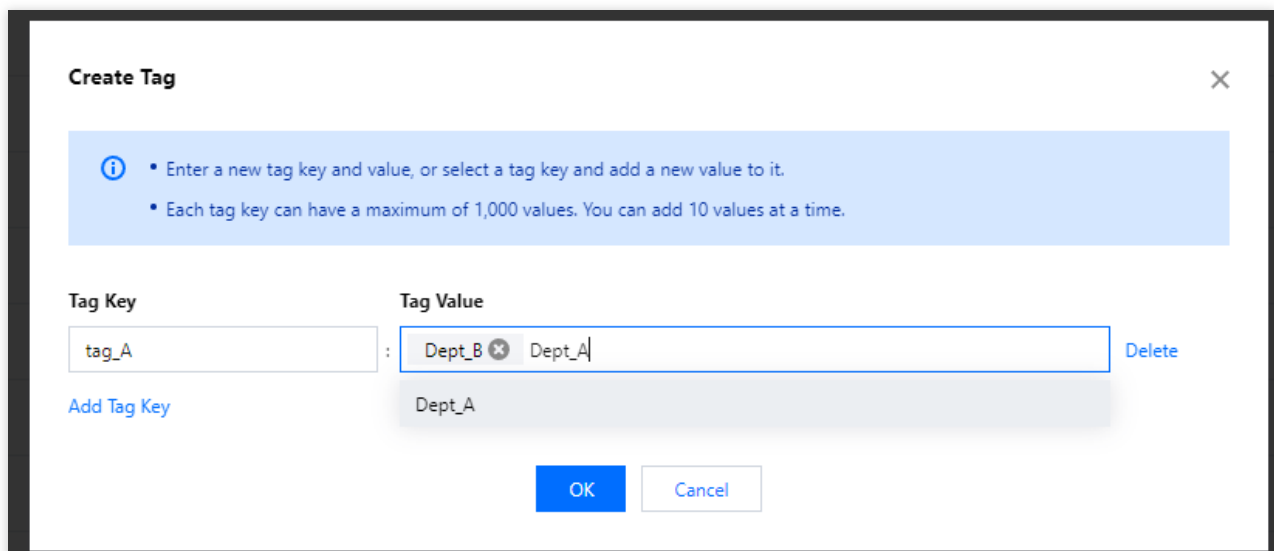
**Note**

The `tag_A` tag is used as an example here and below.



Enter the tag key and tag value and click **OK**.



On the **Cluster List** page, select the target cluster, click **More** at the top, and click **Edit Tag**.
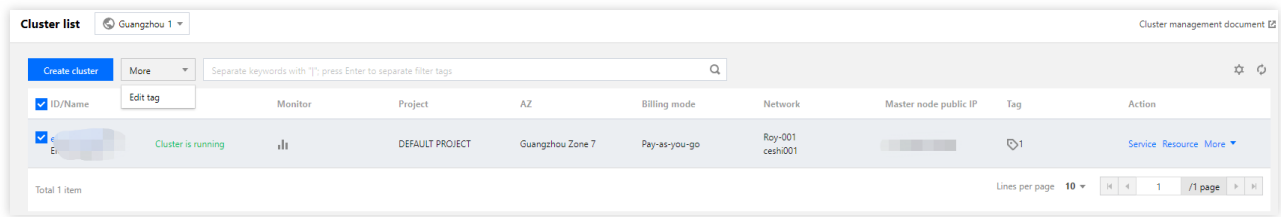
On the **Edit Tag** page, select the created `tag_A` tag and click **Complete**.
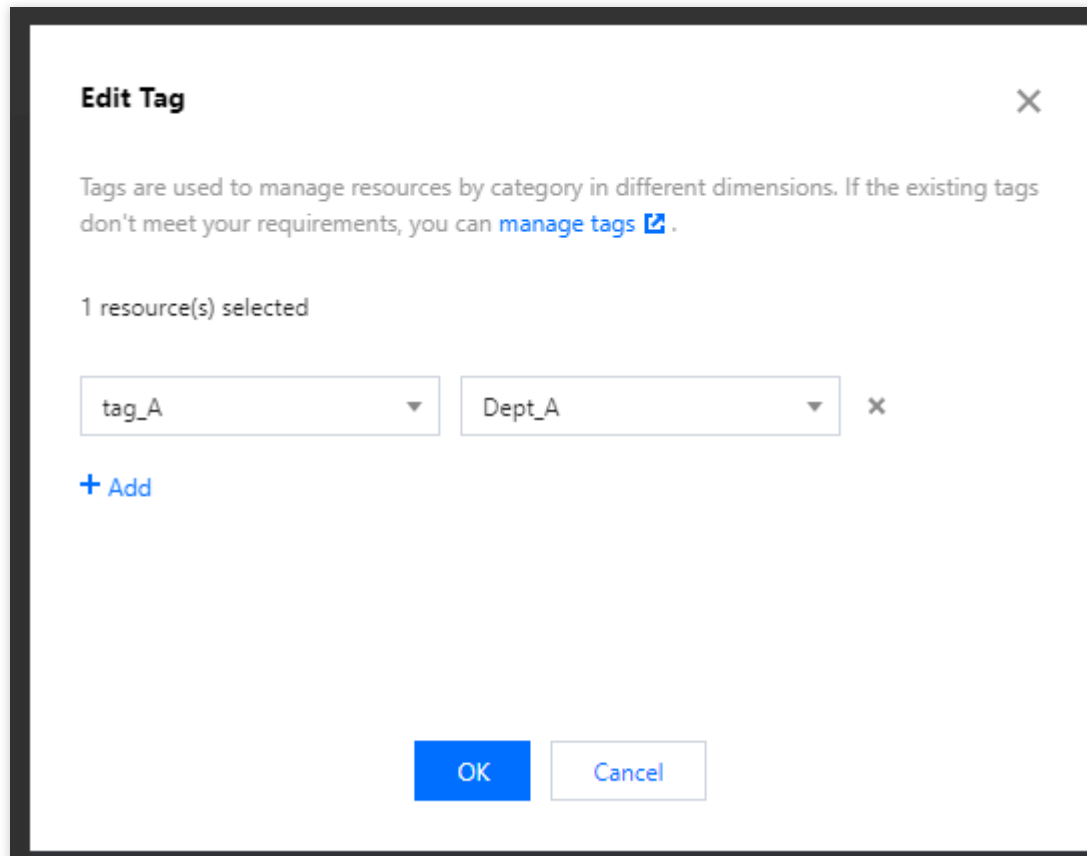


## 2. Create a custom policy

Currently, certain EMR APIs don't support authentication by tag; therefore, you need to set two policies: `Policy_Dept_A1` (for APIs that support authentication by tag) and `Policy_Dept_A2` (for APIs that don't).

**Create custom policy** `Policy_Dept_A1`

Go to the Policies page in the CAM console and click **Create Custom Policy**.

Select **Authorize by Tag**.

In **Tag Policy Generator**, select **Dept_A** for **Authorized Users**, leave **User Groups** empty, select the tag key and tag value of the planned tag for **Tag Key** and **Tag Value** respectively, add services and operations as needed, and click **Next**.

Create the `Policy_Dept_A1` policy, replace the content of the `action` and `resource` fields with the following in **Policy**, and click **Complete**.
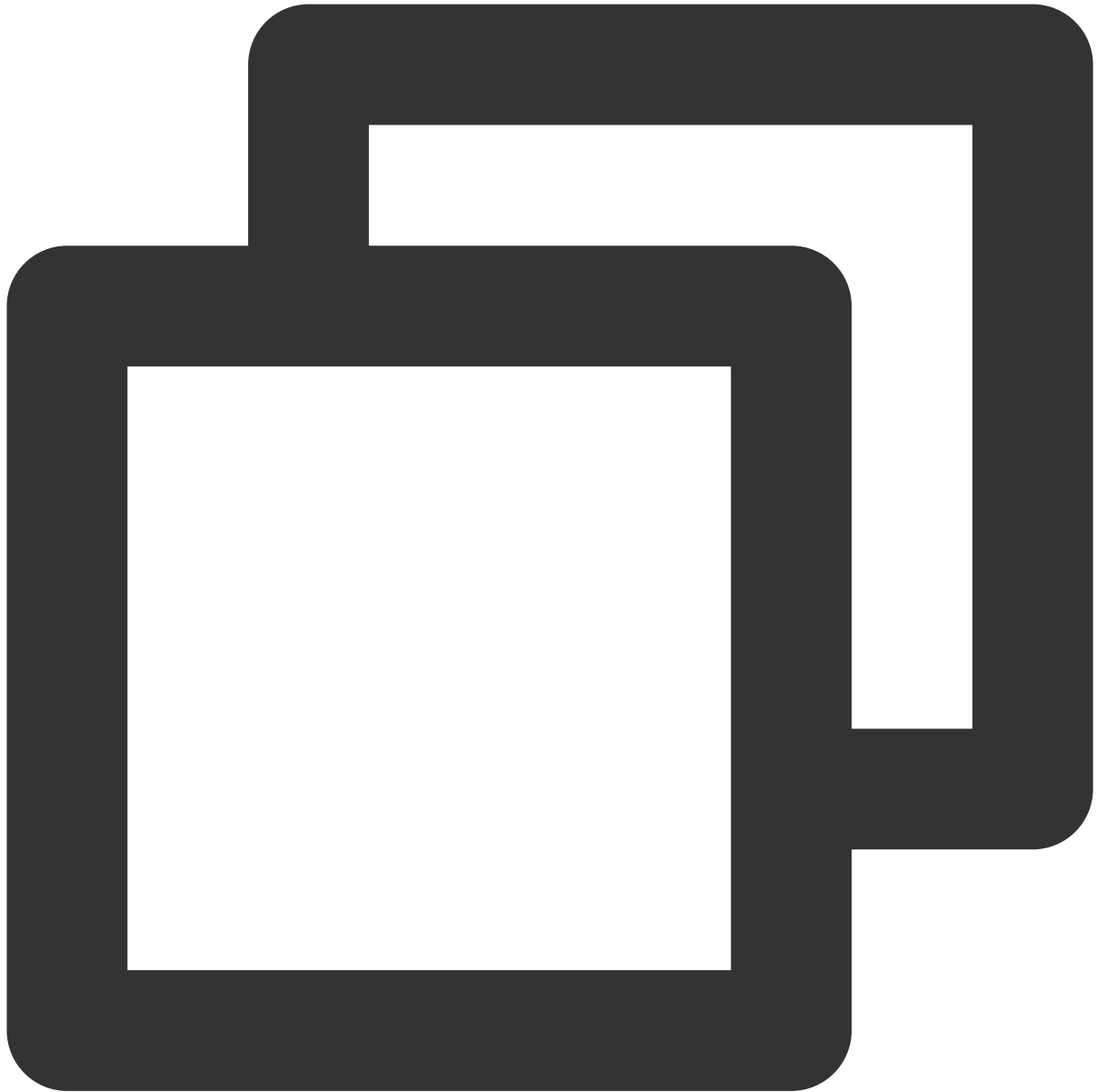


```
{
"effect": "allow",
"action": [
    "emr:DescribeClusterNodes",
    "emr:DescribeInstancesList"
],
"resource": [
    "*"
```

```
    ],
    "condition": {
        "for_any_value:string_equal": {
                "qcs:resource_tag": [
                            "tag_A&Dept_A",
                            "tag_B&Dept_B",
                            "tag_C&Dept_C",
                            "tag_D&Dept_D"
                ]
        }
    }
}
```

**Caution**

The content of the `resource` field needs to be entered based on the actual resource. For more information on how to specify it, see Resource Description Method.

**Create custom policy** `Policy_Dept_A2`

Go to the **Policies** page in the CAM console and click **Create Custom Policy**.

Select **Create by Policy Syntax**.

In **Select Policy Template**, select **Blank Template** and click **Next**.

Create the `Policy_Dept_A2` policy, clear the existing content in **Policy**, paste the following content, and click **Complete**.

```
{
    "version": "2.0",
    "statement": [
            {
                "effect": "allow",
                "action": [
"emr:RunJobFlow","emr:DescribeJobFlow","emr:FindClusterListForMC","emr:EmrScaleoutC
"emr:DescribeModifySpec","emr:DescribeMasterIp","emr:DescribeMetricsDimension"
                ],
                "resource": "*"
            }
```

```
        ]
    }
```

### 3. Authorize and authenticate the sub-user

On the **User List** page, find the target sub-user and click **Authorize** on the right. For more information on how to create a sub-user, see Creating Sub-user.

Select **Policy_Dept_A1** and **Policy_Dept_A2** and click **Complete**.

Log in as a sub-user for authentication.

At this point, when logging in to EMR, the sub-user can see and manage the `test_A` cluster only through the `tag_A` tag.

**Caution**

As the APIs in the `Policy_Dept_A2` policy can only be fully listed, a permission error may occur when new APIs are subsequently added. In case of such permission error, submit a ticket or contact us for assistance.

# Resource-Level and API-Level Authentication Operation Guide

This section describes how to authorize EMR resources. When using resource-level authentication to authorize an EMR feature, you need to plan resources for your departments or organizations first; for example, you can plan usernames, APIs, and clusters for departments A, B, and C as shown below.

After completing the planning, to grant department A only access to the `DescribeRegionAndZoneSaleInfo` API of the cluster, you need to perform the following operations in sequence:

### 1. Create a custom policy

Go to the Policies page in the CAM console and click **Create Custom Policy**.

Select **Create by Policy Syntax**.

In **Select Policy Template**, select **Blank Template** and click **Next**.

Create a custom policy named `DescribeRegionAndZoneSaleInfo-test`, clear the existing content in **Policy**, configure the access `deny` policy for EMR's `DescribeRegionAndZoneSaleInfo` API, and save the configuration.

```
{
        "version": "2.0",
        "statement": [
                {
                        "effect": "allow",
                        "action": [
"emr:DescribeRegionAndZoneSaleInfo"
                        ],
                        "resource": "*"
                }
        ]
```

```
    }
```

**Caution**

For APIs that support resource-level authentication, see **List of APIs supporting resource-level authorization**. As the APIs can only be fully listed, a permission error may occur when new APIs are subsequently added. In case of such permission error, you can add the missing API permission in the policy based on the error message.

## 2. Authorize and authenticate the sub-account

On the **User List** page, find the target sub-user and click **Authorize** on the right. For how to create a sub-user, see Creating Sub-user.

Select the `DescribeRegionAndZoneSaleInfo-test` custom policy and click **Complete**.

After you log in with the sub-account and call the above API, if a pop-up window similar to the following requesting authorization is displayed, the authentication has been successfully configured.

# Custom Service Roles

Last updated：2023-12-27 10:01:27

The custom service role feature enables you to create a CAM service role for granting access to COS resources. Please select **Tencent Cloud Product Service** as the service type and **Elastic MapReduce** as the service supporting the role. If no custom service role is configured, the system uses the **EMRCosRole** role by default to access COS resources.

## Step 1. Customize a Permission Policy

1. Log in to the CAM console, click **Create Custom Policy**, and select **Create by Policy Syntax** in the **Select Policy Creation Method** pop-up window.
2. On the **Create by Policy Syntax** page, select **Blank Template** as the template type.
3. Set the syntax policy as follows:

```
{
    "version": "2.0",
    "statement": [
        {
            "action": "cos:*",
            "effect": "allow",
            "resource": "qcs::cos::uid/appId:bucketName/*"
        }
    ]
}
```

Where `appId` is the AppID of the root account, and `bucketName` is the name of the target bucket. Generate a policy named `TestPolicy`. In practical cases, you can customize the name.

# Step 2. Create a Custom Role

1. In the CAM console, click **Create Role** to open the **Select role entity** pop-up window, select **Tencent Cloud Product Service** to go to the **Create Custom Role** page, and select **Elasticsearch MapReduce (emr)** in **Product Service**.

2. Associate the `TestPolicy` policy generated in step 1. In practical cases, you can associate a policy as needed.

3. Set the role tag keys and values and click **Next**.

4. Generate a custom role named `EMRCosRole`. In practical cases, you can customize the name.

# Step 3. Bind the Role with an EMR Cluster

In the EMR console, click a cluster ID/name to go to the instance information page, select **Instance info** > **Basic configuration** > **Custom Service Role** and then click **Set**. Set the custom service role to the `EMRCosRole` role generated in step 2. In practical cases, you can customize the name.
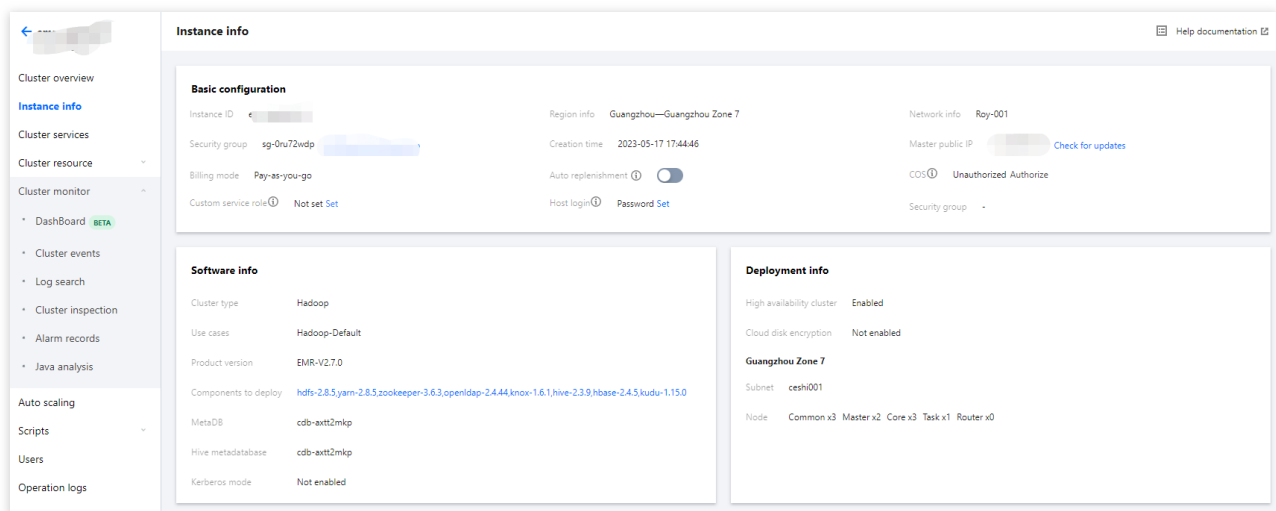
# Setting Tags

Last updated：2023-12-27 10:01:45

## Overview

[Tags](#) are key-value pairs provided by Tencent Cloud EMR to identify cluster types or node resources.
You can manage clusters or node resources in a categorized manner by using tags in various dimensions such as business, purpose, and responsible person, or use tags to easily identify them. Tag key-value pairs have no semantic meaning for EMR and are parsed and matched strictly based on strings.

## Limits

A tag is a key-value pair. You can set tags for EMR clusters or node resources to manage them in a categorized manner. Then, you can easily view and identify clusters and node resources under the corresponding tags. You can edit tags in the EMR console.
There are several limits on tags:

Quantity limit: Each cluster or node can have up to 50 tags. (**Up to five tags can be added at a time**.)
Tag key limits:

You cannot use "qcloud", "tencent", or "project" at the beginning of a tag key because they are reserved by the system. A tag key can contain up to 255 characters. Only digits, letters, and the + = . @ - symbols are supported.
Tag value limit: A tag value can contain up to 127 characters. Only digits, letters, symbols + = . @ -, and a null string are supported.

## Directions

### Editing cluster tags

1. Log in to the [EMR console](#), select the target cluster on the **Cluster list** page and click **More** > **Edit tag**.
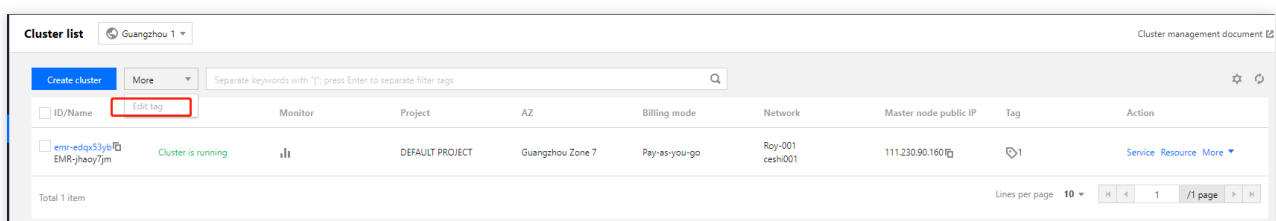
2. Add, modify or delete tags as needed in the pop-up dialog box.

**Note**

You can edit tags for up to 20 clusters at a time.
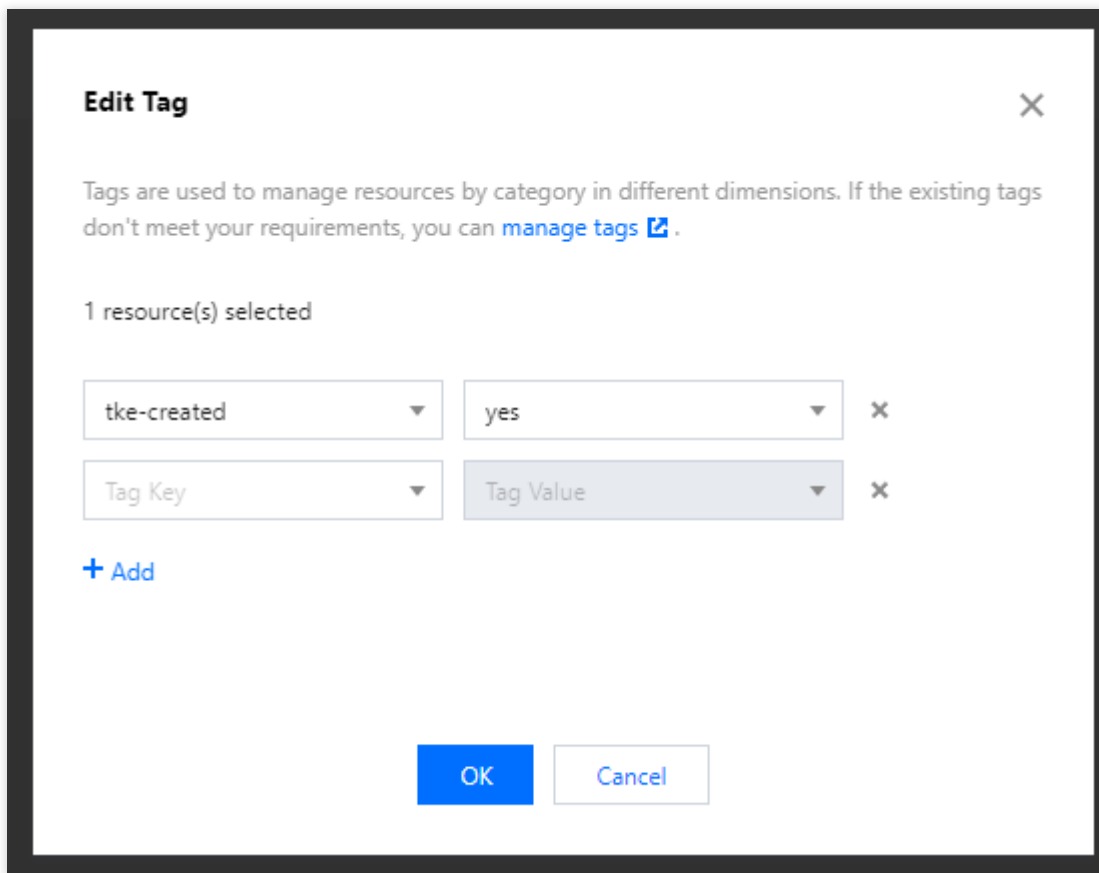


## Editing node tags

1. Log in to the EMR console and click a cluster ID/name to go to the cluster details page. Select **Cluster resources** > **Resources**, select the target node, and click **More** > **Edit tag**.
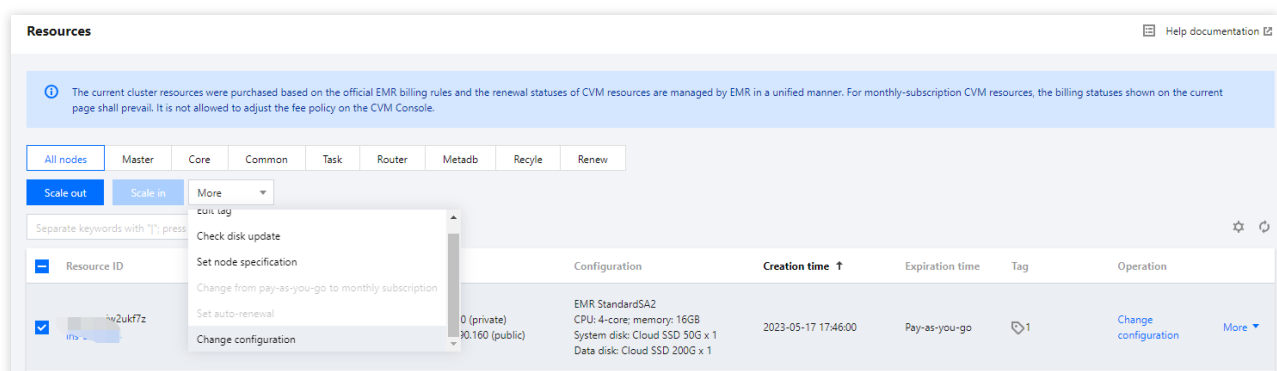


2. Add, modify or delete tags as needed in the pop-up dialog box.

**Note**

You can edit tags for up to 20 clusters at a time.

# Bootstrap Actions

Last updated：2023-12-27 10:02:08

## Overview

A bootstrap action involves executing custom scripts during cluster creation so that you can modify the cluster environment, install third-party software, and utilize proprietary data. This action runs the bootstrap script during cluster creation (including cluster scale-out) and cluster termination (including cluster scale-in), except for router nodes.

Currently, you can specify a bootstrap action only for cluster creation and termination in the console. You can also specify a bootstrap action for the scale-out or scale-in process via API. By default, if no bootstrap action is specified via API, the bootstrap action specified for cluster creation is executed during the scale-out process, and the bootstrap action specified for cluster termination is executed during the scale-in process.

1. The bootstrap action specified for cluster creation (including cluster scale-out) can be executed in any of the following circumstances:

 a. After server initialization: After server resource initialization and before EMR cluster software installation.

 b. Before cluster start: Before cluster service start.

 c. After cluster start: After cluster service start.

2. The bootstrap action specified for cluster termination (including cluster scale-in) can be executed in the following circumstance:

 a. Before service deactivation: Before cluster service deactivation.

 Bootstrap actions run scripts during cluster creation and scale-out in the order in which the scripts are added. The number of bootstrap actions cannot exceed 16.

**Note:**

We recommend that you create a small pay-as-you-go cluster first to test whether the bootstrap action works properly, and if yes, create a cluster for actual business.

## Directions

**Option 1: Add a bootstrap action when creating a cluster on the** purchase page

1. Choose **Basic Configuration** > **Advanced Settings** > **Add Bootstrap Action** to add a bootstrap action.

2. Edit or delete the added bootstrap action as needed.

Select a time for running the script and enter relevant parameters.

Name: We recommend you enter your object name.

Script Location: We recommend you copy the location information from the COS details page. Go to the COS console, click **Bucket List**, select the target script, and choose **Operation** > **Details**.

On the **Details** page, you can see the object name and object address.

Parameter: This refers to the parameters for running the script. Separate multiple parameters by spaces, and do not add spaces in individual parameters. The total length of **Parameter** and **Name** cannot exceed 240 characters.

**Option 2: Add a bootstrap action on the Basic information page of the cluster**

1. Log in to the EMR console and click the **ID/Name** of the target cluster in the cluster list to enter the cluster details page. Then, choose **Basic information** > **Bootstrap Actions** and click **Add Bootstrap Action**.

2. Edit or delete the added bootstrap action as needed. Select a time for running the script and enter relevant parameters.

Name: We recommend you enter your object name.

Script Location: We recommend you copy the location information from the COS details page. Go to the COS console, click **Bucket List**, select the target script, and choose **Operation** > **Details**.

Parameter: This refers to the parameters for running the script. Separate multiple parameters by spaces, and do not add spaces in individual parameters. The total length of **Parameter** and **Name** cannot exceed 240 characters.

## Viewing Bootstrap Result

Currently, you can specify a bootstrap action for scale-out via API but not in the console. If a bootstrap action is specified, it will be executed during scale-out; otherwise, the one specified for cluster creation will be executed.

1. View the bootstrap result in the script's system log.

Logs and script files to be executed are stored in the `/usr/local/service/scripts/` directory. The script system log is `script_syslog` . Naming convention 1: "Execution order" + "*" + "Time" + "Script name" + "*" + stderr.

Naming convention 2: "Execution order" + "*" + "Time" + "Script name" + "*" + stdout.

**Note:**

The scripts will be executed on all types of nodes, and the script files and log files output by script execution will be stored on each node.

Bootstrap scripts need to be encoded in UTF-8.

2. View the bootstrap result in the Task Center.

Log in to the EMR console, click **Task Center** on the left sidebar, or enter a cluster, click **Task** in the top-right corner, and select the target process (creating cluster, scaling out cluster, or initializing node). Then, you can click **Run Details** in the service initialization step of **Task details** to view the bootstrap result.

# Software Configuration

Last updated：2023-12-27 10:02:29

## Feature

Software configuration enables you to customize configurations of components such as HDFS, YARN, and Hive when creating a cluster.

## Custom Software Configuration

Software programs such as Hadoop and Hive have many configuration items. By using the software configuration feature, you can customize component parameters when creating a cluster. During the configuration, you need to provide the corresponding JSON files as required. You can customize the files or generate them by exporting software configuration parameters of an existing cluster for quick cluster creation. For more information on how to export software configuration parameters, please see Exporting Software Configuration.

> Currently, only parameters in the following files can be customized:
>
> HDFS: core-site.xml, hdfs-site.xml, hadoop-env.sh, log4j.properties
>
> YARN: yarn-site.xml, mapred-site.xml, fair-scheduler.xml, capacity-scheduler.xml, yarn-env.sh, mapred-env.sh
>
> Hive: hive-site.xml, hive-env.sh, hive-log4j2.properties

**Sample JSON file and description:**
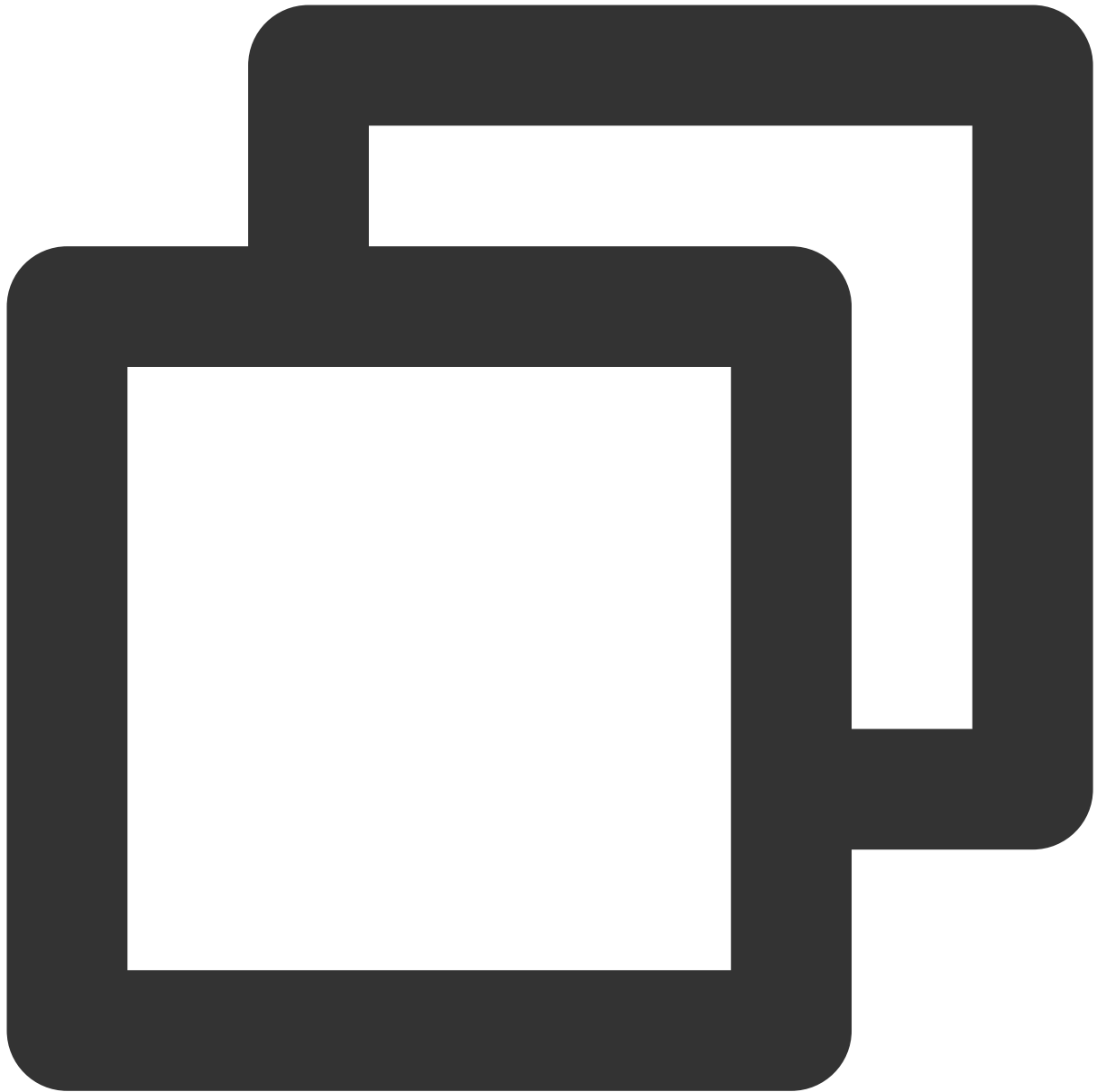
```
[
    {
        "serviceName": "HDFS",
        "classification": "hdfs-site.xml",
        "serviceVersion": "2.8.4",
        "properties": {
            "dfs.blocksize": "67108864",
            "dfs.client.slow.io.warning.threshold.ms": "900000",
            "output.replace-datanode-on-failure": "false"
        }
    },
```

```
    {
        "serviceName": "YARN",
        "classification": "yarn-site.xml",
        "serviceVersion": "2.8.4",
        "properties": {
            "yarn.app.mapreduce.am.staging-dir": "/emr/hadoop-yarn/staging",
            "yarn.log-aggregation.retain-check-interval-seconds": "604800",
            "yarn.scheduler.minimum-allocation-vcores": "1"
        }
    },
    {
        "serviceName": "YARN",
        "classification": "capacity-scheduler.xml",
        "serviceVersion": "2.8.4",
        "properties": {
            "content": "<?xml version=\\"1.0\\" encoding=\\"UTF-8\\"?>\\n<?xml-styl
        }
    }
  ]
```

**Configuration parameter descriptions:**

serviceName: component name, which must be in uppercase.

classification: filename, which must be a full name with file extension.

serviceVersion: component version, which must be the same as the corresponding component version in the EMR product version.

properties: parameters that need to be customized.

If you want to modify configuration parameters in `capacity-scheduler.xml` or `fair-scheduler.xml` , set `key` in `properties` to `content` , and set `value` to the content of the entire file.

If you want to adjust the component configuration of an existing cluster, you can configure component parameters.

# Accessing External Clusters

After configuring the HDFS access address information of an external cluster, you can read data in it.

## Configuration during purchase

EMR allows you to configure access to external clusters when you create an EMR cluster. On the purchase page, you only need to import a JSON file that meets the requirements in the software configuration section. Below is an example based on assumption:

**Assumption**

Assume that the `nameservice` of the external cluster to be accessed is `HDFS8088` , and the access method is as follows:

```
<property>
    <name>dfs.ha.namenodes.HDFS8088</name>
    <value>nn1,nn2</value>
</property>
<property>
    <name>dfs.namenode.http-address.HDFS8088.nn1</name>
    <value>172.21.16.11:4008</value>
</property>
<property>
    <name>dfs.namenode.https-address.HDFS8088.nn1</name>
    <value>172.21.16.11:4009</value>
```

```
</property>
    <name>dfs.namenode.rpc-address.HDFS8088.nn1</name>
    <value>172.21.16.11:4007</value>
<property>
    <name>dfs.namenode.http-address.HDFS8088.nn2</name>
    <value>172.21.16.40:4008</value>
</property>
<property>
    <name>dfs.namenode.https-address.HDFS8088.nn2</name>
    <value>172.21.16.40:4009</value>
</property>
<property>
    <name>dfs.namenode.rpc-address.HDFS8088.nn2</name>
    <value>172.21.16.40:4007</value>
<property>
```

**JSON file and description:**

Taking the assumption as an example, import the JSON file (the requirements for its content are the same as those for custom software configuration) in the box.
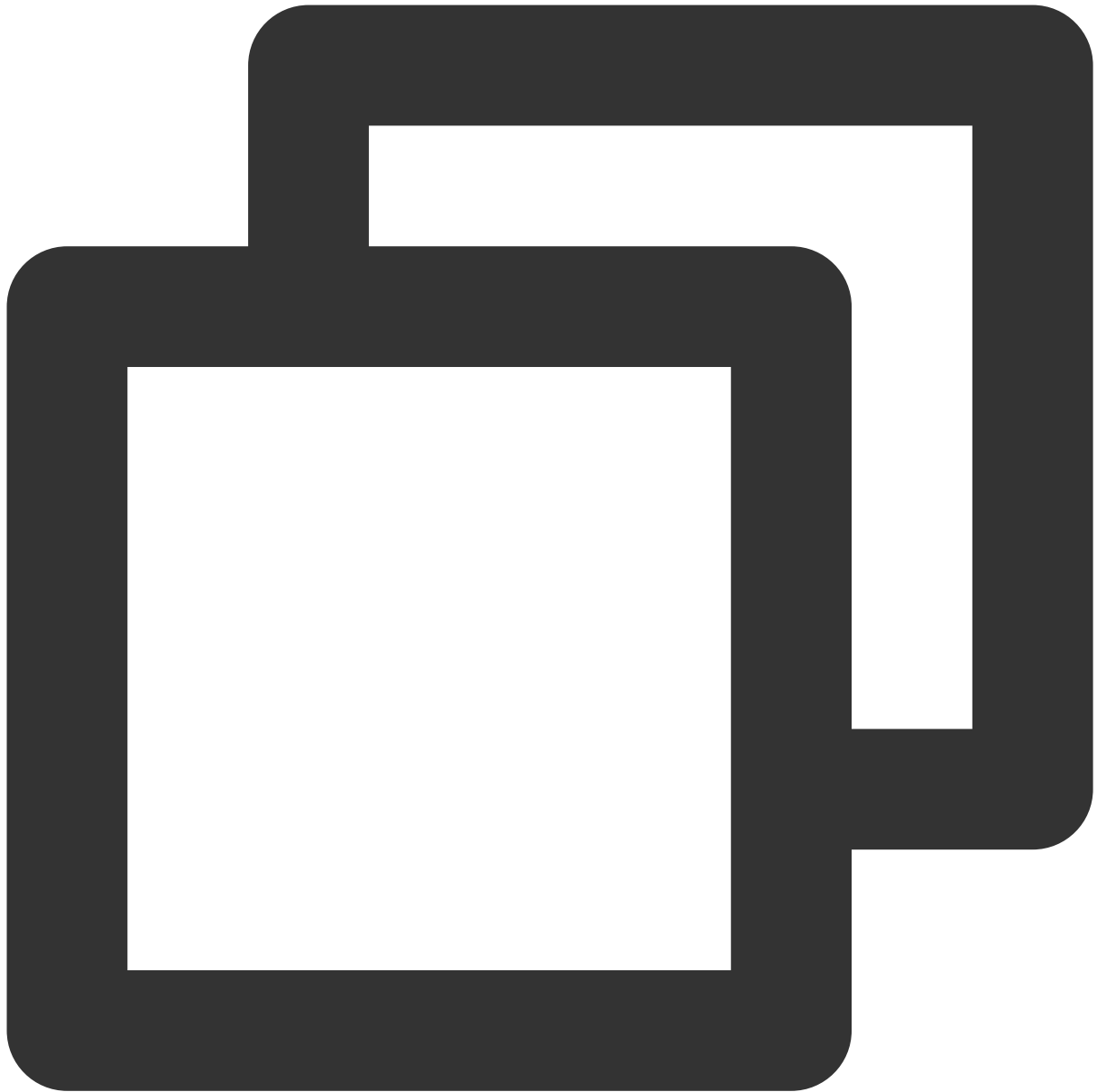
```
[
    {
        "serviceName": "HDFS",
        "classification": "hdfs-site.xml",
        "serviceVersion": "2.7.3",
        "properties": {
            "newNameServiceName": "newEmrCluster",
            "dfs.ha.namenodes.HDFS8088": "nn1,nn2",
            "dfs.namenode.http-address.HDFS8088.nn1": "172.21.16.11:4008",
            "dfs.namenode.https-address.HDFS8088.nn1": "172.21.16.11:4009",
            "dfs.namenode.rpc-address.HDFS8088.nn1": "172.21.16.11:4007",
```

```
            "dfs.namenode.http-address.HDFS8088.nn2": "172.21.16.40:4008",
            "dfs.namenode.https-address.HDFS8088.nn2": "172.21.16.40:4009",
            "dfs.namenode.rpc-address.HDFS8088.nn2": "172.21.16.40:4007"
        }
    }
]
```

**Configuration parameter description**

serviceName: component name, which must be "HDFS".

classification: filename, which must be "hdfs-site.xml".

serviceVersion: component version, which must be the same as the corresponding component version in the EMR product version.

properties: the content must be the same as that in the assumption.

newNameServiceName: it indicates the `nameservice` of the newly created cluster, which is optional. If this parameter is left empty, its value will be generated by the system; if it is not empty, its value must consist of a string, digits, and hyphen.

> Access to external clusters is supported only for high-availability clusters.
>
> Access to external clusters is supported only for clusters with Kerberos disabled.

## Configuration after purchase

EMR allows you to use the configuration distribution feature to access external clusters after creating an EMR cluster.

**Below is the assumption:**

Assume that the `nameservice` of the cluster is `HDFS80238` (if it is not a high-availability cluster, the `nameservice` will usually be `masterIp:rpcport`, such as 172.21.0.11:4007).

The `nameservice` of the external cluster to be accessed is `HDFS8088`, and the access method is as follows:

```
<property>
    <name>dfs.ha.namenodes.HDFS8088</name>
    <value>nn1,nn2</value>
</property>
<property>
        <name>dfs.namenode.http-address.HDFS8088.nn1</name>
        <value>172.21.16.11:4008</value>
</property>
<property>
        <name>dfs.namenode.https-address.HDFS8088.nn1</name>
        <value>172.21.16.11:4009</value>
```

```
</property>
  <name>dfs.namenode.rpc-address.HDFS8088.nn1</name>
        <value>172.21.16.11:4007</value>

        <property>
        <name>dfs.namenode.http-address.HDFS8088.nn2</name>
        <value>172.21.16.40:4008</value>
</property>
<property>
        <name>dfs.namenode.https-address.HDFS8088.nn2</name>
        <value>172.21.16.40:4009</value>
</property>
<property>
        <name>dfs.namenode.rpc-address.HDFS8088.nn2</name>
        <value>172.21.16.40:4007</value>
<property>
```

If the information above is in the EMR cluster, you can view it on the management page of configuration distribution or log in to the server to view the `/usr/local/service/hadoop/etc/hadoop/hdfs-site.xml` file.

1. Enter the Configuration Distribution page and select the `hdfs-site.xml` file of the HDFS component.

2. Change the value of the configuration item `dfs.nameservices` to `HDFS80238,HDFS8088`.

3. Add configuration items and their values.

| Configuration Item | Value |
| --- | --- |
| dfs.ha.namenodes.HDFS8088 | nn1, nn2 |
| fs.namenode.http-address.HDFS8088.nn1 | 172.21.16.11:4008 |
| dfs.namenode.https-address.HDFS8088.nn1 | 172.21.16.11:4009 |
| dfs.namenode.rpc-address.HDFS8088.nn1 | 172.21.16.11:4007 |
| fs.namenode.http-address.HDFS8088.nn2 | 172.21.16.40:4008 |
| dfs.namenode.https-address.HDFS8088.nn2 | 172.21.16.40:4009 |
| dfs.namenode.rpc-address.HDFS8088.nn2 | 172.21.16.40:4007 |
| dfs.client.failover.proxy.provider.HDFS8088 | org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverPr |
| dfs.internal.nameservices | HDFS80238 |

`dfs.internal.nameservice` needs to be added; otherwise, if the cluster is scaled out, the "datanode" may report an error and be marked as "dead" by the `namenode` .

4. Distribute the configuration by using the Configuration Distribution feature.

For more information on relevant configurations and principles, please see the community documentation.

# Mounting CHDFS Instance

Last updated：2023-12-27 10:02:48

## Overview

CHDFS is a high-performance distributed file system with standard HDFS access protocols and hierarchical namespaces. EMR supports read/write of data in CHDFS. This document describes how to mount a CHDFS instance to an EMR cluster.

## Directions

### Scenario 1. Mounting a CHDFS instance to a new cluster

**Note:**

New cluster: This refers to clusters created on or after December 31, 2019. For new clusters, the default CHDFS mounting address of EMR is `/data/emr/hdfs/tmp/chdfs` .

An EMR cluster is automatically adaptive to CHDFS. Create a CHDFS instance and set permissions reasonably to interconnect the CHDFS instance and EMR cluster. The configuration steps are as follows:

1. Create a CHDFS instance in the same region as the EMR cluster as instructed in Creating CHDFS Instance.
2. Create a permission group as needed as instructed in Creating Permission Group.
3. Create permission rules as needed as instructed in Creating Permission Rule.
4. Create a mount point on the same network as the EMR cluster as instructed in Creating Mount Point.
5. Check the connectivity between the CHDFS instance and the EMR cluster. Use the `hadoop fs` command line tool to run the `hadoop fs -ls ofs://${mountpoint}/` command. Here, `mountpoint` is the mounting address. If the file list is output properly, the CHDFS has been successfully mounted.

### Scenario 2. Mounting a CHDFS instance to a legacy cluster

**Note:**

 Legacy cluster: This refers to clusters created before December 31, 2019.

For more information on how to mount a CHDFS instance to a legacy EMR cluster, see Mounting CHDFS Instance.

# Unified Management of Hive Metadata

Last updated：2023-12-27 10:03:06

## Overview

When you choose to deploy the Hive component, the system provides two storage methods for Hive metadata: you can store the metadata in a MetaDB instance separately purchased for the cluster (default method) or associate the metadata with EMR-MetaDB or a self-built MySQL database. In the latter case, metadata will be stored in the associated database and will not be deleted when the cluster is terminated. This way, Hive metadata can be managed in a unified manner.

## Prerequisites

**Cluster default**: A MetaDB instance is automatically purchased to store Hive metadata, alongside metadata of other components. The MetaDB instance will also be terminated when the cluster is terminated. If you need to retain the metadata, please go to the TencentDB console and save it manually in advance.

Hive metadata is stored together with the metadata of Hue, Ranger, Oozie, Presto, Druid, and Superset.

A MetaDB instance needs to be purchased separately for the cluster as the metadatabase.

The MetaDB instance will be terminated when the cluster is terminated, i.e., the metadata stored in it will be deleted.

**Associate EMR-MetaDB**: When a cluster is created, the system will pull a MetaDB instance available in the cloud as the Hive metadatabase, so you don't need to purchase a MetaDB instance separately, which helps reduce your storage costs. Moreover, Hive metadata will not be deleted when the cluster is terminated.

The ID of the available MetaDB instance is the ID of an existing MetaDB instance in the EMR cluster under the same account.

If one or more components among Hue, Ranger, Oozie, Druid, and Superset are selected, a MetaDB instance will be automatically purchased to store the metadata of such components other than Hive.

To terminate the associated EMR-MetaDB instance, you need to go to the TencentDB console to do so. Hive metadatabases cannot be recovered once terminated.

Make sure that the associated EMR-MetaDB instance is in the same network as the newly created cluster.

**Associate self-built MySQL**: You can associate with your local self-built MySQL database to store Hive metadata, so you don't need to purchase a MetaDB instance separately, which helps reduce your storage costs. Please enter the address beginning with "jdbc:mysql://", username, and login password of the database correctly and make sure that the database can be connected with the cluster.

Make sure that the self-built database is in the same network as the EMR cluster.

Enter the username and password of the database correctly.

If one or more components among Hue, Ranger, Oozie, Druid, and Superset are selected, a MetaDB instance will be automatically purchased to store the metadata of such components other than Hive.

Make sure that the Hive metadata version in the self-built database is not below the Hive version in the new cluster.

# Directions

## Installing Hive when creating a cluster

1. Log in with your Tencent Cloud account and click Buy Now. Select the Hive component in **Components to Deploy** on the **Software Configuration** page.



2. Select one of the following values for **Hive metadatabase** as needed: **Cluster default**, **Associate EMR-MetaDB**, and **Associate self-built MySQL**.

3. Restrictions vary depending on the selected value. For more information, see **Prerequisites**.

## Installing Hive after a cluster is created

1. Log in to the EMR console and click **Cluster List** on the left sidebar. On the page that appears, click the **ID/Name** of the target cluster.

2. Choose **Cluster Service** > **Add Component** and select the Hive component.



3. Select one of the following values for **Hive metadatabase** as needed: **Cluster default**, **Associate EMR-MetaDB**, and **Associate self-built MySQL**.

4. Restrictions vary depending on the selected value. For more information, see **Prerequisites**.

# Setting Security Groups

Last updated：2024-01-10 10:02:25

EMR uses Tencent Cloud VPC as the underlying network. Security groups in EMR are used as virtual firewalls to control the access between the internal nodes in a cluster and access from external nodes to internal nodes. This document describes the best practices of using security groups in EMR to help you choose security group policies.

## Security Groups

A security group is a virtual firewall for stateful data packet filtering. As an important network isolation approach provided by Tencent Cloud, it is used to control network access to CVM instances (nodes). When creating an EMR cluster, you can select an existing security group. If there is no existing security group, EMR will automatically create one for you. If the number of security groups has reached the upper limit and you want to create a new one, delete those you no longer use. You can view existing security groups in the VPC console.

## Use Limits and Rules

For use limits and quotas of security groups, see the **Security Group Limits** section in Use Limits Overview.
A security group rule consists of:
Source: IP address of the source data (inbound) or target data (outbound)
Protocol type and protocol port: protocol type such as TCP and UDP
Policy: allow or reject access requests

## Rules for Selecting a Security Group

By default, **Select an existing security group** is selected and an EMR security group is selected. You can create a new EMR security group or select a non-EMR security group.
1. For a new EMR security group, port 22, port 30001, and necessary private network IP ranges are enabled. The security group is named in the format of "emr-xxxxxxxx_yyyyMMdd". Do not modify its name.
2. Select an existing security group available in the current region for the current instance. A security group starting with "emr" is recommended, as EMR service has been activated and necessary policies are running properly for such security groups. Security groups not starting with "emr" may lack necessary inbound and outbound rules. This may cause cluster creation failure or cluster unavailability.
3. If you add nodes in a cluster, by default, the nodes inherit the security group policy selected for the cluster when it is created.

# Details of EMR Security Group Policies

If you select a non-EMR security group when creating an EMR cluster, the following inbound and outbound rules must be included. Otherwise, the cluster creation will fail.

## Inbound rules

| Source | Protocol Port | Policy | Note |
|---|---|---|---|
| 10.0.0.0/8 | ALL | ACCEPT | Opens IP range A. |
| 172.16.0.0/12 | ALL | ACCEPT | Opens IP range B. |
| 192.168.0.0/16 | ALL | ACCEPT | Opens IP range C. |
| 0.0.0.0/0 | ICMP | ACCEPT | Opens local ICMP. |

## Outbound rules

| Source | Protocol Port | Policy | Note |
|---|---|---|---|
| 0.0.0.0/0 | ALL | ACCEPT | Opens all outbound ports. |

## Inbound rules for accessing webUI

To access the cluster service WebUI normally when using a non-EMR security group, the inbound rules must include the following policies:

| Source | Protocol Port | Policy | Note |
|---|---|---|---|
| 0.0.0.0/0 | TCP:13000 | ACCEPT | Opens port 13000 and port hue. |
| 0.0.0.0/0 | TCP:30001 | ACCEPT | Opens port 30001. |
| 0.0.0.0/0 | TCP:30002 | ACCEPT | Opens port 30002. |
| 0.0.0.0/0 | TCP:22 | ACCEPT | Opens the remote login port. |

For more information, see Security Group.

# Component Configuration Sharing

Last updated：2023-12-27 10:14:00

## Overview

You can share the components of an existing cluster with other clusters without deploying such components again.

This makes it easier to manage multiple clusters with the same component configurations.

Currently supported components include Ranger, ZooKeeper, and Kerberos.

Currently supported versions include all EMR versions containing the above components.

Dependent: The current cluster is using components of other existing clusters.

Shared: One or more components of the current cluster are added to other newly created clusters.

**Caution**

Shared components such as ZooKeeper, Kerberos, and Ranger are no longer deployed in dependent clusters separately, and dependencies cannot be canceled once set. When a cluster with shared components deployed is terminated, all its dependent clusters need to be terminated first. Note that if your account has any overdue payment, dependent clusters won't work properly.

## Limits

1. Your account must have the management permissions of one existing cluster.

2. When Kerberos is enabled, other clusters under the current account must have Kerberos deployed before you can enable the dependency.

3. If a dependent component involves user identity management, you need to manually sync the users in the shared cluster to the new cluster after creating it or adding a user. Note that as the current cluster doesn't deploy Kerberos, krb5 users cannot be created.

## Service Architecture

## Directions

### Purchasing a cluster

1. Go to the purchase page and select multiple components under Hadoop.

2. Click **Enable** and select the shared cluster.

3. After the new cluster is successfully purchased, relevant components will automatically depend on the selected cluster.
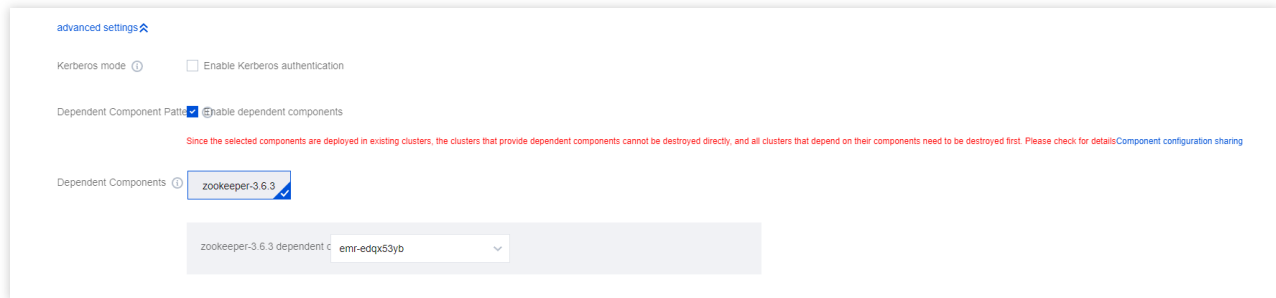


## Adding a component

1. Select **Cluster Service** > **Add Component** on the cluster details page in the console

2. Select the components to be added.

3. Click **Enable** and select the dependent components and cluster.

4. Click **OK** and wait for the components to be successfully added to the cluster.

# Cluster Information

If a cluster has dependent or shared components, the component information is displayed in **Instance info**.

**Caution**

 The management features of dependent components can be used only on the **Components** page of the depended-on cluster.

# Account Sync

User accounts need to be created in both the dependent and depended-on clusters.

1. Log in to the depended-on cluster and click **Create user** to create an account.

2. Log in to the dependent cluster and click **Create user** to create an account with the same username, user group, and password.

3. The dependent cluster will automatically sync the permission and usage information of the account.

← **emr-edqx53yb**
EMR-jhaoy7jm

Cluster overview

Instance info

Cluster services

Cluster resources

Cluster monitor

• DashBoard BETA

• Cluster events

• Logs

• Cluster inspection

• Alarm records

• Java analysis

Auto-scaling

Scripts

**Users**

Operation logs

**Users**

Help documentation ⧉

ⓘ The user list shows only the user information added via the Users page.

Create user   Delete

Separate keywords with "|"; press Enter to separate filter tags

| | Username | User group | Remarks | Creation time | Operation |
|---|---|---|---|---|---|
| ☐ | root | root | - ✎ | 2023-05-17 17:46:42 | Reset password  Download Keytab  Delete |
| ☐ | hadoop | hadoop | - ✎ | 2023-05-17 17:46:42 | Reset password  Download Keytab  Delete |

Total 2 items

Lines per page  10 ▾  ⊩  ◂  1  /1 page  ▸  ⊪

# Managing Cluster
# Instance Information

Last updated：2023-12-27 10:26:23

## Overview

Instance information records the basic information of your EMR cluster. You can view the basic configuration, software information, and hardware information of the cluster on the **Instance Information** page.

The **Basic Configuration** section displays basic information of the cluster, such as network information, creation time, security group, high-availability switch status, master public IP, billing mode, host login method, automatic service role definition switch status, and COS switch status.

The **Software Info** section displays the cluster type, use case, product version, deployed component, MetaDB, Hive metadatabase, and Kerberos mode switch status.

The **Hardware Configuration** section displays the quantity of nodes in each type.

This document describes how to view the cluster instance information in the console.

## Directions

1. After successfully creating a cluster, log in to the EMR console and click the **ID**/**Name** of the cluster.
2. If COS is not enabled for the cluster, click **Authorize**.

If you need refined authorization, you can set a custom service role for accessing Tencent Cloud resources during the execution of big data jobs. You should select **Tencent Cloud product service** as the service role type and **Elastic MapReduce** as the service supporting the role.
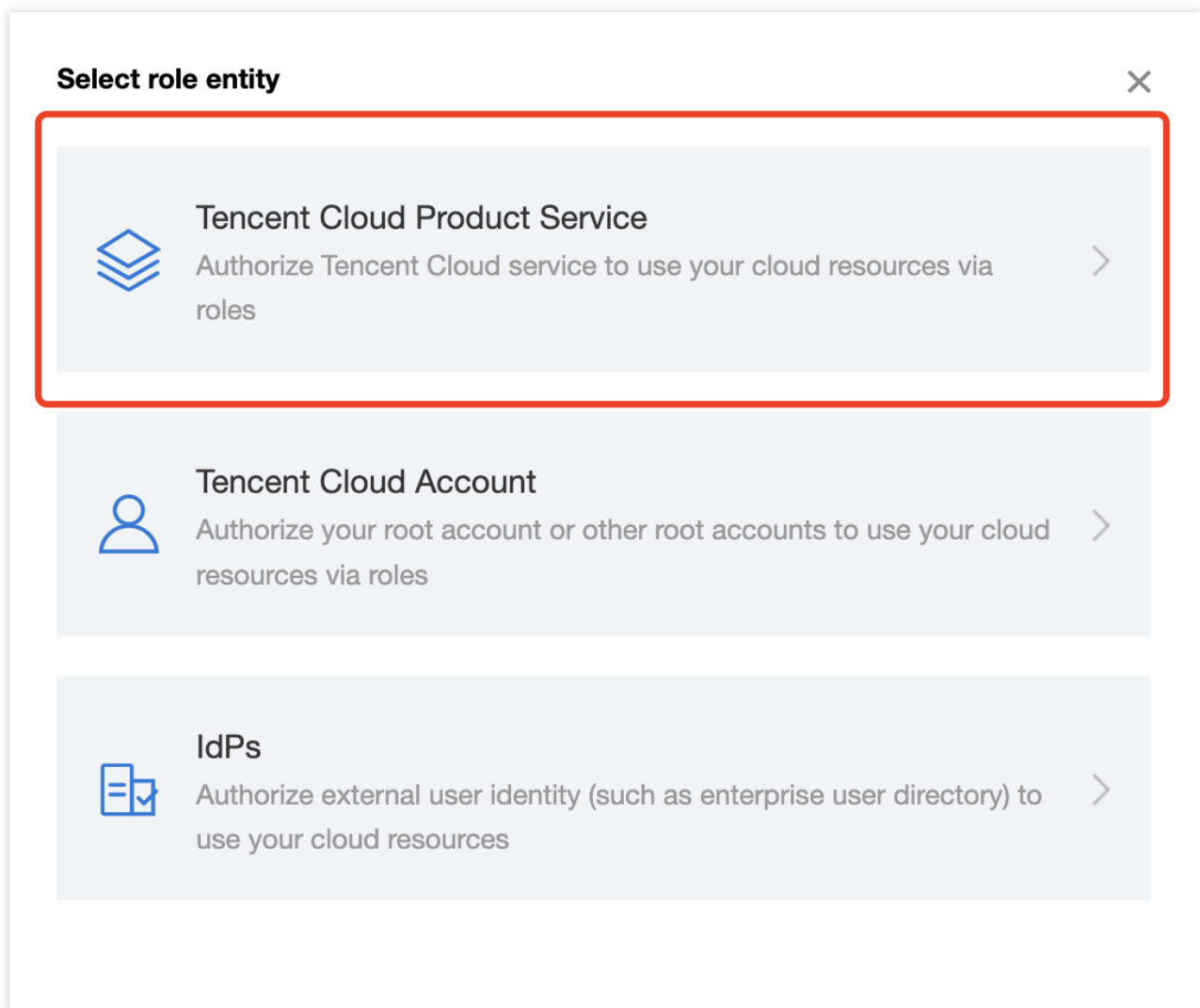
To change the host login password or method of cluster nodes after scale-out, click **Change**.



**Note**

The login method set for initializing nodes in an EMR cluster. After the cluster is created, modification of the setting here applies only to new nodes.

You can go to the CVM console to change host passwords/keys of existing nodes of the cluster. Before change, see **Shutting Down Instances** to understand the potential shutdown risks.

# Node Specification Management

Last updated：2023-12-27 10:26:39

# Feature Overview

Node specification management is used to set hardware specifications when adding more core, task, or router nodes. Each node type can have up to 3 specifications and will automatically select the current default specification during scaling. If the current default specification is sold out or unavailable, you can set a new default specification. You are recommended to keep the configuration of resources such as CPU cores, memory size, and disk capacity in the new specification the same as that of the existing nodes for optimal compatibility with current component parameters.

# Directions

1. Log in to the EMR Console, select **Cluster List**, and click the ID/Name of the target cluster to enter the cluster details page.

2. Select **Cluster Resource** > **Resource Management** > **More** > **Set Node Specification** on the cluster details page
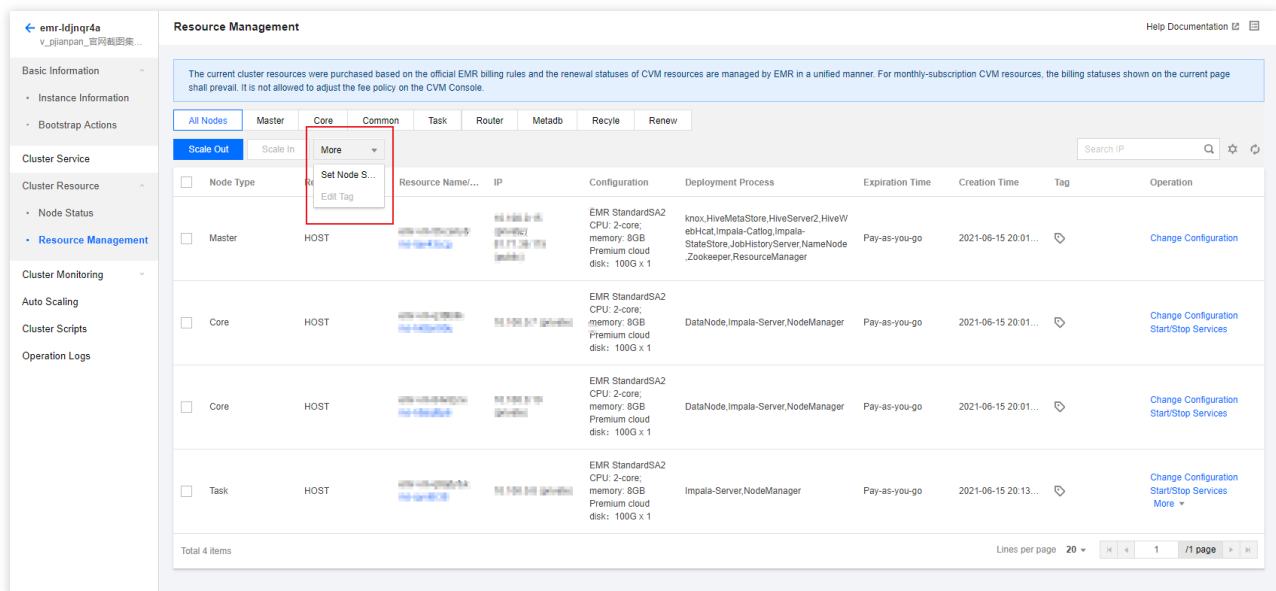


3. The default initial specification of the core nodes is the specification selected when the cluster is created. You can set or delete the default one for normal scaling.

4. The task and router nodes have no default specification. Please add the specification as prompted.

**Set Node Specification**

Node specification management is used to set the specifications for core, task, and router node scale-out. Up to 3 node specifications can be set for each node type. The default specifications for each node type are automatically selected for scale-out. If there is no quota of default specifications, you can set a new default specification. Please try to make the CPU, memory, disk, and other resources of the new specifications consistent with the existing nodes so that the component configuration parameters are compatible.

**Pay-as-you-go Specification**

**Core**

| Type | Model | Disk | Add Time | Operation |
|------|-------|------|----------|-----------|
| EMR StandardSA2 | 2-core 8 GB | Premium cloud disk 100G * 1 | 2021-06-15 20:00:53 | Set as default  Delete |

+ New Specification

**Task**

| Type | Model | Disk | Add Time | Operation |
|------|-------|------|----------|-----------|
| EMR StandardSA2 | 2-core 8 GB | Premium cloud disk 100G * 1 | 2021-06-15 20:12:47 | Set as default  Delete |

+ New Specification

**Router**

| Type | Model | Disk | Add Time | Operation |
|------|-------|------|----------|-----------|

+ New Specification

Confirm

# Checking and Updating Public IP

Last updated：2023-12-27 10:26:58

## Feature Overview

The EMR console supports checking and updating the public IP on master1. After you update or add the master1 public IP, the master public IP in the cluster configuration information and the WebUI access address will be updated too.

## Directions

1. Log in to the EMR console and click a **cluster ID** in the cluster list to go to the instance information page. Click **Check for Updates** next to **Master Public IP** in the **Basic Configuration** section.



2. If the actual master public IP is different from the public IP saved in EMR, you can update the public IP saved in EMR to the actual master public IP. This page will refresh automatically after synchronization finishes.

Check and Update Master Public IP

If the master public IP configured in EMR is not the actual public IP, you can update the master public IP to the actual public IP.

Current Public IP

Actual Public IP    Consistent

No need to update    Cancel
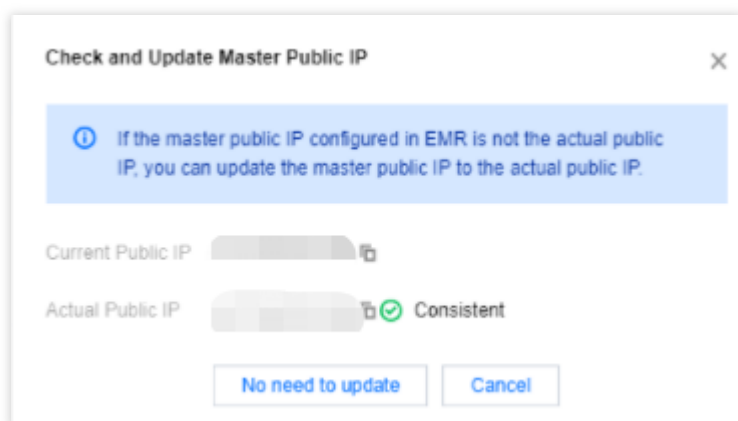
# Cluster Scale-Out

Last updated：2023-12-27 10:27:17

## Feature overview

When the compute and storage resources of your EMR cluster are insufficient, you can add more core and task nodes in the console. When the master node is overloaded or runs out of resources, you can scale out the cluster or add router nodes to share the load of the master node or to serve as task submitters in the cluster. You can scale the cluster at any time.

**Note**

By default, the current instance specification for scale-out is the instance specification selected when the cluster was created. If the current default specification is unavailable, or if you want to adjust the scale-out configuration, you need to set it in **Node spec** as instructed in Node Specification Management.

By default, the selected component will inherit the cluster-level configuration and fall into the default configuration group for that node type. You can also set the configuration of the target component through the **Specify configuration** parameter.

You cannot specify a configuration group for a ClickHouse cluster to be scaled out.

## Prerequisites

Add router nodes: A router node can be used as a submitter, through which you can submit YARN, Hive, and Spark computing tasks to the cluster. We recommend you select a model with larger memory, preferably not lower than the master node specification.

For a pay-as-you-go cluster, all newly added nodes support the pay-as-you-go billing mode.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.

2. On the cluster details page, select **Cluster Resource** > **Resource Management** > **Scale Out** and select the type of nodes to be added (core, task, or router), billing mode, optional services, and scale-out quantity as needed.

**Cluster Scale-out**

| | |
|---|---|
| Name | panjian123 |
| Project | All projects |
| Node Type | Core Node / Task Node / Router Node |
| Billing Mode | Pay-as-you-go |
| Scale Out Components | HDFS-3.1.2 / YARN-3.1.2 |
| Deployment Process | Show deployment process |
| Current Specification | EMR StandardS5 / 2-core 8 GB / Efficient cloud disk 100G * 1 |
| | The default node specification for scale-out. You can adjust the specification in Node Specification. |
| Scale-out Quantity | − 1 + |
| Tag | + New Tag |
| | Up to 5 tags can be bound |
| Cost | |

Scale-out Service: The clients of selected services will be deployed to the new node by default.

Specify configuration: Find the target component and select the level from which to inherit the configuration.

If you choose to inherit the configuration of the cluster, an added node will inherit the cluster-level configuration and fall into the default configuration group for that node type.

If you choose to inherit the configuration of the configuration group, an added node will inherit the configuration group-level configuration and fall into the selected configuration group.

Deployment Process: Displays the information of service deployment processes after components are selected for the nodes to be added. You can edit the deployment process as needed.

Do not start services after scaling: If this option is selected during scaling, added nodes will not start the service. When needed, you can start the service as instructed in Service Start/Stop.

Tag: Used to identify the added node resources.

Current Specification: The default specification.

You can set the default specification for scale-out as instructed in Node Specification Management.

The default node specification is used for scale-out. You can adjust it by clicking **Set Node Specification**.

3. After selecting the desired components and number of nodes to be added, click **Confirm** and make the payment. Then, the cluster will start scaling out, which usually takes 10 to 20 minutes.

4. For ClickHouse cluster scale-out, you can add an even number of nodes in a high-availability (HA) instance, and unlimited nodes in a non-HA instance. You can choose an existing cluster or a new one for the nodes to be added.

**Caution**

There will be no data in the newly added nodes after the virtual ClickHouse cluster is scaled out. You need to migrate the data manually as the system won't automatically do it for you. In order to achieve data balance and improve resource utilization, migrate your data in time after successful cluster scaling.

# Cluster Scale-in

Last updated：2023-12-27 10:27:35

## Overview

When the resources of your EMR cluster are excessive, you can remove task nodes in the console. When the cluster has router nodes that do not share the load of the master node or that serve as task submitters in the cluster, you can remove nodes to achieve cluster scale-in.

## Preparations

Since core nodes store data, we recommend you not terminate them to avoid data security issues. After a node is terminated, the data stored in it will be deleted. By terminating a node, you confirm that the data in it can be deleted. Pay-as-you-go cluster: Once terminated, the cluster will not be retained in the recycle bin but will be permanently terminated. Please proceed with caution.

**Caution**

Before terminating a cluster, please make sure that your data has been backed up as it cannot be recovered after cluster termination.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list.

2. On the **Cluster details** page, select **Cluster resources > Resources**, select cloud resources (task or router nodes) that can scale in from the node list. If two or more types of nodes are selected, the **Scale in** button will be unavailable.

3. In the pop-up window, confirm the instances to be terminated (removed). If the cluster metaDB is a shared metadatabase, the disk will be retained in this operation and needs to be manually removed in the CDB console.

# Auto Scaling
# Overview

Last updated：2023-12-27 10:27:53

## Overview

You can configure scaling rules in EMR to automatically increase or reduce the computing resources of task nodes as your cluster load changes. This saves costs while quickly responding to changes in the computing needs. Automatic scaling supports two scaling policies: load-based scaling (for YARN-enabled clusters only) and time-based scaling.

## Notes

1. Automatic scaling is disabled by default. It can be either custom scaling or managed scaling. Currently, managed scaling is made available through an allowlist. To use it, submit a ticket for application.

2. Custom scaling can be either load-based or time-based. If you switch the scaling policy, the original scaling rules will be retained. However, they will be in an invalid state and will not be triggered or executed. The added nodes will be retained unless the scale-in rule is triggered.

3. In managed scaling, only **HOST** is supported as the resource type; in custom scaling, **HOST** and **POD** are supported and cannot be used at the same time. If you switch the resource type, the resource specification and instance deployment methods set for the original resource type will be retained. However, they will be in an invalid state and will not be triggered or executed. The added nodes will be retained unless a scale-in rule is triggered. Pod resources are currently made available through an allowlist. To use them, submit a ticket for application.

4. **Pay-as-you-go** and **Spot instances preferred** are supported as the instance deployment policies. However, Pod resources can be deployed only on a pay-as-you-go basis.

# Custom Scaling Configuration

Last updated：2023-12-27 10:28:14

## Basic Settings

The **Basic settings** section allows you to set the node quantity range of the custom scaling feature, configure the elastic resource type, and configure whether to support graceful scale-in. It also displays the number of elastic node resources in the current cluster and supports quick release of elastic instances.



Min node count: The minimum number of elastic task nodes to be retained in the cluster after the auto scale-in policy is triggered.

Max node count: The maximum number of elastic task nodes to be retained in the cluster after the auto scale-out policy is triggered. The total number of nodes to be added based on one or more types of specifications cannot exceed the max node count.

Release all: Clears all nodes added in an auto-scaling action with one click.

Release spot instances: Clears all spot instance nodes added in an auto-scaling action with one click.

Release pay-as-you-go instances: Clears all pay-as-you-go nodes added in an auto-scaling action with one click.



Global graceful scale-in: This feature is disabled by default. It applies to all scale-in rules and can be disabled for individual scale-in rules.

Resource type: HOST resources are billed on a pay-as-you-go or spot basis, whereas POD resources are billed on a pay-as-you-go basis and can only be used to assume the NodeManager role of YARN.

**Caution**

When you switch the resource type, the scaling specifications and node selection policy of the new resource type apply.

# Scaling Specifications Management

Scaling specifications refer to node specifications for custom scaling. You can configure each cluster with up to five scaling specifications. When a scale-out rule is triggered, the scaling will be carried out according to the specification priority. When the high-priority specification has fewer resources than the number of resources to be added, the specification of the next priority will be used for supplement. In order to maintain the linear change of the cluster load, we recommend you keep the CPU and memory of the scaling specifications consistent.

Node selection policy: It can be **Pay-as-you-go** or **Spot instances preferred**. **Pay-as-you-go**: Only pay-as-you-go nodes are added to provide the required computing power when a scale-out rule is triggered.

**Spot instances preferred**: Spot instances will be preferred if they are available to provide the required computing power when a scale-out rule is triggered. **Min proportion of pay-as-you-go nodes**: The minimum proportion of pay-as-you-go nodes to the scale-out quantity.

**Example:**

If 10 nodes are to be added and the **Min proportion of pay-as-you-go nodes** is 20%, at least two pay-as-you-go nodes will be added when a scale-out rule is triggered, with the remaining eight being spot instance nodes. When spot instance nodes are insufficient, additional pay-as-you-go nodes will be added to make up the difference.

You can add, delete, change, and query the nodes in the scaling specifications, and adjust the priority of scaling specifications as needed.

The sequence of the five specifications for scale-out is as follows (same for pay-as-you-go and spot instances): When the resources are sufficient: 1 > 2 > 3 > 4 > 5.

**Example:**

If five preset specifications are available and the resources are sufficient, when a scale-out rule is triggered to add 10 nodes, 10 nodes with specification 1 will be added according to the sequence, and other preset specifications will not be selected.

When the resources are insufficient: 1 + 2 > 1 + 2 + 3 > 1 + 2 + 3 + 4 > 1 + 2 + 3 + 4 + 5.

**Example:**

If there are eight nodes with specification 1, four with specification 2, and three with specification 3, when a scale-out rule is triggered to add 13 nodes, eight nodes with specification 1, four with specification 2, and one with specification 3 will be added following the sequence.

When a resource specification is out of stock (take specification 2 as an example): 1 + 3 > 1 + 3 + 4 > 1 + 3 + 4 + 5.

**Example:**

If there are eight nodes with specification 1, no nodes with specification 2 (out of stock), and three nodes with specification 3, when a scale-out rule is triggered to add 10 nodes, eight nodes with specification 1 and two nodes with specification 3 will be added according to the sequence, while specification 2 will not be selected.

If there are eight nodes with specification 1 and nodes with other preset specifications are all out of stock, when a scale-out rule is triggered to add 10 nodes, eight nodes with specification 1 will be added, with the scale-out being partially successful.



# Scaling Rule Management

Scaling rules are business policies used to configure the trigger conditions of scaling actions and the number of nodes to be added or removed. Automatic scaling supports two types of policies: load-based scaling and time-based scaling. You can choose to use either type, but not both at the same time. When you switch the policies, the original scaling rules will be retained in an invalid state and will not be triggered. Those already added nodes will also be retained unless a scale-in rule is triggered. Each policy can be configured with up to 10 scaling rules. When two rules are triggered at the same time, the rule of the higher priority will be executed first.
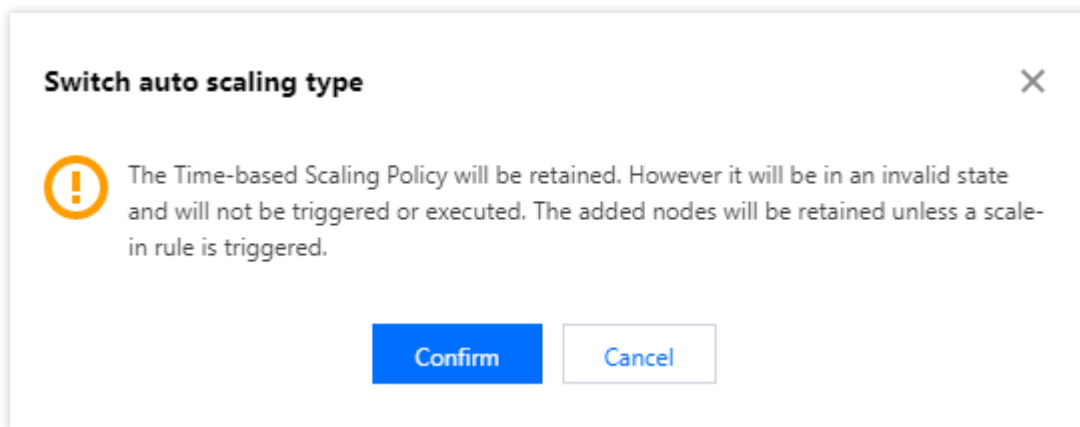


To set a scaling rule, select **Load-based scaling** or **Time-based scaling** for **Auto-scaling type** in the **Scaling rule management** pane and click **Add**.

**Load-based scaling**

When it is impossible to accurately estimate the peaks and valleys of cluster computing, you can configure the scaling policy based on the load to ensure that important jobs are completed on time. The load is mainly based on the preset YARN metric statistics rules and the task nodes are automatically adjusted when the preset conditions are triggered.
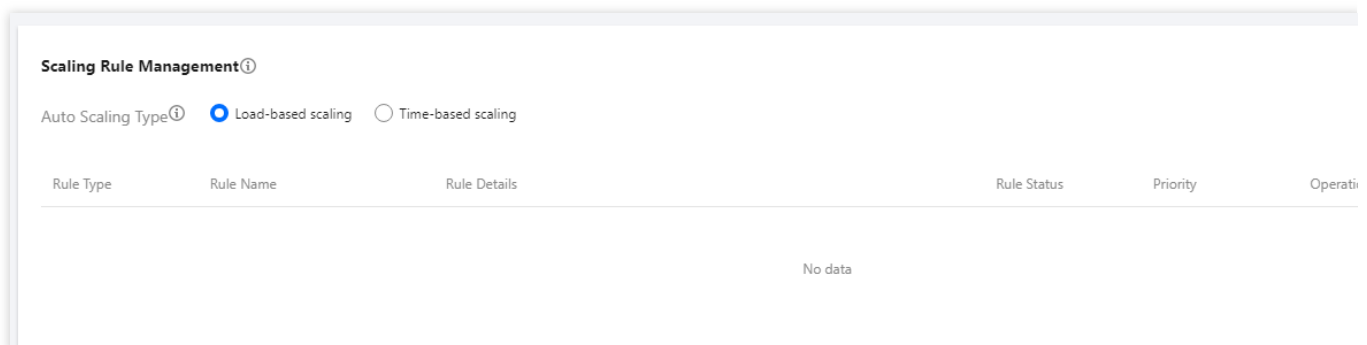


To add a load-based scaling rule, select **Load-based scaling** as the **Auto-scaling type**, click **Add**, and configure the following fields:

Rule status: Used to mark whether the rule is enabled. The default status of a rule is "Enabled". When you don't want a rule to be executed but still want to retain it, you can set the rule status to "Disabled".

Rule name: The name of the scaling rule. The scaling rule names in the same cluster must be unique (including scale-out and scale-in rules).

Validity: The load-based scaling rule can be triggered only within the validity. **Unlimited** is selected by default, and you can customize this field.

Target service: By default, the selected component will inherit the cluster-level configuration and fall into the default configuration group for that node type. You can also set the configuration of the target component through the **Specify configuration** parameter.

Node label: This field is empty by default, and the added nodes will be assigned the default label. If you have set a label, resources will be added to the node with this label.

Load metric: YARN load metric. You can set the condition rule for triggering the threshold based on the load metric selected here.

| Category | Dimension | EMR Auto-Scaling Metric | Description |
|---|---|---|---|
| AvailableVCores | root | AvailableVCores#root | Number of virtual cores available in the root queue |
| | root.default | AvailableVCores#root.default | Number of virtual cores available in the root.default queue |
| | Custom subqueue | Example: AvailableVCores#root.test | Number of virtual cores available in the root.test queue |

| | | | |
|---|---|---|---|
| PendingVCores | root | PendingVCores#root | Number of virtual cores waiting to be available in the root queue |
| | root.default | PendingVCores#root.default | Number of virtual cores waiting to be available in the root.default queue |
| | Custom subqueue | Example: PendingVCores#root.test | Number of virtual cores waiting to be available in the root.test queue |
| AvailableMB | root | AvailableMB#root | Amount of memory available in the root queue, in MB |
| | root.default | AvailableMB#root.default | Amount of memory available in the root.default queue, in MB |
| | Custom subqueue | Example: AvailableMB#root.test | Amount of memory available in the root.test queue, in MB |
| PendingMB | root | PendingMB#root | Amount of memory waiting to be available in the root queue, in MB |
| | root.default | PendingMB#root.default | Amount of memory waiting to be available in the root.default queue, in MB |
| | Custom subqueue | Example: PendingMB#root.test | Amount of memory waiting to be available in the root.test queue, in MB |
| AvailableMemPercentage | Clusters | AvailableMemPercentage | Available memory in percentages |
| ContainerPendingRatio | Clusters | ContainerPendingRatio | Ratio of the number of containers to be allocated to the number of allocated containers |
| AppsRunning | root | AppsRunning#root | Number of running tasks in the root queue |
| | root.default | AppsRunning#root.default | Number of running tasks in |

| | | | |
|---|---|---|---|
| | | | the root.default queue |
| | Custom subqueue | Example: AppsRunning#root.test | Number of running tasks in the root.test queue |
| AppsPending | root | AppsPending#root | Number of pending tasks in the root queue |
| | root.default | AppsPending#root.default | Number of pending tasks in the root.default queue |
| | Custom subqueue | Example: AppsPending#root.test | Number of pending tasks in the root.test queue |
| PendingContainers | root | PendingContainers#root | Number of containers to be allocated in the root queue |
| | root.default | PendingContainers#root.default | Number of containers to be allocated in the root.default queue |
| | Custom subqueue | Example: PendingContainers#root.test | Number of containers to be allocated in the root.test queue |
| AllocatedMB | root | AllocatedMB#root | Amount of memory allocated in the root queue |
| | root.default | AllocatedMB#root.default | Amount of memory allocated in the root.default queue |
| | Custom subqueue | Example: AllocatedMB#root.test | Amount of memory allocated in the root.test queue |
| AllocatedMB | root | AllocatedVCores#root | Number of virtual cores allocated in the root queue |
| | root.default | AllocatedVCores#root.default | Number of virtual cores allocated in the root.default queue |
| | Custom subqueue | Example: AllocatedVCores#root.test | Number of virtual cores allocated in the root.test queue |

| ReservedVCores | root | ReservedVCores#root | Number of virtual cores reserved in the root queue |
| | root.default | ReservedVCores#root.default | Number of virtual cores reserved in the root.default queue |
| | Custom subqueue | Example: ReservedVCores#root.test | Number of virtual cores reserved in the root.test queue |
| AllocatedContainers | root | AllocatedContainers#root | Number of containers allocated in the root queue |
| | root.default | AllocatedContainers#root.default | Number of containers allocated in the root.default queue |
| | Custom subqueue | Example: AllocatedContainers#root.test | Number of containers allocated in the root.test queue |
| ReservedMB | root | ReservedMB#root | Amount of memory reserved in the root queue |
| | root.default | ReservedMB#root.default | Amount of memory reserved in the root.default queue |
| | Custom subqueue | Example: ReservedMB#root.test | Amount of memory reserved in the root.test queue |
| AppsKilled | root | AppsKilled#root | Number of tasks terminated in the root queue |
| | root.default | AppsKilled#root.default | Number of tasks terminated in the root.default queue |
| | Custom subqueue | Example: AppsKilled#root.test | Number of tasks terminated in the root.test queue |
| AppsFailed | root | AppsFailed#root | Number of tasks failed in the root queue |

| | root.default | AppsFailed#root.default | Number of tasks failed in the root.default queue |
|---|---|---|---|
| | Custom subqueue | Example: AppsFailed#root.test | Number of tasks failed in the root.test queue |
| AppsCompleted | root | AppsCompleted#root | Number of tasks completed in the root queue |
| | root.default | AppsCompleted#root.default | Number of tasks completed in the root.default queue |
| | Custom subqueue | Example: AppsCompleted#root.test | Number of tasks completed in the root.test queue |
| AppsSubmitted | root | AppsSubmitted#root | Number of tasks submitted in the root queue |
| | root.default | AppsSubmitted#root.default | Number of tasks submitted in the root.default queue |
| | Custom subqueue | Example: AppsSubmitted#root.test | Number of tasks submitted in the root.test queue |
| AvailableVCoresPercentage | Clusters | AvailableVCoresPercentage | Percentage of virtual cores available in the cluster |
| MemPendingRatio | root | MemPendingRatio#root | Percentage of memory waiting to be available in the root queue |
| | root.default | MemPendingRatio#root.default | Percentage of memory waiting to be available in the root.default queue |
| | Custom subqueue | Example: MemPendingRatio#root.test | Percentage of memory waiting to be available in the root.default queue |

Statistical rule: The cluster load metric selected by the user is triggered once when the threshold is reached according to the selected aggregate dimension (average value) within a specific statistical period.

Statistical period: The statistical duration of the metric. Currently, three statistical periods are supported: 300s, 600s, and 900s.

Repeat count: The number of times that the threshold of the aggregated load metric is reached. When the repeat count is reached, the cluster will be automatically scaled.

Scale-out mode: You can select **Node**, **Memory**, or **Core**, and their values must be a positive integer.

In case of scale-out in the core or memory mode, the quantity of nodes guaranteeing the maximum computing power will be added.

**Example:**

1. In the core mode, if 10 cores are to be added, but the specification for auto-scaling is to add 8-core nodes in order of priority, then **two 8-core nodes** will be added when the rule is triggered.

2. In the memory mode, if 20 GB memory is to be added, but the specification for auto-scaling is to add 16 GB memory nodes in order of priority, then **two 16 GB MEM nodes** will be added when the rule is triggered.

Rule Type          ○ Scale Out     ○ Scale In

Rule Name ⓘ       [                          ]

1-64 characters; supports Chinese characters, letters, digits, -, and _

Valid Time ⓘ       [ Unlimited    ▼ ]

Scale-out Service   [ HDFS-2.8.5 ✓ ]   [ YARN-2.8.5 ✓ ]

Specify configuration   The component inherits the cluster-level configuration by default. To configuration, you can specify a configuration group.

Deployment Process   NodeManager
A deployment process refers to the process information of node deployment.

Node Label ⓘ       [                          ]

The name of node label can only contain numbers, letters, - or _. It cannot start with - or _ an are supported.

Load Metric ⓘ      [ AvailableVCores#root   ▼ ]

Statistical Rule ⓘ  [ Statistical period 300Second   ▼ ]  [ >          ▼ ]  [                 ]

Repeat Count ⓘ     [ 1              ▼ ]

Scale-out mode ⓘ   [ Node          ▼ ]  [                 ]

                   | Node          |
                                      Enable
                   | Core          |

Resource supplement retry ⓘ

Cooldown Period ⓘ  | Memory        |

Cool-down time range: 0-43200 seconds

Scale-in mode: You can select **Node**, **Memory**, or **Core**, and their values must be a positive integer.

In case of scale-in in the core or memory mode, a minimum quantity of nodes will be released in reverse chronological order while keeping the business running properly, with at least one node released.

**Example:**

1. In the core mode, if 20 cores are to be released, and the cluster has three 8-core 16 GB MEM and two 4-core 8 GB MEM elastic nodes in reverse chronological order when a scale-in rule is triggered, then **two 8-core 16 GB MEM nodes** will be released.

2. In the memory mode, if 30 GB memory is to be released, and the cluster has three 8-core 16 GB MEM and two 4-core 8 GB MEM elastic nodes in reverse chronological order when a scale-in rule is triggered, then **one 8-core 16 GB MEM node** will be released.

Rule Type          ● Scale Out     ○ Scale In

Rule Name ⓘ        [                                    ]

                   1-64 characters; supports Chinese characters, letters, digits, -, and _

Valid Time ⓘ       [ Unlimited          ▼ ]

Scale-out Service  [ HDFS-2.8.5 ✓ ]    [ YARN-2.8.5 ✓ ]

                   Specify configuration   The component inherits the cluster-level configuration by default. To
                   configuration, you can specify a configuration group.

Deployment Process NodeManager

                   A deployment process refers to the process information of node deployment.

Node Label ⓘ       [                                    ]

                   The name of node label can only contain numbers, letters, - or _. It cannot start with - or _ an
                   are supported.

Load Metric ⓘ      [ AvailableVCores#root   ▼ ]

Statistical Rule ⓘ [ Statistical period 300Second  ▼ ]  [ >          ▼ ]  [                ]

Repeat Count ⓘ     [ 1                    ▼ ]

Scale-out mode ⓘ   [ Node        ▼ ]  [                ]

                   ┌─────────────────┐
                   │ **Node**        │ Enable
Resource supplement retry ⓘ
                   │ Core            │
                   │ Memory          │
Cooldown Period ⓘ  └─────────────────┘

                   Cool-down time range: 0-43200 seconds

Resource supplement retry: When auto scale-out is performed during peak hours, the number of nodes added may be less than the target number due to the lack of resources. In such a case, if the resource supplement retry policy is enabled, the system will automatically retry to request resources when there are sufficient resources of the set scaling specifications until the target number is reached or approximated. If frequently encountering this situation, you can

enable this feature. Please note that retry may extend the time of auto scale-out and this policy change may affect your business.

Cooldown period: The interval (60s to 43200s) before carrying out the next auto-scaling action after the rule is successfully executed.

Graceful scale-in: After this feature is enabled, if the scale-in action is triggered when a node is executing tasks, the node will not be released immediately. Instead, the scale-in action will be executed after the tasks are completed. However, the scale-in action will still be executed if the tasks are not completed within the specified time.

**Create Rule**

ⓘ   1. If there are multiple preset resource specifications, we recommend you keep CPUs and memory sizes for different s
consistent and match them with the resource occupation configuration of the service, so as to avoid the service proce
due to the inconsistency between node hardware configuration and service configuration.
2. When the scale-in rule is triggered, nodes will be removed in reverse chronological order of their creation time.

Rule Type     ○ Scale Out    ⦿ Scale In

Rule Name ⓘ   [                                    ]

1-64 characters; supports Chinese characters, letters, digits, -, and _

Valid Time ⓘ   [ Unlimited ▾ ]

Load Metric ⓘ   [ AvailableVCores#root ▾ ]

Statistical Rule ⓘ   [ Statistical period 300Second ▾ ] [ > ▾ ] [                    ]

Repeat Count ⓘ   [ 1 ▾ ]

Scale-in mode ⓘ   [ Node ▾ ] [                    ]

Cooldown Period ⓘ   [                    ]

Cool-down time range: 0-43200 seconds

Graceful Scale-In ⓘ   🔵 [                    ]   Value range: 60-1800 seconds

[ Confirm ]   [ Cancel ]

## Time-based scaling

If there are obvious peaks and valleys in your cluster computing workload during a certain period, you can configure
time-based scaling policies to ensure that important jobs are completed on time. Time-based scaling policies allows
you to automatically add or remove task nodes daily, weekly, or monthly during a set time period.
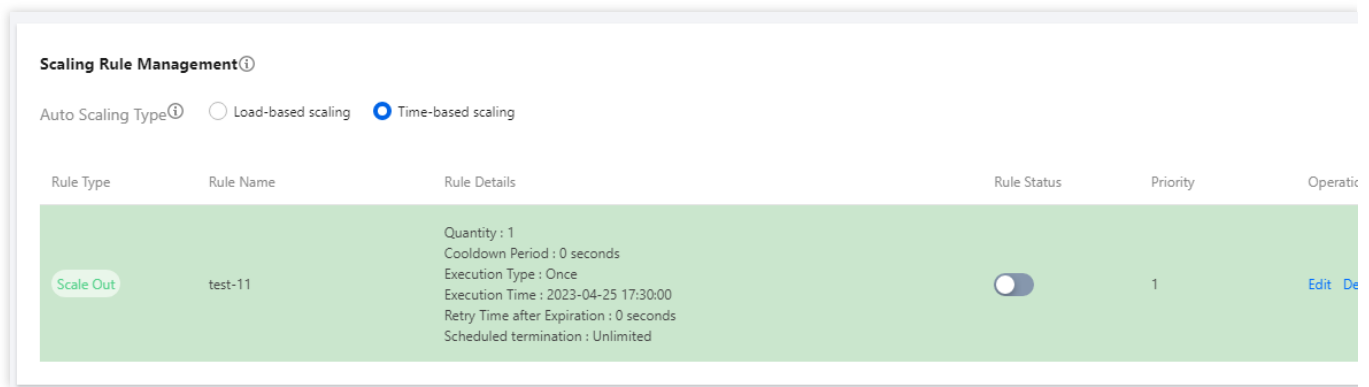
To add a time-based scaling rule, select **Time-based scaling** as the **Auto-scaling type**, click **Add**, and configure the following fields:

Rule name: The name of the scaling rule. The scaling rule names in the same cluster must be unique (including scale-out and scale-in rules).

Target service: By default, the selected component will inherit the cluster-level configuration and fall into the default configuration group for that node type. You can also set the configuration of the target component through the **Specify configuration** parameter.

Once: Triggers a scaling action once at a specific time, accurate to the minute.

Recurring: Triggers a scaling action daily, weekly, or monthly at a specific time or time period.

Retry period: Auto-scaling may not be executed for various reasons at the specified time. After you set the retry period, the system will try to execute the scaling at a certain interval within the period until it is executed when the conditions are met.

Rule expiration: The time-based scaling rule is valid until the end of the date specified here.

Scale-out mode: You can select **Node**, **Memory**, or **Core**, and their values must be an integer other than 0.

In case of scale-out in the core or memory mode, the quantity of nodes guaranteeing the maximum computing power will be added.

**Example:**

1. In the core mode, if 10 cores are to be added, but the specification for auto-scaling is to add 8-core nodes in order of priority, then **two 8-core nodes** will be added when the rule is triggered.

2. In the memory mode, if 20 GB memory is to be added, but the specification for auto-scaling is to add 16 GB memory nodes in order of priority, then **two 16 GB MEM nodes** will be added when the rule is triggered.

Rule Type          ● Scale Out      ○ Scale In

Rule Name (i)      [                                    ]

                   1-64 characters; supports Chinese characters, letters, digits, -, and _

Scale-out Service   [ HDFS-2.8.5 ✓ ]   [ YARN-2.8.5 ✓ ]

                   Specify configuration    The component inherits the cluster-level configuration by default. To
                   configuration, you can specify a configuration group.

Deployment Process  NodeManager

                   A deployment process refers to the process information of node deployment.

Node Label (i)     [                                    ]

                   The name of node label can only contain numbers, letters, - or _. It cannot start with - or _ and
                   are supported.

Execution Type     ● Once      ○ Recurring

Execution Time     [ 2023-04-25 🗓 ]   [ 17:43 ]

Retry Time after Expiration (i)   [            ]

                   The retry time after expiration should be between 0 and 21600 seconds

Scale-out mode (i)    [ Node        ▼ ]   [                ]

                      ┌─────────────────┐
                      │ Node            │
                      ├─────────────────┤ Enable
Resource supplement retry (i)  │ Core    │
                      ├─────────────────┤
Cooldown Period (i)   │ Memory          │
                      └─────────────────┘

                   Cool-down time range: 0-43200 seconds

Scale-in mode: You can select **Node**, **Memory**, or **Core**, and their values must be an integer other than 0.

In case of scale-in in the core or memory mode, a minimum quantity of nodes will be released in reverse chronological

order while keeping the business running properly, with at least one node released.

**Example:**

1. In the core mode, if 20 cores are to be released, and the cluster has three 8-core 16 GB MEM and two 4-core 8 GB MEM elastic nodes in reverse chronological order when a scale-in rule is triggered, then **two 8-core 16 GB MEM nodes** will be released.

2. In the memory mode, if 30 GB memory is to be released, and the cluster has three 8-core 16 GB MEM and two 4-core 8 GB MEM elastic nodes in reverse chronological order when a scale-in rule is triggered, then **one 8-core 16 GB MEM node** will be released.

**Create Rule**

ⓘ    1. If there are multiple preset resource specifications, we recommend you keep CPUs and memory sizes for different
        consistent and match them with the resource occupation configuration of the service, so as to avoid the service proc
        due to the inconsistency between node hardware configuration and service configuration.
        2. When the scale-in rule is triggered, nodes will be removed in reverse chronological order of their creation time.

Rule Type                    ○ Scale Out    ● Scale In

Rule Name ⓘ                 [                                                    ]

                             1-64 characters; supports Chinese characters, letters, digits, -, and _

Execution Type               ● Once    ○ Recurring

Execution Time

                             [ 2023-04-25 📅 ]    [ 17:43 ]

Retry Time after Expiration ⓘ   [                    ]

                             The retry time after expiration should be between 0 and 21600 seconds

Scale-in mode ⓘ              [ Node           ▼ ]    [                    ]

                             | **Node** |
                             | Core |
                             | Memory |

Cooldown Period ⓘ

                                              nge: 0-43200 seconds

Graceful Scale-In ⓘ          ◯○

                                         [ Confirm ]    [ Cancel ]

Resource supplement retry: When auto scale-out is performed during peak hours, the number of nodes added may be less than the target number due to the lack of resources. In such a case, if the resource supplement retry policy is enabled, the system will automatically retry to request resources when there are sufficient resources of the set scaling specifications until the target number is reached or approximated. If frequently encountering this situation, you can enable this feature. Please note that retry may extend the time of auto scale-out and this policy change may affect your business.

Cooldown period: The interval (60s to 43200s) before carrying out the next auto-scaling action after the rule is successfully executed.

Scheduled termination: You can specify the usage duration for the added nodes, so that the nodes will not be affected when a scale-in rule is triggered. **Unlimited** is selected by default. You can enter an integer between 1–24 (in hours) here.

**Use cases:**

This feature is suitable when you need more computing resources for a specified period of time (within one day) and the resources should not be affected by other scale-in rules.



Rule status: Used to mark whether the rule is enabled. The default status of a rule is "Enabled". When you don't want a rule to be executed but still want to retain it, you can set the rule status to "Disabled".

Graceful scale-in: After this feature is enabled, if the scale-in action is triggered when a node is executing tasks, the node will not be released immediately. Instead, the scale-in action will be executed after the tasks are completed. However, the scale-in action will still be executed if the tasks are not completed within the specified time.

**Create Rule**

ⓘ 1. If there are multiple preset resource specifications, we recommend you keep CPUs and memory sizes for different s
consistent and match them with the resource occupation configuration of the service, so as to avoid the service proce
due to the inconsistency between node hardware configuration and service configuration.
2. When the scale-in rule is triggered, nodes will be removed in reverse chronological order of their creation time.

Rule Type          ◯ Scale Out    ⦿ Scale In

Rule Name ⓘ        [                                    ]

                   1-64 characters; supports Chinese characters, letters, digits, -, and _

Execution Type     ⦿ Once    ◯ Recurring

Execution Time     [ 2023-04-25 🗓 ]  [ 17:43 ]

Retry Time after Expiration ⓘ   [                ]

                   The retry time after expiration should be between 0 and 21600 seconds

Scale-in mode ⓘ    [ Node        ▾ ] [                ]

Cooldown Period ⓘ  [                ]

                   Cool-down time range: 0-43200 seconds

Graceful Scale-In ⓘ  🔵 [                ]    Value range: 60-1800 seconds

                   [ Confirm ]  [ Cancel ]

# Viewing Scaling Records

Last updated：2023-12-27 10:28:32

The records of auto scaling actions can be viewed in **Scaling history**. Grading of auto scaling events is supported, and event alarm policies are set based on the event level. For more information about event levels, see Cluster Events. For more information about event alarm configuration, see Alarm Configurations.

# Custom Scaling Records

Supports filtering the scaling records by execution time range and searching by policy name.

Sorts by **Execution time**, and displays **Execution time**, **Policy name**, **Scaling type**, and **Execution status**. You can click **Details** in the **Operation** column to view details.

There are four auto scaling execution statuses:

Executing: The automatic scaling is being executed.

Successful: All nodes are added to or removed from the cluster.

Partially successful: Some nodes are successfully added or removed from the cluster, while others fail due to disk quota management or CVM inventory.

Failed: No node is added to or removed from the cluster.

**Resource supplement retry** displays whether this feature is enabled. If it is enabled, the retries will be displayed.

**Elastic node count** displays the details of the execution result. If the execution fails, it will display the cause of the failure as well as the solution.

**Scaling Specification** displays the instance specification and the number of added or removed nodes after the rule is triggered.

# 自动伸缩

策略管理　　**伸缩记录**

| 今天 | 昨天 | 近7天 | 近30天 | 2022-07-07 00:00:00 至 2022-07-07 20:38:55 📅 | 策略名称 | 请输入策略名称 |

| 执行时间 | 策略名称 | 伸缩类型 | 执行状态 | 操作 |
| --- | --- | --- | --- | --- |
| 2022-07-07 18:06:27 | ask | 扩容 | 成功 | 详情 |
| 2022-07-07 17:20:17 | ask | 扩容 | 成功 | 详情 |
| 2022-07-07 16:49:45 | task | 缩容 | 成功 | 详情 |
| 2022-07-07 16:24:22 | task | 缩容 | 成功 | 详情 |
| 2022-07-07 16:08:52 | task | 缩容 | 成功 | 详情 |
| 2022-07-07 15:48:28 | ask | 扩容 | 成功 | 详情 |
| 2022-07-07 15:28:02 | task | 缩容 | 成功 | 详情 |
| 2022-07-07 15:07:30 | ask | 扩容 | 成功 | 详情 |
| 2022-07-07 14:47:08 | ask | 扩容 | 成功 | 详情 |
| 2022-07-07 14:16:35 | task | 缩容 | 成功 | 详情 |

共 52 项　　　　　　　　　　　　　　　　　　　　　　　　每页显示行

执行策略详情 ✕

| 策略名称 | 负载-1021 |
|---|---|
| 策略类型 | 负载伸缩 |
| 执行状态 | 成功 |
| 开始时间 | 2023-02-22 14:43:19 |
| 结束时间 | 2023-02-22 14:49:22 |
| 资源补足重试 | 未开启 |
| 伸缩节点数 | 扩容信息:[扩容前:2台, 扩容后:12台, 成功:10台, 失败:0台] |
| 伸缩规格 | 规格1:<br>付费模式: 按量计费<br>4台: SA2.LARGE8: CPU: 4核 内存:8GB<br>    系统盘:<br>        高效云盘: 50G<br>    数据盘:<br>        增强型SSD云硬盘: 100G x 1<br>规格2:<br>付费模式: 竞价实例<br>6台: SA2.LARGE8: CPU: 4核 内存:8GB<br>    系统盘:<br>        高效云盘: 50G<br>    数据盘:<br>        增强型SSD云硬盘: 100G x 1 |

关闭

# Managed scaling records

Supports filtering the scaling records by **Execution Time** or **Scaling Type**.

Supports sorting by **Execution Time**, **Scaling Type**, **Model Specification**, **Quantity**, **Execution Status**, or **Reason**.

There are two execution statuses of managed scaling:

Successful: The target node is added to or removed from the cluster based on the cluster load.

Failed: The target node failed to be added to the cluster based on the cluster load due to resource insufficiency. We recommend you modify the preset resource specification.

Model Specification: The target model specification and type after the rule is triggered for a scale-out or scale-in.

Quantity: The number of added or removed nodes in various specifications after the operation.

# Managed Scaling Configuration

Last updated：2023-12-27 10:28:49

After the managed scaling feature is enabled, the system will continuously monitor the load of the YARN cluster and calculate the peak fluctuations in the last 10 minutes, so as to automatically add or remove task nodes. Managed scaling applies only to clusters containing the YARN component.

## Basic Settings

The **Basic settings** section allows you to set the node quantity range of the automatic scaling feature and the minimum number of pay-as-you-go nodes.

Min number of nodes: The minimum number of task nodes a cluster can have when the managed scale-in policy is triggered.

Max number of nodes: The maximum number of task nodes a cluster can have when the managed scale-out policy is triggered. The accumulated number of task nodes under one or more rules cannot exceed the maximum number of nodes.

Min number of pay-as-you-go nodes: The minimum number of pay-as-you-go nodes to be added after a scale-out is triggered, which is used to set the proportion of pay-as-you-go nodes and spot instances. By default, the value is the maximum number of nodes.

**For example:**

If the minimum number of nodes is set to `0`, the maximum number of nodes to `100`, and the minimum number of pay-as-you-go nodes to `10`, at least 10 pay-as-you-go nodes will be added when a scale-out is triggered, with the remaining ones being spot instances. When spot instances are insufficient, additional pay-as-you-go nodes will be added to make up the difference.

## Scaling Specifications Management

Scaling specifications refer to node specifications for managed scaling. You can configure each cluster with up to five scaling specifications. When a scale-out rule is triggered, the scaling will be carried out according to the specification priority. When the high-priority specification has fewer resources than the number of resources to be added, the specification of the next priority will be used for supplement. In order to maintain the linear change of the cluster load, we recommend you keep the CPU and memory of the scaling specifications consistent. Managed scaling is supported only for host resources.

You can add, delete, change, and query the nodes in the scaling specifications, and adjust the priority of scaling specifications as needed.

The sequence of the five specifications for scale-out is as follows (same for pay-as-you-go and spot instances):

When the resources are sufficient: 1 > 2 > 3 > 4 > 5

**For example:**

If five preset specifications are available and the resources are sufficient, when a scale-out rule is triggered to add 10 nodes, 10 nodes with specification 1 will be added according to the sequence, and other preset specifications will not be selected.

When the resources are insufficient: 1 + 2 > 1 + 2 + 3 > 1 + 2 + 3 + 4 > 1 + 2 + 3 + 4 + 5.

**For example:**

If there are eight nodes with specification 1, four with specification 2, and three with specification 3, when a scale-out rule is triggered to add 13 nodes, eight nodes with specification 1, four with specification 2, and one with specification 3 will be added following the sequence.

When a resource specification is out of stock (take specification 2 as an example): 1 + 3 > 1 + 3 + 4 > 1 + 3 + 4 + 5.

**For example:**

If there are eight nodes with specification 1, no nodes with specification 2 (out of stock), and three nodes with specification 3, when a scale-out rule is triggered to add 10 nodes, eight nodes with specification 1 and two nodes with specification 3 will be added according to the sequence, while specification 2 will not be selected.

If there are eight nodes with specification 1 and nodes with other preset specifications are all out of stock, when a scale-out rule is triggered to add 10 nodes, eight nodes with specification 1 will be added, with the scale-out being partially successful.

## Managed scaling monitoring metrics

Managed scaling monitors many metrics and calculates the suggested number of nodes under each metric. Then, it determines on a scale-out or scale-in based on the number of nodes.

| Managed Scaling Monitoring Metric | Description |
| --- | --- |
| AvailableMemPercentage | Available memory in percentages |
| AvailableVCoresPercentage | Proportion of virtual cores available in YARN in percentages |

Statistical Rule: The cluster load metric is set to process the peak loads in the last 10 minutes based on the specified aggregation dimension within a statistical period.

Statistical Period: The statistical duration of the metric, which is 1 minute.

By default, managed scaling applies the principle of quick scale-out and cautious and graceful scale-in.

# Repairing Disks

Last updated：2023-12-27 11:02:43

## Overview

Local disk replacement events are automatically monitored in the EMR console. After disk replacement, you can initialize the new disk in the console on your own.

**Caution**

After you receive a faulty disk notification from the CVM and repair or replace the physical disk as instructed in the notification, the disk repair operation is triggered in the EMR console.

When a disk is replaced, all data on the disk is lost. Make sure that the data on the disk is backed up prior to replacement.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the corresponding cluster in the cluster list.

2. On the cluster details page, choose **Cluster Resource** > **Resource Management** and then repair the disk on the node where the disk is replaced.

3. In the disk repair process, the services on the current node will be restarted, during which both the services and node are unavailable. Therefore, we recommend you repair a disk during off-peak business hours.

## Kudu Service Recovery

**Caution**

If there are multiple local disks and one or more of them use the EMR disk repair feature, the node where the disks are located will run the KuduServer service.

Due to the limitation of Kudu's `fs_data_dirs` feature, to ensure normal startup of KuduServer after one or more disks are formatted, all data directories on the KuduServer node are empty. You should confirm that these directories are not used by any businesses other than Kudu.

Scenario:

Under **Cluster Service** in the EMR console, you can view the health status of the KuduServer on the node where the disk is replaced. The health status is unavailable:

Data consistency check and recovery:

1.1 Ensure that the directory (as described below) is used exclusively for Kudu. If the directory is used for any other purpose, move the relevant data to another directory that is not configured as an `fs_data_dirs` directory before proceeding with the following steps.

Specific directory: View the `/usr/local/service/kudu/conf/tserver.gflags` file:

```
--fs_data_dirs=/data/emr/kudu/tserver,/data1/emr/kudu/tserver
--webserver_doc_root=/usr/local/service/kudu/www
--fs_wal_dir=/data/emr/kudu/tserver/wal
--raft_get_node_instance_timeout_ms=300000
--log_dir=/data/emr/kudu/log
--fs_metadata_dir=/data/emr/kudu/tserver/meta
```

1.2 Log in to the abnormal node of the local disk to view logs: `/data/emr/kudu/log/kudu-tserver.INFO` :

```
I0110 14:16:50.918087 96777 minidump.cc:244] Setting minidump size limit to 20M
I0110 14:16:50.918205 96777 env_posix.cc:2179] Not raising this process' open files per process limit of 1000000; it is already as high as it can go
I0110 14:16:50.918251 96777 file_cache.cc:492] Constructed file cache file cache with capacity 400000
I0110 14:16:50.918663 96777 hybrid_clock.cc:249] auto-selected time source: builtin
I0110 14:16:50.920179 96777 hybrid_clock.cc:583] waiting up to --ntp_initial_sync_wait_secs=60 seconds for the clock to synchronize
I0110 14:16:50.927374 96777 hybrid_clock.cc:601] HybridClock initialized: now 1673331410927647 us; error 18281 us; skew 500 ppm
I0110 14:16:50.927525 96777 webserver.cc:395] Webserver started at http://0.0.0.0:8050/ using document root /usr/local/service/kudu/www and password file <none>
I0110 14:16:50.927855 96777 server_base.cc:612] This appears to be a new deployment of Kudu; creating new FS layout
E0110 14:16:50.938101 96777 tablet_server_main.cc:40] Already present: RunTabletServer() failed: FS layout already exists; not overwriting existing layout: unable to create fi
eady exist: /data/emr/kudu/tserver,/data1/emr/kudu/tserver
```

Run the following commands as a root user to remove inconsistent data:
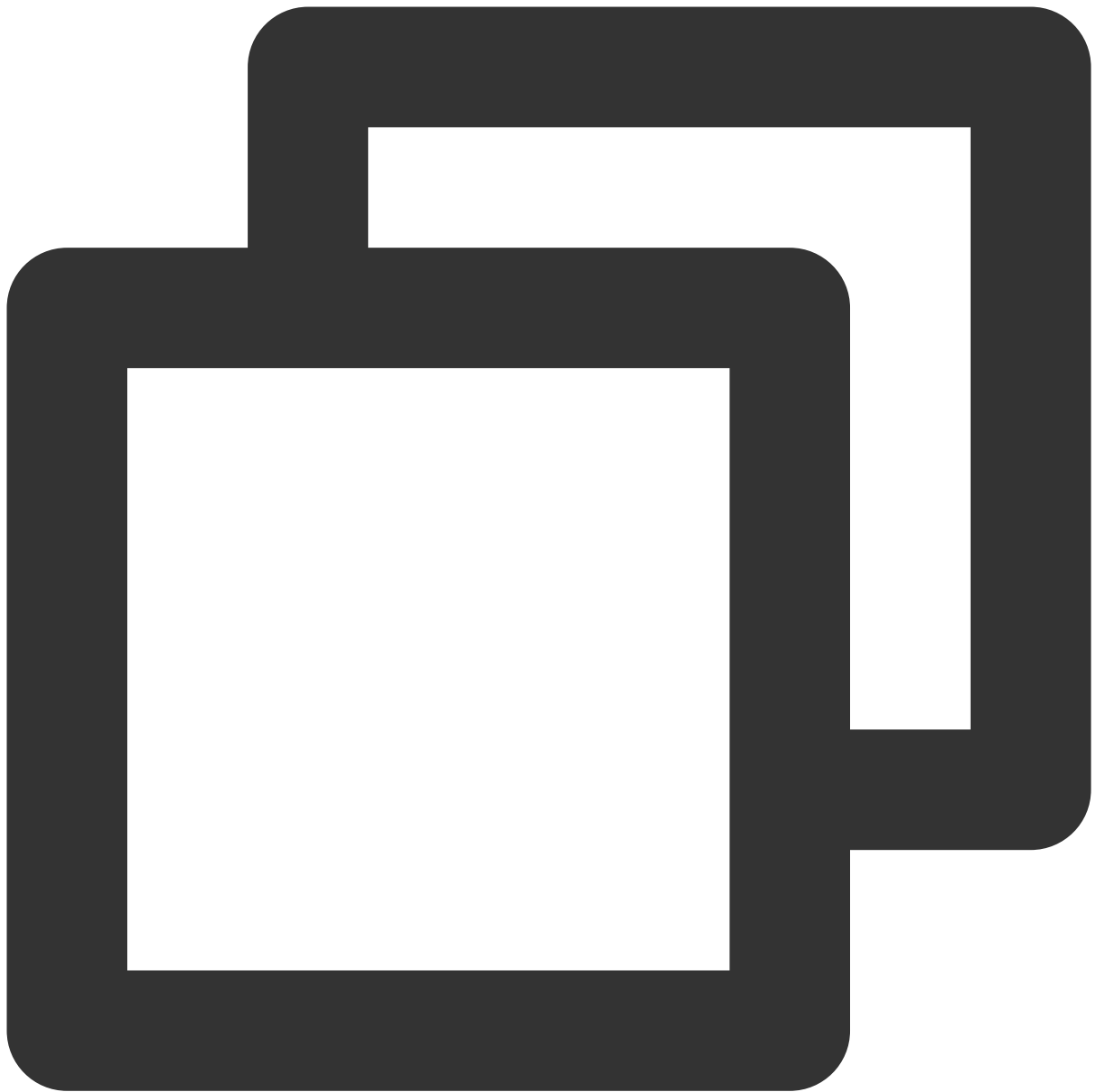
```
rm -rf /data/emr/kudu/tserver/*
rm -rf /data1/emr/kudu/tserver/*
```

The commands assume that `/data/emr/kudu/tserver/` and `/data1/emr/kudu/tserver/` are configured in `fs_data_dirs`. Fore more information, view `/usr/local/service/kudu/conf/tserver.gflags`.

1.3 Observe the service status of KuduServer in the console.

**Note**

If you encounter any issues, submit a ticket.

# Graceful Scale-In

Last updated：2023-12-27 11:06:02

## Overview

After graceful scale-in is enabled, if the scale-in action is triggered when a node is executing tasks, the node will not be released immediately. Instead, it will be released after completing the tasks. Graceful scale-in is available for both automatic scaling and manual scale-in.

**Note**

YARN, HBase, and Presto (renamed Trino in EMR v2.7.0 and v3.40 or later) support graceful scale-in. However, Presto cannot be scaled in gracefully in Ranger, Kerberos, and OpenLDAP integration scenarios.

## Directions

### Auto-scaling

Auto-scaling allows you to enable or disable graceful scale-in for all scale-in rules. This feature is called global graceful scale-in and is disabled by default. When you add or edit a single scale-in rule, graceful scale-in is enabled by default, and the default duration is 60 seconds (valid range: 60–1800 seconds).

**Note**

When global graceful scale-in and a single rule are enabled, graceful scale-in takes effect for the rule.

1. Log in to the EMR console, click the **ID**/**name** of the target cluster to enter the cluster details page, and click **Auto-scaling**.

2. In the **Scaling rule management** section on the **Auto-scaling** page, click **Add rule** and add a scale-in rule.

### Manual scale-in

When you try to manually remove a node, graceful scale-in is disabled by default. When you enable this feature, the default duration is 60 seconds (valid range: 60–1800 seconds).

1. Click the **ID**/**name** of the target cluster to enter the cluster details page. Then, select **Cluster resources** > **Resources**.

2. Select the node to remove and click **Scale in**. In this case, graceful scale-in is disabled by default. You can enable it and set a duration.

3. After finishing your settings, click **Next**, confirm the node information, and click **Start termination**.

# Disk Update Check

Last updated：2023-12-27 11:06:24

## Feature overview

This feature checks whether the cluster disk information is consistent with that displayed in the console and supports performing updates to ensure consistency.

## Directions

1. Log in to the EMR console and click the **ID/Name** of the target cluster in the **Cluster list** to enter the cluster details page.
2. On the cluster details page, select **Cluster Resource** to enter the **Resource Management** page. Select **More** > **Check disk update**. If the cluster disk information is inconsistent with the information displayed in the console, click **Update**, and the disk usage, total disk space, and disk space usage of the cluster will all be updated.
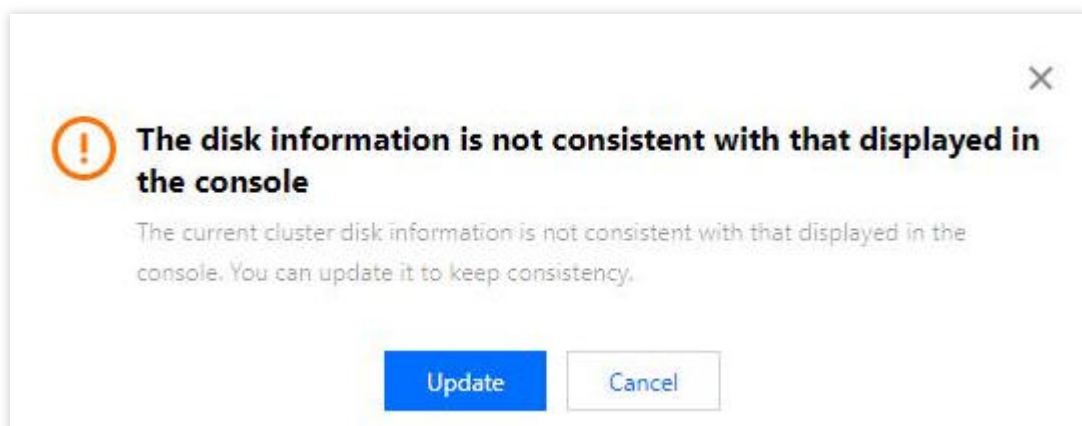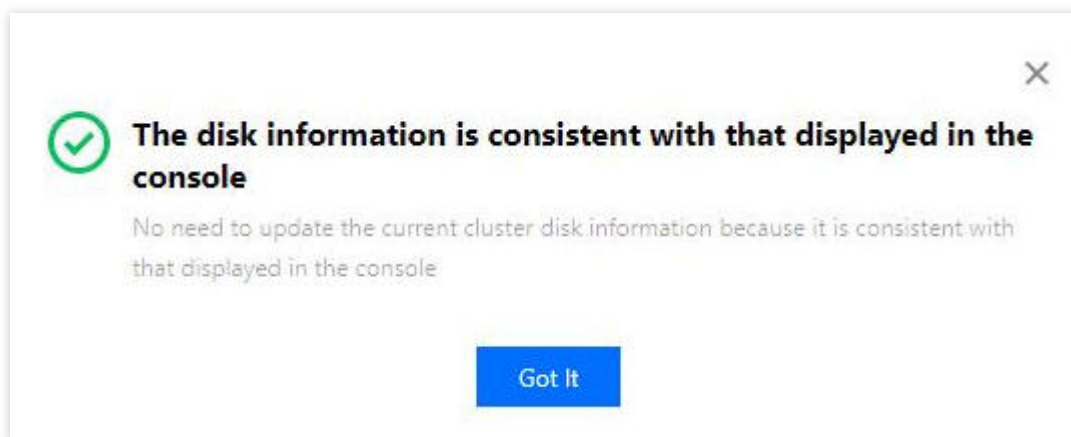
# Scaling up Cloud Disks

Last updated：2023-12-27 11:06:40

## Overview

If the data storage space of a node becomes insufficient as your business grows, you need to expand the space. This document describes how to scale up cloud disks in the EMR console.

**Caution**

You can scale up only cloud data disks but not system disks and local disks.

You cannot batch scale up cloud data disks on multiple nodes.

For your data security, we recommend you create snapshots of your cloud disks before scale-up.

To prevent data loss, a disk can be scaled up only but not scaled down.

## Scaling up Cloud Disks

**Directions**

1. Log in to the EMR console and click the **ID**/**Name** of a target cluster in the cluster list to go to the cluster details page.

2. Click the **Resource Management** tab and then click **Scale up cloud data disk** in the **Operation** column of the target node to enter the settings page.

3. If multiple cloud data disks are mounted to the current node, you can batch scale them up to the same capacity.



4. Disks will be initialized automatically after scale-up, and you don't need to manually update the disk information.

# Changing Configurations

Last updated：2023-12-27 11:06:56

## Overview

In actual use, you may find that the configurations of the nodes in your cluster need to be upgraded, especially when the CPU or memory resources of the master nodes are insufficient. This document describes how to change the instance configuration in the EMR console.

**Caution**

The node will be shut down during the change. Note that the shutdown may affect the normal use of the cluster and even interrupt your business. Proceed with caution.

The size of data disks and system disks cannot be changed.

## Prerequisites

1. After the configurations of pay-as-you-go nodes are changed, the billing tier will restart from the first tier. For monthly subscribed clusters, you need to make up the difference.

2. The configurations of local disk models, Pod resources, and spot instance models cannot be changed.

3. If you batch adjust configurations, the system will automatically deduct fees one by one. Make sure your account has sufficient balance.

4. The refund will be credited to your Tencent Cloud account at the ratio of cash to trial credit paid upon purchase, but the discount amount or voucher (if any) will not be refunded.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.

2. Select **Cluster Resource** on the cluster details page to enter the **Resource Management** page. Select the target node and **change the configuration**. Batch change is supported, but you can only change the configurations of nodes in the same billing mode to the same configuration.
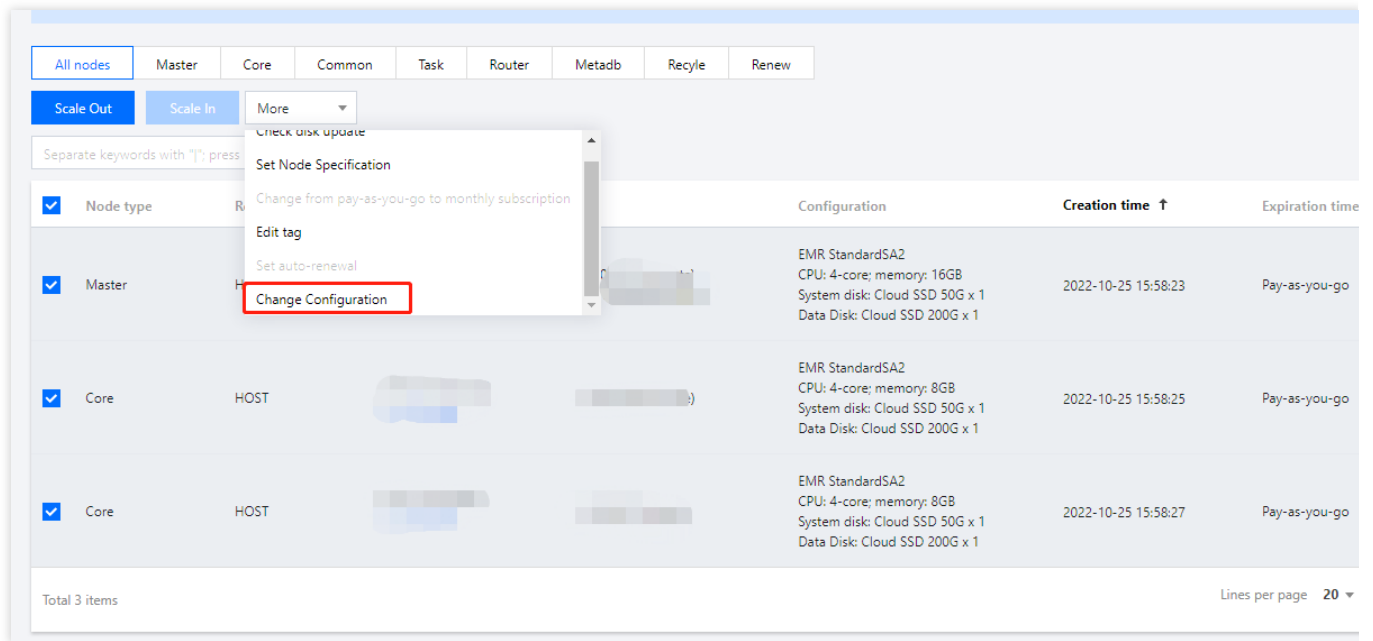
3. On the configuration adjustment page, confirm the relevant information, read carefully the notes, and select **I have read and understood the notes and agree to the operation**.

4. On the **Select Target Configuration** tab, select the model, instance type, model list, and other configuration items. After confirming the cost, click **Start Adjustment**.

5. The fees incurred by adjusting the configurations of different nodes to the same configuration will be displayed on the billing details page.

6. (Optional) To adjust the resources of the component after configuration adjustment, you need to deliver the configuration again in **Configuration Management** and restart the service.

**Caution**

The YARN resources will be automatically adjusted according to the model and specification by default. After configuration adjustment, the size of the resources will change as the specification changes and does not need to be adjusted manually.

If you have manually adjusted the configuration of YARN resources, then after configuration adjustment, you need to modify the parameter values of the `yarn.nodemanager.resource.cpu-vcores` and `yarn.nodemanager.resource.memory-mb` configuration items in **Configuration Management**, click **Save configuration** to deliver the configuration, and restart the NodeManager service for the configuration of YARN resources to be updated.

# Automatic Replacement

Last updated：2023-12-27 11:07:18

## Overview

The system continuously monitors task nodes and router nodes in a cluster for any abnormal running status. If any exceptions are detected, the system automatically purchases nodes of the same model to replace the affected nodes. Meanwhile, alarms are sent to notify users of the replacement result.

**Caution**

When "Instance Termination Protection" is enabled for nodes, this action cannot be triggered.

## Replacement Principle

1. Only pay-as-you-go and spot instance nodes support automatic replacement.
2. The system purchases only nodes of the same model in the same billing mode as abnormal nodes for automatic replacement.
3. When automatic replacement is not enabled, you can still receive alarms in the event of any node abnormalities.
4. Automatic replacement and exception monitoring are not supported for Pod nodes.

## Directions

1. Log in to the EMR console and click **Cluster List** in the left sidebar. On the page that appears, click the **ID/Name** of the cluster for which you want to enable automatic replacement.
2. Choose **Instance Information** > **Basic Configuration** and enable **Automatic Replacement**.
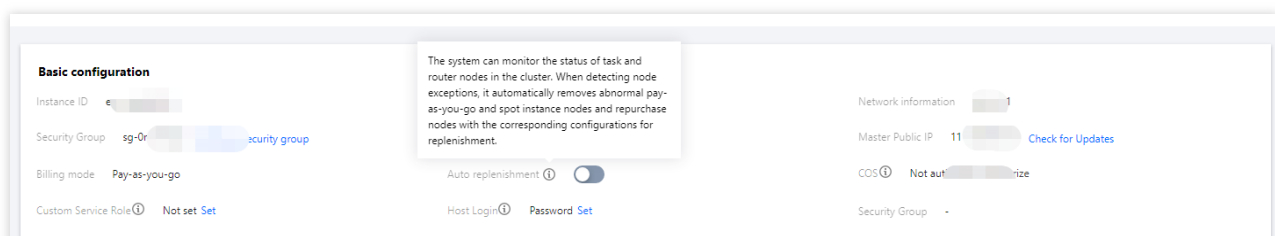
# Exporting Software Configuration

Last updated：2023-12-27 11:07:38

## Overview

You can export the software configuration parameters of existing clusters in the EMR console, so that you can reuse these parameters for software configuration to quickly create a cluster in the future.

**Note**

You can export configuration files at the cluster but not configuration group or node level.

To reuse the configuration of an existing cluster for a new cluster, we recommend you export only the modified configuration files of the existing cluster but not all the configuration files.

## Directions

1. Log in to the EMR console and go to the cluster list page.

2. In the **Action** column of the target cluster, select **More** > **Export Software Configuration**.
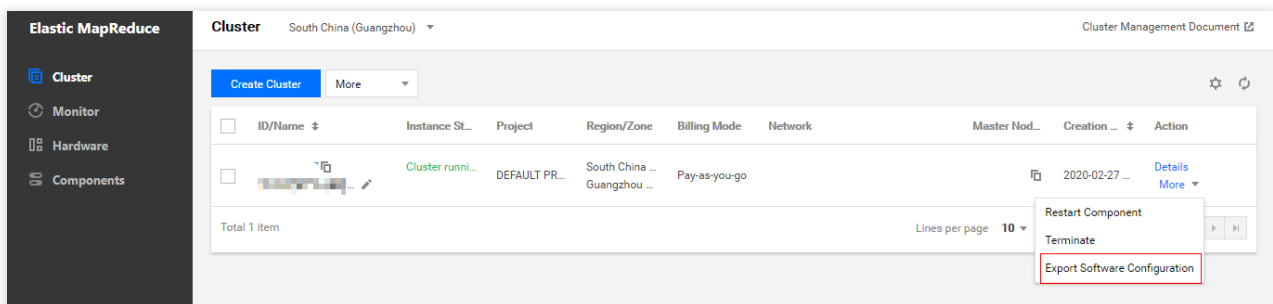


3. Select the files to be exported and click **Export Configuration** to download the software configuration files.

**Export Software Configuration**                                    ×

Components Involved in Cluster Software Configuration Parameters: HDFS, YARN
**Select configuration files:**

☐ Select All

☐ HDFS

☐ core-site.xml

☐ hadoop-env.sh

☐ hdfs-site.xml

☐ log4j.properties

☐ YARN

**Export Configuration**    Cancel
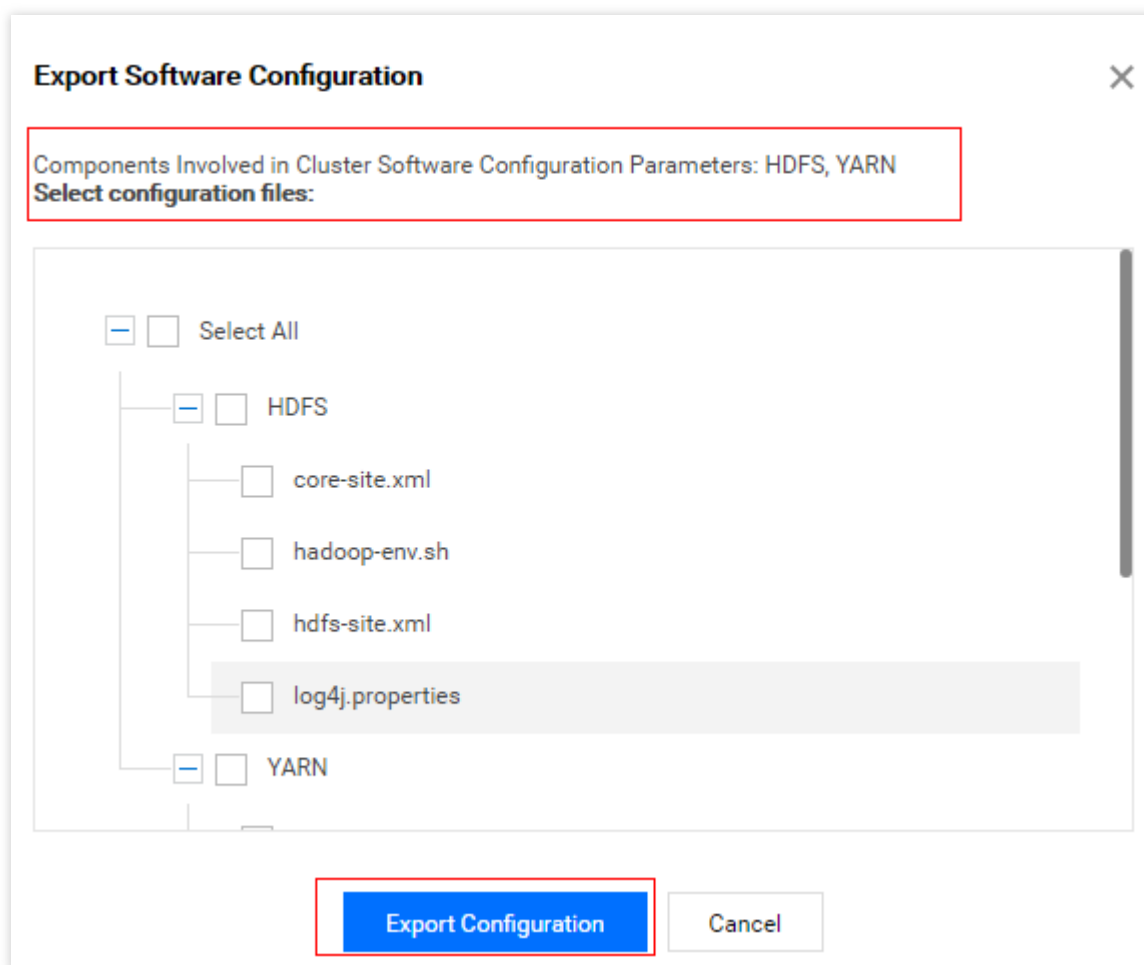
# Cluster Scripts

Last updated：2023-12-27 11:07:55

## Feature Overview

The cluster script feature allows you to run a script on multiple nodes at a time for higher efficiency. Only one cluster script can be run in a cluster at a time. If a cluster script is running (such as installing a third-party software or modifying the running environment of the cluster), you cannot submit and execute new cluster scripts.

**Caution**

Only files in COS can be selected as the script to run.

Only nodes in the current cluster can be selected to run the script on.

You can set custom parameters based on your business requirements.

The EMR console does not verify the scripts you run. Therefore, running a custom script is risky, so please operate with caution.

## Prerequisites

The cluster script feature is only available for clusters that are running.

The script to be run must be a shell script file in the COS STANDARD storage class.

To use the cluster script feature, you must grant EMR the permission to access COS.

## Directions

1. Log in to the EMR console and click a cluster ID in the cluster list to go the instance information page. Click **Cluster Scripts** > **Set Task**.
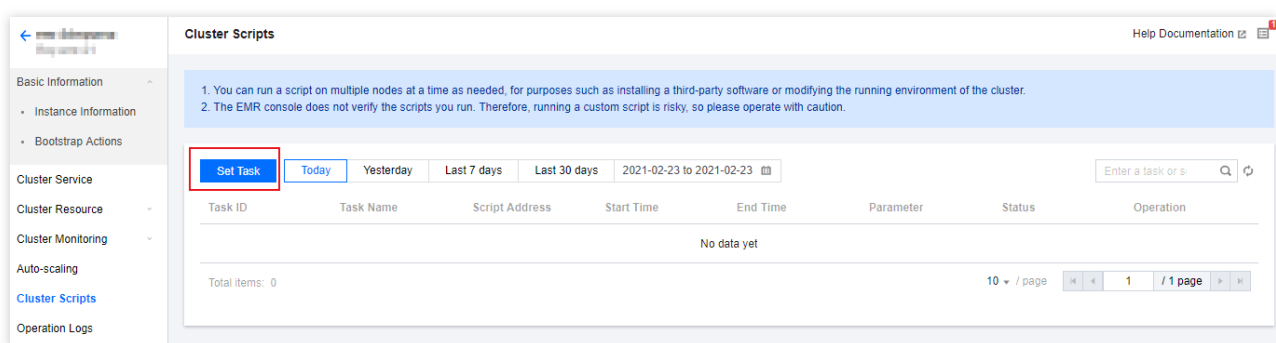
2. Set the **Task Name**, **Script**, **Nodes**, and **Custom Parameter** fields. After that, click **Run** to generate a task in the task list.



3. There are various task statuses depending on the execution results, including successful, failed, partially failed, etc. You can click **Details** to view details.

4. A cluster script may run successfully on some nodes and fail on other nodes. You can batch copy the failed nodes to run the script again.

# Cluster Termination

Last updated：2023-12-27 11:08:14

## Feature Overview

When you no longer need an EMR cluster, you can terminate it in the EMR console.

## Prerequisites

Pay-as-you-go cluster: once terminated, the cluster will not be retained in the recycle bin but will be completely terminated and cannot be recovered. Please do so with caution.

Before terminating a cluster, please make sure that your data has been backed up as it cannot be recovered after termination.

If there is an EIP (an IP with a secondary ENI), the instance will be retained once returned, and the idle IP will continue to incur fees. If you don't need to retain it, please release it on the corresponding resource management page.

## Directions

**Caution**

If MetaDB in the cluster to be terminated is associated with an external cluster as a Hive metadatabase, it will be retained when the cluster is terminated; if you no longer need it, please go to the TencentDB console to terminate it. Hive metadatabases cannot be recovered once terminated. Please do so with caution.

Log in to the EMR console, select **Action** > **More** > **Terminate** to enter the cluster termination page. Confirm the information of the cluster that needs to be terminated. After confirming that everything is correct, check **I have read and agree to** and click **Next**.
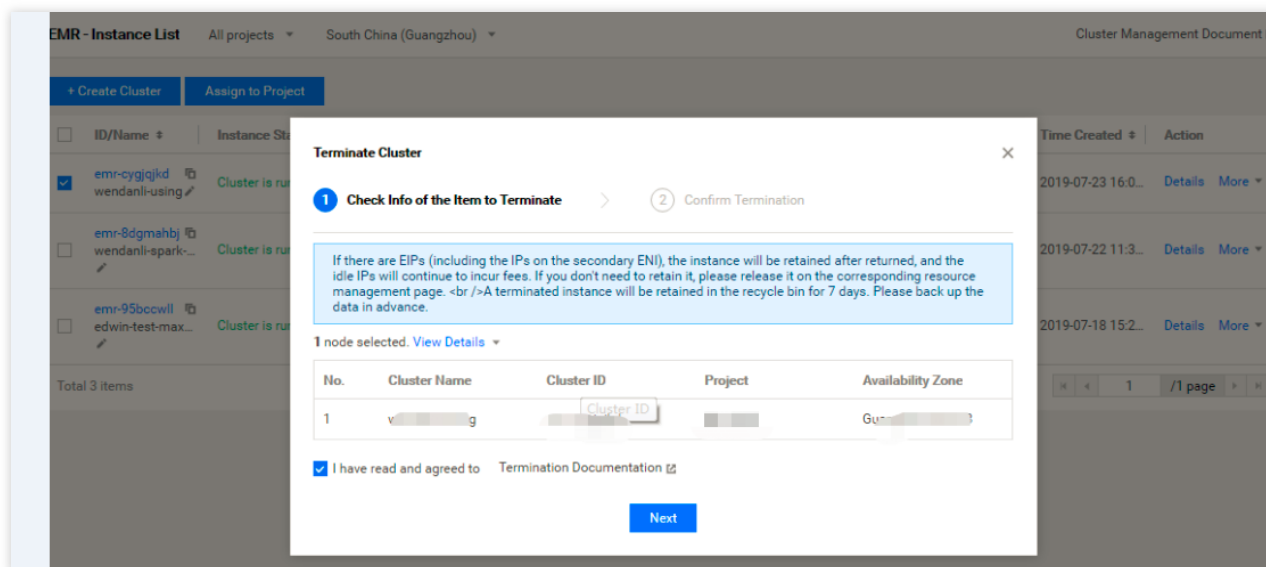
After confirming that everything is correct on the **Confirm Termination** tab, click **Start Termination** to terminate the cluster.
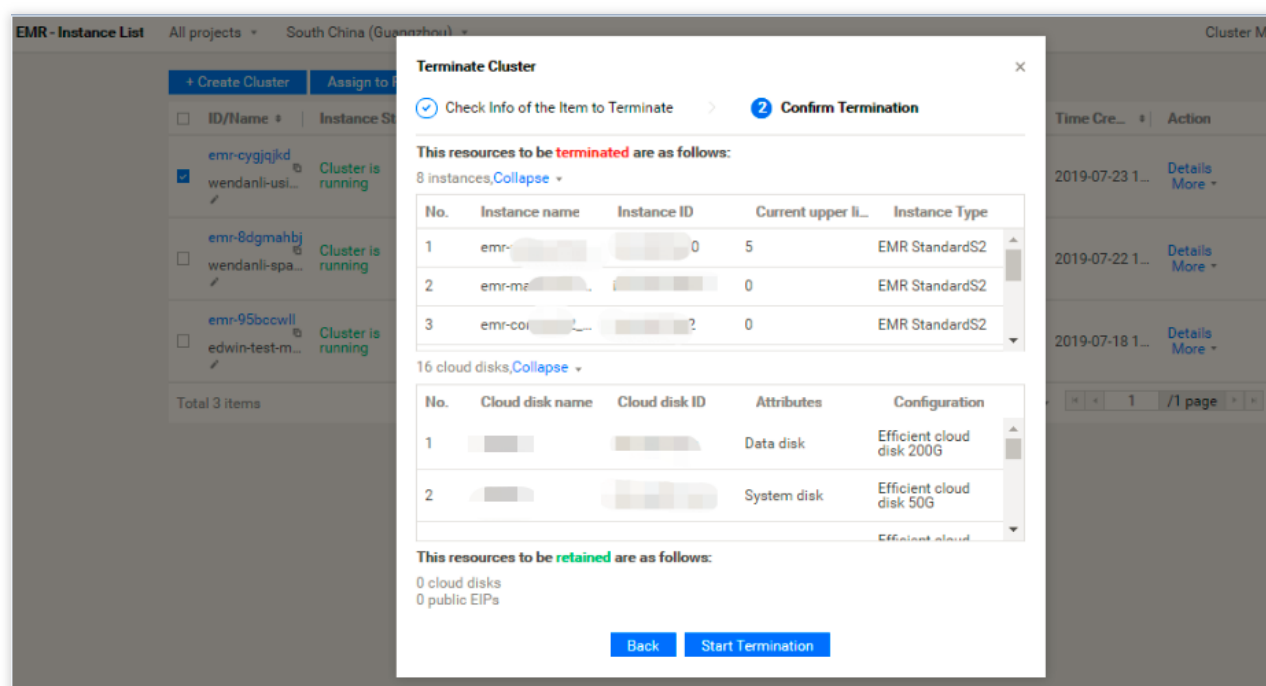
# Operation Logs

Last updated：2023-12-27 11:08:30

## Feature Description

Operation logs record the operations you performed on clusters in the EMR console, such as creating and scaling clusters.

## Directions

After successfully creating a cluster, log in to the EMR console, **select a region in the top-left corner of the cluster list**, click the **ID/Name** f the target cluster to enter the cluster details page, and click **Operation Logs** on the left.
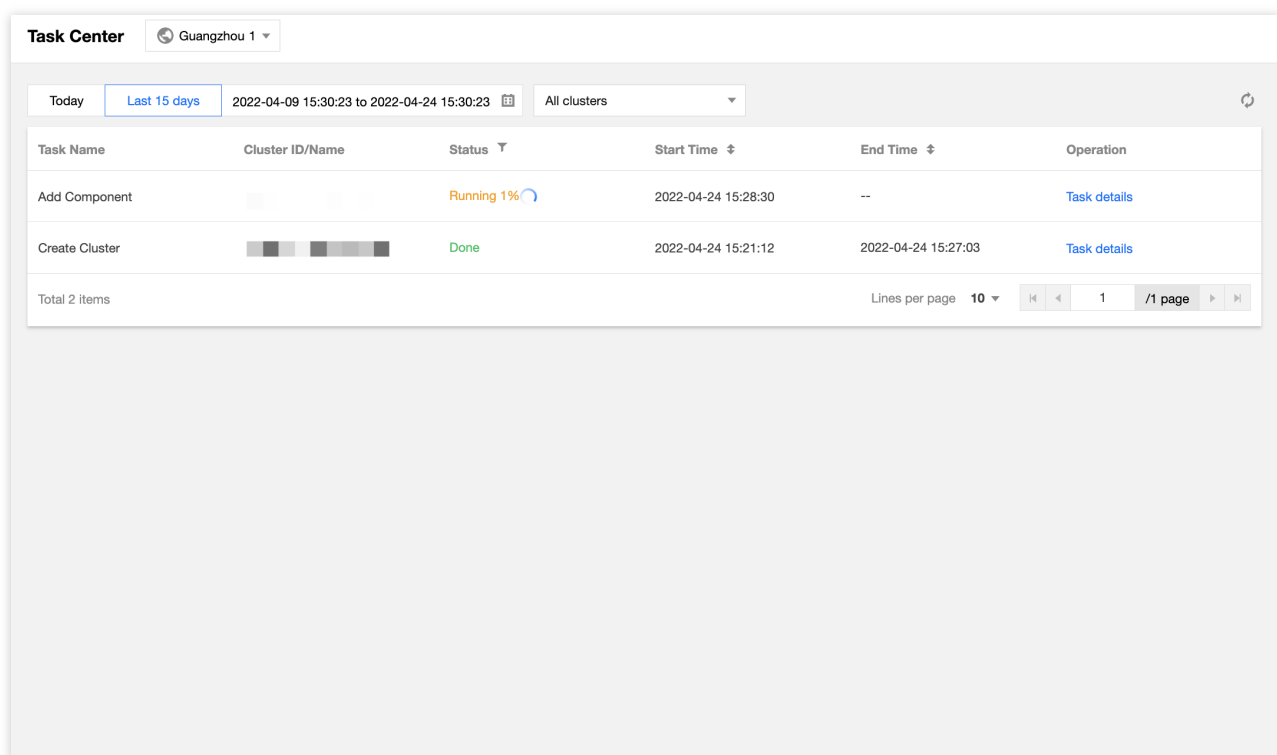
# Task Center

Last updated：2023-12-27 11:08:48

EMR has a **Task Center** where you can view the execution status of all tasks in your clusters and retry, cancel, and display error information for tasks.

# Task List

Log in to the EMR console and click **Task Center** on the left sidebar to view the status, start time, end time, and operation information of all tasks in one or all clusters in the current region.
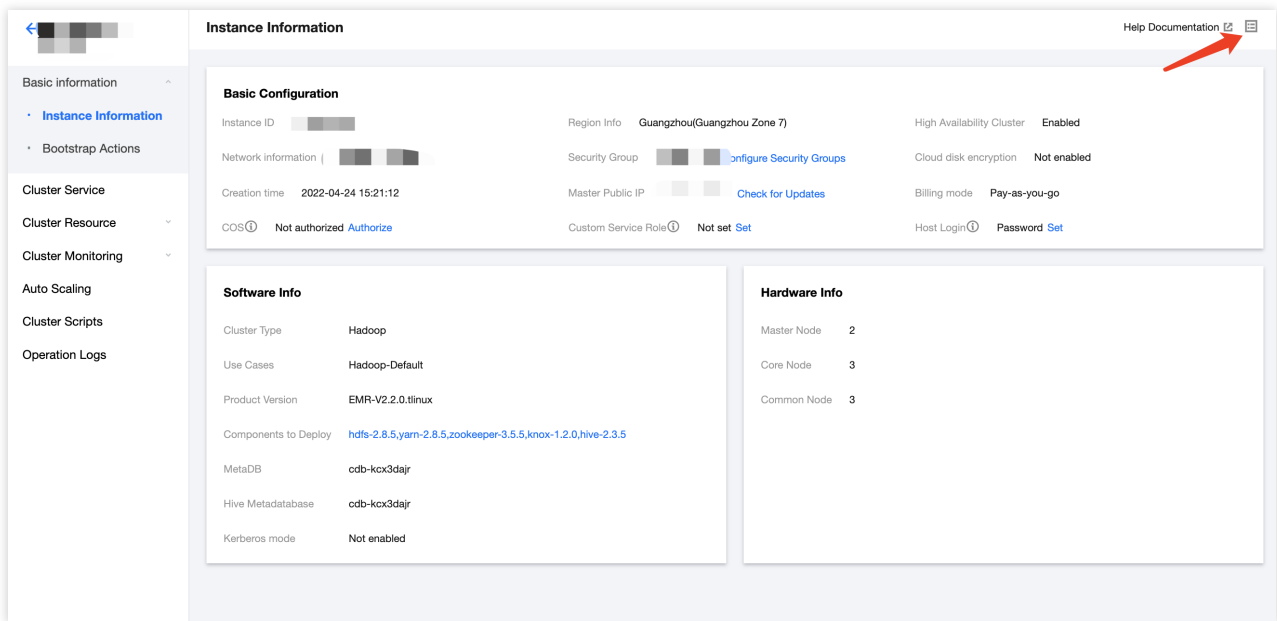


Log in to the EMR console, click **Cluster List** on the left sidebar, select a cluster, and click **Task** in the top-right corner to view the status, start time, end time, and operation information of all tasks in the cluster.

# Task Details

Click **Task Details** to view task parameters and completion status of execution steps.

**Task details** ✕

**Task Parameters**

CoreSize    1                                      DeployProcess    DataNode,NodeManager,Filebeat

Node Type    Core

| No. | Step Name | Status | Start Time | End Time | Operation |
|-----|-----------|--------|------------|----------|-----------|
| 1 | Create Order | Done | 2022-04-24 15:36:17 | 2022-04-24 15:36:18 | - |
| 2 | Apply For Resource | Done | 2022-04-24 15:36:19 | 2022-04-24 15:36:37 | - |
| 3 | Bind tag | Done | 2022-04-24 15:36:38 | 2022-04-24 15:36:39 | - |
| 4 | Initialize security group | Done | 2022-04-24 15:36:39 | 2022-04-24 15:36:41 | - |
| 5 | Initialize resource | Done | 2022-04-24 15:36:41 | 2022-04-24 15:36:42 | - |
| 6 | Prepare for scale-out | Done | 2022-04-24 15:36:42 | 2022-04-24 15:36:42 | - |
| ▸ 7 | Initialize service | Done | 2022-04-24 15:36:43 | 2022-04-24 15:40:01 | Run Details |
| 8 | End task | Done | 2022-04-24 15:40:03 | 2022-04-24 15:40:03 | - |

Confirm    Cancel

# Run Details

In some operation steps in **Task Details**, you can click **Run Details** to view the completion status of the corresponding steps.

← Task details (Scale Out Core)

**Task Parameters**

CoreSize 1                    DeployProcess   DataNode,NodeManager,Filebeat          Node Type   Core

| No. | Step Name | Status | Start Time | End Time | Operation |
|---|---|---|---|---|---|
| 1 | Create Order | Done | 2022-04-24 15:36:17 | 2022-04-24 15:36:18 | - |
| 2 | Apply For Resource | Done | 2022-04-24 15:36:19 | 2022-04-24 15:36:37 | - |
| 3 | Bind tag | Done | 2022-04-24 15:36:38 | 2022-04-24 15:36:39 | - |
| 4 | Initialize security group | Done | 2022-04-24 15:36:39 | 2022-04-24 15:36:41 | - |
| 5 | Initialize resource | Done | 2022-04-24 15:36:41 | 2022-04-24 15:36:42 | - |
| 6 | Prepare for scale-out | Done | 2022-04-24 15:36:42 | 2022-04-24 15:36:42 | - |
| ▾ 7 | Initialize service | Done | 2022-04-24 15:36:43 | 2022-04-24 15:40:01 | See Less |

| Node's IP | State ▼ | Start Time | End Time | ErrorDetails |
|---|---|---|---|---|
| | | No data yet | | |

| 8 | End task | Done | 2022-04-24 15:40:03 | 2022-04-24 15:40:03 | - |

---

**Task details**                                                                          ✕

**Task Parameters**

CoreSize    1                              DeployProcess    DataNode,NodeManager,Filebeat

Node Type    Core

| No. | Step Name | Status | Start Time | End Time | Operation |
|---|---|---|---|---|---|
| 1 | Create Order | Done | 2022-04-24 15:36:17 | 2022-04-24 15:36:18 | - |
| 2 | Apply For Resource | Done | 2022-04-24 15:36:19 | 2022-04-24 15:36:37 | - |
| 3 | Bind tag | Done | 2022-04-24 15:36:38 | 2022-04-24 15:36:39 | - |
| 4 | Initialize security group | Done | 2022-04-24 15:36:39 | 2022-04-24 15:36:41 | - |
| 5 | Initialize resource | Done | 2022-04-24 15:36:41 | 2022-04-24 15:36:42 | - |
| 6 | Prepare for scale-out | Done | 2022-04-24 15:36:42 | 2022-04-24 15:36:42 | - |
| ▾ 7 | Initialize service | Done | 2022-04-24 15:36:43 | 2022-04-24 15:40:01 | See Less |

| Node's IP | State ▼ | Start Time | End Time | ErrorDetails |
|---|---|---|---|---|
| | | No data yet | | |

Confirm        Cancel

# Managing Service
# Managing Users

Last updated：2023-12-27 11:09:05

## Overview

This feature allows you to manage (add, edit, or delete) users in an EMR cluster easily. New users can submit Hadoop cluster tasks.

**Note:**

User management is available only for EMR-v2.6.0 and EMR-v3.2.1 and above.

You need to manually trigger the delivery of configurations in `ranger-ugsync-site.xml` and restart the EnableUnixAuth service to sync user configurations.

Deleting users and resetting passwords may cause a running task to fail.

## Directions

1. Log in to the EMR console and click the **ID/Name** of a cluster in the cluster list to enter the cluster details page.
2. Click **Users**. On the displayed page, you can add and delete users in batches, reset passwords, and download keytabs.

3. Click **Create user** to create a user. **Username**, **User group**, and **Password** are required and **Remarks** is optional.



4. Manually trigger the delivery of configuration for ranger-ugsync-site.xml and restart the service.

Log in to the EMR console, select the target cluster in the cluster list, and click *Cluster services* to enter the cluster service list.

Select **Operation > Configurations** in the upper-right corner of the Ranger service panel.

On the **Configurations** page, select `ranger-ugsync-site.xml` , click **Edit configuration** and make no modification. Then, click **Save configuration** below, and select **Save and deliver** in the pop-up window.

Return to the **Roles** page, select the EnableUnixAuth service, and click **Restart service**.

5. Reset a password.

On the user management page, find the user whose password needs to be changed and click **Reset password** on the right. Enter the new password and confirm, and click **Confirm**.

6. Download a keytab.

On the user management page, find the user for whom you want to download the keytab and click **Download keytab**.

The keytab can be used to log in to the cluster.



**Caution**

 Keytab downloading is available only for Kerberos clusters.

7. Delete a user.

On the user management page, find the user to be deleted and click **Delete** on the right. Then click **Delete**.

**Confirm deletion** ✕

⊘ Deleting a user may cause running tasks to fail. Please proceed with caution.

Are you sure you want to delete user test-11?

Confirm   Cancel
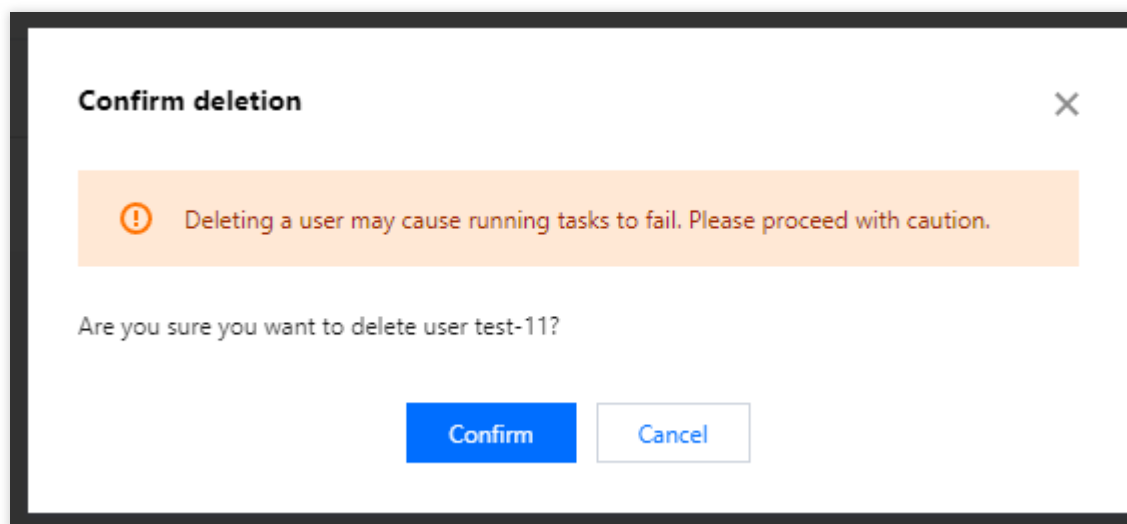
# Adding Components

Last updated：2023-12-27 11:09:22

## Feature Description

This document describes how to add components in the EMR console.

**Caution**

You need to manage the components through **Cluster Service**. Component changes made directly by logging in to a node (such as adding a component) will not be synced to the console and thus cannot be further managed. Components on a different EMR version or component version cannot be added. Instead, you can only add components on the current EMR version.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.

2. On the cluster details page, select **Cluster Service** > **Add Component** to add a component not installed in the cluster.

3. If there is no metadatabase in the cluster and Hue, Ranger, Oozie, Druid, and Superset components are to be installed, you need to purchase a TencentDB instance as the metadata store. You cannot add the Kudu component to a non-HA cluster.

4. There are two storage options for Hive metadata: you can store the metadata in the MetaDB (default option) or associate the metadata with EMR-MetaDB or a self-built MySQL database. In the latter case, metadata will be stored in the associated database and will not be deleted when the cluster is terminated.

## Add Component

The current cluster has no metadatabase. If you want to add a component such as Hive, Sqoop, Hue, Ranger, Oozie, and P purchase a new TencentDB instance to store metadata.

| Cluster Version | EMR-V3.0.0.tlinux | | | |
|---|---|---|---|---|
| Optional Components | ☑ zookeeper-3.4.9 | ☑ hadoop-3.1.2 | ☑ knox-1.2.0 | ☑ sy |
| | ☐ hbase-2.2.0 | ☑ hive-3.1.1 | ☑ hue-4.4.0 | ☐ oo |
| | ☐ presto-0.222 | ☐ ranger-2.0.0 | ☐ spark_hadoop3.1-2.4.3 ... | ☐ sc |
| | ☐ tez-0.9.2 | ☐ flume-1.9.0 | ☐ impala-2.10.0 | ☐ al |
| | ☐ flink-1.8.1 | | | |

| Hive Metadatabase | Default | Associate EMR-MetaDB | Associate self-built MYSQL |
|---|---|---|---|

Instance ID     Select data ▼

MetaDB ⑦     Pay-as-you-go 1 high IO TencentDB with 4,000 MB memory and 100 GB disk

Cost

Confirm     Cancel

5. The selection for MetaDB purchase and Hive metadatabase is the same as when you purchase a new cluster.

6. After selecting the component, click **OK**.

# Restarting Service

Last updated：2023-12-27 11:09:40

## Overview

After modifying the configuration of a component, you need to restart the service for the configuration to take effect. To minimize or avoid the impact of the service restart on your business, restart the service on a rolling basis. For instances with active and standby attributes, standby instances will be restarted first before active instances. Rolling restart takes longer than regular restart.

Services can be restarted in the console, and rolling restart is selected by default. Note that when rolling restart is disabled, restarting all nodes at the same time may make the service unavailable.

There are two policies for failure handling when node restart fails: waiting for processing in case of failure or continuing processing in case of single node failure.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.

2. To restart the entire component, select **Operation** > **Restart Service** on the target component block of the **Cluster Service** page, or go to the component details page and select **More** > **Restart All Services** in the top-right corner. To restart a role instance, select **Operation** > **Role Management** on the target component block of the **Cluster Service** page, select the target service role, and click **Restart Service**.

When restarting a service on the component block or component details page, you need to select the target service role, restart method, rolling restart switch status, and failure handling policy.

If you set **Service Role** to **All**, the entire component will be restarted.

**Restart Service**                                                                    ✕

Service name *          HDFS

Service Role *          | NameNode                          ▼ |

Restart Range          All eligible nodes (total 2)

Restart Method         | Quick restart mode                ▼ |

Rolling Restart        Each restart: | 1 |  nodes, interval: | 5 |  seconds

                       Up to 1 nodes can be restarted and the maximum interval is 5

                       minutes.

Failure-Handling Policy  | Wait for processing in case of failure          ▼ |

Operation Reason       |                                                   |
                       |                                                   |
                       |                                                   |

                       Enter no more than 200 characters

                       [ Confirm ]   [ Cancel ]

When restarting the service on the **Role Management** page, you only need to select the restart method and failure

handling policy.

**Restart Service**                                              ✕

Service name *          HDFS

Service Role *          NameNode            ▼

Restart Range          All eligible nodes (total 2)

Restart Method         Quick restart mode          ▼

Rolling Restart        Each restart:  1    nodes, interval:  5    seconds

                       Up to 1 nodes can be restarted and the maximum interval is 5
                       minutes.

Failure-Handling Policy    Wait for processing in case of failure          ▼

Operation Reason

                       Enter no more than 200 characters

                       Confirm        Cancel

3. The restart methods supported by service components are as follows:

| Component | Service | Restart Mode | Description | Remarks |
|---|---|---|---|---|
| HDFS | NameNode | Quick restart mode | It can be restarted by running hadoop-daemon.sh stop \| start namenode. | - |
| | NameNode | Safe restart mode | For an HA cluster, run `saveNameSpace` on the standby NameNode.Then, restart the component by | Only rolling restart is supported. |

| | | | running hadoop-daemon.sh stop \| start namenode. For a non-HA cluster, the quick restart mode is used. | |
|---|---|---|---|---|
| | DataNode | Default restart mode | It can be restarted by running hadoop-daemon.sh stop \| start datanode. | - |
| | JournalNode | Default restart mode | It can be restarted by running hadoop-daemon.sh stop \| start journalnode. | - |
| | zkfc | Default restart mode | It can be restarted by running hadoop-daemon.sh stop \| start zkfc. | - |
| YARN | ResourceManager | Default restart mode | It can be restarted by running sbin/yarn-daemon.sh stop \| start resourcemanager. | Only rolling restart is supported. |
| | NodeManager | Default restart mode | It can be restarted by running sbin/yarn-daemon.sh stop \| start nodemanager. | - |
| | JobHisotryServer | Default restart mode | It can be restarted by running sbin/yarn-daemon.sh stop\| start historyserver. | - |
| | TimeLineServer | Default restart mode | It can be restarted by running sbin/yarn-daemon.sh stop \| start timelineserver. | - |
| HBase | HbaseThrift | Default restart mode | It can be restarted by running hbase-daemon.sh stop \| start thrift. | - |
| | HMaster | Default restart | It can be restarted by | - |

| | | mode | running hbase-daemon.sh stop \| start master. | |
| | HRegionServer | Quick restart mode | It can be restarted by running hbase-daemon.sh stop \| start regionserver. | - |
| | HRegionServer | Safe restart mode | It can be restarted by running graceful_stop.sh --restart --reload. | - |
| Hive | HiveMetaStore | Default restart mode | It can be restarted by running hcat_server.sh stop \| strat. | - |
| | HiveServer2 | Default restart mode | It can be restarted by running hive-daemon.sh stop-h2 \| start-h2. | - |
| | HiveWebHcat | Default restart mode | It can be restarted by running webhcat_server.sh stop \| start. | - |
| Presto | PrestoCoordinator | Default restart mode | It can be restarted by running bin/launcher stop \| start. | Only rolling restart is supported. |
| | PrestoWorker | Default restart mode | It can be restarted by running bin/launcher stop \| start. | - |
| ZooKeeper | QuorumPeerMain | Default restart mode | It can be restarted by running bin/zkServer.sh stop \| start. | - |
| Spark | SparkJobHistoryServer | Default restart mode | It can be restarted by running sbin/stop-history-server.sh \| sbin/start-history-server.sh. | - |
| Hue | Hue | Default restart mode | It can be restarted by running build/env/bin/start.sh | - |

| | | | | |
|---|---|---|---|---|
| | | | and build/env/bin/stop.sh. | |
| Oozie | Oozie | Default restart mode | It can be restarted by running oozied.sh stop \| start. | - |
| Storm | Nimbus | Default restart mode | It can be restarted by running bin/storm-daemon.sh nimbus stop \| start. | - |
| | Supervisor | Default restart mode | It can be restarted by running bin/storm-daemon.sh supervisor stop \| start. | - |
| | LogViewer | Default restart mode | It can be restarted by running bin/storm-daemon.sh nimbus stop \| start. | - |
| | UI | Default restart mode | It can be restarted by running bin/storm-daemon.sh nimbus stop \| start. | - |
| Ranger | Ranger | Default restart mode | It can be restarted by running sbin/ranger-daemon.sh stop \| start. | - |
| Alluxio | AlluxioMaster | Default restart mode | It can be restarted by running bin/alluxio-stop.sh master and bin/alluxio-start.sh master. | - |
| | AlluxioWorker | Default restart mode | It can be restarted by running bin/alluxio-stop.sh worker and bin/alluxio-start.sh worker. | - |
| Ganglia | Httpd | Default restart mode | It can be restarted by running /sbin/service http stop \| start. | - |

| | Gmetad | Default restart mode | It can be restarted by running /sbin/service gmetad stop \| start. | - |
| --- | --- | --- | --- | --- |
| | Gmond | Default restart mode | It can be restarted by running /sbin/service gmon stop \| start. | - |

# Starting/Stopping Services

Last updated：2023-12-27 11:09:56

## Overview

The Start/Stop Services feature can stop or start all services on a node. This document describes how to use Start/Stop Services in the EMR console.

## Directions

1. Log in to the EMR console and click the **Cluster ID**/**Name** of the target cluster in the **Cluster list** to enter the cluster details page.
2. Select **Cluster Resource** and click **Resource Management**. Then, select the target node and click **More** > **Start/Stop Services**.
3. Check the service information in the **Start/Stop Services** window.
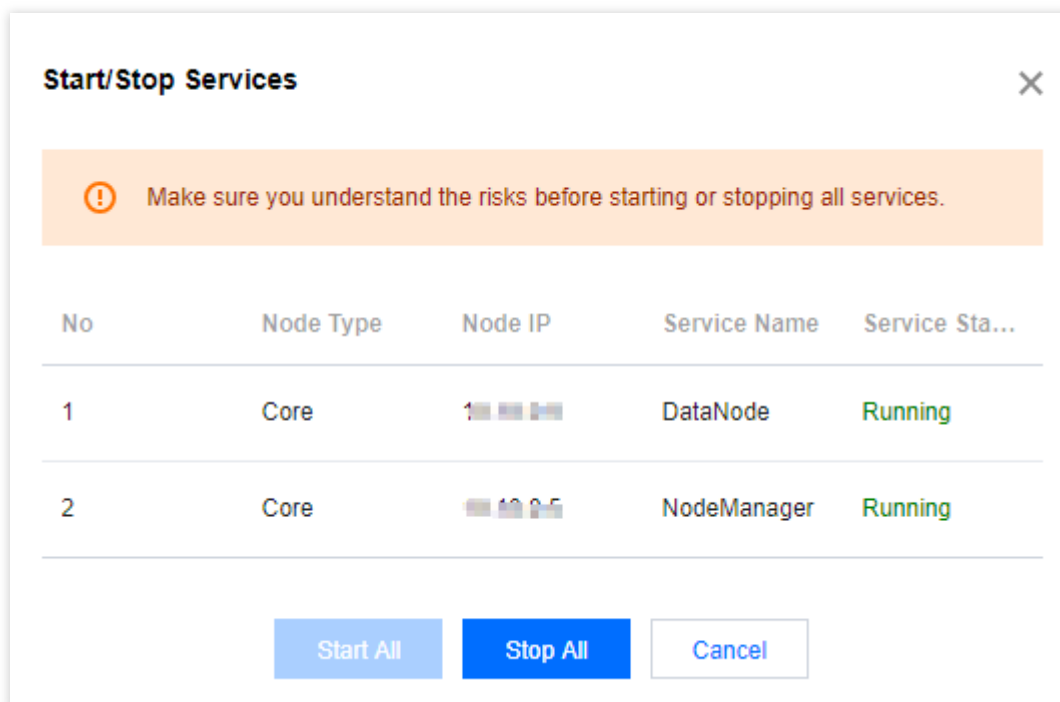


4. After confirming that the information is correct, click **Start All** or **Stop All**.

# WebUI Access

Last updated：2023-12-27 11:10:15

## Overview

The software WebUI entry feature is the accessibility to native component UIs provided by EMR. Such UIs can be quickly accessed through the public IP address of a master node (it is recommended to configure a security policy in a timely manner). If the private network of your cluster is interconnected with your organizational public network, you can disable this public IP address and access the UIs directly through the private network.

## Viewing WebUI Address

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. Click **Cluster Services** and then click **WebUI address** below a target component to access its WebUI.
The access address requires authentication. The username is "root", and the default password is the one entered when the cluster was created. To change the password, click **Reset WebUI password** on the page.

## WebUI Access over Private Network

If **Enable public network for master nodes** is not selected during the cluster creation, you cannot access the WebUIs of components via the WebUI addresses given on the **Cluster Services** page. You can access these WebUIs with a browser via the following links in a private network environment.

| Component | Link |
|-----------|------|
| HDFS UI | `http://{cluster IP in private network}:4008` |
| YARN UI | `http://{cluster IP in private network}:5004` |
| HBase UI | `http://{cluster IP in private network}:6001` |
| Hive UI | `http://{cluster IP in private network}:7003` |
| Hue UI | `http://{cluster IP in private network}:13000` |
| Ranger UI | `http://cluster IP in private network}:6080` |
|  |  |

| Storm UI | `http://{cluster IP in private network}:15001` |
| --- | --- |
| Oozie UI | `http://{cluster IP in private network}:12000` |
| Ganglia UI | `http://{cluster IP in private network}:1800` |
| Presto UI | `http://{cluster IP in private network}:9000` |
| Allexio UI | `http://{cluster IP in private network}:19999` |

To access the WebUI of a component over a public network after the cluster is created, you can set an EIP for the master node as instructed below:

1. Log in to the EMR console, click the **Cluster ID** of the target cluster to enter the cluster details page, then select **Cluster Resources > Resource Management**, and click the **Resource name/ID** of the target master node to go to the CVM console.
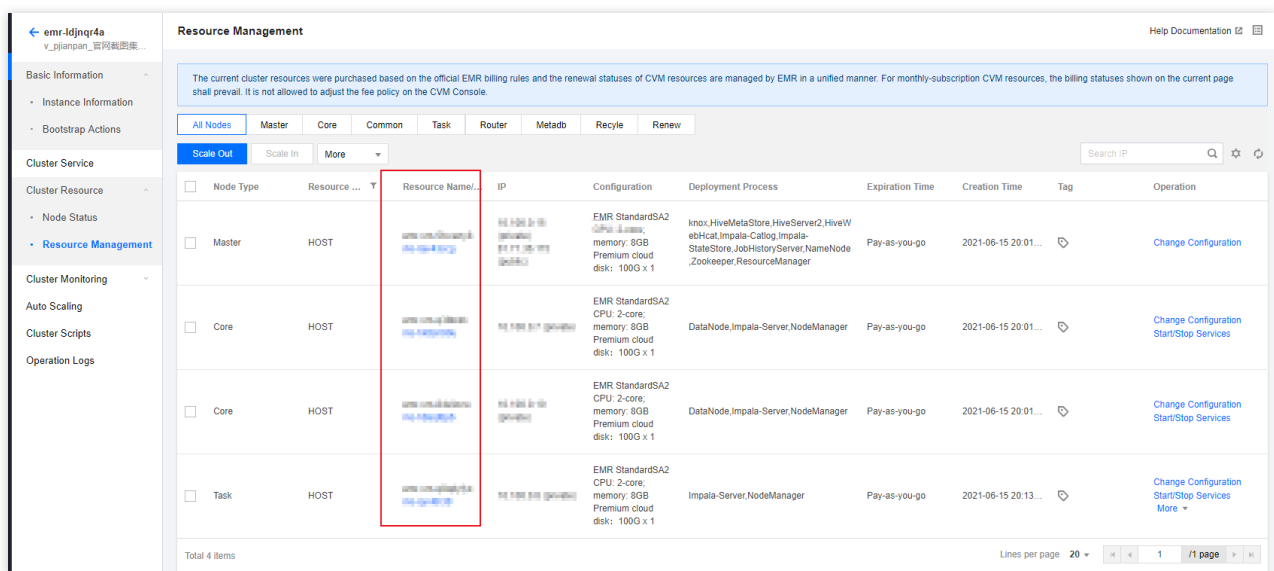


2. Change the network bandwidth setting of the CVM instance to be bound with an EIP, so that its bandwidth is not 0 to ensure connection.

In the CVM console, select **More actions > Resource adjustment > Adjust network** against the target instance in the CVM instance list.
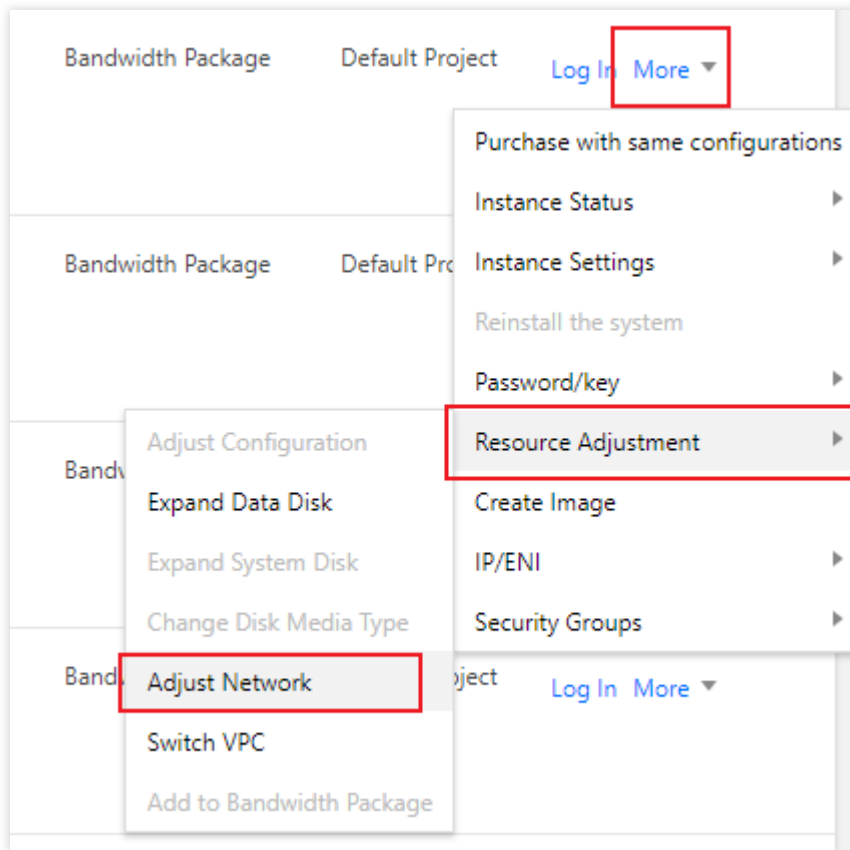
Set an appropriate target bandwidth cap to ensure the bandwidth of the CVM instance is greater than 0.

3. Click **Instance ID** of the CVM instance to enter its basic information page, and select the ENI tab.
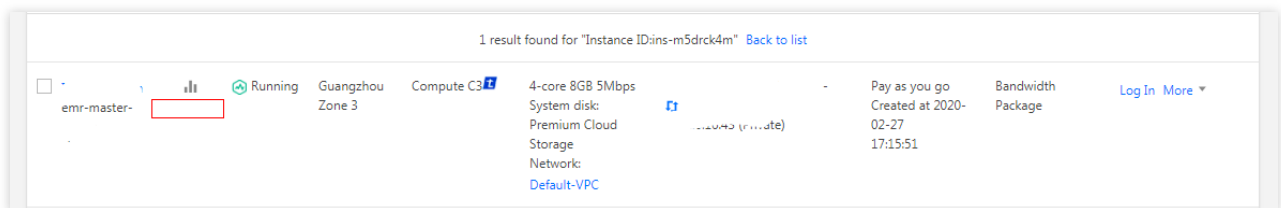


4. Click **Bind** to bind an existing EIP or create a new EIP for the CVM instance.
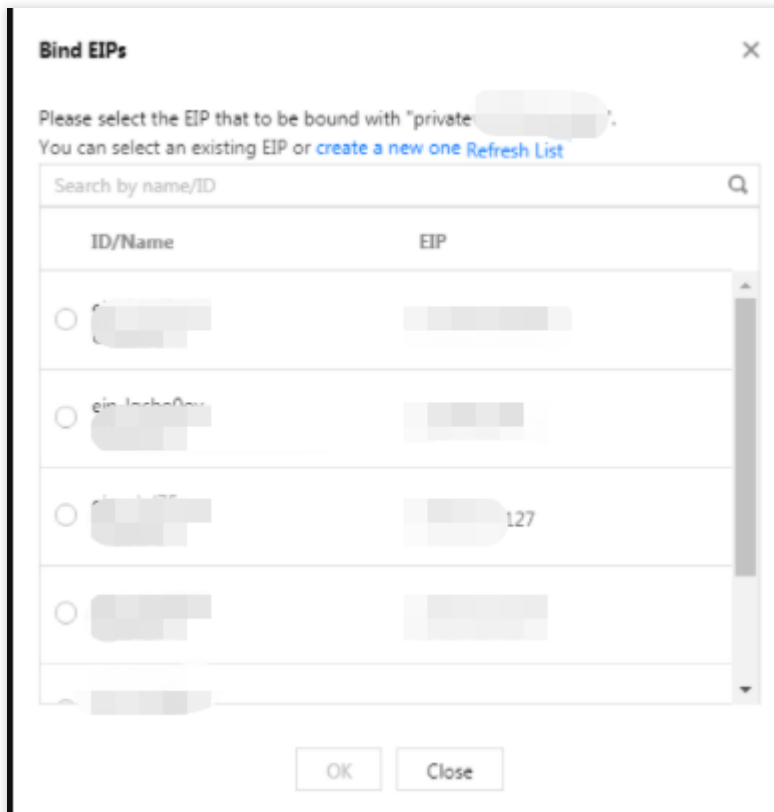
After the EIP is bound, you can see the EIP information beside the public network IP bound to the primary ENI on the ENI page.

5. Check the CVM access over public network.

6. Use the `ping` or `ssh` command to check whether the EIP is effective. Ensure the security group inbound rule is available to ICMP and port 22. Then, access WebUIs of components.

EMR v1.3.1, EMR v2.0.1, EMR v2.1.0, and EMR v3.0.0 now support Apache Knox. By default, you can access WebUIs of components over a public network with Knox as the proxy. For UI details of all components and Knox usage, see Knox Development Guide.

**Note**

After an EIP is bound, the WebUI addresses given in the EMR console will remain unchanged. To change these addresses, please contact us.

# Resetting WebUI Password

The access address requires authentication. The username is "root", and the default password is the one entered when the cluster was created. You can change the password as instructed below.

1. Log in to the EMR console and click the **ID/Name** of a target cluster in the cluster list to enter the cluster details page.

2. Click **Cluster Services** on the cluster details page and click **Reset WebUI password** in the top-left corner to reset the password.

**Caution**

The password must contain 8 to 16 characters in at least three of the following character types: uppercase letters, lowercase letters, digits, and special symbols (! @ # % ^ *). The first character cannot be a special symbol.

If OpenLDAP is installed in the cluster (EMR v2.6.0 and EMR v3.2.1 or later), you can change the password only on the **User Management** page. To reset the WebUI password, go to **User Management** and create a user.

# Resetting WebUI Password

Last updated：2023-12-27 11:10:59

## Overview

EMR provides a software WebUI entry to access the WebUIs of components. It is required to authenticate access requests. The username is "root", and the default password is the one you entered when creating the cluster. You can reset the WebUI password in the EMR console if you forgot it. This document describes how to do so.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. Click **Cluster Service** on the cluster details page and click **Reset WebUI Password** in the top-left corner to reset the password.

**Caution**

The password must contain 8 to 16 characters with at least three of the following character types: uppercase letters, lowercase letters, digits, and special symbols (! @ # % ^ *). The first character cannot be a special symbol.

If OpenLDAP is installed in the cluster (EMR v2.6.0 and EMR v3.2.1 or later), you can change the password only on the **User Management** page. To reset the WebUI password, go to **User Management** and create a user.

# Software WebUI Entry

Last updated：2023-12-27 11:11:14

## Feature Overview

EMR provides a software WebUI entry for you to access the native component UIs. To do so, you can use the public IP of the primary node (we recommend configuring the security policy promptly). If your cluster's private network and the organization network are connected, you can disable the public IP and directly access the UIs over the private network.

## Directions

1. Log in to the EMR Console, select **Cluster List**, and click the ID/Name of the target cluster to enter the cluster details page.
2. On the cluster details page, select **Cluster Service** and click the **WebUI address** of the target component for access.

The access address requires authentication. The username is "root", and the default password is the one you entered when creating the cluster. To change the password, click **Reset Native UI Password** on the page.

**Note:**

Access incurs fees that are billed by traffic.

# Operation Guide for Access to WebUI over Private Network

Last updated : 2023-12-27 11:11:33

When creating an EMR cluster, if you do not check **Enable Public Network Access for Cluster Master Node**, you cannot access the WebUIs of relevant components through the native WebUI access addresses on the component management page. This document describes how to view native component WebUIs in a cluster where the public network access to master node is not enabled.

## Private Network Access

You can use a browser to access a component's WebUI over the private network. Below are the links to native WebUIs of each component:

| Component | Link |
| --- | --- |
| HDFS UI | `http://{cluster private IP}:4008` |
| YARN UI | `http://{cluster private IP}:5004` |
| HBASE UI | `http://{cluster private IP}:6001` |
| HIVE UI | `http://{cluster private IP}:7003` |
| HUE UI | `http://{cluster private IP}:13000` |
| RANGER UI | `http://{cluster private IP}:6080` |
| STROM UI | `http://{cluster private IP}:15001` |
| OOZIE UI | `http://{cluster private IP}:12000` |
| GANGLIA UI | `http://{cluster private IP}:1800` |
| PRESTO UI | `http://{cluster private IP}:9000` |
| ALLUXIO UI | `http://{cluster private IP}:19999` |

## Binding an EIP

You can bind an elastic public IP (EIP) to the master node by following the steps below to access component WebUIs from a browser over the public network:

1. Log in to the EMR Console, select **Cluster List**, and click the **ID/Name** of the target cluster to enter the cluster details page. Select the primary node to which an EIP binds in **Cluster Resources** ->**Resource Management** and click its Instance ID/Resource Name to enter the CVM Console.



2. Adjust the network bandwidth settings of the CVM instance to which the EIP will be bound and make sure that its bandwidth is not 0; otherwise, the node cannot be connected.

In the CVM instance list in the CVM Console, select **More** > **Resource Adjustment** > **Adjust Network** for the target instance.
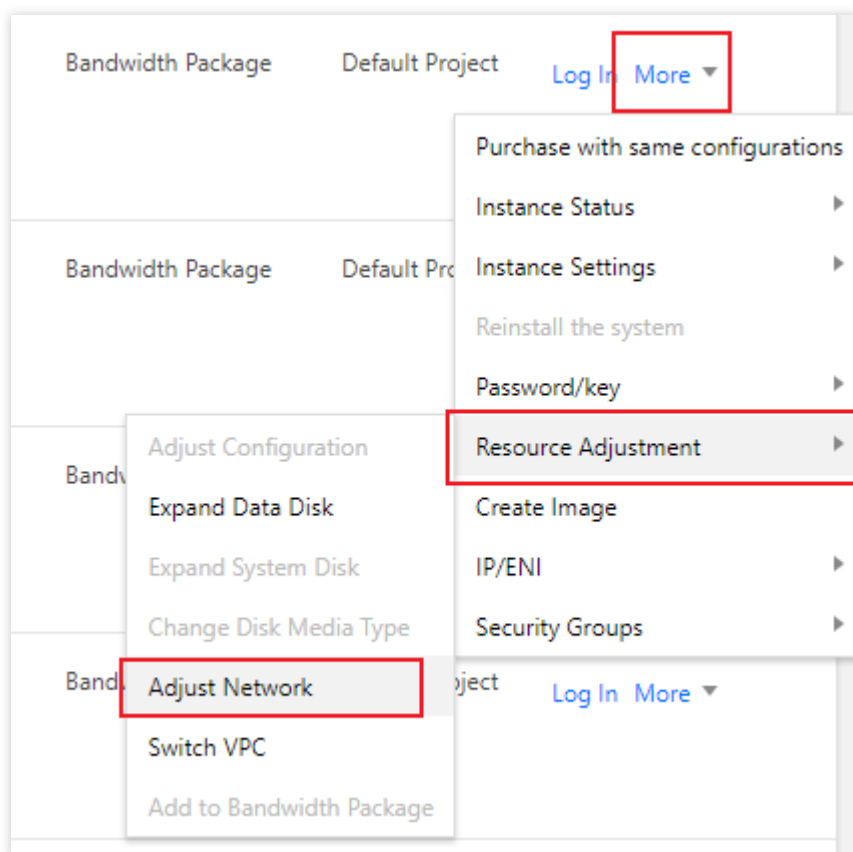
Adjust the bandwidth upper limit to an appropriate value to ensure that the bandwidth of the CVM instance is greater than 0.

3. Click the ID of the CVM instance to enter its basic information page and switch to the ENI tab.
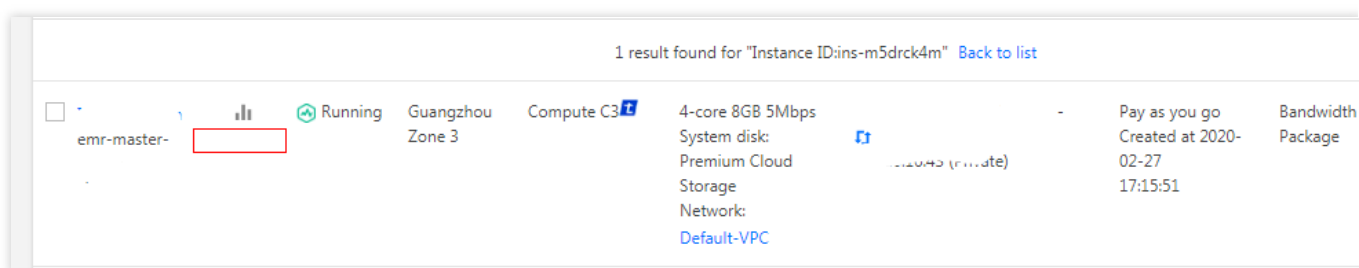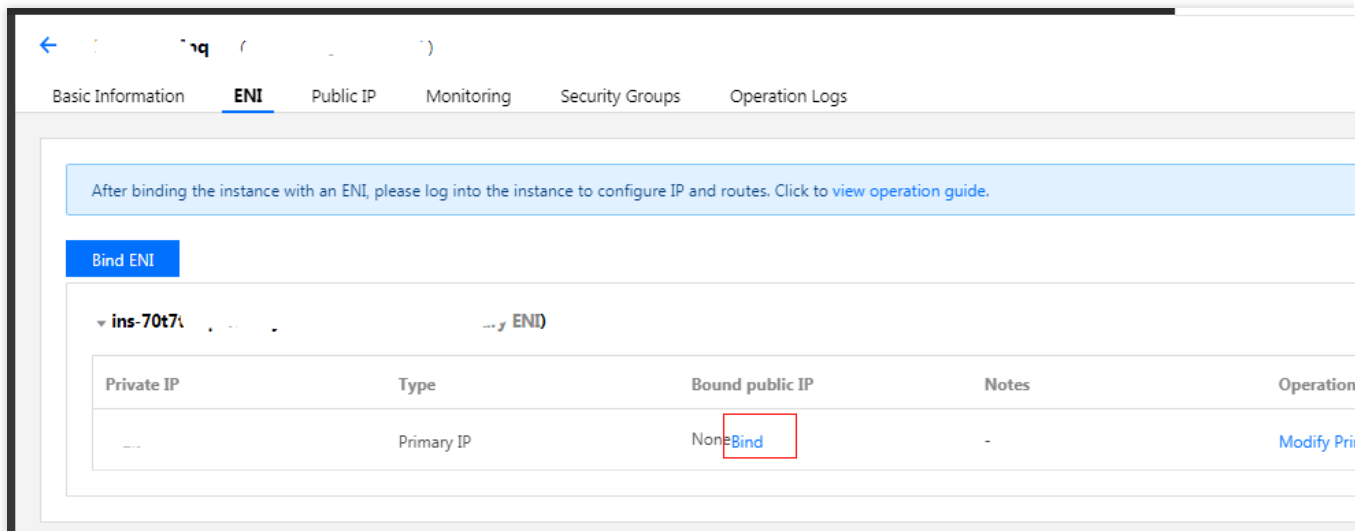
4. Click **Bind** to bind an existing EIP to the current CVM instance or create an EIP for binding.
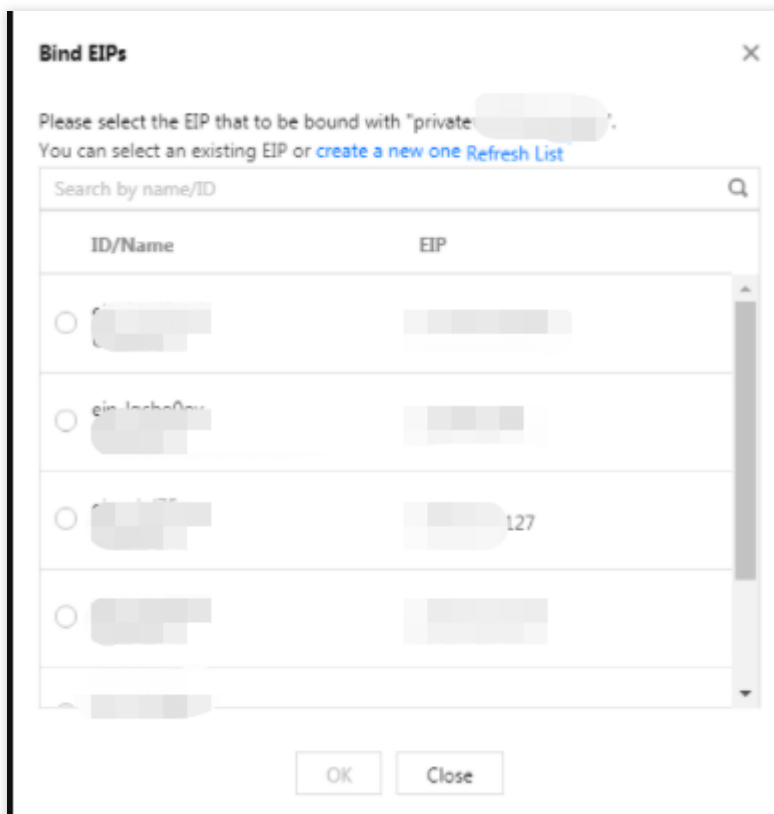


After binding the EIP, in the ENI tab, you can see that there is EIP information in the "Bound Public IP" section of the primary ENI.

5. Check whether the CVM instance can be accessed over the public network.

You can use the `ping` or `ssh` command to check whether the EIP has taken effect. **Make sure that the inbound rules of the security group allow ICMP and port 22**.

6. Access the native WebUIs of components.

EMR v1.3.1, v2.0.1, v2.1.0, and v3.00 support Apache Knox, and access requests to native component WebUIs over the public network will pass through Knox by default. For specific UI links to each component and Knox usage, please see Knox Development Guide.

**Note**

After an EIP is bound, the access addresses of native component WebUIs in the EMR Console will stay unchanged. If you need to change the addresses in the console, please submit a ticket for assistance.

# Role Management

Last updated：2023-12-27 11:12:20

## Overview

Role management provides Ops and operation features at the service role level. You can restart, pause, and maintain service roles at the node level. Role status can be monitored to help you stay on top of the real-time status of role processes.

**Glossary**

Restart: The nodes under the selected service role will be restarted on a rolling basis.
Pause: The nodes under the selected service role will be paused. They can be resumed by using the **Start** feature.
Maintenance: The process daemon will be stopped on the nodes under the selected service role, so when a process becomes abnormal for various reasons, no alarm or automatic recovery will occur. This is suitable for node debugging. You can use the **Exit Maintenance** feature to resume the daemon.

## Directions

1. Log in to the [EMR console](#) and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, select **Cluster Service** and click **Operation** > **Role Management** in the top-right corner of the target component block. The following uses HDFS as an example.
3. The **Role Management** list displays the current service role's health status, operation status, configuration group, node type, maintenance status, server IP, and last restarted time. You can select a role and **restart, maintain, start, or pause** it.
**Note**
 The **Health Status** column displays the running status of the current role. The **Operation Status** column displays user operations. The **Maintenance Status** column displays whether the role is under maintenance.

# Health Status

| Health Status | Description |
|---|---|
| Good | Port check is responded to within 5s. |
| Risky | Port check is responded to within 5s–10s. |
| Unavailable | Port check is not responded to within 10s. |
| Not checked | Roles in the maintenance mode or in Stopped operation status are not checked. |
| Unknown | The checker is abnormal or down. |

# Client Management

Last updated：2023-12-27 11:50:33

## Overview

Client management provides the information of the client deployed in the component, including the configuration status, configuration group, node type, and node IP.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, select **Cluster Service** and click **Operation** > **Client Management** in the top-right corner of the target component block to enter the **Client Management** page.

# Configuration Management
# Configuration Management

Last updated：2023-12-27 14:12:56

## Overview

Configuration management allows you to modify key configuration parameters of popular open-source components like HDFS, YARN, Hive, and Spark. The configuration of services can be modified as needed at the level of cluster, node, or configuration group. This document describes how to configure service parameters in the console.

**Caution**

For security reasons, when a custom configuration file is deleted in **Configuration Management** in the console, it will not be deleted from the client.

If you need to delete a custom configuration file from the client, you can use a cluster script to batch perform the operation on existing clusters.

After configuration modification, if the configuration status in **Role Management** changes to **Configuration expired**, you need to restart the service to make the configuration effective.

## Directions

### Editing a configuration item

1. Log in to the EMR console, select the target cluster in the cluster list, and click **Service** to enter the cluster service list.
2. In the cluster service list, select **Operation** > **Configuration Management** in the top-right corner of the target service panel.
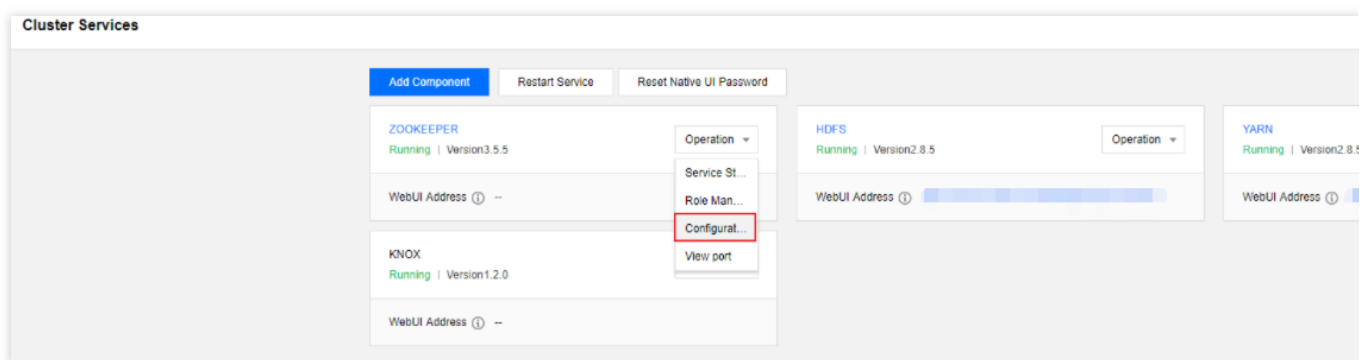


3. On the **Configuration Management** page, select the **Level** as needed. By default, **Cluster level** is selected.

To configure the parameters of all nodes of the selected service in the cluster, select **Cluster level**.

To configure the parameters of multiple specified nodes of the selected service, create a configuration group and use it as the level.

To configure the parameters of a specified node of the selected service, select **Node level**.
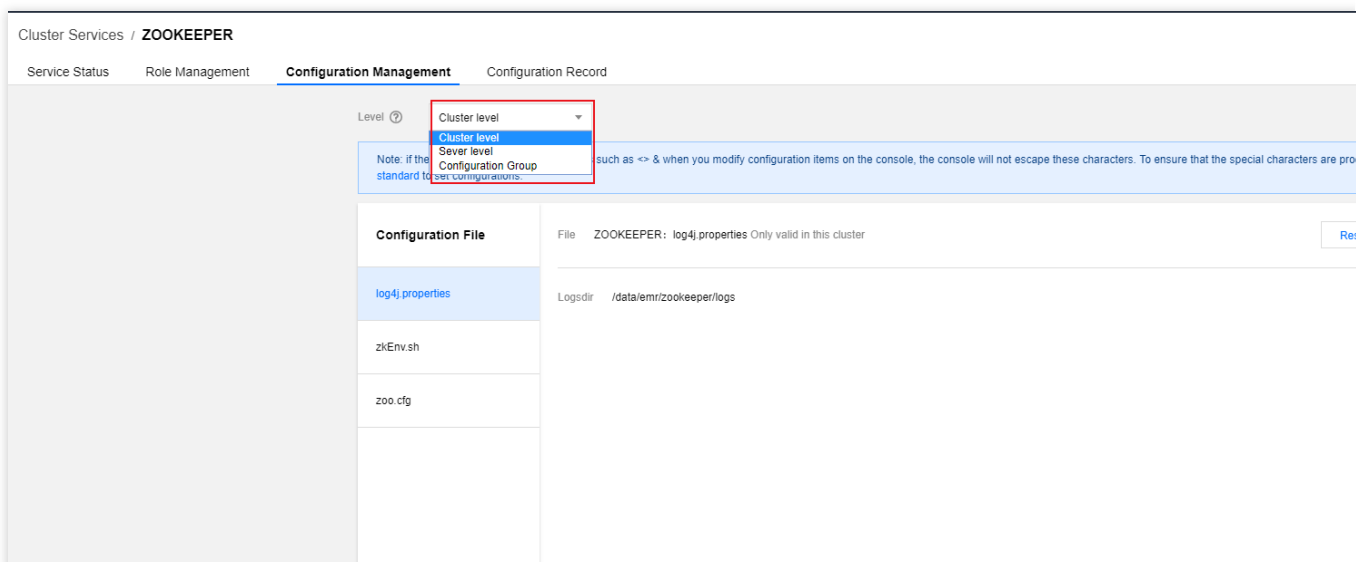


4. You can search for a configuration item or narrow down the search scope with the filter on the left.

5. Select the target configuration file, click **Edit Configuration**, and perform operations such as adding, editing, and deleting configuration items as needed.

Enter the new parameter value for the target parameter. If needed, you can click **Recover** to recover a parameter to its original value. You can also click **Default Value** to restore the parameter to the default value recommended by the system.

You can delete certain configuration items by clicking **Delete** > **OK**.

If the file does not have the parameter you want to configure, click **Add a configuration item** and enter the new parameter name and value in the pop-up window.

6. After confirming that the modification is correct, click **Save configuration**. After the configuration is delivered successfully, click **Restart Service**.

**Note**

If the service process configuration is modified, the configuration will not take effect unless you restart the service.

If the client configuration is modified, the configuration will take effect immediately after being saved.

## Adding a configuration file

1. Log in to the EMR console, select the target cluster in the cluster list, and click **Service** to enter the cluster service list.

2. In the cluster service list, select **Operation** > **Configuration Management** in the top-right corner of the target service panel.

3. If the configuration file you want to configure does not exist, you can click **Add configuration file** on the right to configure a configuration file.

4. After you click **Save configuration**, the parameters will be delivered, and the configuration file name will be updated in the configuration file list.



5. After the custom configuration file is delivered and takes effect, it can be modified and deleted.

## Comparing configurations at different levels

After configurations are modified at different levels, click **Switch to dimension-based comparisons** to compare them at the cluster, configuration group, and node levels. To make the configuration at a level the same as that at the higher level, overwrite the difference and save the modification.

Cluster Service / HDFS ▾

Service Status     Role management     Client management     **Configuration Management**

ⓘ  For configuration adjustment in the console, special characters contained in values such as "<", ">", and "&" are not escaped. To ensure proper processing of special characters, please follow the **XML standard** to set configurations.

Configuration Record

**Configuration filter**

Enter a parameter name 🔍

**Range**                     Clear all
○ NameNode
○ DataNode
○ Client

**Category**                  Clear all
☐ About Address Or Port
☐ Advanced
☐ Basic
☐ CHdfs
☐ Compression
☐ About Cos
☐ About Data Store
☐ About Disk
☐ group
☐ About HA
☐ Host
☐ About JVM OR Memory
☐ About Log
☐ About Network
☐ Path
☐ About Performance
☐ Pid
☐ Proxy
☐ About Security
☐ About Time
☐ uri

**All**     core-site.xml     hadoop-env.sh     hdfs-site.xml     httpfs-env.sh     httpfs-site.xml     log4j.properties

Save configuration     Cancel

| File | Parameter | Value (cluster/configuration group/node dimension) ⓘ |
|------|-----------|------------------------------------------------------|
| | | No data |

Total items: 0

# Configuration Status

Last updated：2023-12-27 14:30:32

## Overview

After the component configuration is modified, saved, and delivered, you can view the configuration status to check for service configuration delivery failures or configuration expiration. You can view the configuration failure and expiration details to understand the modified configuration items of each role.

## Specification items

Synced: No configuration item is modified, or the service is restarted after the modification.

Configuration expired: The service is not restarted after the configuration item is modified and delivered, which means the configuration does not take effect.

Configuration failed: The modified configuration item fails to be delivered to the node configuration.

## Status

1. A role instance has three configuration statuses: **Synced**, **Configuration failed**, and **Configuration expired**.

The initial configuration status is **Synced**. If the role process configuration fails to be delivered, the configuration status of the role will become **Configuration failed**.

After the configuration is delivered successfully, the configuration status of the role will become **Configuration expired**.

In the absence of a configuration failure, the configuration status of the role will become **Synced** after the service is restarted successfully.

2. A client has two configuration statuses: **Synced** and **Configuration failed**.

The initial configuration status is **Synced**. If the client configuration or role process configuration fails to be delivered, the configuration status of the client will become **Configuration failed**.

After the configuration is delivered successfully, the configuration status of the client will become **Synced**.

3. A component has two configuration statuses: **Configuration failed** and **Configuration expired**.

The initial status is not displayed. If the role instance or client fails to be configured, the configuration status of the component will become **Configuration failed**.

If the role instance configuration expires, the configuration status of the component will become **Configuration expired**.

If the role instance or client has both failed configuration and expired configuration, the configuration status of the component will be displayed as **Configuration failed**.

# Directions

1. Log in to the [EMR console](#) and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, select **Cluster Service** and click the component block whose configuration status is **Configuration failed** or **Configuration expired**.
3. Here takes **Configuration expired** as an example. On the component details page, click the configuration status icon in the top-right corner to view the expired configuration details.
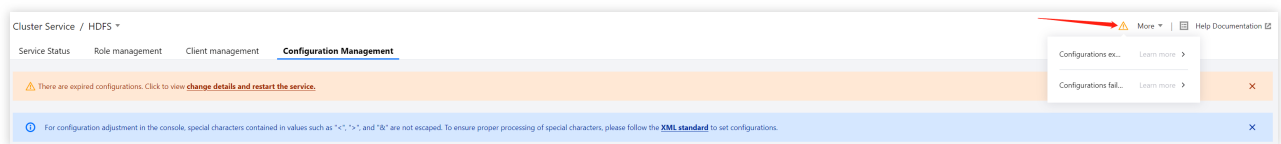


4. Confirm the target expired configuration and restart the service.

# Configuration Rollback

Last updated：2023-12-27 14:30:49

## Overview

This document describes how to roll back operations on component configuration items, such as adding, modifying, and deleting configuration items, in the Elastic MapReduce (EMR) console.

## Directions

1. Log in to the EMR console and click **ID**/**Name** of the target cluster in the cluster list to go to the cluster details page.

2. On the cluster details page, click **Cluster Service** and choose **Operation** > **Configuration Management** in the top-right corner of the target component block. The following uses HDFS as an example:
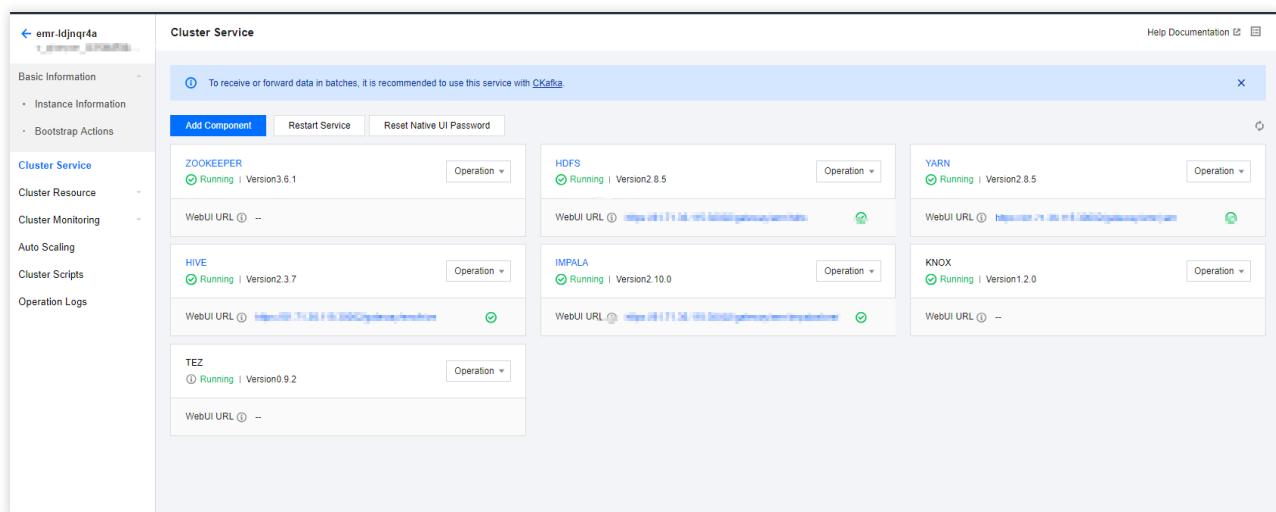


3. Select **Configuration Record** on the **Configuration Management** page to display the configuration record of the current component by default. You can also switch to view the configuration record of all components. Click **Details** to view the parameter values before and after the configuration change. Click **Rollback** > **Confirm rollback** to roll back the configuration change of a parameter. After the rollback is successful, the component will be restarted, and the previous configuration will take effect in a short time.

**Note**

Added, modified, or deleted configuration items can be rolled back, while added or deleted configuration files cannot.

# Configuration Group Management

Last updated：2023-12-27 14:31:16

## Overview

A configuration group is used for group-based configuration management of nodes that have different specifications or purposes but are deployed with the same component. This document describes how to use a configuration group to manage service configuration.

A configuration item modified at the cluster level will overwrite that at the configuration group and node levels if you have not separately modified it at the configuration group and node levels.

A configuration item modified at the configuration group level will overwrite that at the node level within the configuration group if you have not separately modified it at the node level. After the modification, the configuration at the cluster level will not overwrite the configuration at the configuration group level.

A configuration item modified at the node level will be updated only for the node. After the modification, the configuration delivered from the cluster and configuration group levels will not overwrite the configuration at the node level.

**Note**

A node can belong to only one configuration group, and nodes in different groups are different.

## Directions

1. Log in to the EMR console and click **ID**/**Name** of the target cluster in the cluster list to go to the cluster details page.

2. On the cluster details page, click **Cluster services** and select **Operation** > **Configurations** in the top-right corner of the target component block. The following uses HDFS as an example:

3. Click **Configurations**, set **Level** to **Configuration group**, select the target configuration group, and click **Manage configuration groups**.

**Note**

After a cluster is created, several default configuration groups based on node type will be generated for each component, which can only be modified but not deleted and will inherit the cluster configuration upon initialization. You can add configuration groups as needed, with a maximum of 15 configuration groups allowed for each component.
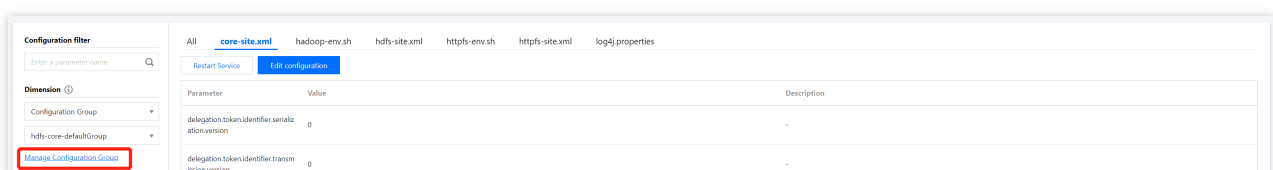
4. On the **Manage configuration groups** page, you can view, add, modify, and delete configuration groups for the current component.

**Note**

You can select an inherited configuration group for a new configuration group. If no inherited configuration group is selected, the cluster-level configuration is inherited by default.

After a custom configuration group is deleted, nodes in it will be transferred to the default configuration group for the same node type.
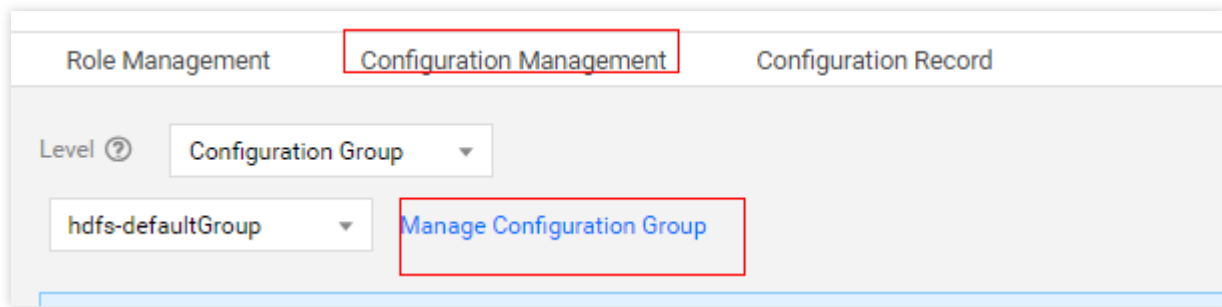


5. If the configuration of a configuration group is different from that of the cluster, click **Save** at the cluster level to overwrite the configuration group configuration. Similarly, if the configuration of a node is different from that of the configuration group to which the node belongs, click **Save** at the configuration group level to overwrite the node configuration.

# YARN Resource Scheduling Overview

Last updated：2023-12-27 14:31:38

## Overview

YARN resource scheduling supports interactive YARN resource queue scheduling management, which is more convenient than file-based configuration management. Currently, it supports two types of scheduling configurations: Fair Scheduler and Capacity Scheduler.

## Notes

1. Fair Scheduler is used by default. Therefore, you need to configure parameters in the `fair-scheduler.xml` configuration file for the YARN component. If you switch to Capacity Scheduler, configure parameters in the `capacity-scheduler.xml` configuration file. No matter which scheduler you use, parameter configurations must be consistent with those on the **Resource Scheduling** page.

2. After switching the scheduler or setting the policy on the **Resource Scheduling** page, you need to click **Apply** to deliver the configuration, which may restart the ResourceManager.

3. If you have enabled label-based scheduling in **Configuration Management**, the sync operation will be performed when Capacity Scheduler is enabled.

4. Subpools can be nested inside a resource pool, and the nested subpools are subject to the rules set for the parent pool.

5. If you specify to sync the modified configuration when disabling the resource scheduler, the configuration file and parameters of the scheduler in **Configuration Management** will be overwritten.

# Configuring Fair Scheduler

Last updated：2023-12-27 14:31:57

## Overview

Fair Scheduler allocates resources fairly to each job on YARN based on weight.

## Definitions

Configuration Set: Specifies the resource assignment among active resource pools during a specified period.

Execution Plan: Specifies when the configuration set will be active.

Placement Rule: Automatically allocates jobs submitted by different users to specified resource pools.

User Limits: Specifies the maximum number of applications that can be submitted by a user at a time.

## Directions

### Creating a resource pool

1. Log in to the EMR console and click **Details** of the target Hadoop cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, select **Cluster services** and click **Operation** > **Resource scheduling** in the top-right corner of the YARN component block to enter the **Resource scheduling** page.



3. Toggle on **Resource scheduler** and configure the scheduler.

4. Create a resource pool for Fair Scheduler.

Select **Fair Scheduler**. On the displayed page, click **Create resource pool**. You can edit, clone, and delete an existing resource pool as well as create a subpool for it.

Scheduling Policy ⓘ

~~DRP: Dominant Resource Fairness. This fairly schedules resources based on C~~
and memory.

🔘 Fair: fairly schedules resources based on memory only.

⚪ FIFO: schedules resources on a first-in, first-out basis.

**Preemption**

Preemption Mode ⓘ                    🔵⚪

Fair Share Preemption Threshold ⓘ        [ − ____ + ]

Fair Share Preemption Timeout ⓘ         [ − ____ + ]  seconds

Min Share Preemption Timeout ⓘ         [ − ____ + ]  seconds

**Access Control**

Submission ⓘ        🔘 Allow all users to submit jobs to this resource pool
                   ⚪ Allow the following users to submit jobs to this resource pool

Management ⓘ        🔘 Allow all users to manage this pool
                   ⚪ Allow the following users to manage this pool

[ Confirm ]   [ Cancel ]

**Fields and parameters**:

| Field | Parameter | Description |
|---|---|---|
| Resource Pool Name | name | Name of the resource pool or queue. It can contain only letters, digits, hyphens, and underscores and cannot start with a hyphen or underscore. |
| Parent Pool | The value of `type` is `parent`. | Means that although the resource pool has no subpools, it is not a leaf node. A parent pool cannot have subpools in Hadoop v2.8 and later. |
| Configuration Set | None | YARN does not have this parameter, which means a collection of scheduled tasks. |
| Weight | weight | Percentage of resources in the parent pool. A greater weight means more resources allocated. |
| Min Resources | minResources | The minimum amount of resources guaranteed. When the minimum amount of resources guaranteed for a queue is not met, it has a higher priority than other queues at the same level to obtain resources. |

| Max Resources | maxResources | The maximum amount of resources that can be used. The amount of resources available for each queue cannot exceed this value. |
|---|---|---|
| Max Concurrent Running Apps | maxRunningApps | The maximum number of concurrent running applications allowed. This limitation can prevent intermediate outputs generated by excessive concurrent running map tasks from filling up the disks. |
| Max Share for App Master | maxAMShare | The maximum percentage of resources that can be used to run Application Master. This attribute only applies to leaf queues. |
| Scheduling Policy | schedulingPolicy | You can set a scheduling policy for any queue. Valid values include `Fifo`, `Fair`, and `Drf`. If the value is `Fifo` or `Fair`, only memory is taken into account in resource allocation. If `Drf`, both memory and the number of cores are taken into account. |
| Preemption Mode | allowPreemptionFrom | This field applies only to Hadoop v3.x and later. In v2.x, you can only configure `yarn.scheduler.fair.preemption` to set the preemption mode. |
| Fair Share Preemption Threshold | fairSharePreemptionThreshold | The fair share preemption threshold for the queue. If the queue waits `fairSharePreemptionTimeout` without receiving `fairSharePreemptionThreshold*fairShare` resources, it is allowed to preempt resources from other queues. If this field is not set, the queue will inherit the value from its parent queue. |
| Fair Share Preemption Timeout | fairSharePreemptionTimeout | Number of seconds the queue is under its fair share threshold before it will try to preempt resources from other queues. If this field is not set, the queue will inherit the value from its parent queue. |
| Min Share Preemption Timeout | minSharePreemptionTimeout | Number of seconds the queue is under its minimum share before it will try to preempt resources from other queues. If this field is not set, the queue will inherit the value from its parent queue. |
| Submission | aclSubmitApps | List of users that can submit apps to the queue |
| Management | aclAdministerApps | List of users that can manage the queue |

## Configuring an execution plan

1. In the **Policy settings** section, click **Execution plans** > **Create execution plan** to create an execution plan. **Configuration set status** indicates whether the execution plan is enabled, which defaults to "Enabled". When you don't want to enable an execution plan but still want to retain the configuration set, you can set the plan status to "Disabled".

2. In the **Create execution plan** window, select/enter the **Configuration set**, **Name**, and **Plan validity**.



## Sample configuration set

1. Log in to the EMR console and click **Details** of the target Hadoop cluster in the cluster list to enter the cluster details page.

2. On the cluster details page, select **Cluster services** and click **Operation** > **Resource scheduling** in the top-right corner of the YARN component block to go to the **Resource scheduling** page.

3. Toggle on **Resource scheduler** and select **Fair Scheduler**.

4. Click **Create resource pool** and configure parameters as needed.

5. On the **Resource Scheduling** page, click **Execution Plans** > **Create Execution Plan** and adjust the **Plan Validity** as needed.

**Note**

 If scheduled scale-out is configured for the EMR cluster, we recommend you set the **Plan Validity** to a time period after the scheduled scale-out.



6. On the **Resource Scheduling** page, click **Resource Pools** and select an item from the **Configuration Set** dropdown list.

**Note**

 The resource limits of a resource pool can be configured by configuration set.



7. Select the created resource pool from **Configuration Set** and adjust the **Resource Limits** as needed.

**Resource Pools**  Execution Plans  Placement Rules  User Limits

Create Resource Pool

Configuration Set 08to12

| Resource Pool Name | Weight | Min Resources | | Max Resources | | Max Running Apps | Scheduling Policy | Preem |
|---|---|---|---|---|---|---|---|---|
| | | CPU | Memory | CPU | Memory | | | |
| root | -- | -- | -- | -- | -- | -- | -- | -- |
| flink | 2 | -- | -- | -- | -- | -- | drf | -- |
| hive | 2 | -- | -- | -- | -- | 100 | fair | -- |
| default | 2 | -- | -- | 50 | 150 | 10 | fair | -- |

资源池  计划模式  放置规则  用户限制

新建资源池

配置集 20to24

| 资源池名称 | 权重 | 最小资源数 | | 最大资源数 | | 最高处于运行状态Application数目 | 调度策略 | 抢占模 |
|---|---|---|---|---|---|---|---|---|
| | | CPU | 内存 | CPU | 内存 | | | |
| root | -- | -- | -- | -- | -- | -- | -- | -- |
| flink | 8 | -- | -- | -- | -- | -- | drf | -- |
| hive | 1 | -- | -- | -- | -- | 100 | fair | -- |
| default | 1 | -- | -- | 50 | 150 | 10 | fair | -- |

资源池  计划模式  放置规则  用户限制

新建资源池

配置集 default

| 资源池名称 | 权重 | 最小资源数 | | 最大资源数 | | 最高处于运行状态Application数目 | 调度策略 | 抢占模 |
|---|---|---|---|---|---|---|---|---|
| | | CPU | 内存 | CPU | 内存 | | | |
| root | -- | -- | -- | -- | -- | -- | -- | -- |
| flink | 4 | -- | -- | -- | -- | -- | drf | -- |
| hive | 4 | -- | -- | -- | -- | 100 | fair | -- |
| default | 2 | -- | -- | 50 | 150 | 10 | fair | -- |

8. After the resource pool adjustment, click **Apply** to make the settings effective.

## Configuring a placement rule

In the **Policy settings** section, click **Placement rules** > **Create placement rule** to create a placement rule.
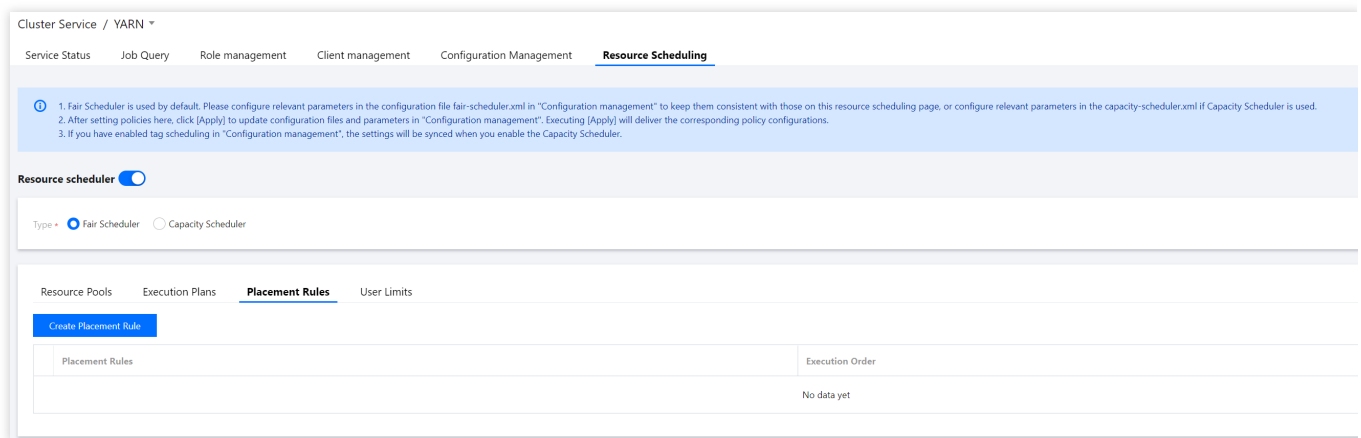
Set **Placement type** and **Pool name**.



**Configuration rule type description:root.[pool name]**: The rule always needs to be met and applies when other rules do not match. Therefore, it needs to be placed after others.

**root.[pool name].[username]**: The rule checks whether the pool name exists in the resource pool, and if so, a resource pool with the same name as the username will be created (when **Create one if the pool does not exist** is selected).

**root.[primary group]**: The rule uses the resource pool matching the user's primary group. On Linux, the user's default primary group has the same name as the username. The names of the user's primary group and resource pool are used for match.

**root.[primary group].[username]**: The rule first uses the resource pool matching the user's primary group and then the subpool matching the username. If the selected pool does not exist but a pool is created, a subpool with the same name as the username will be created under the pool.

**root.[secondarygroup]**: The rule matches the user's secondary groups and uses the resource pool matching one of the secondary groups.

**root.[secondarygroup].[username]**: The rule first matches the user's secondary groups and then the resource pool matching the username.

**root.[username]**: The rule matches the resource pool with the same name as the username (not recommended).

Specified at runtime: The rule mainly uses the resource pool specified at runtime.

Placement rules are matched in the order of 1, 2, 3... If a rule is matched, rules after it will not be matched.

## Configuring user limits

In the **Policy settings** section, click **User limits** > **Create user limit** to create a user limit.
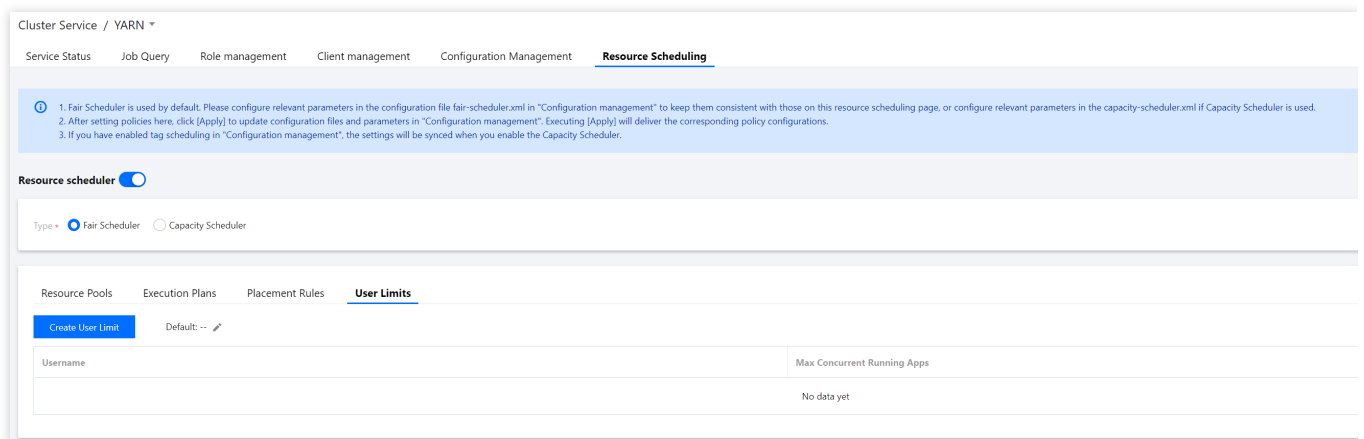


Set **Username** and **Max Concurrent Running Apps**.

# Configuring Capacity Scheduler

Last updated：2023-12-27 14:32:16

## Overview

Capacity Scheduler organizes resources in a hierarchical manner, allowing multiple users to share cluster resources based on multi-level resource restrictions.

## Directions

### Creating a resource pool

1. Log in to the EMR console and click **Details** of the target Hadoop cluster in the cluster list to go to the cluster details page.
2. On the cluster details page, select **Cluster Service** and choose **Operation** > **Resource Scheduling** in the top-right corner of the YARN component block to go to the **Resource Scheduling** page.
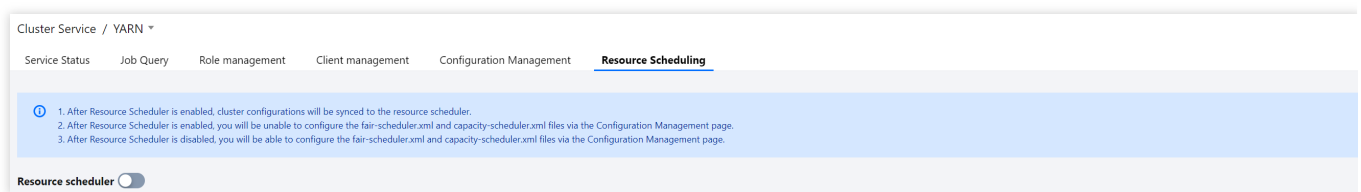


3. Toggle on **Resource scheduler** and configure the scheduler.
4. Create a resource pool for Capacity Scheduler.

Select **Capacity Scheduler** and click **Create Resource Pool** to create a resource pool. You can edit or clone an existing resource pool as well as create a subpool for it. You can also click **Default settings** to set the number of times of delayed scheduling.

Cluster Service / YARN ▾

Service Status    Job Query    Role management    Client management    Configuration Management    **Resource Scheduling**

ⓘ    1. Fair Scheduler is used by default. Please configure relevant parameters in the configuration file fair-scheduler.xml in "Configuration management" to keep them consistent with those on this resource scheduling page, or configure relevant parameters in the capacity-scheduler.xml if Capacity Scheduler is used.
2. After setting policies here, click [Apply] to update configuration files and parameters in "Configuration management". Executing [Apply] will deliver the corresponding policy configurations.
3. If you have enabled tag scheduling in "Configuration management", the settings will be synced when you enable the Capacity Scheduler.

**Resource scheduler** 🔵

Type ∗  ◯ Fair Scheduler    ◉ Capacity Scheduler

**Resource Pools**    Resource Pool Mappings

[ Create Resource Pool ]    [ Default settings ]

Tag scheduling 🔘

| Resource Pool Name | Capacity | Max C... | Min User Ca... | User Resour... | Max vCores per Container | Max Memory per Container |
|---|---|---|---|---|---|---|
| root | -- | -- | -- | -- | -- | -- |
| default | 100% | 100% | -- | 1 | -- | -- |

---

n the configuration file fair-scheduler.xml in "Configuration management" to keep them consistent with those on this resource

## Create Resource Pool

### Basic Settings

Resource Pool Name ⓘ

[ Enter a resource pool name ]

### Resource Limits

Tag settings

| Tag | Capacity | Max Capacity |
|---|---|---|
| <DEFAULT_PARTITION ▾ | Enter    % | Enter |

Min User Capacity ⓘ          [ Enter    % ]

User Resource Factor ⓘ        [ Enter    % ]

Max Memory per Container ⓘ    [ Enter    MB ]

Max vCores per Container ⓘ    [ Enter ]

Resource Pool Status ⓘ    ◉ RUNNING    ◯ STOPPED

### App Limits

Max Apps ⓘ    [ −  10000  + ]

Max Resources for AM ⓘ    [ 25    % ]

Resource pool priority ⓘ            Enter

**Access control**

Submission ⓘ         ◉ Allow all users to submit jobs to this resource pool

                         ○ Allow the following users to submit jobs to this resource pool

Management ⓘ       ◉ Allow all users to manage this pool

                         ○ Allow the following users to manage this pool

<div align="center">

[ Confirm ]   [ Cancel ]

</div>

**Fields and parameters**:

| Field Name | Parameter | Description |
|---|---|---|
| Resource Pool Name | yarn.scheduler.capacity. <queue-path>.queues</queue-path> | Name of the resource pool or queue |
| Label Settings | N/A | Set the specified label that can be accessed by the queue. |
| Capacity | yarn.scheduler.capacity. <queue-path>.capacity</queue-path> | Available resource amount. The total capacity of the subpools of a parent pool is 100. Available resource amount = resource amount of the parent pool * percentage set here. The queue can consume more resources than the queue's capacity if there are idle resources in other queues. |
| Max Capacity | yarn.scheduler.capacity. <queue-path>.maximum-capacity</queue-path> | Maximum queue capacity in percentage. Because of resource sharing, the amount of resources used by a queue may exceed its capacity, and this field specifies the maximum amount of resources that can be used by the queue. |
| Default Label Expression | yarn.scheduler.capacity. <queue-path>.default-node-label-expression</queue-path> | If a resource request does not have a node label specified, the application will be submitted to the corresponding partition specified by this configuration item. By default, the value is empty, i.e., applications will be allocated to containers on nodes with no label. |
| Min User Capacity | yarn.scheduler.capacity. <queue-path>.minimum-user-limit-percent</queue-path> | Minimum resources in percentage guaranteed for each user. Each queue enforces a limit on the percentage of resources allocated to a user at any given time. When multiple users' applications are running in a queue concurrently, the amount |

| | | of resources used by each user varies between a minimum and maximum value. The minimum value depends on the number of running applications, and the maximum value is determined by `minimum-user-limit-percent`. |
|---|---|---|
| User Resource Factor | yarn.scheduler.capacity. <queue-path>.user-limit-factor</queue-path> | Maximum amount of resources in percentage that can be used by each user. For example, if the value is `30`, the amount of resources for each user cannot exceed 30% of the queue capacity at any given time. |
| Max Memory per Container | yarn.scheduler.capacity. <queue-path>.maximum-allocation-mb</queue-path> | Maximum memory that can be allocated to each container. The value will overwrite and cannot be greater than that of the system's `yarn.scheduler.maximum-allocation-mb`. |
| Max vCores per Container | yarn.scheduler.capacity. <queue-path>.maximum-allocation-vcores</queue-path> | Maximum number of CPU cores that can be allocated to each container. The value will overwrite and cannot be greater than that of the system's `yarn.scheduler.maximum-allocation-vcores`. |
| Resource Pool Status | yarn.scheduler.capacity. <queue-path>.state</queue-path> | Status of the queue. The value can be `Running` or `Stopped`. If a queue is in the `Stopped` status, new applications cannot be submitted to it or any of its subqueues. |
| Max Apps | yarn.scheduler.capacity. <queue-path>.maximum-applications</queue-path> | Maximum number of concurrent active (both running and pending) applications allowed in the system |
| Max Resources for AM | yarn.scheduler.capacity. <queue-path>.maximum-am-resource-percent</queue-path> | Maximum percentage of resources in the cluster which can be used to run application masters. It controls the number of concurrent active applications. |
| Resource Pool Priority | yarn.scheduler.capacity.root. <leaf-queue-path>.default-application-priority</leaf-queue-path> | Configure the priority of the resource queue, which is `0` by default. The larger the value, the higher the priority. |
| Submission | yarn.scheduler.capacity.root. <queue-path>.acl_submit_applications | List of users that can submit apps to the queue |
| Management | yarn.scheduler.capacity.root. <queue-path>.acl_administer_queue | List of users that can manage the queue |
| Delayed | yarn.scheduler.capacity.node- | Set the allowed number of times of delayed scheduling to |

| Scheduling | locality-delay | ensure the local execution of tasks. If the value is `-1`, delayed scheduling will be disabled. |
|---|---|---|

## Configuring resource pool mappings

1. In the **Policy Settings** section, choose **Resource Pool Mappings** > **Create Resource Pool Mapping** to create a resource pool mapping.
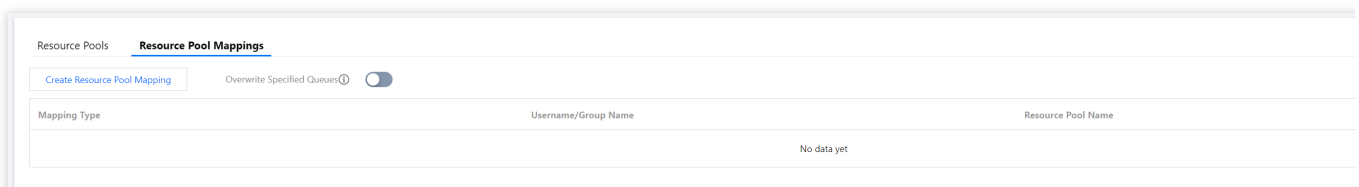




2. Configure **Overwrite Specified Queues**.

This feature is disabled by default. For example, you have defined a mapped queue in **Resource Pool Mappings** and specified a queue other than the mapped queue when submitting a job; if the specified queue is `default` and **Overwrite Specified Queues** is enabled, the mapped queue will be used; otherwise, the specified queue will be used.

## Sample label-based scheduling

1. Log in to the EMR console and click **Details** of the target Hadoop cluster in the cluster list to go to the cluster details page.

2. On the cluster details page, select **Cluster Service** and choose **Operation** > **Resource Scheduling** in the top-right corner of the YARN component block to go to the **Resource Scheduling** page.

3. Toggle on **Resource scheduler** and select **Capacity Scheduler**.

4. Toggle on **Label-based scheduling** and click **Label management** to go to the label management** page.
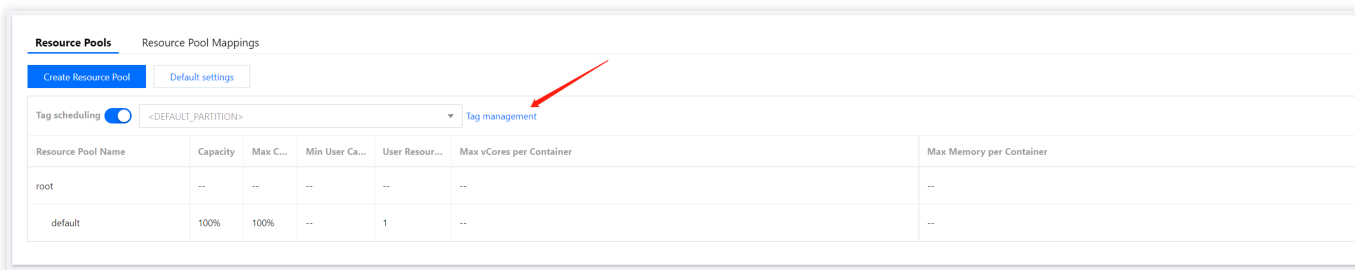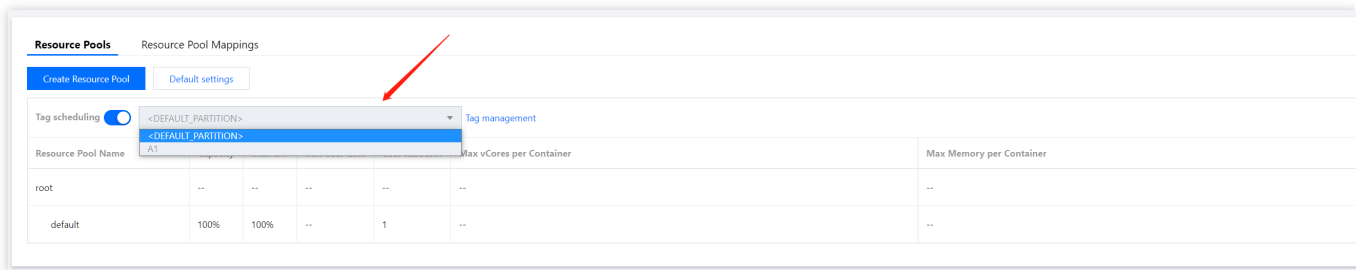


5. Click **Create Label**, enter the label name, and set the label type and node to be bound to as needed.



6. After the label is set, click **Apply**. Then, you can view and edit the resource queue of the label in the resource pool.
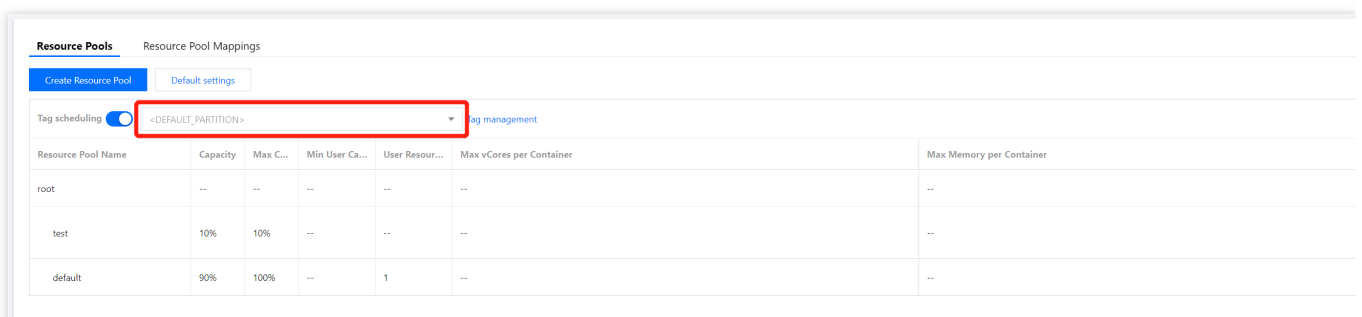
7. On the **Resource Scheduling** page, click **Create Resource Pool** and set **Label**, **Capacity**, **Max Capacity**, and other parameters as needed.

**Note**

 The capacity and maximum capacity of a resource pool can be configured by label.



8. After the resource pool is set, click **Apply** to submit the task to the backend.

**Caution**

 Proceed with caution when restarting the ResourceManager. If you are prompted that the ResourceManager will be restarted after you click **Apply**, check whether the operation is successful in **Scheduling History** and whether the ResourceManager is healthy in **Role Management**.

# Label Management

Last updated：2023-12-27 14:34:47

## Overview

Label management allows you to create, edit, or delete a label or bind it to a node. Adding different labels to cluster nodes helps allocate resources at a finer granularity based on Capacity Scheduler and specify the running locations of applications.

## Prerequisites

You have enabled YARN resource scheduling, switched to Capacity Scheduler, and enabled label-based scheduling.

## Directions

1. Log in to the EMR console and click **Details** of the target Hadoop cluster in the cluster list to enter the cluster details page.

2. On the cluster details page, select **Cluster Service** and click **Operation** > **Resource Scheduling** in the top-right corner of the YARN component block to enter the **Resource Scheduling** page.

3. Toggle on **Resource scheduler** and select **Capacity Scheduler**.

4. Toggle on **Label-based scheduling** and click **Label Management** to enter the **Label Management** page.
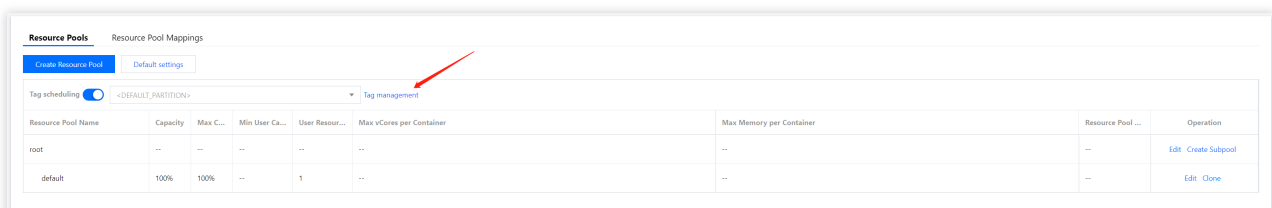


5. On the **Label Management** page, view all labels of the cluster, create, edit, delete, or sync a label, or redirect to the web UI to view the node to which the label is bound.

**Tag management**

Apply  Reset  ✕

Create tag    Sync tag

View nodes ↗

| Tag | Operation |
| --- | --- |
| <DEFAULT_PARTITION> | Edit |
| A1 | Edit  Delete |

Total items: 2

10 ▼ / page    |◁  ◁    1    / 1 page    ▷  ▷|

# Viewing Scheduling History

Last updated：2023-12-27 14:35:03

## Overview

The scheduling history displays information such as operation records and task status of the resource queue configuration.

## Directions

1. Log in to the EMR console and click **Details** of the target Hadoop cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, select **Cluster Service** and click **Operation** > **Resource Scheduling** in the top-right corner of the YARN component block to enter the **Resource Scheduling** page.
3. On the **Resource Scheduling** page, click **Scheduling History** to view the start time, end time, status, and operation information of a task.

4. You can filter scheduling records by execution time and click **Details** in the **Operation** column to view detailed information.

# HBase RIT Fixing

Last updated：2023-12-27 14:35:21

## Feature overview

The HBase RIT fixing feature is suitable for fixing regions in the `FAILED_OPEN` , `FAILED_CLOSED` , or `CLOSING` status when an HBase 2.2.0 or later cluster encounters prolonged Region-In-Transition (RIT) issues.

## Directions

1. Log in to the [EMR console](#) and click a **cluster ID**/**name** in the **Cluster list** to enter the cluster details page.

2. On the cluster details page, click **Cluster Service** and select **Operation** > **RIT fixing** in the top-right corner of the HBase component panel to view and fix regions in the RIT status.



3. In the list, click **Fix** in the **Operation** column, or select the target regions and click **Batch fix**, and confirm the information in the **RIT fixing** pop-up window.

4. After confirming that everything is incorrect, click **Confirm**.

5. You can view the progress and result of RIT fixing in the **Task Center**.

# Component Port Information

Last updated：2023-12-27 14:35:38

This document describes common component ports.

## Common HDFS ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| HDFS | 4007 | NameNode RPC port, which is used for:<br>1. The communication between the HDFS client and NameNode.<br>2. The connection between the DataNode and NameNode. |
| HDFS | 4008 | HDFS HTTP port (NameNode), which is used for:<br>1. Point-to-point NameNode checkpoint operations.<br>2. Connecting the remote web client to the NameNode UI. |
| HDFS | 4009 | HDFS HTTPS port (NameNode), which is used for:<br>1. Point-to-point NameNode checkpoint operations.<br>2.  Connecting the remote web client to the NameNode UI. |
| HDFS | 4004 | DataNode IPC server port, which is used for connecting the client to the DataNode to perform RPC operations. |
| HDFS | 4001 | DataNode data transfer port, which is used for:<br>1. The data transfer from or to the DataNode to or from the HDFS client.<br>2. Point-to-point DataNode data transfer. |
| HDFS | 4002 | DataNode HTTP port, which is used for connecting the remote web client to the DataNode UI in safe mode. |
| HDFS | 4003 | DataNode HTTPS port, which is used for connecting the remote web client to the DataNode UI in safe mode. |
| HDFS | 4005 | JournalNode RPC port, which is used for client communication to access multiple types of information. |
| HDFS | 4006 | JournalNode HTTP port, which is used for connecting the remote web client to the JournalNode in safe mode. |
| HDFS | 4032 | The listener port of the HttpFS HTTP server, which is used for connecting the remote RESTful API to HttpFS. |

# Common YARN ports

| Component | Port Number | Description |
|---|---|---|
| YARN | 5024 | The web HTTP port of the JobHistory server, which is used for viewing the webpage of the JobHistory server. |
| YARN | 5022 | JobHistory server port, which is used for:<br>1. Recovering task data on the MapReduce client.<br>2. Getting the task report on the JobHistory client. |
| YARN | 5003 | The web HTTP port of the ResourceManager service. |
| YARN | 5002 | The web HTTPS port of the ResourceManager service, which is used for connecting to the ResourceManager web application in safe mode. |
| YARN | 5004 | The web HTTP port of the NodeManager. |
| YARN | 5005 | The web HTTPS port of the NodeManager, which is used for connecting to the NodeManager web application in safe mode. |
| YARN | 5001 | ResourceManager scheduler port. |
| YARN | 5000 | ResourceManager RPC port, which is used for submitting tasks on the client. |

# Common Hive ports

| Component | Port Number | Description |
|---|---|---|
| Hive | 7010 | The port for WebHCat to provide RESTful services, which is used for the communication between the WebHCat client and WebHCat server. |
| Hive | 7001 | The port for HiveServer to provide Thrift services, which is used for the communication between the HiveServer client (Beeline) and HiveServer. |
| Hive | 7004 | The port for MetaStore to provide Thrift services, which is used for the communication between the MetaStore client and MetaStore, i.e., the communication between HiveServer and MetaStore. |
| Hive | 7003 | The web UI port of Hive, which is used for the HTTPS/HTTP communication between web requests and the Hive UI server. |

# Common Spark ports

| Component | Port Number | Description |
|---|---|---|
| Spark | 10000 | HTTP port. |

# Common Presto ports

| Component | Port Number | Description |
|---|---|---|
| Presto | 9000 | The HTTP port for the Presto coordinator or worker to provide external services. |

# Common PrestoSQL ports

| Component | Port Number | Description |
|---|---|---|
| PrestoSQL | 9000 | The HTTP port for the PrestoSQL coordinator or worker to provide external services. |

# Common Trino ports

| Component | Port Number | Description |
|---|---|---|
| Trino | 9000 | The HTTP port for the Trino coordinator or worker to provide external services. |

# Common Impala ports

| Component | Port Number | Description |
|---|---|---|
| Impala | 27000 and 27009 | The ports for Impala application communication. The former is for Impala 2.x, and the latter is for Impala 3.x. |
| Impala | 27001 | The port for impala-shell communication. |

| Impala | 27004 | The web port of the Impala server. |
|--------|-------|-------------------------------------|
| Impala | 27007 | The web port of the Impala Catalog. |
| Impala | 27005 | The web port of the Impala StateStore. |

## Common Kudu ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| Kudu | 7051 | The PRC port of the Kudu master. |
| Kudu | 7050 | The RPC port of the Kudu server. |
| Kudu | 8051 | The HTTP port of the Kudu master. |
| Kudu | 8050 | The HTTP port of the Kudu server. |

## Common ClickHouse ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| ClickHouse | 9000 | The TCP port for accessing the business client. |
| ClickHouse | 8123 | The HTTP port for accessing the business client. |
| ClickHouse | 9009 | The HTTPS port for accessing the business client. |

## Common Kylin ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| Kylin | 16500 | Kylin HTTP port. |

## Common Doris ports

| Component | Port Number | Description |
|-----------|-------------|-------------|

| Doris | 8000 | The Thrift server port on the broker. |
|-------|------|---------------------------------------|
| Doris | 9060 | The Thrift server port on the backend, which is used for receiving frontend requests. |
| Doris | 8060 | The BRPC port on the backend, which is used for the communication between backend instances. |
| Doris | 9050 | The heartbeat service port (Thrift) on the backend, which is used for receiving frontend heartbeats. |
| Doris | 8040 | The Thrift server port on the broker, which is used for receiving requests. |
| Doris | 9010 | The port on the frontend for the communication between metadata management modules (BDBJE). |
| Doris | 8030 | The HTTP server port on the frontend. |
| Doris | 9020 | The Thrift server port on the frontend. |
| Doris | 9030 | The port on the frontend for receiving MySQL client access requests. |

## Common StarRocks ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| StarRocks | 8000 | The Thrift server port on the broker. |
| StarRocks | 9060 | The Thrift server port on the backend, which is used for receiving frontend requests. |
| StarRocks | 8060 | The BRPC port on the backend for the communication between backend instances. |
| StarRocks | 9050 | The heartbeat service port (Thrift) on the backend, which is used for receiving frontend heartbeats. |
| StarRocks | 8040 | The Thrift server port on the broker, which is used for receiving requests. |
| StarRocks | 9010 | The port on the frontend for the communication between metadata management modules (BDBJE). |
| StarRocks | 8030 | The HTTP server port on the frontend. |
| | | |

| StarRocks | 9020 | The Thrift server port on the frontend. |
| StarRocks | 9030 | The port on the frontend for receiving MySQL client access requests. |

## Common Druid ports

| Component | Port Number | Description |
| --- | --- | --- |
| Druid | 8082 | The listener port of the broker server ( `broker-runtime.properties` ), which is used for receiving client queries. |
| Druid | 8081 | The listener port of the coordinator server ( `coordinator - runtime.properties` ) for the communication with other components. |
| Druid | 8083 | The listener port of the historical server ( `historical-runtime.properties` ) for the communication with other components. |
| Druid | 8091 | The listener port of the middleManager server ( `middleManager- runtime.properties` ) for the communication with other components. |
| Druid | 8090 | The listener port of the Overlord server ( `overlord-runtime.properties` ) for the communication with other components. |
| Druid | 18888 | The listener port of the router server ( `router-runtime.properties` ), which is used for routing client queries to the broker. |

## Common HBase ports

| Component | Port Number | Description |
| --- | --- | --- |
| HBase | 6000 | HMaster RPC port, which is used for connecting the HBase client to HMaster. |
| HBase | 6001 | HMaster HTTPS port, which is used for connecting the remote web client to the HMaster UI. |
| HBase | 6002 | The RPC port of the RegionServer (RS), which is used for connecting the HBase client to the RegionServer. |
| HBase | 6003 | RegionServer HTTPS port, which is used for connecting the remote web client to the RegionServer UI. |

| HBase | 6005 | The Info Server listener port of the HBase Thrift server. |
|-------|------|-----------------------------------------------------------|
| HBase | 6004 | The HBase Thrift server listener port. |

## Common Flink ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| Flink | 16001 | Flink web UI port, which is used for the HTTP/HTTPS communication between client web requests and the Flink server. |

## Common Storm ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| Storm | 15002 | Logviewer service port. |
| Storm | 15000 | Nimbus service port. |
| Storm | 15001 | Storm web UI port. |

## Common Hue ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| Hue | 13000 | The port for Hue to provide HTTPS web services, which can be modified. |

## Common Oozie ports

| Component | Port Number | Description |
|-----------|-------------|-------------|
| Oozie | 12000 | HTTP port, which is used for client access. |

## Common Superset ports

| Component | Port Number | Description |
|---|---|---|
| Superset | 18088 | Superset service port, which is used for client connection. |

## Common Zeppelin ports

| Component | Port Number | Description |
|---|---|---|
| Zeppelin | 18000 | Zeppelin service port, which is used for client connection. |

## Common Kafka ports

| Component | Port Number | Description |
|---|---|---|
| Kafka | 9092 | The port for the broker to provide data reception and acquisition services. |

## Common Kafka Manager ports

| Component | Port Number | Description |
|---|---|---|
| Kafka Manager | 9000 | The listener port of the Kafka Manager server, which is used for client connection. |

## Common Ranger ports

| Component | Port Number | Description |
|---|---|---|
| Ranger | 6080 | Ranger admin web UI port. |
| Ranger | 5151 | The port of Ranger UserSync for authentication services. |

## Common COSRanger ports

| | | |
|---|---|---|

| Component | Port Number | Description |
|---|---|---|
| COSRanger | 9999 | The listener port of the COSRangerServer server, which is used for client connection. |

## Common KRB5 ports

| Component | Port Number | Description |
|---|---|---|
| KRB5 | 749 | Kadmin service port. |
| KRB5 | 754 | Kprop service port. |
| KRB5 | 88 | Kerberos server port for component authentication in the Kerberos service, which may be used in configuring mutual trust of clusters. |

## Common Knox ports

| Component | Port Number | Description |
|---|---|---|
| Knox | 30002 | HTTP port, which is usually used for browser connection. |
| Knox | 33389 | The LDAP port of Knox, which is used for Knox authentication. |

## Common ZooKeeper ports

| Component | Port Number | Description |
|---|---|---|
| ZooKeeper | 2181 | ZooKeeper client port, which is used for connecting the ZooKeeper client to the ZooKeeper server. |

## Common OpenLDAP ports

| Component | Port Number | Description |
|---|---|---|

| | | |
|---|---|---|
| OpenLDAP | 389 | The port for client access. |

## Common Tez ports

| Component | Port Number | Description |
|---|---|---|
| Tez | 2000 | The web UI port of Tez. |

## Common Livy ports

| Component | Port Number | Description |
|---|---|---|
| Livy | 8998 | The listener port of the Livy server, which is used for client connection. |

## Common Kyuubi ports

| Component | Port Number | Description |
|---|---|---|
| Kyuubi | 10009 | The listener port of the Kyuubi server, which is used for client connection. |

## Common Alluxio ports

| Component | Port Number | Description |
|---|---|---|
| Alluxio | 20001 | The RPC port of the Alluxio Job Master, which is used by the Alluxio Master to assign tasks to the Alluxio Job Master. |
| Alluxio | 30001 | The RPC port of the Alluxio Job Worker, which is used by the Alluxio Job Master to assign tasks to the Alluxio Job Worker. |
| Alluxio | 19998 | The RPC port of the Alluxio Master, which is used for connecting the client to the Alluxio Master. |
| Alluxio | 29998 | The RPC port of the Alluxio Worker, which is used by the Alluxio Master to assign read-write tasks to the Alluxio Worker. |

# Common GooseFS ports

| Component | Port Number | Description |
|---|---|---|
| GooseFS | 9206 | The RPC port of the Alluxio Job Master, which is used by the Alluxio Master to assign tasks to the Alluxio Job Master. |
| GooseFS | 9210 | The RPC port of the Alluxio Job Worker, which is used by the Alluxio Job Master to assign tasks to the Alluxio Job Worker. |
| GooseFS | 9201 | The RPC port of the Alluxio Master, which is used for connecting the client to the Alluxio Master. |
| GooseFS | 9211 | The listener port of the GooseFSProxy server, which is used for client proxy. |
| GooseFS | 9204 | The RPC port of the Alluxio Worker, which is used by the Alluxio Master to assign read-write tasks to the Alluxio Worker. |

# Common Ganglia ports

| Component | Port Number | Description |
|---|---|---|
| Ganglia | 1602 | Gmetad server port. |
| Ganglia | 1603 | Gmetad server port, which is used for receiving HTTPd data queries. |
| Ganglia | 1601 | The port for gmond process communication and gmetad access. |

# Service Operation

Last updated：2023-12-27 14:35:59

## Overview

Service operation is a collection of convenient Ops and management tools provided for components, including general service restart and command-based Ops operations specific to certain components, such as HDFS NameNode active/standby switch, HDFS data rebalance, YARN ResourceManager active/standby switch, and YARN queue refresh.

**Note**

1. YARN ResourceManager active/standby switch requires you to disable `yarn.resourcemanager.ha.automatic-failover.enabled` . If it is not displayed in the operation drop-down list on the YARN block, you need to find `yarn.resourcemanager.ha.automatic-failover.enabled` in the `yarn-site.xml` configuration file of YARN configuration management and disable it.

2. Do not delete the queues that have already taken effect in the scheduler's configuration file when refreshing YARN queues.

3. Ranger metadatabase change is currently supported only for MySQL databases, and only the connections of admin users will be tested.

## Specification Items

**HDFS NameNode active/standby switch**: This operation switches the active NameNode to standby status and the standby NameNode to active status.

**HDFS data rebalance**: This operation is usually required when a new DataNode is added. It will make the data evenly distributed to avoid hotspotting and make the cluster read/write load more balanced.

**YARN ResourceManager active/standby switch**: This operation switches the active ResourceManager to the standby status and the standby ResourceManager to active status. It is allowed only when `yarn.resourcemanager.ha.automatic-failover.embedded` is disabled.

**YARN queue refresh**: This operation enables additions or updates to `capacity-scheduler.xml` or `fair-scheduler.xml` to take effect in the ResourceManager. Do not delete queues defined in them.

**Ranger metadatabase change**: To change the underlying database of Ranger, you need to modify the `conf/install.properties` file and then run the `setup.sh` script locally. This script can synchronize the database information to the local `ranger-admin-site.xml` configuration file without modifying the content of `ranger-admin-site.xml` in **Configuration Management**. When you modify and deliver `ranger-admin-`

`site.xml` based on your needs, the database information will be overwritten, resulting in an exception. This operation allows you to quickly configure a Ranger metadatabase, so as to avoid exceptions caused by missing configuration items when you change the metadatabase address.

# Directions

1. Log in to the EMR console and click the **ID/Name** of the corresponding cluster in the cluster list to go to the cluster details page.
2. On the cluster details page, select **Cluster Service** and click the target component block.
3. Taking HDFS NameNode active/standby switch as an example, in **Cluster Service**, select **HDFS** > **Operation** > **NN Failover** to perform the active/standby switch operation.

---

**NN Failover**                                                                    ✕

ⓘ   Backend Command: hdfs haadmin -failover

Service name          HDFS

Service Role          NameNode

Operation Parameters

     Select a NameService *    [ HDFS1000348                    ▾ ]

Operation Scope       ◉ Default node

Operation Reason *    [ Enter the operation reason                              ]

Enter no more than 200 characters

[ Confirm ]    Cancel

---

# HBase Table-Level Monitoring
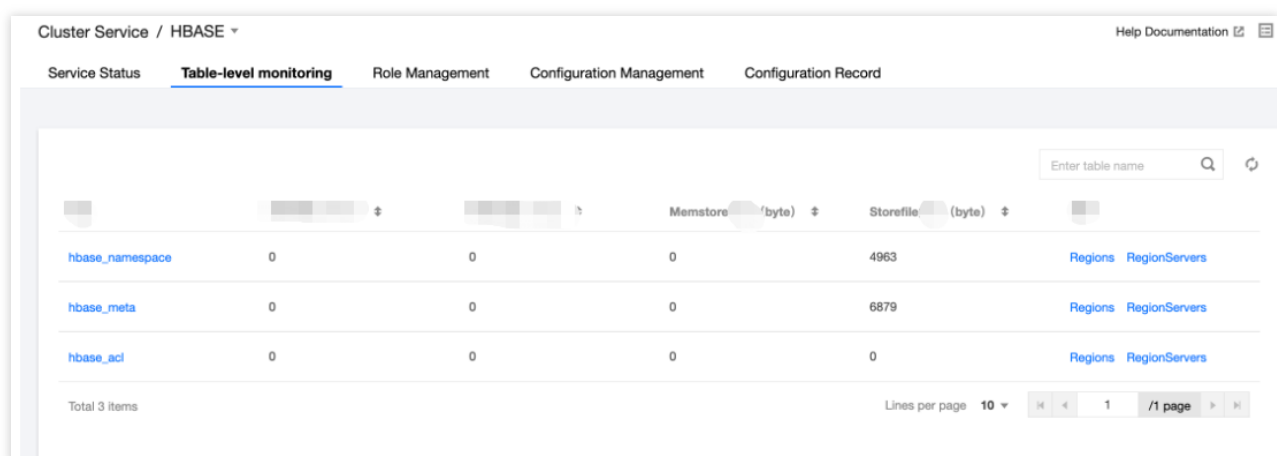
Last updated：2023-12-27 14:36:15

## Description

HBase table-level monitoring enables you to monitor the numbers of read and write requests and storage usage of each table in HBase.

## HBase Table Load List

1. Log in to the EMR console and click a **Cluster ID**/**Name** on the **Cluster List** page to enter the cluster details page.
2. On the cluster details page, click **Cluster Service** and select **Operation** > **Table-Level Monitoring** in the top-right corner of the HBase component block to query HBase table loads.



## Viewing Table Details

Click a table name to pop up the table details. The details page can display the numbers of read and write requests and size of store (including MemStore and StoreFile) of the selected table in the dimension of the entire table or node. You can switch between nodes by using the **Node Filter** in the top-right corner.
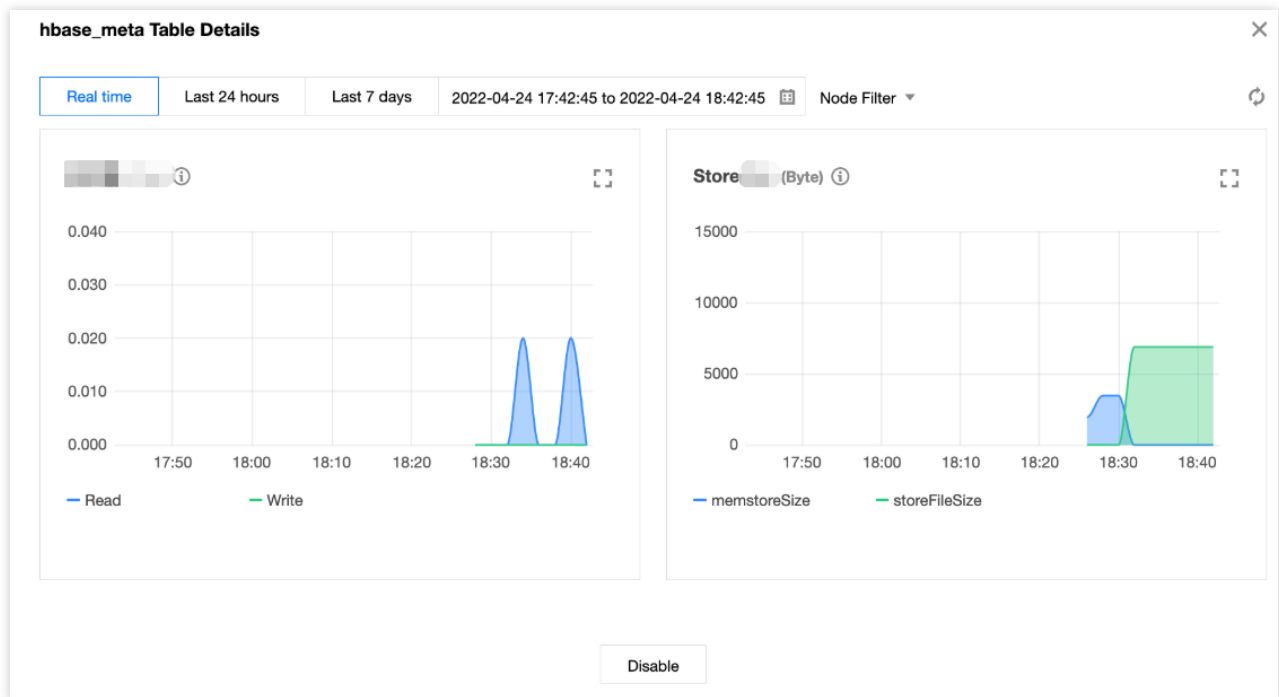
## Regions Operation

Click **Regions Operation** to view the numbers of read and write requests of each region contained in the table.



## Region Details

Click a region name to pop up the region details page, where the numbers of read and write requests of the selected table are displayed at different time granularities. Click **Time Granularity** in the top-right corner to switch between time granularities.

## RegionServers Operation

Click **RegionServers Operation** to view the request delay of each RegionServer where the table is distributed.

# Component Health Status

Last updated：2023-12-27 14:37:08

The component health status shows whether the current component is running normally. It is aggregated from the health statuses of all roles and can be viewed on the **Cluster Service** page.
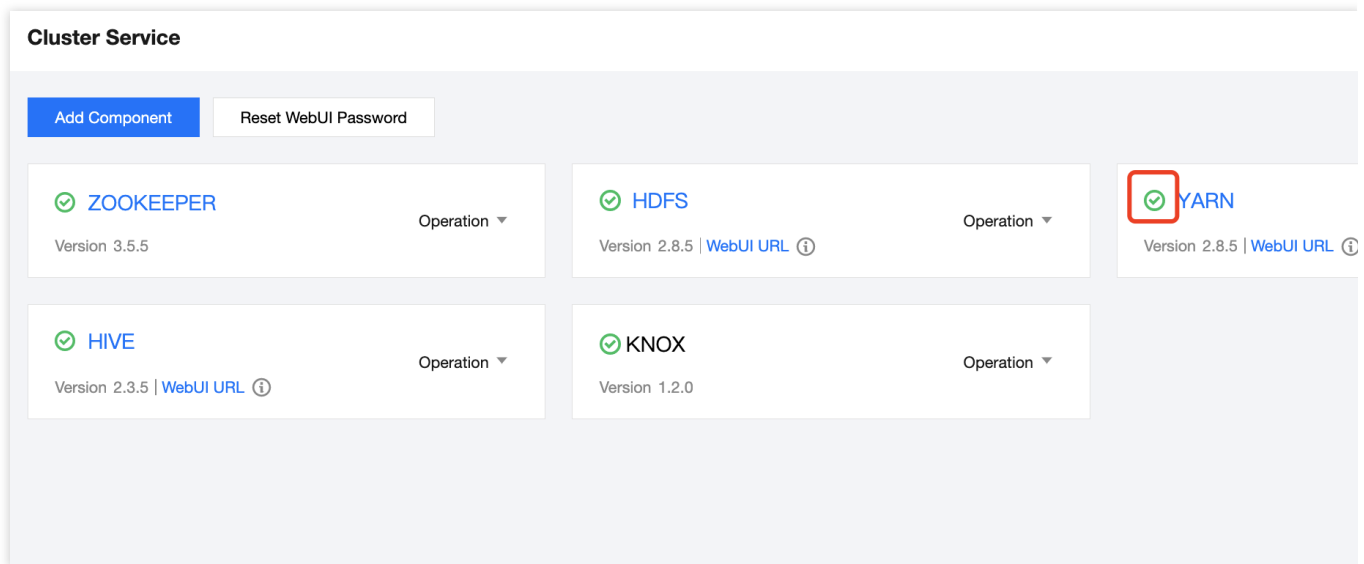


## Component Health Status

There are four component health statuses: Good, Risky, Unavailable, and Unknown or Not Checked. Different statuses are displayed in different colors.

| Component Health Status | Description | Status Aggregation Rule |
|---|---|---|
| Green: Good | The service is running normally. | The health statuses of all role instances are Good. |
| Orange: Risky | Service available. The health status of some role instances is Unavailable or Risky. Please handle. | The health statuses of some instances of a role under the component are Unavailable or Risky. For example, if HDFS has one NameNode role instance and two DataNode role instances, the health status of one DataNode role instance is Unavailable, and the health statuses of the other DataNode role instance and the NameNode role instance are Good, then the health status of HDFS is Risky. |
| Red: | Service unavailable. The health | The health statuses of all instances of a role under |

| Unavailable | status of all instances of a role is Unavailable. Please handle in time. | the component are UNAVAILABLE. For example, if HDFS has one NameNode role instance and two DataNode role instances, the health statuses of the two DataNode role instances are Unavailable, and the health status of the NameNode role instance is Good, the health status of HDFS is Unavailable. |
| --- | --- | --- |
| Gray: Unknown or Not Checked | Service health status Unknown or Not Checked. If there is no component, the health status will be Not Checked because no check is conducted; if there is a component, but it is in the maintenance mode or its operation status is Stopped, the health status will be Not Checked; if there is a component, but the health status of role instances cannot be obtained correctly, the health status will be Unknown. Ignore this if no problem is found in troubleshooting. | 1. The health statuses of all role instances of the component are not Risky or Unavailable, and the health status of at least one role instance is Unknown or Not Checked. For example, if HDFS has one NameNode role instance and two DataNode role instances, the health status of one DataNode role instance is Unknown or Not Checked, and the health statuses of the other DataNode role instance and the NameNode role instance are Good, the health status of HDFS is Unknown or Not Checked.<br>2. The component has no processes, so no check is conducted, such as Iceberg, Hudi, and Flink. |

# Monitoring and Alarms
# Cluster Overview

Last updated：2023-12-27 14:37:51

## Feature overview

Cluster overview displays a summary of cluster operation status, where you can view the cluster operation status, core service metrics, core node metrics, and top 10 nodes.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, select **Cluster Monitor** > **Cluster Overview** to view the overall status of the current cluster. The **Cluster Overview** page provides four monitoring views in the cluster dimension, namely, overall status of cluster, key cluster metrics, cluster deployment status, and top 10 nodes.
Overall cluster conditions: This view displays the current cluster status, node quantity, metadatabase online status, component health status, and events in the cluster within one hour.
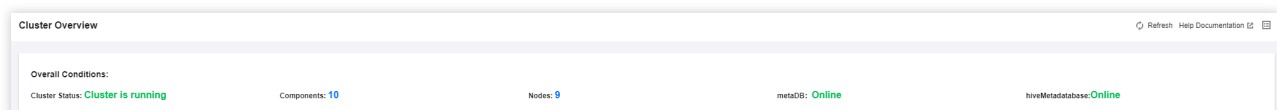


Key cluster metrics: This view displays the total utilization of HDFS, YARN, CPU, memory, and disk as well as total network traffic of the current cluster. You can click **View Trend** in the top-right corner of each metric and select a target time period to view the corresponding usage trend.
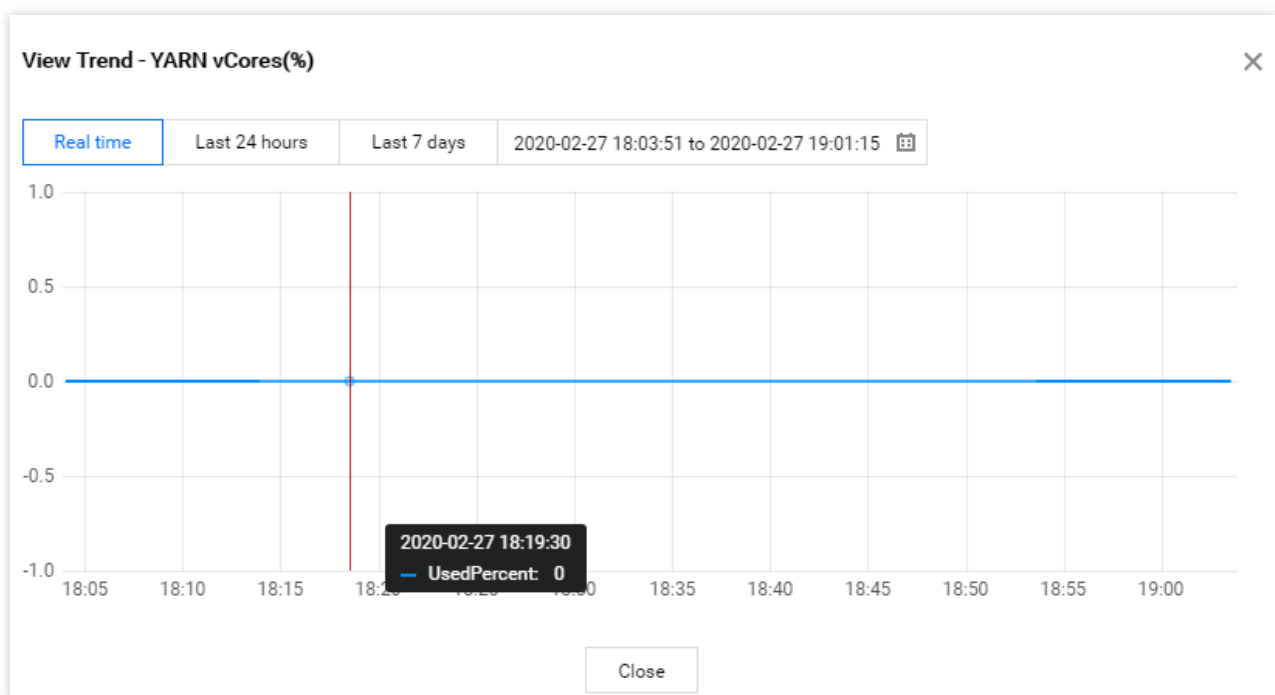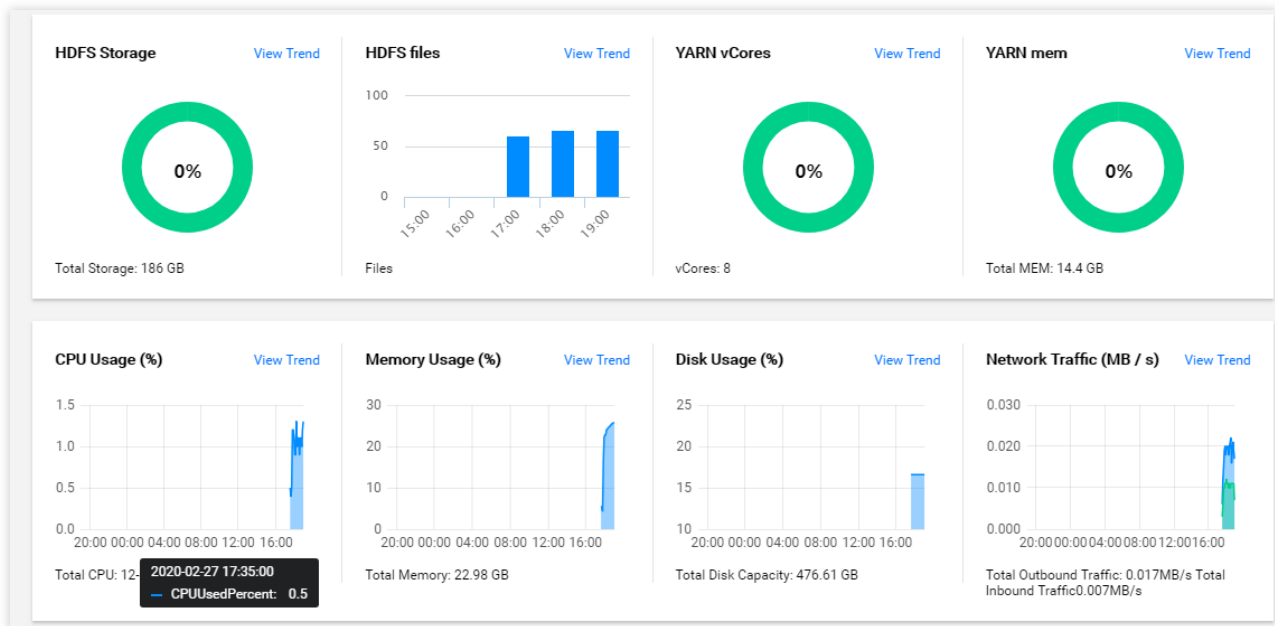
Cluster deployment status: This view displays abnormal, missing, and invalid processes deployed on nodes and the number of nodes in the current cluster, so that you can correct the configuration accordingly.

**Deployment Status**

| Node Type | Standard process (deployed) | Standard process (missing) | Invalid Process | Nodes |
|-----------|------------------------------|-----------------------------|------------------|-------|
| MASTER | HMaster,HistoryServer,Knox,NameNode,QuorumPeerMain,ResourceManager,SparkJobHistoryServer,ThriftServer | | | 1 |
| CORE | DataNode,NodeManager,RegionServer | | | 2 |

Top 10 nodes: This view displays the trend of node loads in the current cluster under core metrics. You can select multiple nodes to compare their load trends under the same metric.
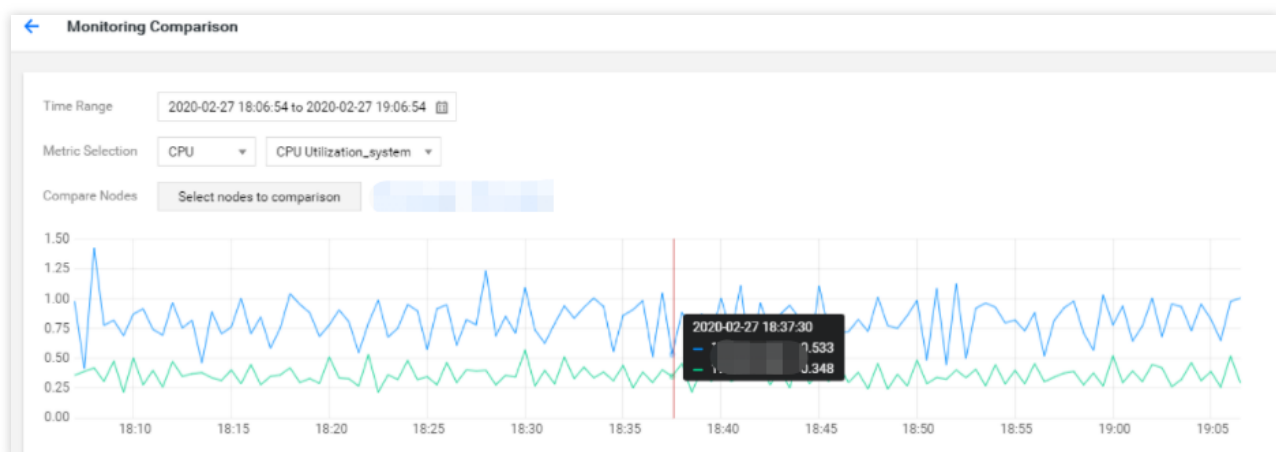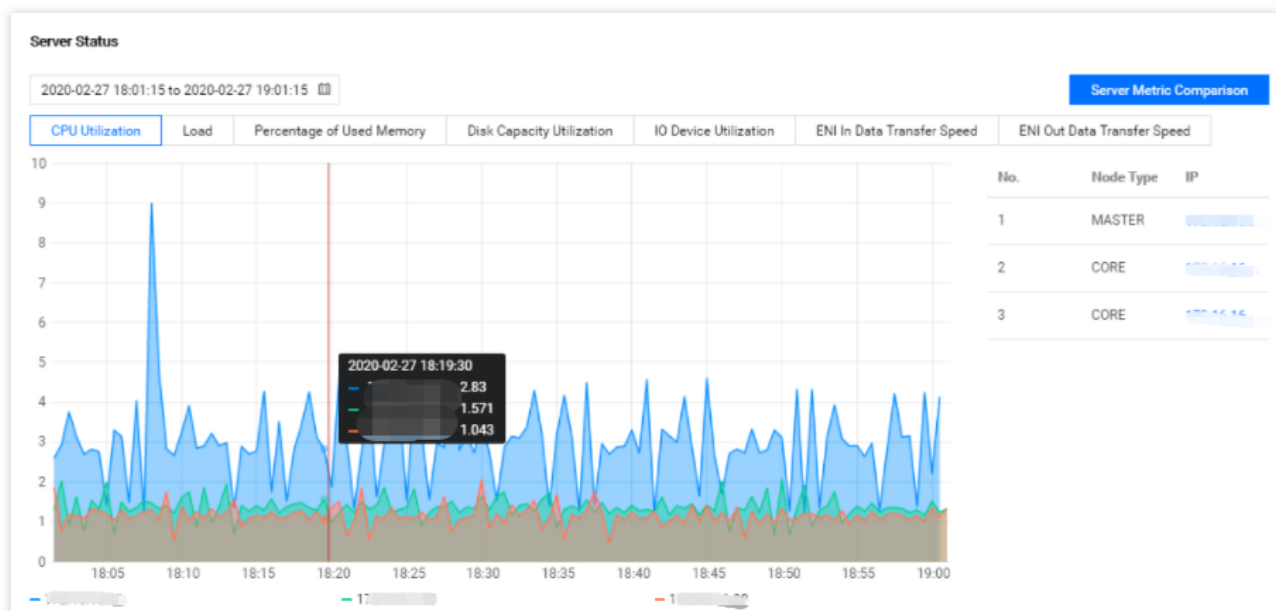
# Node Status

Last updated：2023-12-27 14:38:20

## Overview

The **Node status** page displays the monitoring overview and list of all nodes in the current cluster. It enables you to view heat maps of all nodes. You can manage the status information and metrics of nodes in the EMR console.

## Directions

1. Log in to the EMR console and click the **ID/Name** of the target cluster in the cluster list to enter the cluster details page.
2. Select **Cluster resources > Node status** to view the monitoring information of all nodes in the cluster.
3. On the **Node status** page, view aggregated monitoring metrics and list of all nodes in the cluster.
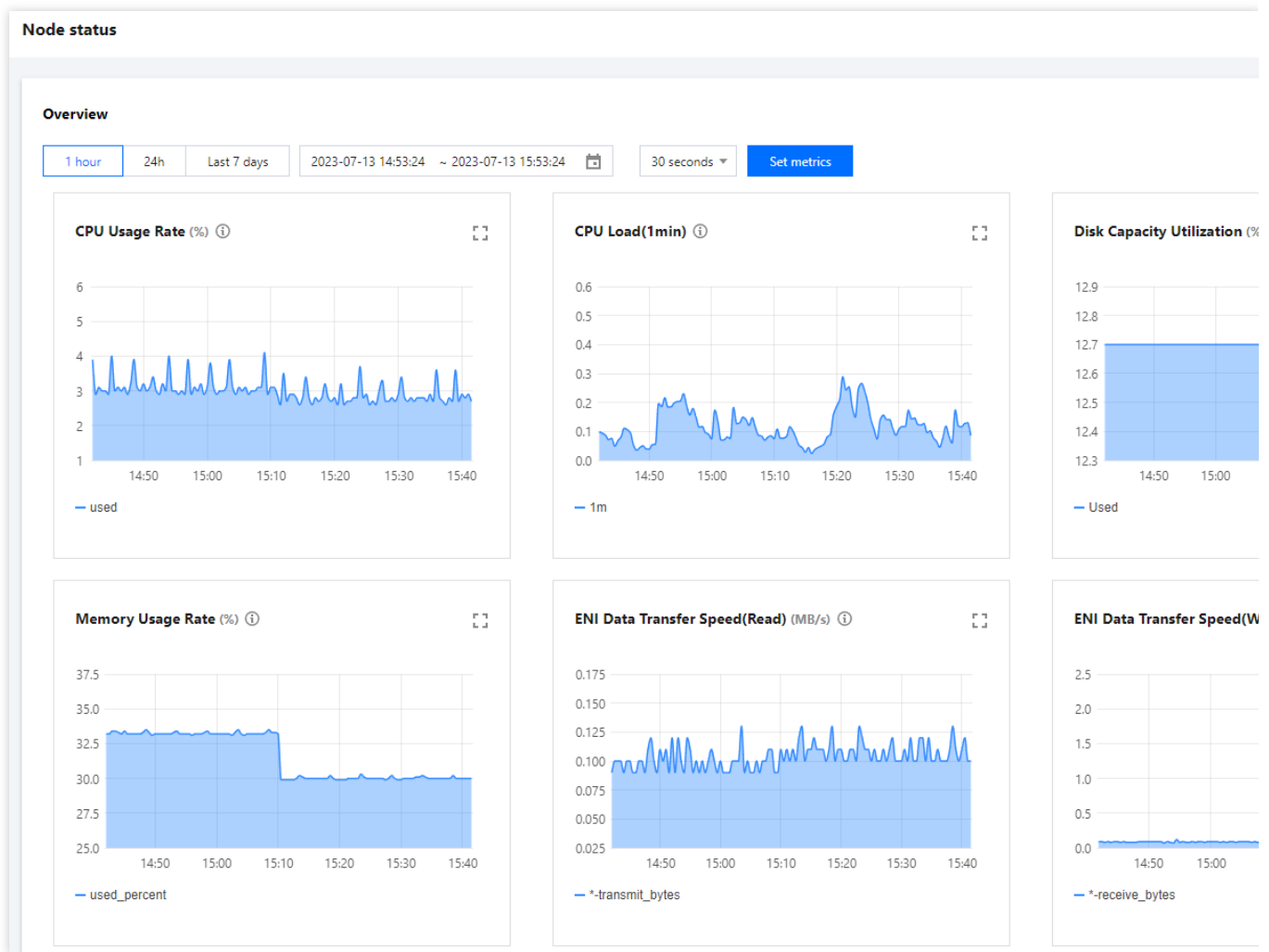
**Overview**: Displays aggregated monitoring metrics of all nodes during a specified period of time and the statistical rule of each metric. You can click **Set metrics** to customize the displayed metrics.

0.25

0.00

15:00          15:10          15:20          15:30          15:40

☐ CPU Load(5min)          Preview

☐ Device IOPS          Preview

☐ IO Operation Time          Preview

☑ Disk Capacity Utilization          Preview

☐ IO Device Utilization          Preview

☑ Memory Usage Rate          Preview

☑ ENI Data Transfer Speed(Read)          Preview

☑ ENI Data Transfer Speed(Write)          Preview

☐ Established TCP Connections          Preview

Confirm          Cancel
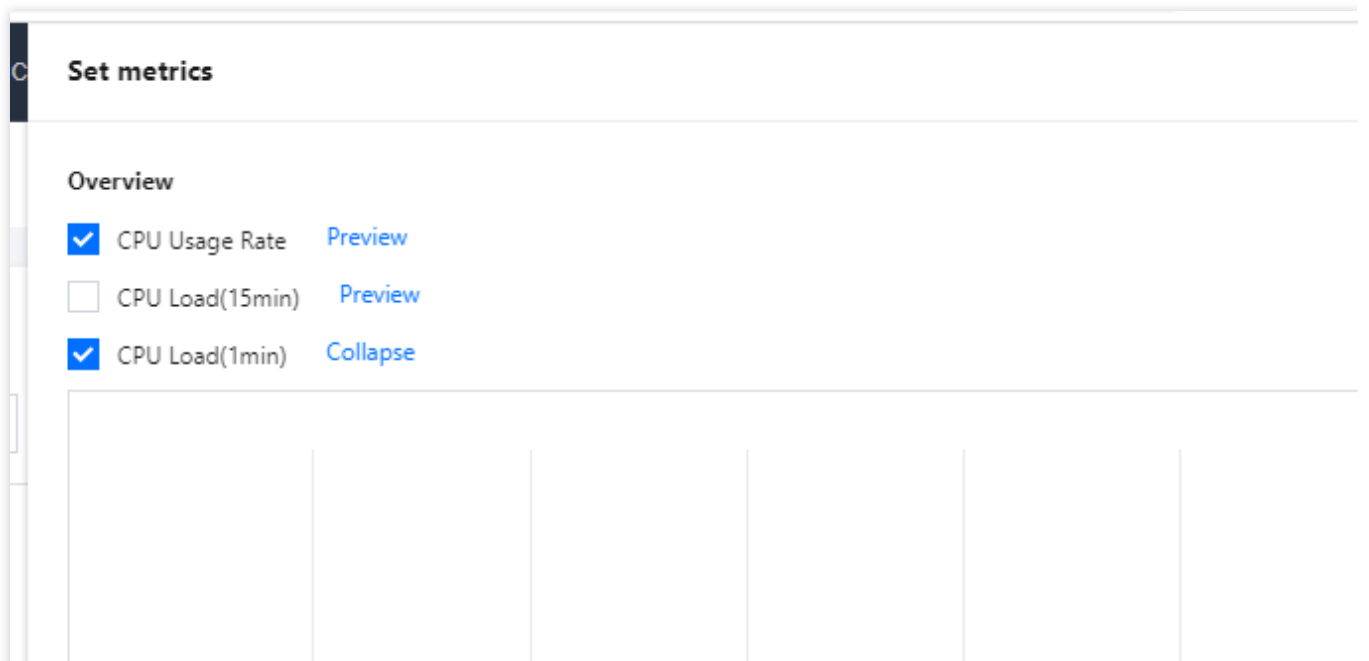
**Heat map**: Displays the loads of nodes. You can view the maps for a specified period of time and under specified load conditions. Load heat maps consist of two parts: the aggregated map of all nodes in the current cluster and individual heat map of each node.
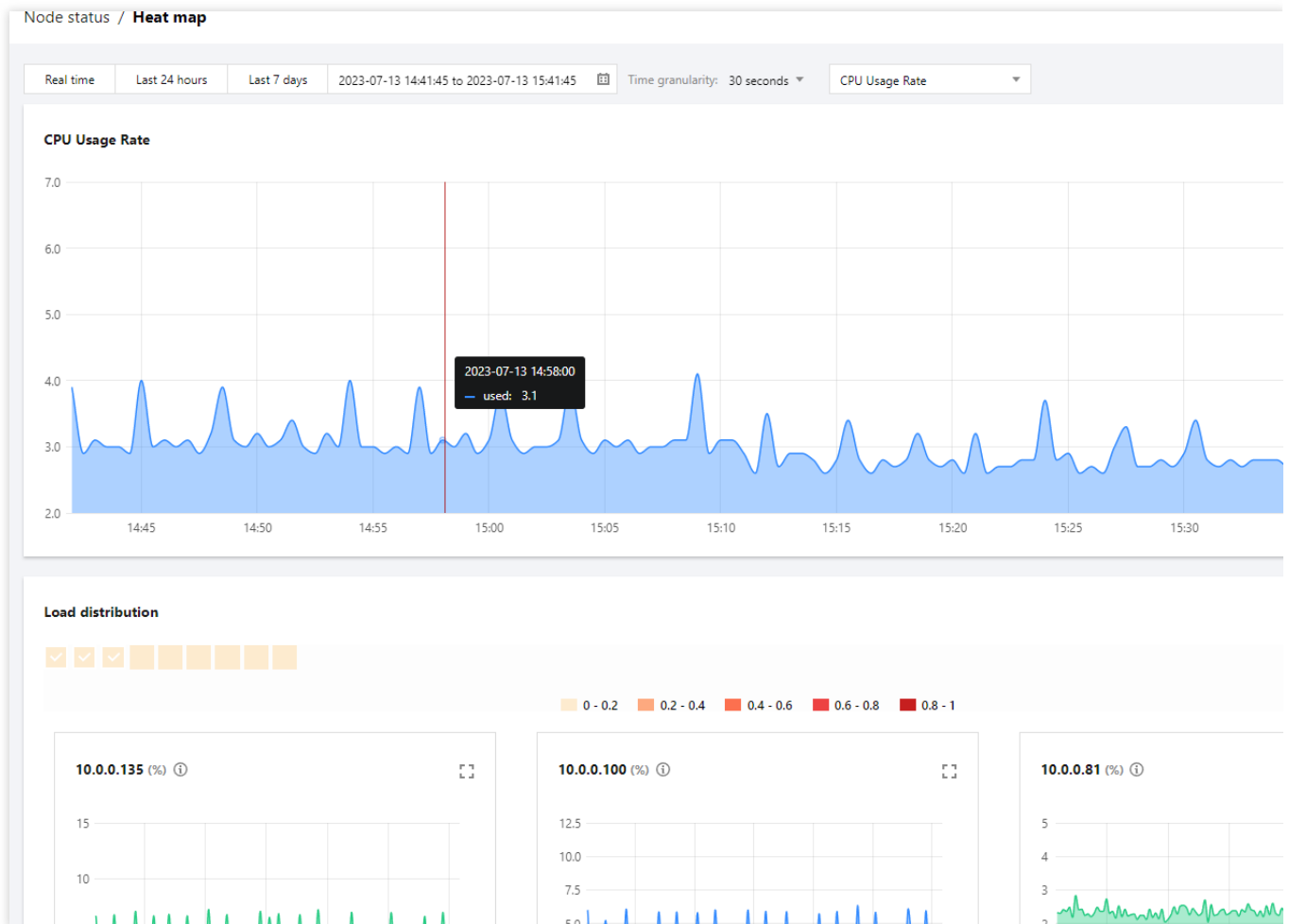
**Node list**: Displays all nodes in the cluster, including type, CPU utilization, memory utilization, disk utilization, and other items. You can click the IP of a node to view its basic configurations, deployment status, load status, and monitoring information.

Basic configurations: Display basic information of the node, such as type, resource name, resource ID, billing mode, and spec.
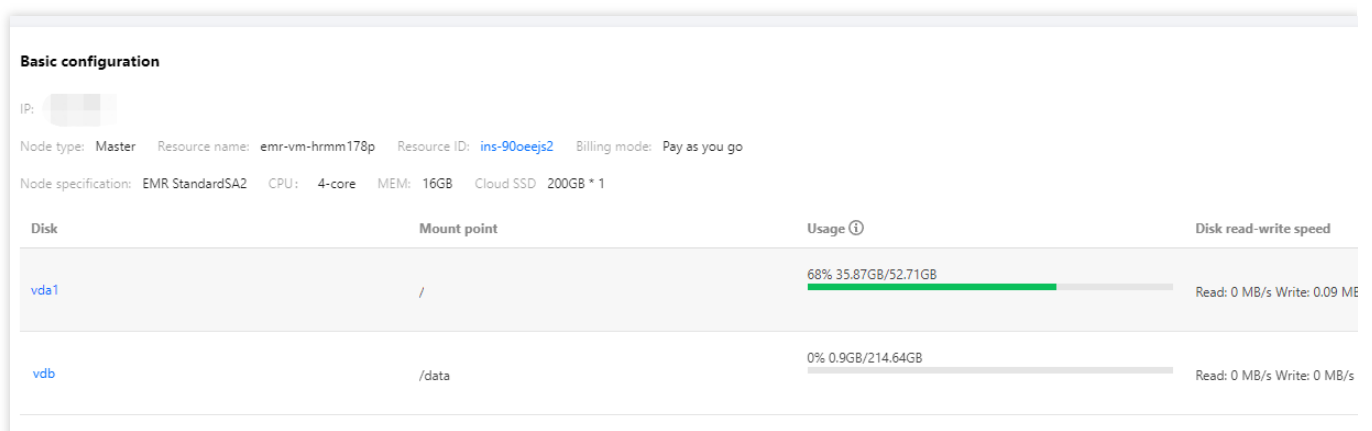


Deployment status: Display services deployed in the node, process type, and process status.

Load status: Displays top N processes in the node. You can view the data on a specified date.



Node monitoring: Displays load maps of metrics (6 by default and max 12) of the node. You can click **Set metrics** to customize the displayed metrics.

**Node monitoring**

| 1 hour | 24h | Last 7 days | 2023-07-13 14:55:41 ~ 2023-07-13 15:55:41 📅 | Time granularity | 30 seconds ▼ | Set metrics |

**CPU Utilization** (%) ⓘ

⌗

```
200

150

100

50

0
        14:50   15:00   15:10   15:20   15:30   15:40
```

— steal        — system        — guest
— iowait        — nice        — irq

**Disk Capacity Utilization** (%) ⓘ

⌗

```
13.05

13.00

12.95

12.90

12.85

12.80
        14:50   15:00   15:10   15:20   15:30   15:40
```

— Used(all)
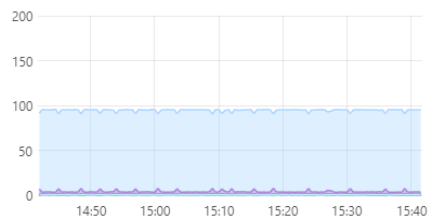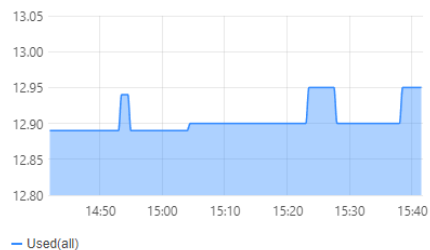
# Service Status

Last updated：2023-12-27 14:38:36

## Overview

The **Service status** page provides detailed monitoring for main services installed in the cluster, including HDFS, YARN, Hive, ZooKeeper, Spark, HBase, and Presto. This document describes how to view cluster service status in the console.

## Directions

1. Log in to the EMR console and click the **ID/Name** of the target cluster in the cluster list.
2. On the cluster details page, select **Cluster services** and click **Operation** > **Service status** in the top-right corner of the target component block. The following uses HDFS as an example.
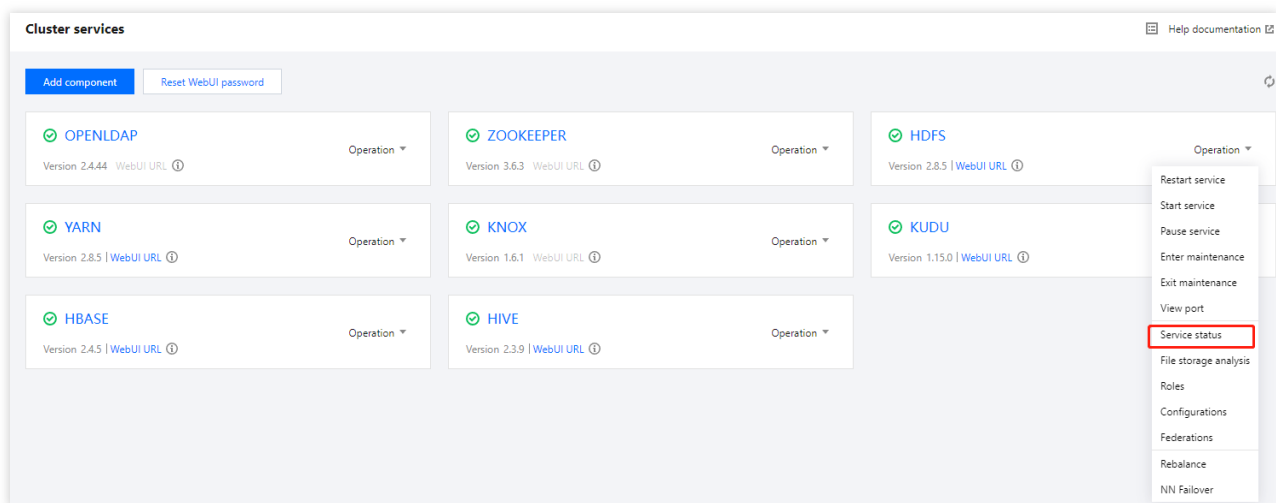


3. The service status page provides monitoring views of three service dimensions, namely, service summary, health status, and service overview. As different service components have different services, the dimensions of display vary.
4. **Service overview** displays 6 metrics of service components by default and their statistical rules during a specified time period. You can click **Set metrics** to customize the displayed metrics.

Service overview

1 hour | 24h | Last 7 days | 2023-07-13 13:18:01 ~ 2023-07-13 14:18:01 | Time granularity | 30 seconds ▼ | Set metrics

**Blocks** ⓘ

```
1500

1000

 500

   0
    13:20  13:30  13:40  13:50  14:00  14:10
```
— PendingDeletion...  — PendingReplicati...  — UnderReplicated...
— BlocksTotal  — ScheduledReplic...  — ExcessBlocks

**Cluster Storage Capacity** (GB) ⓘ

```
1250

1000

 750

 500

 250

   0
    13:20  13:30  13:40  13:50  14:00  14:10
```
— CapacityRemaini...  — CapacityUsed  — CapacityTotal
— CapacityUsedNo...

**Cluster Data Nodes** ⓘ

```
6

5

4

3

2

1

0
 13:20  13:30  13:40  13:50  14:00  14:10
```
— NumDecomDead...  — NumLiveDataNod...  — NumDecommissi...
— NumDecomLiveD...  — NumDeadDataNo...  — NumStaleDataNo...

**Total Cluster Files** ⓘ

```
775

770

765

760

755
 13:20  13:30  13:40  13:50  14:00  14:10
```
— FilesTotal

**Cluster Load** (1) ⓘ

```
45

40

35

30

25

20
 13:20  13:30  13:40  13:50  14:00  14:10
```
— TotalLoad

**Storage Usage Rate** (%) ⓘ

```
 1.0

 0.5

 0.0

-0.5

-1.0
 13:20  13:30  13:40  13:50  14:00  14:10
```
— CapacityUsedRate

**Set metrics**                                                              ✕

**Cluster Overview**

☑ Snapshots        Preview

☑ Block Capacity        Preview

☑ Blocks        Preview

☑ Cluster Storage Capacity        Preview

☑ Cluster Data Nodes        Preview

☑ Total Cluster Files        Preview

☑ Cluster Load        Preview

☑ Master/slave Conditions        Preview

☑ Storage Usage Rate        Preview

**Disk**

☑ Disk Failures        Preview

[ Confirm ]    [ Cancel ]

5. **Service summary** displays the current overall status of the service.

6. **Health status** displays running overview of each component in the current service. Clicking a role name or running overview will redirect you to **Roles** or **Role status** page.
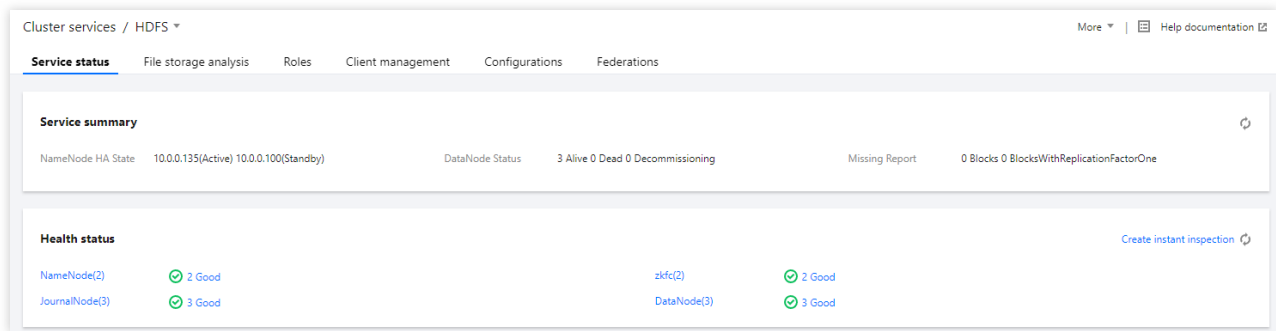
On a role status page, you can click **Set metrics** to customize the displayed metrics.
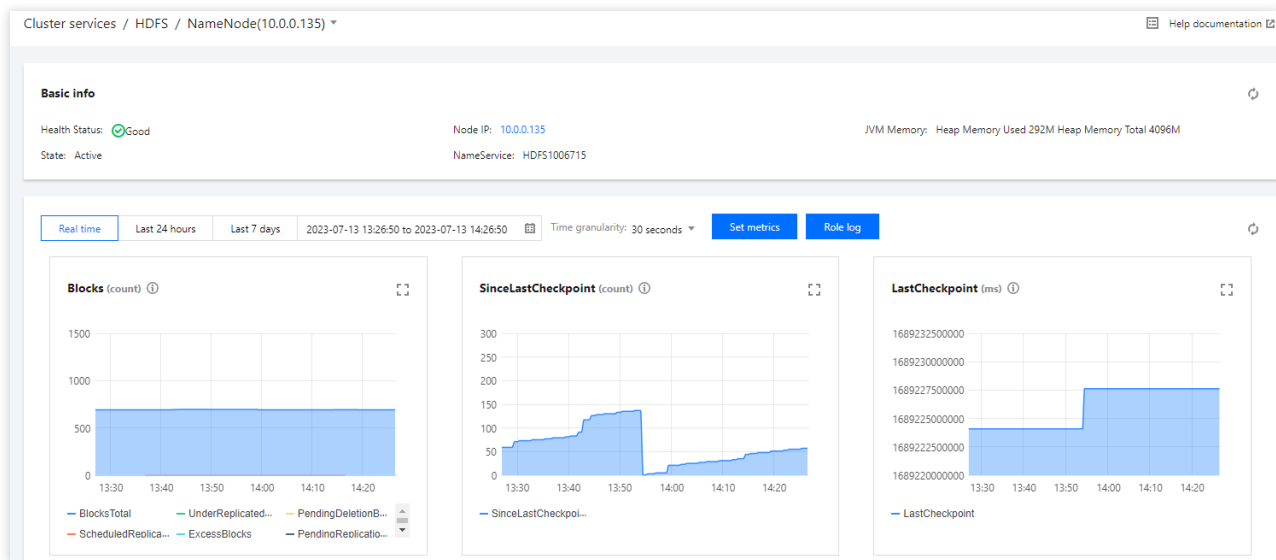


7. **Service overview** enables you to view cluster-level metrics. You can click **Set metrics** to customize the displayed metrics.
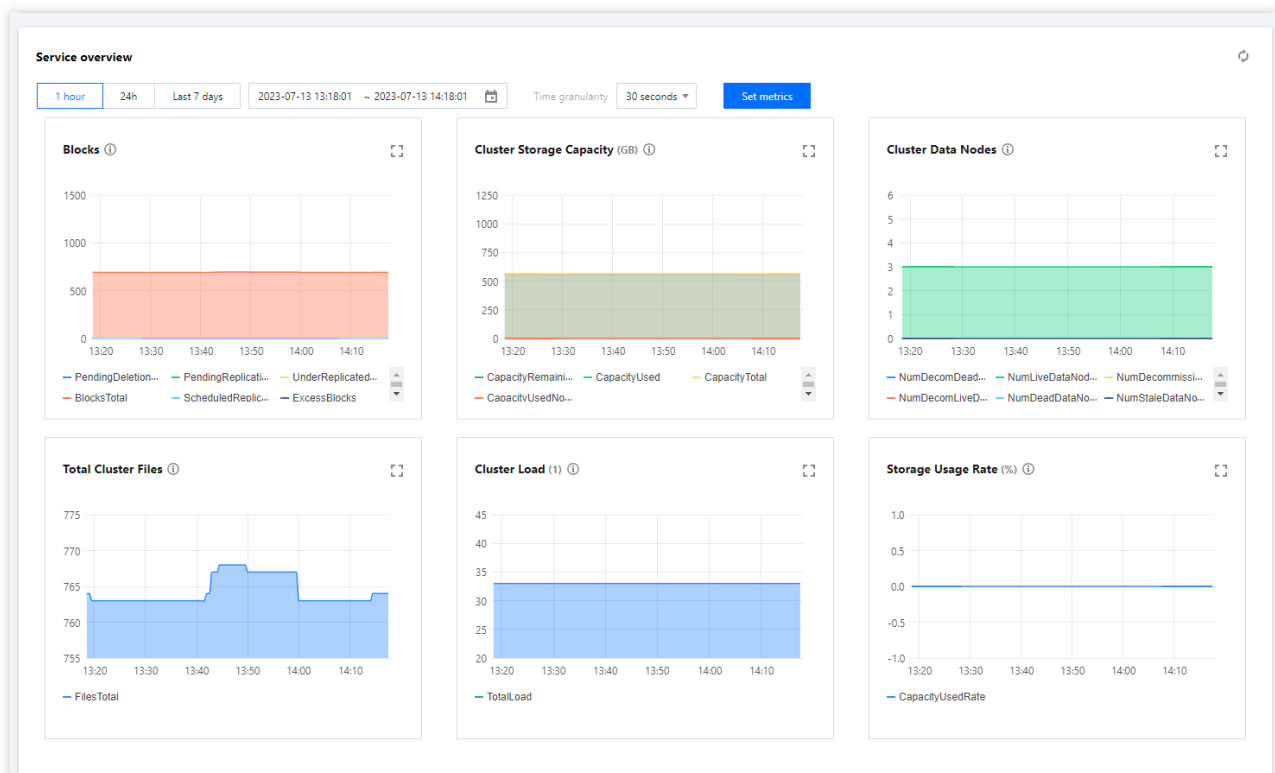
## Caution

**Service monitoring** displays the HDFS service information by default. You can manually adjust to view other service components.

Service monitoring dimensions vary by service component due to different service types. For example, HBase supports table-level monitoring.
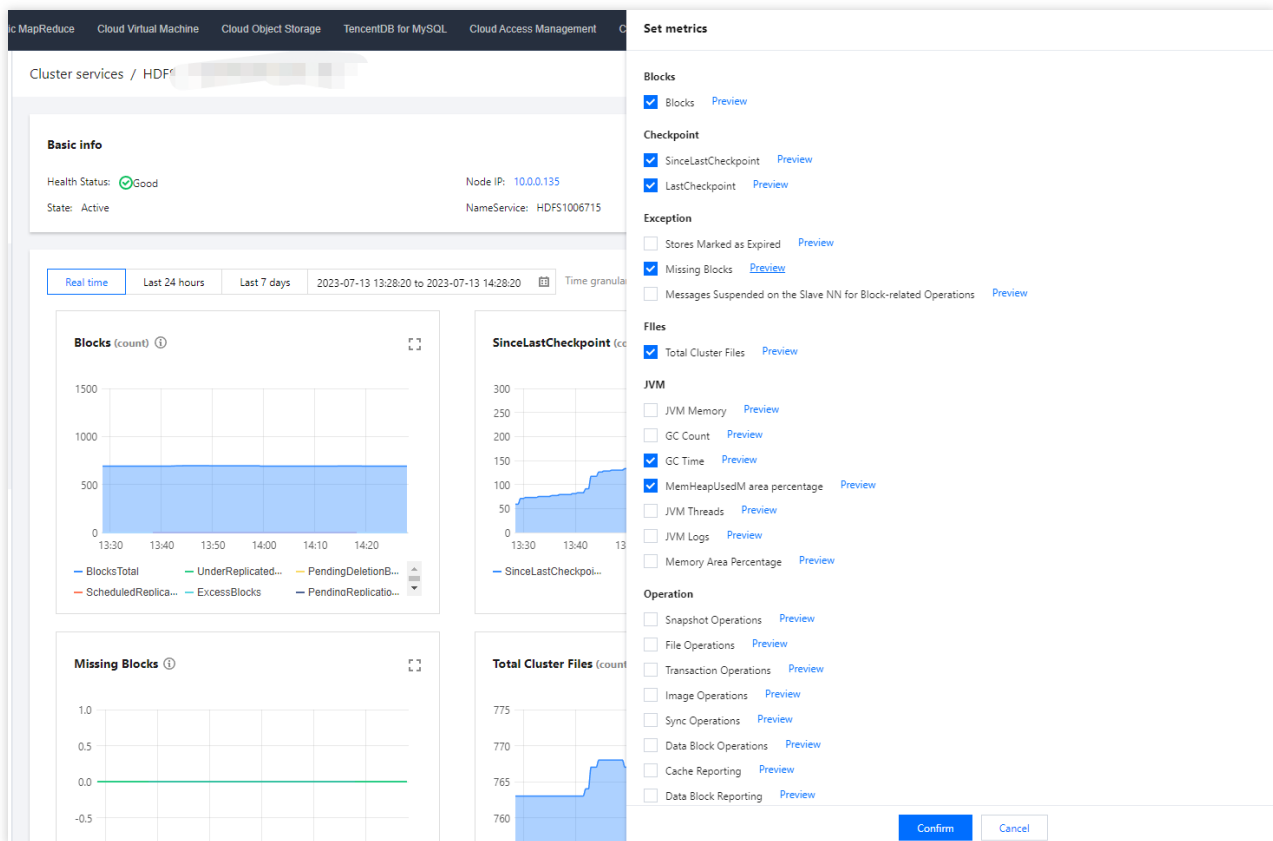
# Cluster Event

Last updated：2023-12-27 14:38:57

## Overview

Cluster events include event lists and event policies.

Event list: It records key change events and abnormal events occurring in the cluster.

Event policy: Event monitoring trigger policies can be customized based on the actual business conditions. Events with monitoring enabled can be set as cluster inspection items.

## Viewing Event List

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.

2. On the cluster details page, select **Cluster monitoring** > **Cluster events** > **Event list** to view all operation events in the current cluster.
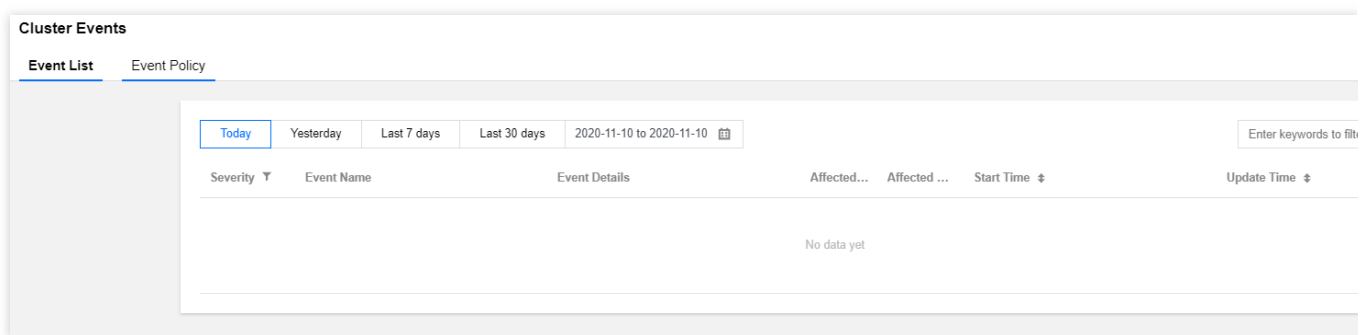


The severity divides into the following:

Fatal: Exception events of a node or service that require manual intervention and will cause service interruption if left unattended. Such events may last for a period of time.

Severe: Alert events that currently have not caused service or node interruption but will cause fatal events if left unattended.

Moderate: Regular events occurring in the cluster that generally do not require special processing.

3. Click the value in the **Triggers today** column to view the event's trigger records, metrics, logs, and snapshots.

# Setting Event Policies

1. Log in to the [EMR console](#) and click the **ID**/**Name** of the target cluster in the cluster list.

2. On the cluster details page, select **Cluster monitoring** > **Cluster events** > **Event policy** and you can customize the event monitoring trigger policies.

3. The event configuration list contains the event name, event trigger policy, severity (fatal, severe, and moderate), and option to enable/disable monitoring, which can be modified and saved.



4. Event trigger policies cover two types of events: fixed system policy events (which cannot be modified) and custom events (which can be configured based on the business standards).

5. You can select whether to enable event monitoring in an event policy. Only events with monitoring enabled can be selected as cluster inspection items. Monitoring is enabled by default for some events and is enabled by default and cannot be disabled for some other events. The following are the specific rules:

| Category | Event Name | Description | Recommendations and Measure | Default Value |
|---|---|---|---|---|
| Node | The CPU utilization exceeds the threshold continuously | The server CPU utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=85, t=1,800 |
| | The average CPU utilization exceeds the threshold | The average server CPU utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=85, t=1,800 |

| | | | |
|---|---|---|---|
| The average CPU iowait utilization exceeds the threshold | The average CPU iowait utilization of the server in the last t (300 ≤ t ≤ 2,592,000) seconds has been greater than or equal to m | Manually troubleshoot the issue | m=60, t=1,800 |
| The 1-second CPU load exceeds the threshold continuously | The 1-minute CPU load has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=8, t=1,800 |
| The 5-second CPU load exceeds the threshold continuously | The 5-minute CPU load has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=8, t=1,800 |
| The memory utilization exceeds the threshold continuously | The memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=85, t=1,800 |
| The swap space exceeds the threshold continuously | The server swap memory has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=0.1, t=300 |
| The total number of system processes exceeds the threshold continuously | The total number of system processes has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=10,000, t=1,800 |
| The average total number of fork subprocesses exceeds the threshold | The average total number of fork subprocesses in the last t (300 ≤ t ≤ 2,592,000) seconds | Manually troubleshoot the issue | m=5,000, t=1,800 |

| | | | |
|---|---|---|---|
| | has been greater than or equal to m | | |
| No process OOM | An OOM error occurred in the process | Adjust the process heap memory size | - |
| A disk I/O error occurred (this event is not supported currently) | A disk I/O error occurred | Replace the disk | - |
| The average disk utilization exceeds the threshold continuously | The average disk space utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=85, t=1,800 |
| The average disk I/O utilization exceeds the threshold continuously | The average disk I/O utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=85, t=1,800 |
| The node file handle utilization exceeds the threshold continuously | The node file handle utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=85, t=1,800 |
| The number of TCP connections to the node exceeds the threshold continuously | The number of TCP connections to the node has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Check whether there are connection leaks | m=10,000, t=1,800 |
| The configured node memory utilization exceeds the threshold | The memory utilization configured for all roles on the node exceeds the node's physical memory threshold | Adjust the allocated node process heap memory | 90% |
| The node process is | The node service | View the service logs | - |

| unavailable | process is unavailable | to find out why the service failed to be pulled | |
|---|---|---|---|
| The node heartbeat is missing | The node heartbeat failed to be reported regularly | Manually troubleshoot the issue | - |
| The hostname is incorrect | The node's hostname is incorrect | Manually troubleshoot the issue | - |
| Failed to ping the metadatabase | The TencentDB instance heartbeat failed to be reported regularly | - | - |
| The utilization of a single disk exceeds the threshold continuously | The single disk space utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=0.85, t=1,800 |
| The I/O utilization of a single disk exceeds the threshold continuously | The single disk I/O device utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=0.85, t=1,800 |
| The single disk inodes utilization exceeds the threshold continuously | The single disk inodes utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=0.85, t=1,800 |
| The difference between the UTC time and NTP time of the server exceeds the threshold | The difference between the UTC time and NTP time of the server exceeds the threshold (in ms) | 1. Make sure that the NTP daemon is running 2. Make sure that the network communication with the NTP server is normal | Difference=3000( |
| Automatic node | If automatic node | 1. If the replenishment | - |

| | | | | |
|---|---|---|---|---|
| | replenishment | replenishment is enabled, when any exceptions in task and router nodes are detected, the system automatically purchases nodes of the same model to replace the affected nodes. | is successful, no more attention is required. 2. If the replenishment fails, manually terminate the affected nodes in the console and purchase new nodes to replace them. | |
| | Node failure | Faulty nodes exist in a cluster | Handle the failure in the consoleor submit a ticket to contact us. | - |
| HDFS | The total number of HDFS files exceeds the threshold continuously | The total number of files in the cluster has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Increase the NameNode memory | m=50,000,000, t=1,800 |
| | The average total number of HDFS files exceeds the threshold | The average total number of files in the cluster in the last t (300 ≤ t ≤ 2,592,000) seconds has been greater than or equal to m | Increase the NameNode memory | m=50,000,000, t=1,800 |
| | The total number of HDFS blocks exceeds the threshold continuously | The total number of blocks in the cluster has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Increase the NameNode memory or the block size | m=50,000,000, t=1,800 |
| | The average total number of HDFS blocks exceeds the threshold | The average total number of HDFS blocks in the last t (300 ≤ t ≤ 2,592,000) seconds has been greater than or equal to m | Increase the NameNode memory or the block size | m=50,000,000, t=1,800 |
| | | | | |

| The number of HDFS data nodes marked as dead exceeds the threshold continuously | The number of data nodes marked as dead has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=1, t=1,800 |
|---|---|---|---|
| The HDFS storage space utilization exceeds the threshold continuously | The HDFS storage space utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Clear files in HDFS or expand the cluster capacity | m=85, t=1,800 |
| The average HDFS storage space utilization exceeds the threshold | The average HDFS storage space utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Clear files in HDFS or expand the cluster capacity | m=85, t=1,800 |
| Active/Standby NameNodes were switched | Active/Standby NameNodes were switched | Locate the cause of NameNode switch | - |
| The NameNode RPC request processing latency exceeds the threshold continuously | The RPC request processing latency has been greater than or equal to m milliseconds for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=300, t=300 |
| The number of current NameNode connections exceeds the threshold continuously | The number of current NameNode connections has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=2,000, t=1,800 |
| A full GC event occurred on a NameNode | A full GC event occurred on a NameNode | Fine-tune the parameter settings | - |
| The NameNode JVM | The NameNode JVM | Adjust the NameNode | m=85, t=1,800 |

| memory utilization exceeds the threshold continuously | memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | heap memory size | - |
|---|---|---|---|
| The DataNode RPC request processing latency exceeds the threshold continuously | The RPC request processing latency has been greater than or equal to m milliseconds for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=300, t=300 |
| The number of current DataNode connections exceeds the threshold continuously | The number of current DataNode connections has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=2,000, t=1,800 |
| A full GC event occurred on a DataNode | A full GC event occurred on a NameNode | Fine-tune the parameter settings | - |
| The DataNode JVM memory utilization exceeds the threshold continuously | The NameNode JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the DataNode heap memory size | m=85, t=1,800 |
| Both NameNodes of HDFS are in Standby service status | Both NameNode roles are in Standby status at the same time | Manually troubleshoot the issue | - |
| The number of HDFS missing blocks exceeds the threshold | The number of missing blocks in the cluster has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | We recommend you troubleshoot HDFS data block corruption and run the hadoop fsck / command to check the HDFS file distribution | m=1, t=1,800 |
| The HDFS NameNode entered the safe mode | The NameNode entered the safe mode | We recommend you troubleshoot HDFS | - |

| | | (for 300 seconds continuously) | data block corruption and run the hadoop fsck / command to check the HDFS file distribution | |
|---|---|---|---|---|
| YARN | The number of currently missing NodeManagers in the cluster exceeds the threshold continuously | The number of currently missing NodeManagers in the cluster has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Check the NodeManager process status and check whether the network connection is smooth | m=1, t=1,800 |
| | The number of pending containers exceeds the threshold continuously | The number of pending containers has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Reasonably specify resources that can be used by YARN jobs | m=90, t=1,800 |
| | The cluster memory utilization exceeds the threshold continuously | The memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Scale out the cluster | m=85, t=1,800 |
| | The average cluster memory utilization exceeds the threshold | The average memory utilization in the last t (300 ≤ t ≤ 2,592,000) seconds has been greater than or equal to m | Scale out the cluster | m=85, t=1,800 |
| | The cluster CPU utilization exceeds the threshold continuously | The CPU utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Scale out the cluster | m=85, t=1,800 |
| | The average cluster CPU utilization exceeds the threshold | The average CPU utilization in the last t (300 ≤ t ≤ 2,592,000) seconds has been | Scale out the cluster | m=85, t=1,800 |

| | greater than or equal to m | | |
|---|---|---|---|
| The number of available CPU cores in each queue is below the threshold continuously. | The number of available CPU cores in each queue has been less than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Assign more resources to the queue | m=1, t=1,800 |
| The available memory in each queue is below the threshold continuously | The available memory in each queue has been less than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Assign more resources to the queue | m=1,024, t=1,800 |
| Active/Standby ResourceManagers were switched | Active/Standby ResourceManagers were switched | Check the ResourceManager process status and view the standby ResourceManager logs to locate the cause of active/standby switch | - |
| A full GC event occurred in a ResourceManager | A full GC event occurred in a ResourceManager | Fine-tune the parameter settings | - |
| The ResourceManager JVM memory utilization exceeds the threshold continuously | The ResourceManager JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the ResourceManager heap memory size | m=85, t=1,800 |
| A full GC event occurred in a NodeManager | A full GC event occurred in a NodeManager | Fine-tune the parameter settings | - |
| The available memory in NodeManager is | The available memory in a single NodeManager has | Adjust the NodeManager heap memory size | m=1, t=1,800 |

| | | | | |
|---|---|---|---|---|
| | below the threshold continuously | been less than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | | |
| | The NodeManager JVM memory utilization exceeds the threshold continuously | The NodeManager JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the NodeManager heap memory size | m=85, t=1,800 |
| HBase | The number of regions in RIT status in the cluster exceeds the threshold continuously | The number of regions in RIT status in the cluster has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | If the HBase version is below 2.0, run hbase hbck -fixAssigment | m=1, t=60 |
| | The number of dead RegionServers exceeds the threshold continuously | The number of dead RegionServers has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=1, t=300 |
| | The average number of regions in each RegionServer in the cluster exceeds the threshold continuously | The average number of regions in each RegionServer in the cluster has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Expand the node capacity or upgrade the node | m=300, t=1,800 |
| | A full GC event occurred on HMaster | A full GC event occurred on HMaster | Fine-tune the parameter settings | m=5, t=300 |
| | The HMaster JVM memory utilization exceeds the threshold continuously | The HMaster JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the HMaster heap memory size | m=85, t=1800 |

| The number of current HMaster connections exceeds the threshold continuously | The number of current HMaster connections has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=1000, t=1800 |
|---|---|---|---|
| A full GC event occurred in RegionServer | A full GC event occurred in RegionServer | Fine-tune the parameter settings | m=5, t=300 |
| The RegionServer JVM memory utilization exceeds the threshold continuously | The RegionServer JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the RegionServer heap memory size | m=85, t=1800 |
| The number of current RPC connections to RegionServer exceeds the threshold continuously | The number of current RPC connections to RegionServer has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=1000, t=1800 |
| The number of RegionServer StoreFiles exceeds the threshold continuously | The number of RegionServer StoreFiles has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Run the major compaction | m=50,000, t=1,800 |
| A full GC event occurred in HBase Thrift | A full GC event occurred in HBase Thrift | Fine-tune the parameter settings | m=5, t=300 |
| The HBase Thrift JVM memory utilization exceeds the threshold continuously | The HBase Thrift JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the HBase Thrift heap memory size | m=85, t=1,800 |
| Both HMasters of | Both HMaster roles | Manually troubleshoot | - |

| | | | | |
|---|---|---|---|---|
| | HBase is in Standby service status | are in Standby status at the same time | the issue | |
| Hive | A full GC event occurred in HiveServer2 | A full GC event occurred in HiveServer2 | Fine-tune the parameter settings | m=5, t=300 |
| | The HiveServer2 JVM memory utilization exceeds the threshold continuously | The HiveServer2 JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Adjust the HiveServer2 heap memory size | m=85, t=1,800 |
| | A full GC event occurred in HiveMetaStore | A full GC event occurred in HiveMetaStore | Fine-tune the parameter settings | m=5, t=300 |
| | A full GC event occurred in HiveWebHcat | A full GC event occurred in HiveWebHcat | Fine-tune the parameter settings | m=5, t=300 |
| ZooKeeper | The number of ZooKeeper connections exceeds the threshold continuously | The number of ZooKeeper connections has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=65,535, t=1,800 |
| | The number of ZNodes exceeds the threshold continuously | The number of ZNodes has been greater than or equal to m for t (300 ≤ t ≤ 2,592,000) seconds continuously | Manually troubleshoot the issue | m=2,000, t=1,800 |
| Impala | The ImpalaCatalog JVM memory utilization exceeds the threshold continuously | The ImpalaCatalog JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the ImpalaCatalog heap memory size | m=0.85, t=1,800 |
| | The Impala daemon JVM memory | The Impala daemon JVM memory | Adjust the Impala daemon heap memory | m=0.85, t=1,800 |

| | | | | |
|---|---|---|---|---|
| | utilization exceeds the threshold continuously | utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | size | |
| | The number of Impala Beeswax API client connections exceeds the threshold | The number of Impala Beeswax API client connections has been greater than or equal to m | Adjust the value of fs_sevice_threads in the impalad.flgs configuration in the console | m=64, t=120 |
| | The number of Impala HiveServer2 client connections exceeds the threshold | The number of Impala HiveServer2 client connections has been greater than or equal to m | Adjust the value of fs_sevice_threads in the impalad.flgs configuration in the console | m=64, t=120 |
| | The query execution duration exceeds the threshold | The query execution duration exceeds m seconds | Manually troubleshoot the issue | - |
| | The total number of failed queries exceeds the threshold | The total number of failed queries has been greater than or equal to m for t seconds (300 ≤ t ≤ 604,800) | Manually troubleshoot the issue | m=1, t=300 |
| | The total number of committed queries exceeds the threshold | The total number of committed queries has been greater than or equal to m for t seconds (300 ≤ t ≤ 604,800) | Manually troubleshoot the issue | m=1, t=300 |
| | The query execution failure rate exceeds the threshold | The query execution failure rate has been greater than or equal to m for t seconds (300 ≤ t ≤ 604,800) | Manually troubleshoot the issue | m=1, t=300 |
| PrestoSQL | The current number of failed PrestoSQL | The current number of failed PrestoSQL nodes has been | Manually troubleshoot the issue | m=1, t=1,800 |

| | | | | |
|---|---|---|---|---|
| | nodes exceeds the threshold continuously | greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | | |
| | The number of queuing resources in the current PrestoSQL resource group exceeds the threshold continuously | The number of queuing tasks in the PrestoSQL resource group has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Fine-tune the parameter settings | m=5,000, t=1,800 |
| | The number of failed PrestoSQL queries exceeds the threshold | The number of failed PrestoSQL queries is greater than or equal to m | Manually troubleshoot the issue | m=1, t=1,800 |
| | A full GC event occurred in a PrestoSQLCoordinator | A full GC event occurred in a PrestoSQLCoordinator | Fine-tune the parameter settings | - |
| | The PrestoSQLCoordinator JVM memory utilization exceeds the threshold continuously | The PrestoSQLCoordinator JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the PrestoSQLCoordinator heap memory size | m=0.85, t=1,800 |
| | A full GC event occurred on a PrestoSQL worker | A full GC event occurred on a PrestoSQL worker | Fine-tune the parameter settings | - |
| | The PrestoSQLWorker JVM memory utilization exceeds the threshold continuously | The PrestoSQLWorker JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the PrestoSQLWorker heap memory size | m=0.85, t=1,800 |
| Presto | The current number of failed Presto nodes | The current number of failed Presto nodes has been greater than | Manually troubleshoot the issue | m=1, t=1,800 |

| | | | | |
|---|---|---|---|---|
| | exceeds the threshold continuously | or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | | |
| | The number of queuing resources in the current Presto resource group exceeds the threshold continuously | The number of queuing tasks in the Presto resource group has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Fine-tune the parameter settings | m=5,000, t=1,800 |
| | The number of failed Presto queries exceeds the threshold | The number of failed Presto queries is greater than or equal to m | Manually troubleshoot the issue | m=1, t=1,800 |
| | A full GC event occurred on a PrestoSQL coordinator | A full GC event occurred on a PrestoSQL coordinator | Fine-tune the parameter settings | - |
| | The Presto coordinator JVM memory utilization exceeds the threshold continuously | The Presto coordinator JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the Presto coordinator heap memory size | m=0.85, t=1,800 |
| | A full GC event occurred on a Presto worker | A full GC event occurred on a Presto worker | Fine-tune the parameter settings | - |
| | The Presto worker JVM memory utilization exceeds the threshold continuously | The Presto worker JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the Presto worker heap memory size | m=0.85, t=1,800 |
| Alluxio | The current total number of Alluxio workers is below the threshold continuously | The current total number of Alluxio workers has been smaller than or equal to m for t (300 ≤ t ≤ | Manually troubleshoot the issue | m=1, t=1,800 |

| | | | |
|---|---|---|---|
| | | | 604,800) seconds continuously | | |
| | The utilization of the capacity on all tiers of the current Alluxio worker exceeds the threshold | The utilization of the capacity on all tiers of the current Alluxio worker has been greater than or equal to the threshold for t (300 ≤ t ≤ 604,800) seconds continuously | Fine-tune the parameter settings | m=0.85, t=1,800 |
| | A full GC event occurred on an Alluxio master | A full GC event occurred on an Alluxio master | Manually troubleshoot the issue | - |
| | The Alluxio master JVM memory utilization exceeds the threshold continuously | The Alluxio master JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the Alluxio worker heap memory size | m=0.85, t=1,800 |
| | A full GC event occurred on an Alluxio worker | A full GC event occurred on an Alluxio worker | Manually troubleshoot the issue | - |
| | The Alluxio worker JVM memory utilization exceeds the threshold continuously | The Alluxio worker JVM memory utilization has been greater than or equal to m for t (300 ≤ t ≤ 604,800) seconds continuously | Adjust the Alluxio master heap memory size | m=0.85, t=1,800 |
| kudu | The cluster replica skew exceeds the threshold | The cluster replica skew has been greater than or equal to the threshold for t (300 ≤ t ≤ 3,600) seconds continuously | Run the rebalance command to balance the replicas | m=100, t=300 |
| | The hybrid clock error exceeds the threshold | The hybrid clock error has been greater than or equal to the threshold for t (300 ≤ t | Make sure that the NTP daemon is running and the network | m=5,000,000, t=300 |

| | | ≤ 3,600) seconds continuously | communication with the NTP server is normal | |
| --- | --- | --- | --- | --- |
| | The number of running tablets exceeds the threshold | The number of running tablets has been greater than or equal to m for t (300 ≤ t ≤ 3,600) seconds continuously | Too many tablets on a node can affect the performance. We recommend you clear unnecessary tables and partitions or expand the capacity as needed. | m=1,000, t=300 |
| | The number of failed tablets exceeds the threshold | The number of failed tablets has been greater than or equal to m for t (300 ≤ t ≤ 3,600) seconds continuously | Check whether any disk is unavailable or data file is corrupted | m=1, t=300 |
| | The number of failed data directories exceeds the threshold | The number of failed data directories has been greater than or equal to m for t (300 ≤ t ≤ 3,600) seconds continuously | Check whether the path configured in the fs_data_dirs parameter is available | m=1, t=300 |
| | The number of full data directories exceeds the threshold | The number of full data directories has been greater than or equal to m for t (120 ≤ t ≤ 3,600) seconds continuously | Clear unnecessary data files or expand the capacity as needed | m=1, t=120 |
| | The number of write requests rejected due to queue overloading exceeds the threshold | The number of write requests rejected due to queue overloading has been greater than or equal to m for t (300 ≤ t ≤ 3,600) seconds continuously | Check whether the number of write hotspots or worker threads is small | m=10, t=300 |
| | The number of expired scanners exceeds the threshold | The number of expired scanners has been greater than or equal to m for t (300 ≤ t ≤ | Be sure to call the method for closing a scanner after reading data | m=100, t=300 |

| | | 3,600) seconds continuously | | |
|---|---|---|---|---|
| | The number of error logs exceeds the threshold | The number of error logs has been greater than or equal to m for t (300 ≤ t ≤ 3,600) seconds continuously | Manually troubleshoot the issue | m=10, t=300 |
| | The number of RPC requests that timed out while waiting in the queue exceeds the threshold | The number of RPC requests that timed out while waiting in the queue has been greater than or equal to m for t (300 ≤ t ≤ 3,600) seconds continuously | Check whether the system load is too high | m=100, t=300 |
| Kerberos | The Kerberos response time exceeds the threshold | The Kerberos response time has been greater than or equal to m (ms) for t (300 ≤ t ≤ 604,800) seconds continuously | Manually troubleshoot the issue | m=100, t=1,800 |
| Cluster | The auto scaling policy has failed | 1. The scale-out rule failed due to insufficient subnet EIPs associated with the cluster.<br>2. The scale-out rule failed due to insufficient expansion resource inventory of the preset specifications.<br>3. The scale-out rule failed due to insufficient account balance.<br>4. An internal error occurred. | 1. Switch to another subnet in the same VPC.<br>2. Switch to specifications of resources that are sufficient or submit a ticket to contact developers.<br>3. Top up the account to ensure that the account balance is sufficient.<br>4. Submit a ticket to contact developers. | - |
| | The execution of the auto scaling policy has timed out | 1. Scaling cannot be performed temporarily | 1. Adjust the cooldown period for the scaling rule. | - |

| | | | |
|---|---|---|---|
| | | as the cluster is in the cooldown period.<br>2. Scaling is not triggered because the retry period upon expiration is too short.<br>3. The cluster in the current status cannot be scaled out. | 2. Extend the retry period upon expiration.<br>3. Try again later or submit a ticket to contact developers. | |
| | The auto scaling policy is not triggered | 1. The scale-out rule cannot be triggered because no expansion resource specification is set.<br>2. The scale-out rule cannot be triggered because the maximum number of nodes for elastic resources is reached.<br>3. The scale-in rule cannot be triggered because the minimum number of nodes for elastic resources is reached.<br>4. The time range for scaling has expired.<br>5. The scale-in rule cannot be triggered because there are no elastic resources in the cluster. | 1. Set at least one elastic resource specification for the rule.<br>2. Modify the maximum number of nodes to continue scaling out if the upper limit is reached.<br>3. Modify the minimum number of nodes to continue scaling in if the lower limit is reached.<br>4. Modify the effective time range of the rule if you want to continue using the rule for auto scaling.<br>5. Execute the scale-in rule after adding elastic resources. | - |
| | Auto scaling partially succeeded | 1. Only partial resources were supplemented because the resource inventory was less than the required quantity for scale-out.<br>2. Only partial resources were supplemented because the required | 1. Use the available resources for manual scaling to supplement the resources for auto scaling.<br>2. Use the available resources for manual scaling to supplement the resources for auto scaling. | - |

| | | | |
|---|---|---|---|
| | quantity for scale-out exceeded the actual quantity of resources delivered.<br>3. The scale-out rule was partially successful because the maximum number of nodes for elastic resources was reached.<br>4. The scale-in rule was partially successful because the minimum number of nodes for elastic resources was reached.<br>5. The resource supplement failed due to insufficient subnet EIPs associated with the cluster.<br>6. The resource supplement failed due to insufficient expansion resource inventory of the preset specifications.<br>7. The resource supplement failed due to insufficient account balance. | 3. Modify the maximum number of nodes to continue scaling out if the upper limit is reached.<br>4. Modify the minimum number of nodes to continue scaling in if the lower limit is reached.<br>5. Switch to another subnet in the same VPC.<br>6. Switch to specifications of resources that are sufficient or submit a ticket to contact developers.<br>7. Top up the account to ensure that the account balance is sufficient. | |
| The node process is unavailable | The node process is unavailable | Manually troubleshoot the issue | - |
| The process is killed by OOMKiller | The process OOM is killed by OOMKiller | Adjust the process heap memory size | - |
| A JVM or OLD exception occurred | A JVM or OLD exception occurred | Manually troubleshoot the issue | 1. The OLD utilization reaches 80% for 5 consecutive minutes<br>Or |

| | | | 2. The JVM memory utilization reaches 90% |
|---|---|---|---|
| Timeout of service role health status occurred | The service role health status has timed out for t seconds (180 ≤ t ≤ 604,800) | The service role health status has timed out for minutes. To resolve this issue, check the logs of the corresponding service role and perform necessary actions accordingly. | t=300 |
| A service role health status exception occurred | The service role health status has been abnormal for t seconds (180 ≤ t ≤ 604,800) | The service role health status has been unavailable for minutes. To resolve this issue, check the logs of the corresponding service role and perform necessary actions accordingly. | t=300 |
| Auto scaling failed | This alert indicates that the auto scaling process has failed (either completely or partially) | Manually troubleshoot the issue | - |

# Log

Last updated：2024-01-10 09:56:47

## Overview

The log search feature collects component operation logs. Then, you can search for core service logs of the current cluster and node system logs by keyword to quickly view key service logs without logging in to nodes.

## Directions

1. Log in to the [EMR console](#) and click the **ID/Name** of the target cluster in the cluster list.

2. On the cluster details page, choose **Cluster Monitoring** > **Log Search** to filter and view log content by current cluster, log file, node IP, and time range.

 Alternatively, choose **Cluster Service** > target component block > **Role Management**, click a **Node IP** in **Role List** to enter the page where the node monitoring metrics are displayed, and then click **Role Log** to enter the **Log Search** page.
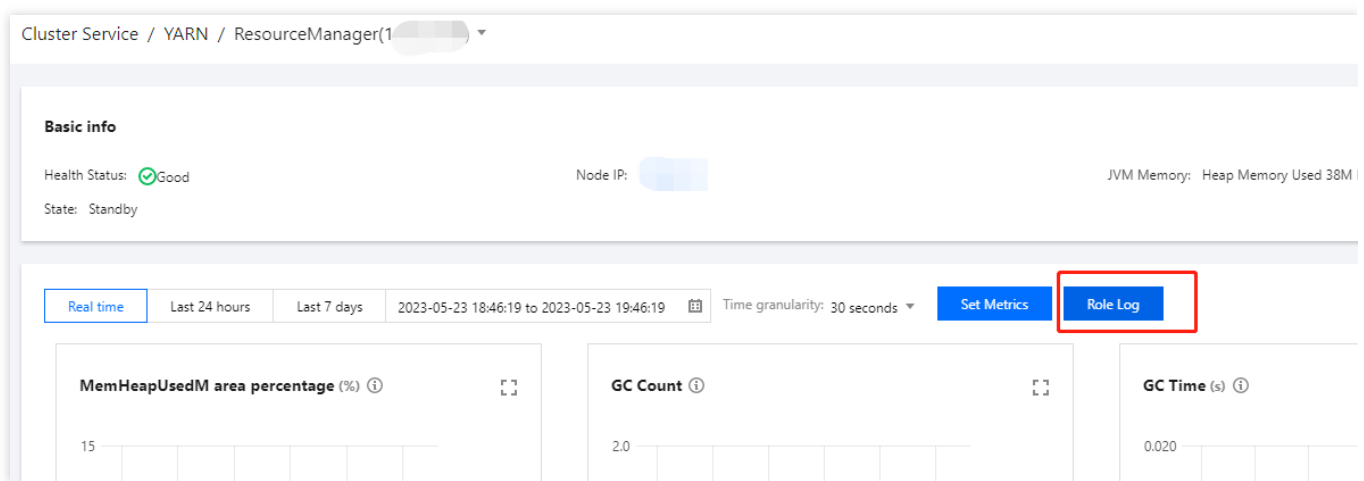


Click a **Node IP** to enter the **Node Status** page, or click **Log Source** to enter the page where the node monitoring metrics are displayed.

**Keyword description:**

Supports full-text search based on keywords.

Supports search by using special symbols, including hyphen (-), period (.), asterisk (*), greater-than sign (>), less-than sign (<), equal sign (=), exclamation mark (!), brackets (() and {}), and forward slash (/).

Supports phrase-based search. Example: `address=/ip:port` .

3. During troubleshooting, you usually need to pay attention to contextual logs of keywords. On the **Log Search** page, click **View Context** to enter the **Log Context** page.

## Service Types Supported for Log Search

**Note:**

Currently, only logs in the past 15 days can be searched for.

If log collection has not been enabled on your cluster, you can contact your Tencent Cloud rep for assistance.

| Component | Role | Log | Description |
|-----------|------|-----|-------------|
| HDFS | NameNode | /data/emr/hdfs/logs/hadoop-hadoop-namenode.log | NameNode runtime log |
| | ZKFC | /data/emr/hdfs/logs/hadoop-hadoop-zkfc.log | ZKFC runtime log |
| | DataNode | /data/emr/hdfs/logs/hadoop-hadoop-datanode.log | DataNode runtime log |
| | JournalNode | /data/emr/hdfs/logs/hadoop-hadoop-journalnode.log | JournalNode runtime log |
| | DFSRouter | /data/emr/hdfs/logs/hadoop-hadoop-dfsrouter.log | DFRouter runtime log |

| | | | |
|---|---|---|---|
| YARN | ResourceManager | /data/emr/yarn/logs/yarn-hadoop-resourcemanager.log | ResourceManager runtime log |
| | NodeManager | /data/emr/yarn/logs/yarn-hadoop-nodemanager.log | NodeManager runtime log |
| | JobHistoryServer | /data/emr/yarn/logs/mapred-hadoop-historyserver.log | JobHistoryServer runtime log |
| HBase | HMaster | /data/emr/hbase/logs/hbase-hadoop-master.log | HMaster runtime log |
| | ThriftServer | /data/emr/hbase/logs/hbase-hadoop-thrift.log | ThriftServer runtime log |
| | RegionServer | /data/emr/hbase/logs/hbase-hadoop-regionserver.log | RegionServer runtime log |
| ClickHouse | ClickHouse-server | /data/clickhouse/clickhouse-server/logs/clickhouse-server.log | ClickHouse-server runtime log |
| Druid | Broker | /data/emr/druid/var/log/druid/broker.log | Broker runtime log |
| | Coordinator | /data/emr/druid/var/log/druid/coordinator.log | Coordinator runtime log |
| | Router | /data/emr/druid/var/log/druid/router.log | Router runtime log |
| | Overload | /data/emr/druid/var/log/druid/overload.log | Overload runtime log |
| | Historical | /data/emr/druid/var/log/druid/historical.log | Historical runtime log |
| | MiddleManager | /data/emr/druid/var/log/druid/middleManager.log | MiddleManager runtime log |
| Zookeeper | Zookeeper | /data/emr/zookeeper/logs/zookeeper-root-server.log | ZooKeeper runtime log |
| Hive | HiveServer2 | /data/emr/hive/logs/hadoop-hive | HiveServer2 runtime log |
| Kudu | KuduMaster | /data/emr/kudu/logs/kudu-master.WARNING | KuduMaster runtime log |
| | KuduServer | /data/emr/kudu/logs/kudu-tserver.WARNING | KuduServer |

| | | | runtime log |
|---|---|---|---|
| Alluxio | AlluxioMaster | /data/emr/alluxio/logs/master.log | AlluxioMaster runtime log |
| | AlluxioWorker | /data/emr/alluxio/logs/worker.log | AlluxioWorker runtime log |
| Ranger | EmbeddedServer | /data/emr/ranger/logs/ranger-admin.log | EmbeddedServer runtime log |
| CosRanger | CosRangerServer | /usr/local/service/cosranger/log/info.log | COSRanger runtime log |
| Impala | Catalogd | /data/emr/impala/logs/catalogd.INFO | Cataloged runtime log |
| | Statestored | /data/emr/impala/logs/statestored.INFO | Statestored runtime log |
| | Impalad | /data/emr/impala/logs/impalad.INFO | Impalad runtime log |
| Spark | HistoryServer | /data/emr/spark/logs/spark-hadoop.log | HistoryServer runtime log |
| Kylin | Kylin | /data/emr/kylin/logs/kylin.log | Kylin runtime log |
| Zeppelin | ZeppelinServer | /data/emr/zepplin/logs/zeppelin-hadoop.log | ZeppelinServer runtime log |
| Knox | Gateway | /data/emr/knox/logs/gateway.log | Gateway runtime log |
| Doris | BrokerBootstrap | /data/emr/doris/broker/log/apache_hdfs_broker.log | BrokerBootstrap runtime log |
| | PaloFe | /data/emr/doris/fe/log/fe.log | PaloFe runtime log |
| | PaloBe | /data/emr/doris/be/log/be.INFO | PaloBe runtime log |
| Kafka | Kafka | /user/local/service/kafka/logs/server.log | Kafka runtime log |

## Minimum Log Level Supported by Services

| Service | Default Minimum Log Level |
|---------|---------------------------|
| Impala and Kudu | INFO |
| Other services | WARN |

# Query Rules for the Minimum Log Level

| Minimum Log Level | Queryable Log Levels |
|-------------------|----------------------|
| INFO | INFO, WARN, ERROR, FATAL |
| WARN | WARN, ERROR, FATAL |
| ERROR | ERROR, FATAL |
| FATAL | FATAL |

# Application Analysis
# YARN Job Query

Last updated：2023-12-27 14:43:45

## Overview

You can view submitted jobs, memory and vCore consumption, and other information at the user granularity, and quickly check the queue, status, duration, and other metrics of YARN jobs in the EMR console. Comparison of historical tasks by job, job insight, task execution information, and other features are also available.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list.
2. On the cluster details page, click **Cluster services** and select **Operation** > **Job query** in the top-right corner of the YARN component to view job statistics and resource consumption trends, query job and task information, get insights into application execution results, and compare application monitoring data.
 i. Submitted jobs, view and distribution of memory and vCores consumed at the user granularity, and recent trends of relevant metrics.

ii. Job filtering by user, application name, queue name, job type, duration, or relevant throughput metrics.



iii. Resource consumption statistics by user, queue, or other metrics, helping check costs (also supported by API).



**Caution**

To use the new features of task information, application insight, and application comparison for applications of Spark type, you first need to check whether the Spark History version is supported by running `curl "http://localhost:10000/api/v1/applications" | json_pp`. If the returned result is in abnormal JSON format, the Spark History version is not supported. In this case, you can submit a ticket to apply for enabling such features.

In job query, ResourceManager data is collected once every 30s. The collection operation has negligible impact on cluster running.

3. In the job list, click **More** > **Application insight** to view detailed application insight items and related insight rules, results, and suggestions.

**Application insight**

| Insight item | Level ▼ | Rule | Result/Recommendat |
|---|---|---|---|
| Map Data Skew | Severe | The data processed by the task is more than twice the … | ✓ |
| Map Memory waste | Medium | The idle Memory resources account for more than 20… | ✓ |
| Map Slow Task | Medium | The task processing time is more than twice the avera… | ✓ |
| MapperTaskGC | Medium | The GC time of executor exceeds 20% | ✓ |
| Reduce Data Skew | Severe | The data processed by the task is more than twice the … | ✓ |
| Reduce Memory waste | Medium | The idle Memory resources account for more than 20… | ✓ |
| Reduce Slow Task | Medium | The task processing time is more than twice the avera… | ✓ |

**Caution**

To ensure stable cluster running, no collection will be performed when any of the following rules is met:

1.1 Apps for which the runtime is less than 10 minutes.

1.2 Apps for which the number of subtasks during collection is greater than 30,000.

1.3 Apps for which the collection delay time is longer than 24 hours.

You can submit a ticket to modify relevant parameters of insight collection policy.

**Risk description**

YARN's application insight feature collects and analyzes the application data of Spark History, Job History, and Timeline Server. If you find the number of requests of this service constantly exceeds the load threshold, you can submit a ticket to disable the feature.

4. In the job list, click **More** > **Application comparison** to compare the business metrics between the current application and another application of the same type.
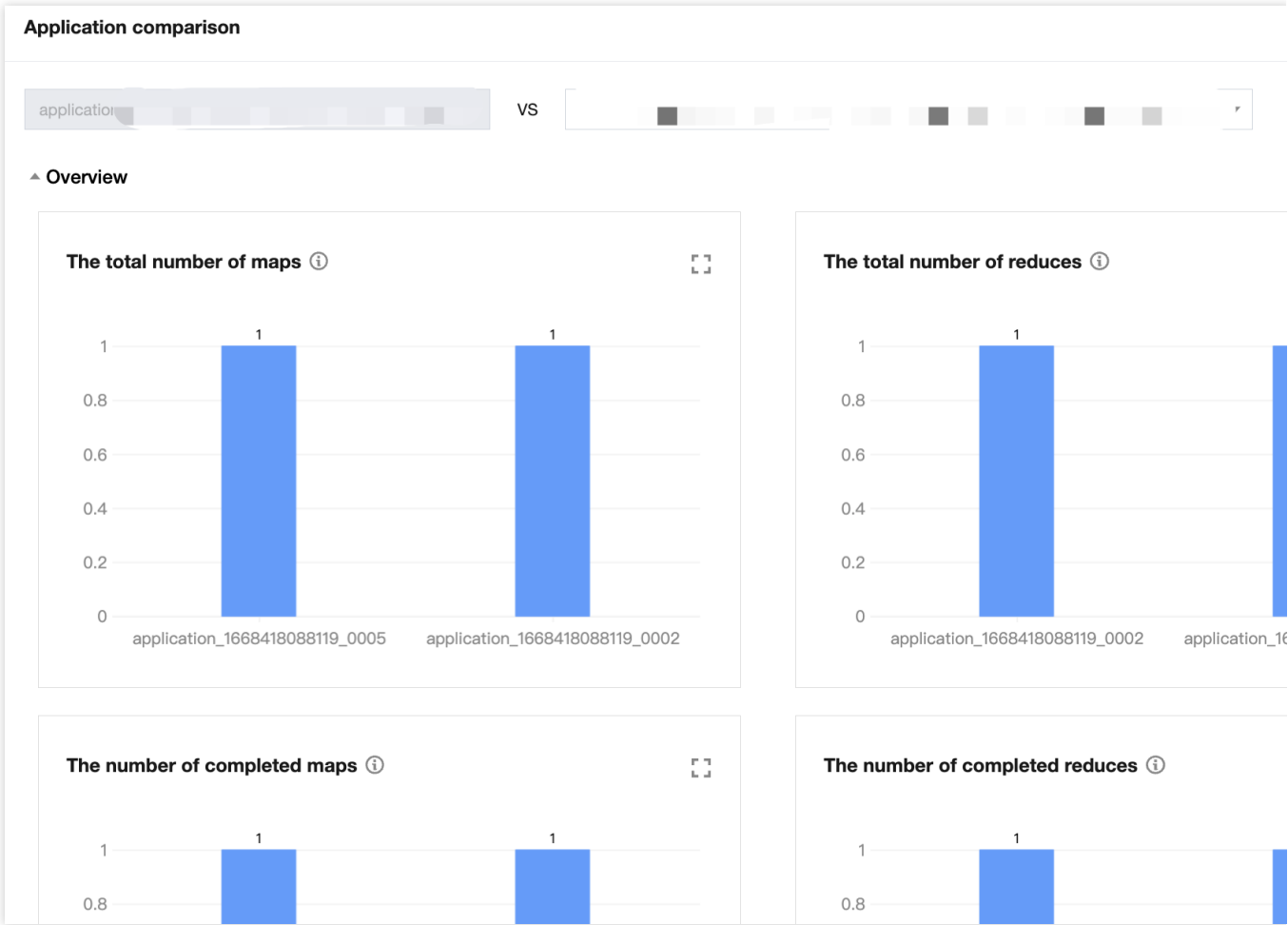
**Caution**

Only applications of MAPREDUCE, Spark, or Tez type in the final status of SUCCEEDED can be compared.

Applications of the same type with the same name are filtered on the page by default. Only applications of the same type can be selected and filtered for comparison, and real-time data can be filtered and queried on the backend.

5. In the job list, click **More** > **Task information** to view the job task list, hosts comparison, and task execution logs.

The feature support is as follows:

| Job Type | Task Information | Hosts Comparison | Task Log |
|----------|-----------------|------------------|----------|
| MAPREDUCE | Supported | Supported | Supported |
| Spark | Supported | Unsupported | Supported |
| Tez | Supported | Supported | Unsupported |
| Others | Unsupported | Unsupported | Unsupported |

# HDFS File Storage Analysis

Last updated：2023-12-27 14:44:48

## Overview

This feature allows you to perform an overall analysis of HDFS storage, including the total number of files, total storage usage, file distribution, and recent trends in file storage as of the day before the current day (T-1). It also provides top directory lists for large and small files.

You can view the daily changes and recent trends in the total number of files and total storage usage based on HDFS storage within a cluster.

The file count and storage usage pie charts can help you understand the proportion of small files and the storage used by them.

The feature also allows you to query and download directory information for the top 1,000 large/small files as of the last collection time.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the corresponding cluster in the cluster list.
2. On the cluster details page, click **Cluster Service** and choose **Operation** > **File Storage Analysis** in the top-right corner of the HDFS component block to view information about relevant files stored on **HDFS** and their directories as of the last collection time.
3. Obtain statistical views.
3.1 You can view the daily increase and day-over-day change in the total number of files stored on HDFS and total storage usage.
3.2 Pie charts are provided to show you the numbers of empty files (size = 0 MB), small files (size ≤ 2 MB), other files (2 MB < size < 128 MB), and large files (size ≥ 128 MB), and the storage used by each of the four categories of files.
4. You can view recent trends in the number and storage usage of different categories of files.
5. You can query and download information about the top 1,000 small/large files as of the day before the current day (T-1), including the file name, file path, file size, user group, owner, and last access time.

**Risk description**

The data required for file storage analysis is collected at 14:00 (Beijing time) every day.

1. The file storage analysis involves collecting and analyzing backup FsImage files, which can increase the memory usage of the local machine (a maximum increase of 4 GB). If the proportion of machines in the cluster that are using memory consistently remains at a high level, you can submit a ticket to disable the feature.

2.. In HA clusters, the analysis feature is executed on the Standby Master node, while in non-HA clusters, the feature is executed on the Master node.

# Impala Query Management

Last updated：2023-12-27 14:45:04

## Feature overview

You can view the percentile distribution of various metrics for Impala queries in multiple dimensions. The Impala list allows you to quickly view detailed metrics such as query statements, query status, users, databases, scanned rows, peak memory usage, total read/sent bytes, and HDFS scanned rows.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.
2. On the cluster details page, click **Cluster Service** and select **Operation** > **Query Management** in the top-right corner of the Impala component to query jobs.
Take the duration as an example. In the currently selected time range, if the TP90 percentile duration is 6.86k (ms), 90% of queries last for within 6.86s.

3. Impala query list information is provided. Certain column header fields can be filtered and sorted, and compound filtering by multiple dimensions is supported.

**Query list**

| | Today | Yesterday | Last 7 days | Last 30 days | | 2022-11-09 00:00:00 ~ 2022-11-09 20:29:24 | | Execution statement or query ID |

| statement | Coordinator ▼ | MaxNodePeakMem... ⇳ ▼ | queryType | PerNodePea... ⇳ ▼ | ScanH |
|---|---|---|---|---|---|
| select * from hivetest.local_Native_TEXT... | ■ ■ ■ | -- | QUERY | -- | -- |
| select count(*),name from hivetest.local_... | | -- | QUERY | -- | -- |
| select count(*),name from hivetest.local_... | | -- | QUERY | -- | -- |
| select a.name,b.age from hivetest.local_... | ■ ■ ! | -- | QUERY | -- | -- |
| select * from hivetest.hbase_test | | -- | QUERY | -- | -- |
| SELECT get_json_object(impala_hdfs_js... | ■ ■ | 199.84 KB | QUERY | 352.17 | 2 |
| select * from hivetest.cos_default_defaul... | ■ ■ | -- | QUERY | -- | -- |
| insert into hivetest.cos_default_default_... | | -- | QUERY | -- | -- |
| SELECT * from impalatest.impala_hdfs_j... | | 199.84 KB | QUERY | 335.88 | 2 |
| CREATE external table if not exists impa... | 2 | -- | DDL | -- | -- |

Total items: 41

10 ▼ / page   |◀ ◀

4. Click **Overview** in the **Operation** column to view the time distribution information throughout the entire lifecycle of an Impala query, key metric information, and certain node information of the runtime.

总览

### Full Query Cycle

**Query Compilation(ms)**



**QueryTimeline(ms)**



### Important Metrics

| Key | Value | Key | Value |
|---|---|---|---|
| Estimated Per-Host Mem | 41113800 | Tables Missing Stats | default.impal |
| Request Pool | default-pool | Admission result | Admitted imr |
| Cluster Memory Admitted | 78.42 MB | Executor Group | default |
| ClientFetchWaitTimer | 4s424ms | NumRowsFetched | 94.21K (942( |
| NumRowsFetchedFromCache | 0 (0) | RowMaterializationRate | 3.93 M/sec |

Click **Details** in the **Operation** column to view query statements, query plans, execution overview, profiles, and
memory information.

**Note:**

For Impala queries that take more than 3s to execute, you can view profiles in **Overview** and **Details**.

# Hive Table Analysis

Last updated：2023-12-27 14:45:23

## Overview

This feature allows you to quickly view multiple detailed metrics of YARN jobs such as the commit queue, status, and duration. Statistics views are also provided for viewing metric statistics in the three dimensions of queue, user, and job type.

Provides data distribution and trend information about the storage of databases and tables.

Provides the distribution of tables based on the last access time as a reference for the distribution of hot and cold data.

Helps you identify data skew in small files and partitions in a table based on the proportion of small files in the table and the amount of data in each partition.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the corresponding cluster in the cluster list.
2. On the cluster details page, click **Cluster Service** and choose **Operation** > **Table Analysis** in the top-right corner of the Hive component block to view table and data information based on Hive MetaStore as of the last collection time.
3. Obtain statistical views.
3.1 You can view the daily increase and day-over-day change in metrics such as the number of Hive databases, total number of Hive tables, and total Hive data storage size.
3.2 You can view the distribution of tables based on the last access time as a reference for the distribution of hot and cold data.
**Note:**

| Time Range | Description |
|---|---|
| Within 3 months | The last access time is within the last 3 months from the current date. |
| 3 months to 1 year | The last access time is more than 3 months ago but within the last year from the current date. |
| 1 to 5 year | The last accesss time is more than 1 year ago but within the last 5 years from the current date. |
| More than 5 years | The last access time is more than 5 years ago from the current date. |
| | |

| Other | Data collection is not enabled on Cos/CHDFS. The system failed to collect the last access time of data. |
|-------|--------------------------------------------------------------------------------------------------------|
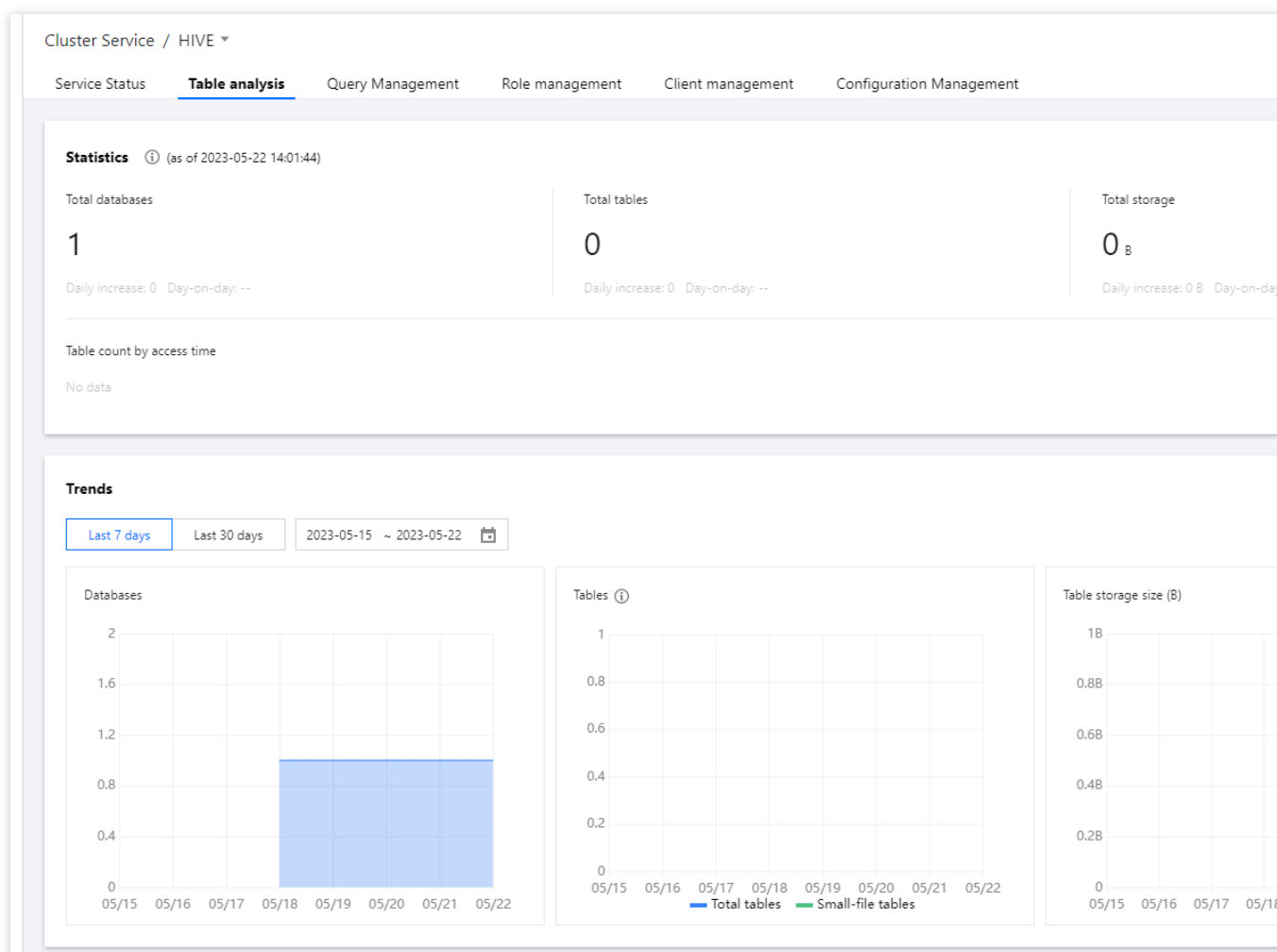


4. The trend view displays the historical trends in the number of databases, number of tables, and storage size of tables. In the table count trend chart, the trend in the number of small files is also shown, providing insights into the distribution and growth trends of small files.
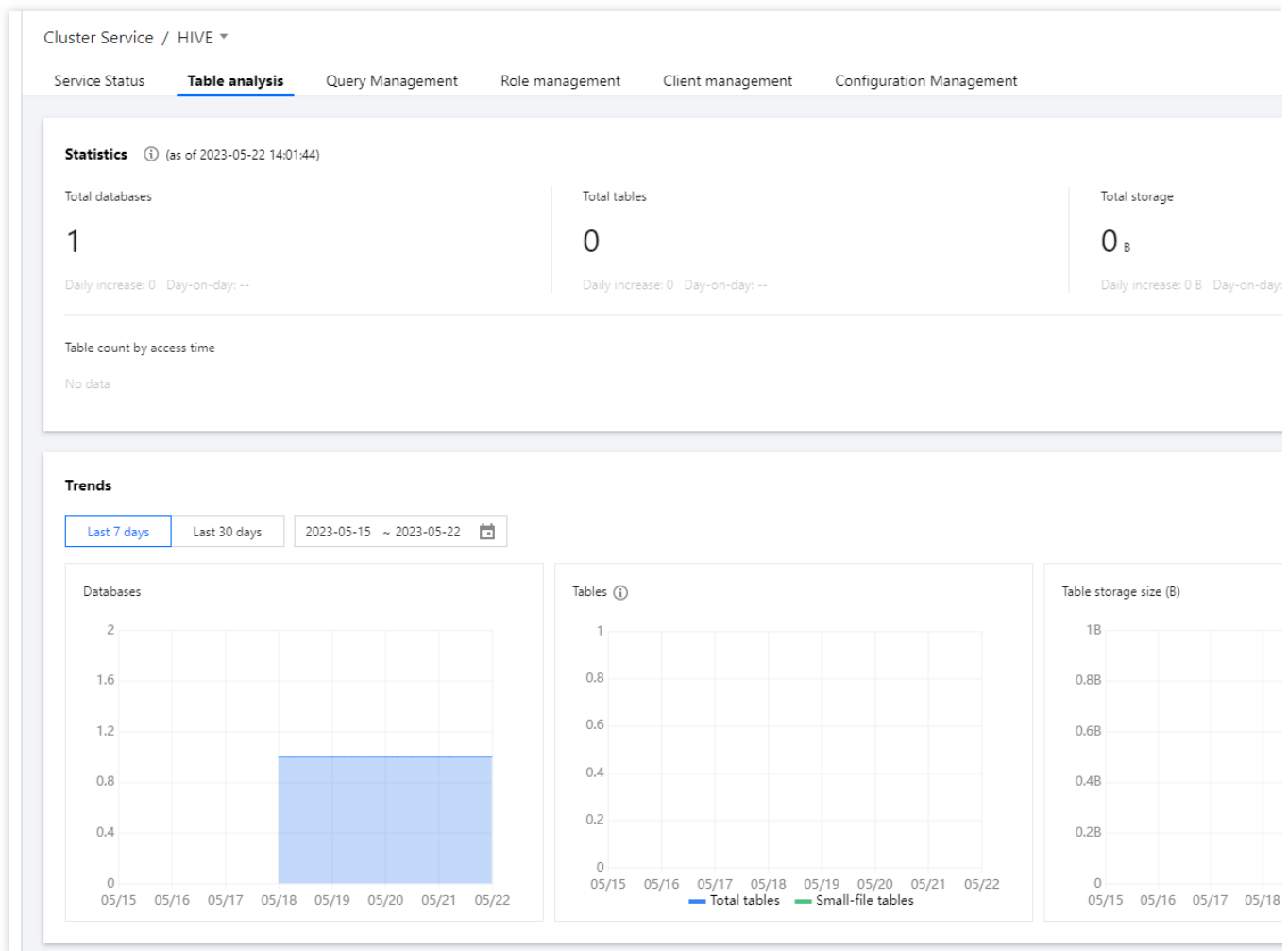
5. You can view the following information about a table: database, storage size, number of files, proportion of small files, and partitioning status. The number of files and proportion of small files provide insights into any issues with small files in Hive.

6. By clicking "View Partition Details", you can view some information about a partition, such as the name, size, and total number of files. This offers insights into the distribution of data across partitions and the number of files in each partition.

**Risk description**

The data required for table analysis is collected at 14:00 every day.

1. Data collected mainly includes metadata on databases, tables, and partitions stored in HMS, as well as directory information stored in NameNode. This may lead to a slight increase in the number of requests made to HMS and NameNode. If the number of requests continues to grow and exceeds the capacity limit, submit a ticket to disable this feature.

2. The collection of data from HMS and NameNode involves only metadata and does not involve specific business data.

# HBase Table Analysis

Last updated：2023-12-27 14:45:41

## Overview

Table analysis provides information such as the numbers of read/write requests and storage information for an entire HBase table, each individual region within the table, and each region server. The feature also supports analyzing the read/write queries per second (QPS) and their trends for the corresponding table or region server in practical scenarios.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the corresponding cluster in the cluster list.
2. On the cluster details page, click **Cluster Service** and choose **Operation** > **Table analysis** in the top-right corner of the HBase component block to query HBase table loads.

## Supported Operations in the Table List

In the HBase table list, you can view information about the read QPS, write QPS, MetaStore storage, and StoreFile size for each table. You can also identify the top tables in the cluster by using the sorting button in each column.

## Viewing table details

Click a table name to obtain a detailed view of the table. The details page displays the numbers of read/write requests, MemStore storage, and storeFile size for the entire table or each individual node. You can switch between nodes by using the node filter in the top-right corner.
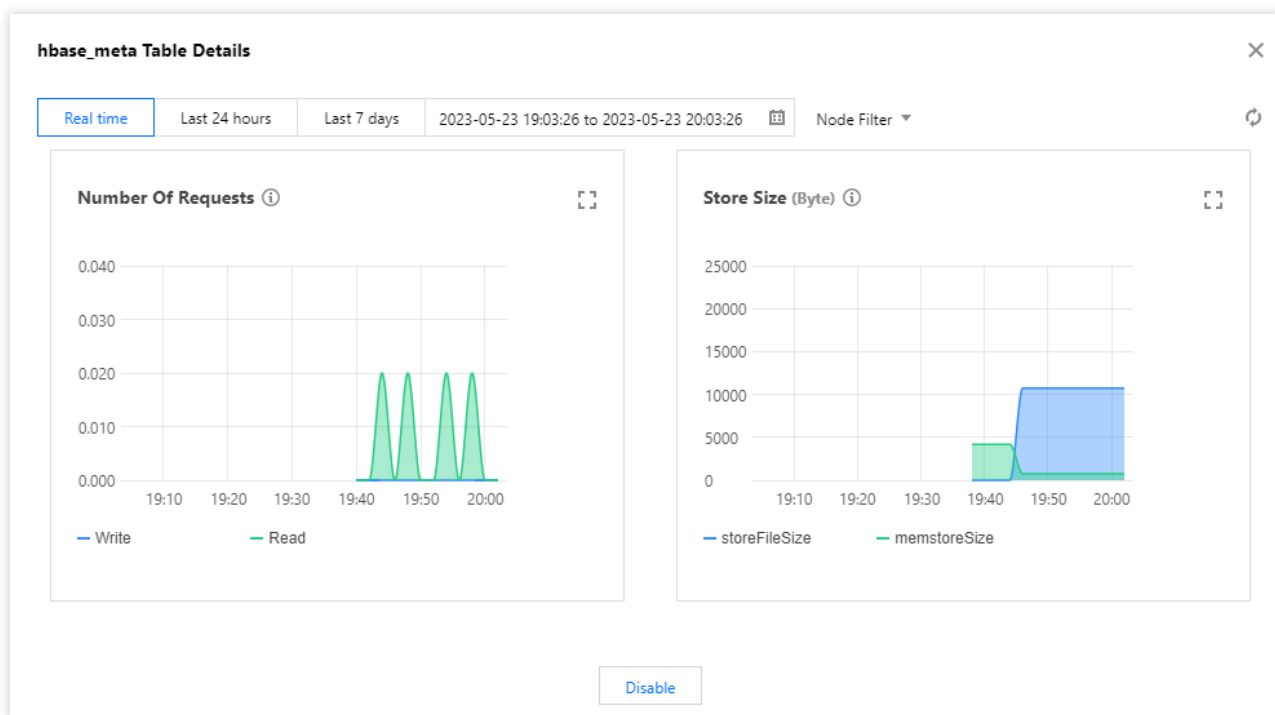


## Viewing the overall region information

Click **Regions** to view the overall region information, including the number of read/write requests for each region in the table, which helps you identify hot spot regions.



## Viewing region details

Click a region name to obtain the detailed view of the region. The details page displays the numbers of read/write requests for the table at different time granularities. Select an option from **Time granularity** in the top-right corner to switch between different time granularities.
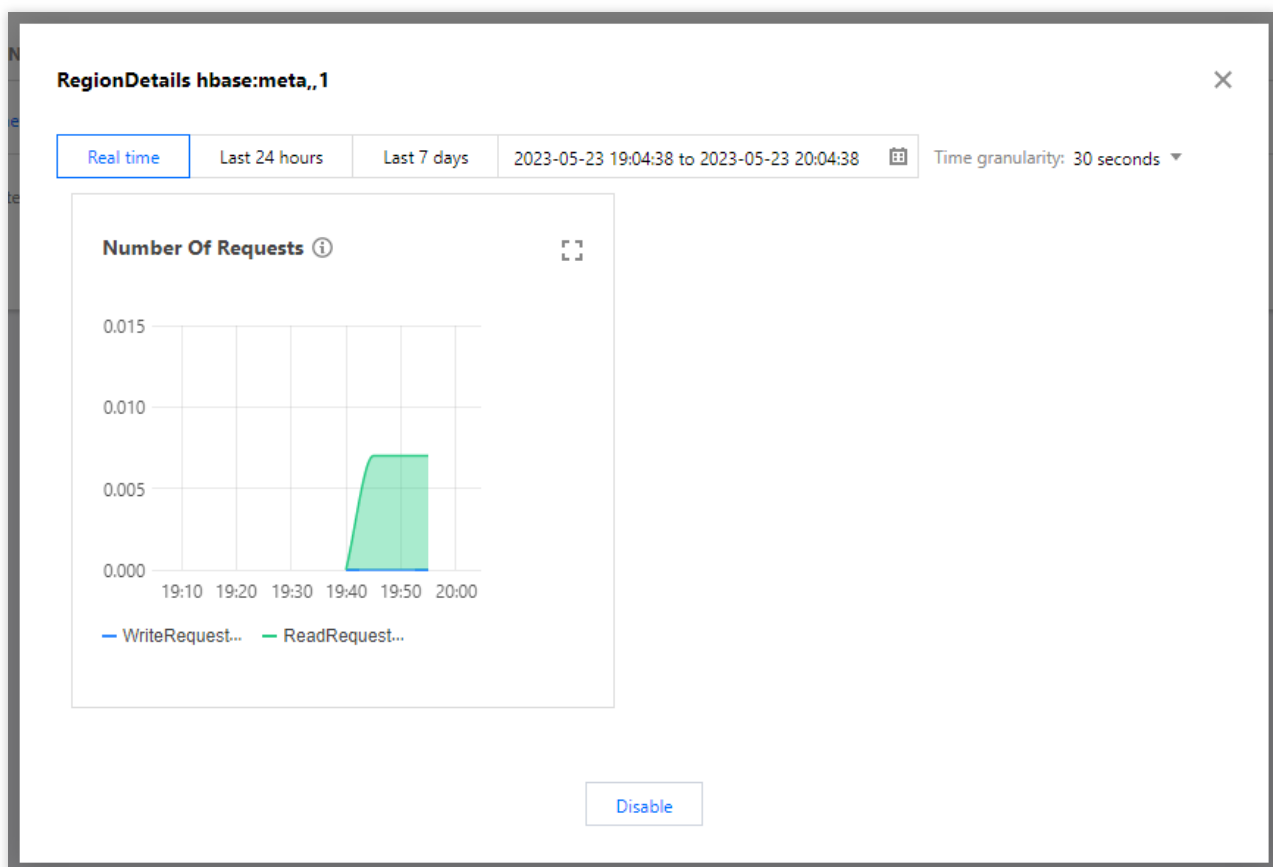


## Viewing the overall region server information

Click **RegionServers** to view the request latencies of each region server that hosts one or more regions in the table.



# Region Analysis

With region analysis, you can search for the table that a specific region belongs to or filter results by the region server that hosts the region. By examining the average request QPS and average read/write QPS, you can identify hot spots in the cluster where a large number of requests are being processed.

By clicking the view button in the "Average read QPS" or "Average write QPS" column header, you can view the trend of read/write QPS for the current region and observe sudden changes in request traffic. You can specify the time range for the information displayed.

# Kudu Table Analysis

Last updated：2023-12-27 14:45:59

## Overview

Kudu table monitoring and Tablet analysis can help you identify data hot spots and skew in tables and Tablet deployment.

1. Kudu table analysis provides load information, including table-level, Tablet, and TabletServer read/write QPS and storage.

2. Through Tablet analysis, you can analyze the historical trends of read/write QPS information for tables or TabletServers, based on real-world scenarios.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the corresponding cluster in the cluster list.

2. On the cluster details page, click **Cluster Service** and choose **Operation** > **Table Analysis** in the top-right corner of the Kudu component block to view the analysis of Kudu table loads.

## Supported Operations in the Table List

In the Kudu table list, you can view information about table-level request QPS, write QPS, and OnDiskDataSize storage. You can also identify the top tables in the cluster by using the sorting button in each column header.
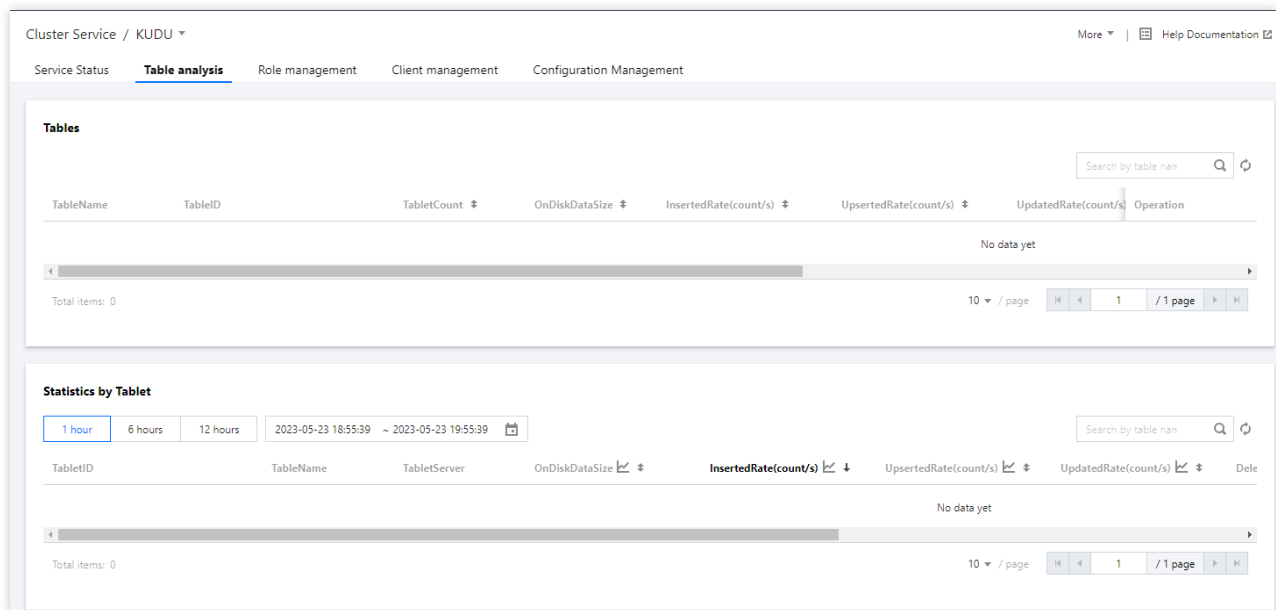
## Viewing table details

Click a table name to obtain a detailed view of the table. The details page displays the numbers of read/write requests and storage size (including OnDiskDataSize) for the entire table or each individual node. You can switch between nodes by using the node filter in the top-right corner.

## Viewing the overall Tablet information

Click **Tablets Operation** to view the overall information about the read/write request volumes of each Tablet in the table, and identify hot spots within the Tablets.

## Viewing Tablet details

Click the corresponding Tablet name to view the Tablet details and metric trends. The details page provides data on request and scan metrics for the selected Tablet at various time granularities. You can change the time granularity in the top-right corner.

## Viewing the overall TabletServer information

Click **TabletServers Operation** to view the overall information about each TabletServer, such as the request latency and data storage.

# Tablet Analysis

With Tablet analysis, you can search for the table that a specific region belongs to or filter results by the TabletServer that hosts the region. By examining the average request QPS and average read/write QPS, you can identify hot spots

in the cluster where a large number of requests are being processed.

By clicking the view button in the "Average read QPS" or "Average write QPS" column header, you can view the trend of read/write QPS for the current Tablet and observe sudden changes in request traffic. You can specify the time range for the information displayed.

# Cluster Inspection

Last updated：2023-12-27 14:46:15

## Overview

You can perform health check on the nodes and services in each cluster instantly or periodically (daily or weekly) with the selected inspection items. Only one periodic inspection task can be configured for each cluster to help you know the cluster health conditions periodically and fix the exceptions and risks in time.

When you set a one-time or periodic inspection task, general inspection items will be selected by default. If you want to check certain service features in special cases, you can select the corresponding inspection items as needed. However, service feature inspection will compromise cluster performance, which is thus not recommended during business peak hours.

After each inspection task is completed, a PDF report will be generated, which can be downloaded and deleted. Up to 50 inspection reports can be retained under each root account. If an excessive report is generated, the oldest one will be deleted.

**Note**

The inspection system supports service inspection items. Currently, only HDFS, YARN, HBase, Hive, and ZooKeeper are supported.

The configuration of a periodic inspection task cannot be changed when it is being executed.

## Directions

1. Log in to the EMR console and click the **ID**/**Name** of the target cluster in the cluster list to enter the cluster details page.

2. On the cluster details page, select **Cluster Monitoring** > **Cluster Inspection** and you can perform health check on the nodes and services in the current cluster. Only one periodic inspection task can be configured for each cluster. You can also click **One-Time Inspection** to perform an inspection. To configure a periodic inspection task, click **Periodic Inspection Settings**.
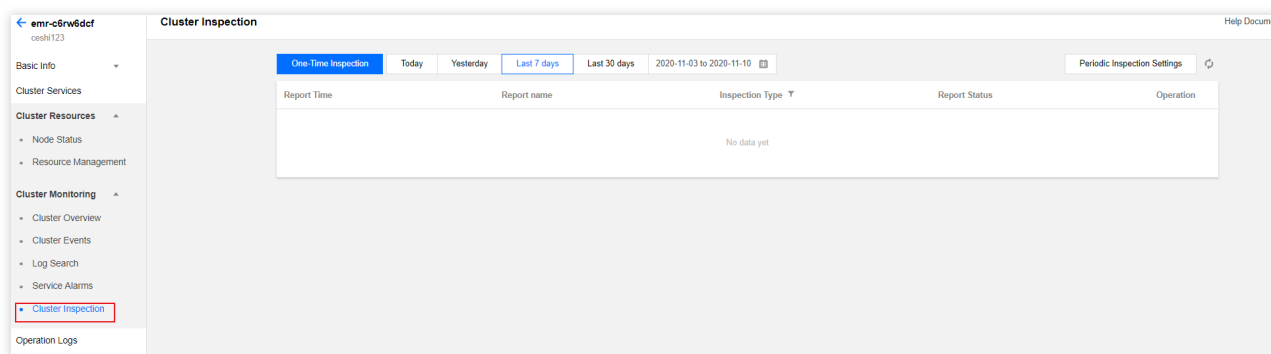
One-Time Inspection: The health status of nodes and services in the cluster from a specified time point to the current time point will be checked, and an inspection report will be generated.

**Start One-Time Inspection**                                                          ✕

ⓘ  1. One-time inspection: check the health status of a cluster's nodes and service from a specified moment to the current time
and generate an inspection report.
2. Periodic inspection: when periodic inspection is enabled, the system automatically checks the health status of nodes and
services in each preset inspection cycle and generate an inspection report. Only one periodic inspection can be configured
for each cluster.
3. Inspection items support all enabled event monitoring policies by default. You can adjust the inspection items in Event
Policy ⤴ .

Inspection Interval    [ 2020-11-10                   📅 ]   [ 00:00          🕐 ]   to date

Inspection Item    ┌─────────────────────────────────────────────────────────────┐
                   │  ☐ Select all                      ☐ Invert All              │
                   │                                                               │
                   │  ▾ ▬ NODE                                                     │
                   │        ☑ CpuUsageMoreThanThreshold                            │
                   │        ☑ CpuIOWaitAvgMoreThanThreshold                        │
                   │        ☑ MemoryFreeException                                  │
                   │        ☑ ProcessCntMoreThanThreshold                          │
                   │        ☑ ProcessOutOfMemory                                   │
                   │        ☑ DiskIOError                                          │
                   │        ☑ DiskSpaceMoreThanThreshold                          │
                   │        ☑ DiskUtilMoreThanThreshold                           │
                   │        ☐ ProcessLost                                          │
                   │        ☑ HeartBeatLost                                        │
                   │        ☑ HostnameChanged                                      │
                   │  ▸ ▬ HDFS                                                     │
                   │  ▸ ☑ YARN                                                     │
                   └─────────────────────────────────────────────────────────────┘

                        [ **Start Inspecting** ]   [ Cancel ]

Periodic Inspection: After a periodic inspection policy is enabled, the system will automatically check the health status
of the nodes and services in the cluster in every inspection cycle and generate an inspection report. One periodic
inspection policy can be configured for each cluster.
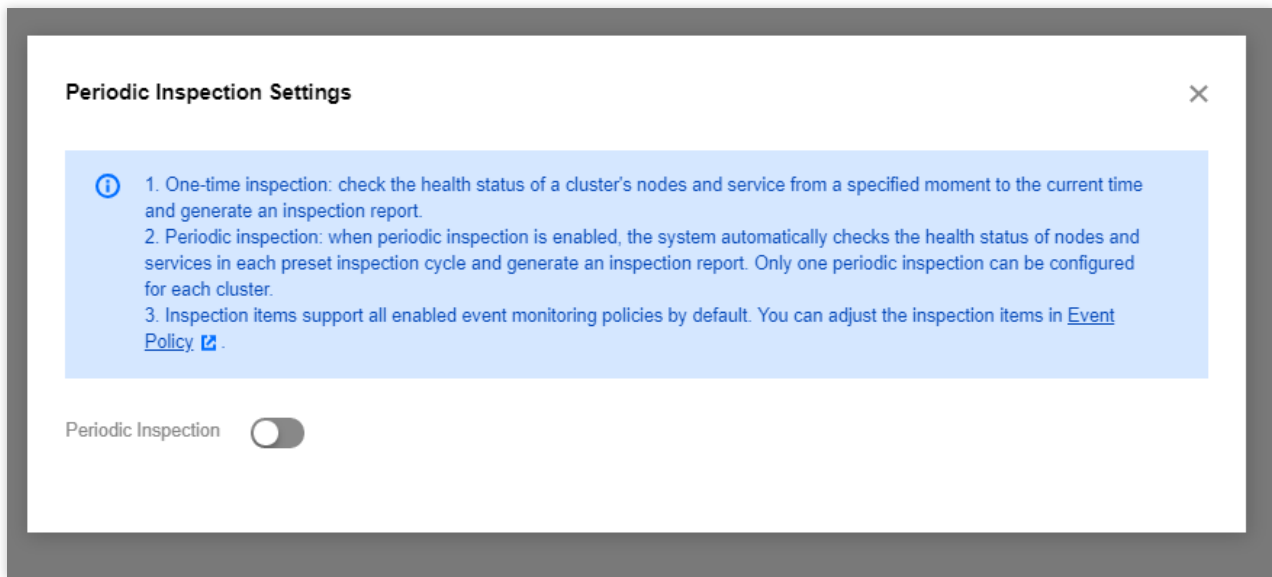
Inspection Item: All enabled event monitoring policies are supported by default. You can adjust the inspection items as instructed in Cluster Event. All events with monitoring enabled are selected by default as the initial inspection items. When you customize the inspection items, those selected last time will be selected by default.
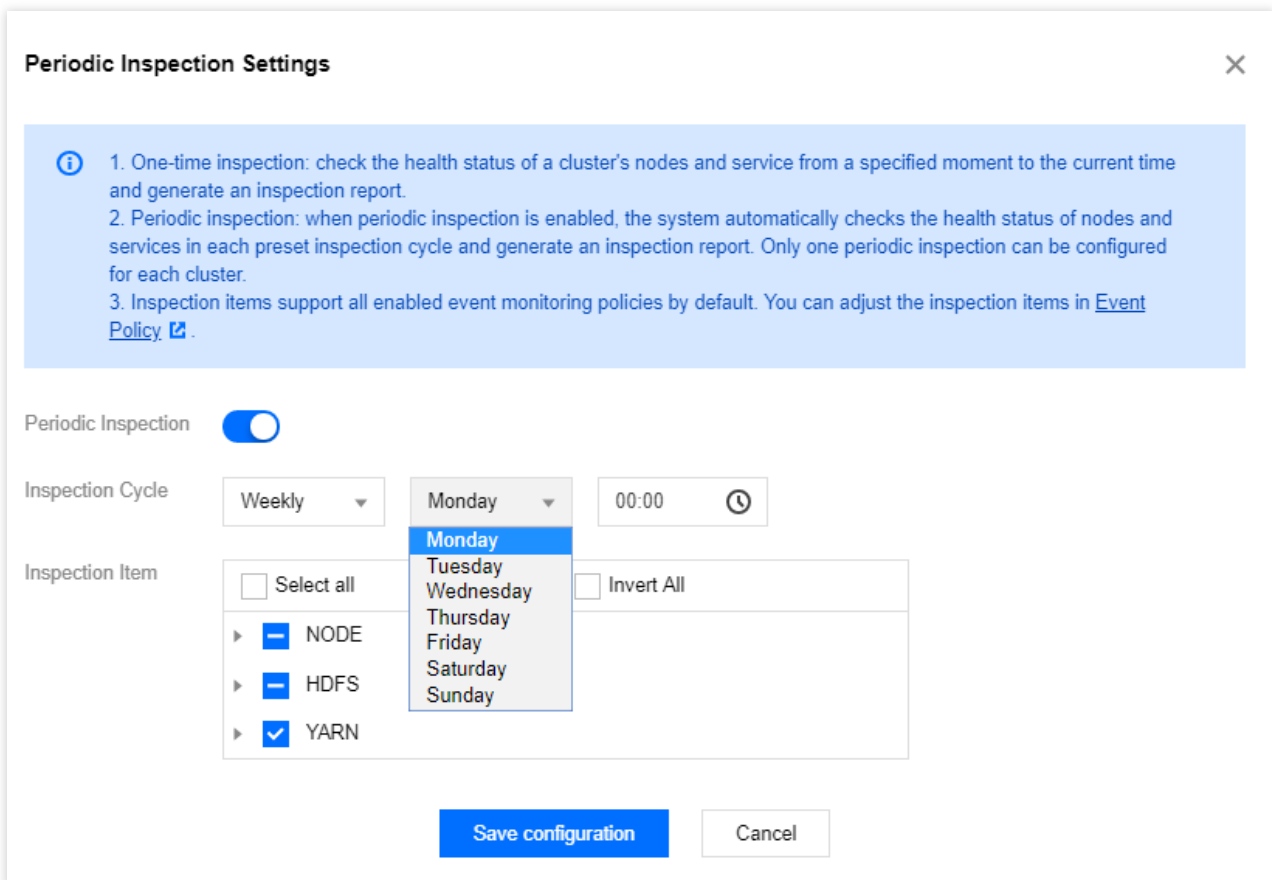
# Monitoring Metrics

# Node Monitoring Metrics

Last updated：2023-12-27 14:46:33

**Node - CPU**

| Title | Metric | Unit | Description |
|---|---|---|---|
| CPU utilization | idle | % | Percentage of CPU idle time |
| | irq | % | Percentage of interrupts |
| | nice | % | Percentage of CPU utilization under nice priority |
| | steal | % | Percentage of wait time by virtual CPUs for physical CPUs |
| | softirq | % | Percentage of CPU soft interrupts |
| | guest | % | Percentage of time spent running virtual processors |
| | system | % | CPU utilization in kernel mode |
| | user | % | CPU utilization in user mode |
| | iowait | % | Percentage of CPU idleness due to process I/O waits |
| Load | 1m | % | 1-minute load |
| | 5m | % | 5-minute load |
| | 15m | % | 15-minute load |
| Cores | cpu_count | - | Number of CPU cores |

**Node - memory**

| Title | Metric | Unit | Description |
|---|---|---|---|
| Memory utilization | MemTotal | GB | Total memory size |
| | MemFree | GB | Total free memory size |
| | MemAvailable | GB | Total available memory size |
| | Buffers | GB | Total memory size used by buffers |

| | Cached | GB | Total memory size used by file cache |
|---|---|---|---|
| | SwapCached | GB | Total swap memory size by anonymous page writes |
| | SwapFree | GB | Total available swap size |
| | AnonPages | GB | Total unmapped memory size |
| | SwapTotal | GB | Total swap size |
| | Dirty | GB | Total memory size to write to disk |
| | Writeback | GB | Total memory size being written back to disk |
| | HardwareCorrupted | GB | Total unavailable memory size due to memory hardware failure |
| | Shmem | GB | Total shared memory size |
| | MemUsed | GB | Total used memory size |
| Percentage of used memory | available_percent | % | Percentage of available memory size out of total memory |
| | used_percent | % | Percentage of used memory size out of total memory |

## Node - disk

| Title | Metric | Unit | Description |
|---|---|---|---|
| Device read/write rate | Read | MB/s | Data read per second |
| | Write | MB/s | Data written per second |
| Device IOPS | all | count/s | Number of I/O operations in progress on current device |
| I/O operation time | Read | ms | Average wait time per device I/O read operation |
| | Write | ms | Average wait time per device I/O write operation |
| | IO | ms | Average processing time per I/O request |
| Device read/write QPS | Read | count/s | Read QPS |
| | Write | count/s | Write QPS |
| | Merge-Read | count/s | Merged read QPS |

| | Merge-Write | count/s | Merged write QPS |
|---|---|---|---|
| I/O device utilization | all | % | Disk busyness |
| Disk space | Free | GB | Free disk storage space |
| | Available | GB | Available disk storage space (for unprivileged users) |
| | Total | GB | Total disk storage space |
| Disk space utilization | Used | % | Disk space utilization |
| INODES | Free | - | Number of remaining disk inodes |
| | Total | - | Total number of disk inodes |
| Inode utilization | Used | % | Disk inode utilization |

## Node - file handle

| Title | Metric | Unit | Description |
|---|---|---|---|
| File handle | allocated | - | Number of allocated file handles |
| | maximum | - | Maximum number of file handles |
| System interrupt | intr_total | count/s | Number of system interrupts |
| System context switch | context_switches_total | count/s | Number of system context switches |
| System process | forks_total | - | Number of new system processes |
| | procs_running | - | Number of running system processes |
| | procs_blocked | - | Number of blocked system processes |
| | procs_total | - | Total number of system processes |
| | thrds_total | - | Total number of system threads |
| Agent version | AgentVersionl | version | Agent version |

## Node - network

| Title | Metric | Unit | Description |
|---|---|---|---|
| TCP LISTEN | ListenDrops | count/s | Number of incoming connections (SYN packets) |

| exception | | | dropped for any reason |
|---|---|---|---|
| | ListenOverflows | count/s | Number of occurrences where the upper limit of the Accept queue is exceeded after the last step of three-way handshake is completed |
| TCPSyncookies | SyncookiesFailed | count/s | Number of packets received with invalid SYN Cookie information |
| | SyncookiesRecv | count/s | Number of packets received with valid SYN Cookie information |
| | SyncookiesSent | count/s | Number of SYN/ACK packets sent through SYN Cookie |
| TCP connection abort exception | TCPAbortOnTimeout | count/s | Number of connections closed because the attempts of retransmissions of various timers (RTO/PTO/keepalive) exceed the upper limit |
| | TCPAbortOnData | count/s | Number of sockets closed due to unknown data received |
| | TCPAbortOnClose | count/s | Number of sockets closed when the user-mode program has data in the buffer |
| | TCPAbortOnMemory | count/s | Number of connections closed due to memory issues |
| | TCPAbortOnLinger | count/s | Number of connections suspended in lingering status after being closed |
| | TCPAbortFailed | count/s | Number of failed attempts to close connection |
| TCP connection establishment | ActiveOpens | count/s | Number of actively established TCP connections |
| | CurrEstab | count/s | Number of TCP connections currently established |
| | PassiveOpens | count/s | Number of passively established TCP connections |
| | AttemptFails | count/s | Number of connection establishment failures |
| | EstabResets | count/s | Number of reset connections |
| TCP packet | InSegs | count/s | Number of received packets, including erroneous |

| | | | |
|---|---|---|---|
| | | | ones |
| | OutSegs | count/s | Number of sent packets |
| | RetransSegs | count/s | Number of received TCP packets |
| | InErrs | count/s | Number of retransmitted packets |
| | OutRsts | count/s | Number of sent RST packets |
| TCP retransmission rate | RetransSegsRate | % | Retransmission rate at TCP layer |
| | ResetRate | % | RESET sending frequency |
| | InErrRate | % | Percentage of erroneous packets |
| TCP TIME-WAIT | TW | count/s | Number of sockets ending TIME_WAIT status after normal timeout |
| | TWKilled | count/s | Number of sockets ending TIME_WAIT status through tcp_tw_recycle mechanism |
| | TCPTimeWaitOverflow | count/s | Number of TIME_WAIT sockets unable to be allocated due to limit exceeding |
| | TWRecycled | count/s | Number of sockets ending TIME_WAIT status through tcp_tw_reuse mechanism |
| TCP RTO | TCPTimeouts | count/s | Number of first RTO timer timeouts |
| | TCPSpuriousRTOs | count/s | Number of spurious timeouts detected through F-RTO mechanism |
| | TCPLossProbes | count/s | Number of Tail Loss Probe (TLP) packets sent due to Probe Timeout (PTO) |
| | TCPLossProbeRecovery | count/s | Number of lost packets just repaired by TLP probes |
| | TCPRenoRecoveryFail | count/s | Number of connections that enter the Recovery phase and then undergo RTO (SACK option not supported by the opposite) |
| | TCPSackRecoveryFail | count/s | Number of connections that enter the Recovery phase and then undergo RTO (SACK option supported by the opposite) |

| | TCPRenoFailures | count/s | Number of failures that enter the TCP_CA_Disorder phase and then undergo RTO (SACK option not supported by the opposite) |
|---|---|---|---|
| | TCPSackFailures | count/s | Number of failures that enter the TCP_CA_Disorder phase and then undergo RTO (SACK option supported by the opposite) |
| | TCPLossFailures | count/s | Number of connections that enter the TCP_CA_Loss phase and then undergo RTO timeout |
| TCP RTO constant | RtoAlgorithm | 1/s | Number of delayed algorithms for forwarding unanswered objects |
| | RtoMax | 1 | Maximum number of retransmissions due to TCP latency |
| | RtoMin | 1 | Minimum number of retransmissions due to TCP latency |
| TCP retransmission | TCPLostRetransmit | count/s | Number of SKB retransmissions due to loss |
| | TCPFastRetrans | count/s | Number of fast SKB retransmissions |
| | TCPForwardRetrans | count/s | Number of regular SKB retransmissions |
| | TCPSlowStartRetrans | count/s | Number of SKB retransmissions with successful slow start |
| | TCPRetransFail | count/s | Number of failed retransmission attempts |
| UDP datagram | OutDatagrams | count/s | Number of sent UDP datagrams |
| | InDatagrams | count/s | Number of received UDP datagrams |
| ENI data receiving/sending rate | eth0-receive_bytes | MB/s | Volume of data received by ENI |
| | eth0-transmit_bytes | MB/s | Volume of data sent by ENI |
| ENI packet receiving/sending rate | eth0-receive_drop | count/s | Volume of data received and then dropped by ENI |
| | eth0-receive_errs | count/s | Volume of data failed to be received by ENI |
| | eth0-transmit_drop | count/s | Volume of data sent and then dropped by ENI |
| | eth0-transmit_errs | count/s | Volume of data failed to be sent by ENI |

| | eth0-transmit_packetsl | count/s | Number of packets sent by ENI |
|---|---|---|---|
| TCP socket | TCP_inuse | - | Number of TCP sockets in use (listening) |
| | TCP_orphan | - | Number of TCP connections waiting to be closed |
| | TCP_tw | - | Number of TCP sockets to be terminated |
| | TCP_alloc | - | Number of TCP sockets allocated (established, sk_buff obtained) |
| | sockets_used | - | Total number of used sockets |
| TCP connection status | ESTABLISHED | - | Number of TCP connections in Established status |
| | SYN-SENT | - | Number of TCP connections in SYN-SENT status |
| | SYN-RECV | - | Number of TCP connections in SYN-RECV status |
| | FIN-WAIT1 | - | Number of TCP connections in FIN-WAIT1 status |
| | FIN-WAIT2 | - | Number of TCP connections in FIN-WAIT2 status |
| | TIME-WAIT | - | Number of TCP connections in TIME-WAIT status |
| | CLOSE | - | Number of TCP connections in CLOSE status |
| | CLOSE-WAIT | - | Number of TCP connections in CLOSE-WAIT status |
| | LAST-ACK | - | Number of TCP connections in LAST-ACK status |
| | LISTEN | - | Number of TCP connections in LISTEN status |
| | CLOSEING | - | Number of TCP connections in CLOSEING status |

## Node - event

| Title | Metric | Unit | Description |
|---|---|---|---|
| CPU utilization | used | % | 1 - (percentage of CPU idle time) |

| 15-minute CPU load | 15m | - | 15-minute load |
|---|---|---|---|
| 1-minute CPU load | 1m | - | 1-minute load |
| 5-minute CPU load | 5m | - | 5-minute load |
| Disk IOPS | all | - | Number of I/O operations in progress on current device |
| Disk I/O operation time | IO | - | Average processing time per I/O request |
| Disk space utilization | Used | - | Disk space utilization |
| Disk I/O device utilization | all | - | Disk busyness |
| Memory utilization | used_percent | - | Percentage of used memory size out of total memory |
| Outbound network traffic rate | *-transmit_bytes | - | Volume of data sent by ENI |
| Inbound network traffic rate | *-receive_bytes | - | Volume of data received by ENI |
| TCP connections | CurrEstab | - | Number of TCP connections currently established |

# HDFS Monitoring Metrics

Last updated：2023-12-27 14:48:18

## HDFS - Overview

| Title | Metric | Unit | Description |
|-------|--------|------|-------------|
| Cluster storage capacity | CapacityTotal | GB | Total cluster storage capacity |
| | CapacityUsed | GB | Used cluster storage capacity |
| | CapacityRemaining | GB | Remaining cluster storage capacity |
| | CapacityUsedNonDFS | GB | Non-HDFS used cluster capacity |
| Cluster load | TotalLoad | 1 | Current connections |
| Total files in cluster | FilesTotal | - | Total number of files |
| Blocks | BlocksTotal | - | Total number of blocks |
| | PendingReplicationBlocks | - | Number of blocks waiting to be backed up |
| | UnderReplicatedBlocks | - | Number of blocks with insufficient replicas |
| | CorruptBlocks | - | Number of corrupted blocks |
| | ScheduledReplicationBlocks | - | Number of blocks arranged for backup |
| | PendingDeletionBlocks | - | Number of blocks waiting to be deleted |
| | ExcessBlocks | - | Number of excess blocks |
| | PostponedMisreplicatedBlocks | - | Number of abnormal blocks postponed to be processed |
| Block capacity | BlockCapacity | - | Block capacity |
| Cluster data node | NumLiveDataNodes | - | Number of live data nodes |
| | NumDeadDataNodes | - | Number of data nodes marked as dead |

| | NumDecomLiveDataNodes | - | Number of decommissioned live nodes |
|---|---|---|---|
| | NumDecomDeadDataNodes | - | Number of decommissioned dead nodes |
| | NumDecommissioningDataNodes | - | Number of decommissioning nodes |
| | NumStaleDataNodes | - | Number of DataNodes marked as stale |
| HDFS storage space utilization | CapacityUsedRate | - | HDFS cluster storage space utilization |
| Snapshots | Snapshots | - | Number of snapshots |
| Disk failure | VolumeFailuresTotal | - | Total number of volume failures across all DataNodes |

## HDFS - NameNode

| Title | Metric | Unit | Description |
|---|---|---|---|
| Data traffic | ReceivedBytes | Bytes/s | Data receiving rate |
| | SentBytes | Bytes/s | Data sending rate |
| QPS | RpcQueueTimeNumOps | 1/s | RPC call rate |
| Request processing latency | RpcQueueTimeAvgTime | ms | Average RPC latency |
| | RpcProcessingTimeAvgTime | ms | Average RPC request processing time |
| Authentication and authorization | RpcAuthenticationFailures | 1 per time | Number of RPC authentica failures |
| | RpcAuthenticationSuccesses | 2 per time | Number of RPC authentica successes |
| | RpcAuthorizationFailures | 3 per time | Number of RPC authorizati failures |
| | RpcAuthorizationSuccesses | 4 per time | Number of RPC authorizati successes |
| Current connections | NumOpenConnections | - | Number of current connect |

| Length of RPC processing queue | CallQueueLength | - | Length of current RPC processing queue |
|---|---|---|---|
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapCommitte configured by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory curre used by JVM |
| | MemHeapCommittedM | MB | Committed size of JVM HeapMemory |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemMaxM | MB | Maximum size of memory available to JVM runtime |
| Block reporting latency | BlockReportAvgTime | count/s | Average latency of process DataNode blocks per seco |
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | ThreadsWaiting | - | Number of threads in WAITING status |
| | ThreadsTimedWaiting | - | Number of threads in TIME WAITING status |
| | ThreadsTerminated | - | Number of threads in Terminated status |
| JVM logs | LogFatal | - | Number of FATAL-level log |
| | LogError | - | Number of ERROR-level lo |
| | LogWarn | - | Number of WARN-level log |

| | LogInfo | - | Number of INFO-level logs |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Surviv memory |
| | S1 | % | Percentage of used Surviv memory |
| | E | % | Percentage of used Eden memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| | CCS | % | Percentage of used compressed class space memory |
| Storages marked as content stale | NumStaleStorages | - | Number of DataNode stora marked as content stale |
| Pending block-related messages for later processing on the standby NameNode | PendingDataNodeMessageCount | count/s | Number of DataNode requ queued on the standby NameNode |
| Missing blocks | NumberOfMissingBlocks | - | Number of missing blocks |
| | NumberOfMissingBlocksWithReplicationFactorOne | - | Number of missing blocks 1) |

| | | | |
|---|---|---|---|
| Snapshot operation | AllowSnapshotOps | count/s | Number of AllowSnapshot operations executed per second |
| | DisallowSnapshotOps | count/s | Number of DisallowSnapsh operations executed per second |
| | CreateSnapshotOps | count/s | Number of CreateSnapsho operations executed per second |
| | DeleteSnapshotOps | count/s | Number of DeleteSnapsho operations executed per second |
| | ListSnapshottableDirOps | count/s | Number of ListSnapshottableDir operations executed per second |
| | SnapshotDiffReportOps | count/s | Number of SnapshotDiffReportOps operations executed per second |
| | RenameSnapshotOps | count/s | Number of RenameSnapshotOps operations executed per second |
| File operation | CreateFileOps | count/s | Number of CreateFile operations executed per second |
| | GetListingOps | count/s | Number of GetListing operations executed per second |
| | TotalFileOps | count/s | Number of TotalFileOps operations executed per second |
| | DeleteFileOps | count/s | Number of DeleteFile operations executed per second |

| | FileInfoOps | count/s | Number of FileInfo operatic executed per second |
|---|---|---|---|
| | GetAdditionalDatanodeOps | count/s | Number of GetAdditionalDatanode operations executed per second |
| | CreateSymlinkOps | count/s | Number of CreateSymlink operations executed per second |
| | GetLinkTargetOps | count/s | Number of GetLinkTarget operations executed per second |
| | FilesInGetListingOps | count/s | Number of FilesInGetListin operations executed per second |
| File statistics | FilesDeleted | count | Number of deleted or renar files and folders |
| | FilesCreated | count | Number of created files and folders |
| | FilesAppended | count | Number of appended 1 |
| Transaction operation | TransactionsNumOps | count/s | Number of journal transact operations processed per second |
| | TransactionsBatchedInSync | count/s | Number of journal transact operations batch processe per second |
| Image operation | GetEditNumOps | count/s | Number of GetEditNumOp operations executed per second |
| | GetImageNumOps | count/s | Number of GetImageNumC operations executed per second |
| | PutImageNumOps | count/s | Number of PutImageNumC operations executed per second |

| Sync operation | SyncsNumOps | count/s | Number of journal sync operations processed per second |
|---|---|---|---|
| Block operation | BlockReceivedAndDeletedOps | count/s | Number of BlockReceivedAndDeleted operations executed per second |
| | BlockOpsQueued | count/s | Number of processed DataNode block reporting operations |
| Cache reporting | CacheReportNumOps | count/s | Number of CacheReport operations processed per second |
| Block reporting | BlockReportNumQps | count/s | Number of DataNode block reporting operations proces per second |
| Sync operation latency | SyncsAvgTime | ms | Average latency of process journal sync operations |
| Cache reporting latency | CacheReportAvgTime | ms | Average latency of cache reporting |
| Image operation latency | GetEditAvgTime | ms | Average latency of reading Edit files |
| | GetImageAvgTime | ms | Average latency of reading image files |
| | PutImageAvgTime | ms | Average latency of writing image files |
| Transaction operation latency | TransactionsAvgTime | ms | Average latency of process journal transaction operatic |
| Start time | StartTime | ms | Process start time |
| Active/Standby status | State | 1 | NameNode HA status |
| Active/Standby status | State | 1: Active. | NameNode active/standby status |

| | | 0: Standby | |
|---|---|---|---|
| Threads | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Number of threads |
| | DaemonThreadCount | - | Number of backend thread |
| Transactions since the last checkpoint | SinceLastCheckpoint | count | Total number of transactior since the last checkpoint |
| Checkpoint time | LastCheckpoint | time | Time since the last checkp |
| Length of the queue waiting for file locks | LockQueueLength | count | LockQueueLength length of the queue waiting file locks |
| Average RPC time (1) | CompleteAvgTime | ms | Average latency of Complete requests |
| | CreateAvgTime | ms | Average latency of Cre requests |
| | RenameAvgTime | ms | Average latency of Rename requests |
| | AddBlockAvgTime | ms | Average latency of AddBlock requests |
| | GetListingAvgTime | ms | Average latency of GetListing requests |
| | GetFileInfoAvgTime | ms | Average latency of GetFileInfo requests |
| | SendHeartbeatAvgTime | ms | Average latency of SendHeartbeat requests |
| Average RPC time (2) | RegisterDatanodeAvgTime | ms | Average latency of RegisterDatanode request |
| | BlockReportAvgTime | ms | Average latency of BlockReport requests |
| | DeleteAvgTime | ms | Average latency of De |

| | RenewLeaseAvgTime | ms | Average latency of RenewLease requests |
| | BlockReceivedAndDeletedAvgTime | ms | Average latency of BlockReceivedAndDeleted requests |
| | FsyncAvgTime | ms | Average latency of fsy requests |
| | VersionRequestAvgTime | ms | Average latency of VersionRequest requests |
| Average RPC time (3) | ListEncryptionZonesAvgTime | ms | Average latency of ListEncryptionZones reque |
| | SetPermissionAvgTime | ms | Average latency of SetPermission requests |
| | SetTimesAvgTime | ms | Average latency of SetTimes requests |
| | SetSafeModeAvgTime | ms | Average latency of SetSafeMode requests |
| | MkdirsAvgTime | ms | Average latency of Mk requests |
| | GetServerDefaultsAvgTime | ms | Average latency of GetServerDefaults request |
| | GetBlockLocationsAvgTime | ms | Average latency of GetBlockLocations reques |
| RPC statistics (1) | CompleteNumOps | count/s | Number of Complete c per second |
| | CreateNumOps | count/s | Number of Create call per second |
| | RenameNumOps | count/s | Number of Rename ca per second |
| | AddBlockNumOps | count/s | Number of AddBlock c per second |

| | GetListingNumOps | count/s | Number of GetListing per second |
|---|---|---|---|
| | GetFileInfoNumOps | count/s | Number of GetFileInfo calls second |
| | SendHeartbeatNumOps | count/s | Number of SendHeart calls per second |
| RPC statistics (2) | RegisterDatanodeNumOps | count/s | Number of RegisterDatanode calls pe second |
| | BlockReportNumOps | count/s | Number of BlockRepo calls per second |
| | DeleteNumOps | count/s | Number of Delete calls second |
| | RenewLeaseNumOps | count/s | Number of RenewLea calls per second |
| | BlockReceivedAndDeletedNumOps | count/s | Number of BlockReceivedAndDeleted calls per second |
| | FsyncNumOps | count/s | Number of fsync calls second |
| | VersionRequestNumOps | count/s | Number of VersionRequest calls per second |
| RPC statistics (3) | ListEncryptionZonesNumOps | count/s | Number of ListEncryptionZones calls second |
| | SetPermissionNumOps | count/s | Number of SetPermiss calls per second |
| | SetTimesNumOps | count/s | Number of SetTimes per second |
| | SetSafeModeNumOps | count/s | Number of SetSafeMo calls per second |
| | MkdirsNumOps | count/s | Number of Mkdirs call per second |

| | GetServerDefaultsNumOps | count/s | Number of GetServerDefaults calls pe second |
|---|---|---|---|
| | GetBlockLocationsNumOps | count/s | Number of GetBlockLocati calls per second |

## HDFS - DataNode

| Title | Metric | Unit | Description |
|---|---|---|---|
| Xceivers | XceiverCount | - | Number of Xceivers |
| Data read/write rate | BytesWrittenMB | Bytes/s | DataNode byte write rate |
| | BytesReadMB | Bytes/s | DataNode byte read rate |
| | RemoteBytesReadMB | Bytes/s | Remote client byte read rate |
| | RemoteBytesWrittenMB | Bytes/s | Remote client byte write rate |
| Client connections | WritesFromRemoteClient | - | Remote client write QPS |
| | WritesFromLocalClient | - | Local client write QPS |
| | ReadsFromRemoteClient | - | Remote client read QPS |
| | ReadsFromLocalClient | - | Local client read QPS |
| Block verification failure | BlockVerificationFailures | count/s | Number of block verification failures |
| Disk failure | VolumeFailures | count/s | Number of disk failures |
| Network error | DatanodeNetworkErrors | count/s | Network error statistics |
| Heartbeat latency | HeartbeatsAvgTime | ms | Average heartbeat time |
| Heartbeat QPS | HeartbeatsNumOps | count/s | Heartbeat QPS |
| Packet transfer RT | SendDataPacketTransferNanosAvgTime | ms | Average time of sending packets |
| Block operation | ReadBlockOpNumOps | count/s | Block read OPS from DataNode |
| | WriteBlockOpNumOps | count/s | Block write OPS to |

| | | | DataNode |
|---|---|---|---|
| | BlockChecksumOpNumOps | count/s | Checksum OPS by DataNode |
| | CopyBlockOpNumOps | count/s | Block copying OPS |
| | ReplaceBlockOpNumOps | count/s | Block replacement OPS |
| | BlockReportsNumOps | count/s | Block reporting OPS |
| | IncrementalBlockReportsNumOps | count/s | Incremental block reporting OPS |
| | CacheReportsNumOps | count/s | Cache reporting OPS |
| | PacketAckRoundTripTimeNanosNumOps | count/s | Number of ACK round trips processed per second |
| Fsync operation | FsyncNanosNumOps | count/s | Number of fsync operations processed per second |
| Flush operation | FlushNanosNumOps | count/s | Number of flush operations processed per second |
| Block operation latency statistics | ReadBlockOpAvgTime | ms | Average block read time |
| | WriteBlockOpAvgTime | ms | Average block write time |
| | BlockChecksumOpAvgTime | ms | Average block check time |
| | CopyBlockOpAvgTime | ms | Average block copy time |
| | ReplaceBlockOpAvgTime | ms | Average block replacement time |
| | BlockReportsAvgTime | ms | Average block reporting time |
| | IncrementalBlockReportsAvgTime | ms | Average time of incremental block reporting |
| | CacheReportsAvgTime | ms | Average time of cache reporting |
| | PacketAckRoundTripTimeNanosAvgTime | ms | Average time of ACK round trip processing |
| Flush latency | FlushNanosAvgTime | ms | Average flush time |

| Fsync latency | FsyncNanosAvgTime | ms | Average fsync time |
|---|---|---|---|
| RamDisk Blocks | RamDiskBlocksWrite | blocks/s | Total number of blocks written to memory |
| | RamDiskBlocksWriteFallback | blocks/s | Total number of blocks failed to be written to memory (failover to disk) |
| | RamDiskBlocksDeletedBeforeLazyPersisted | blocks/s | Total number of blocks deleted before the application is saved to the disk |
| | RamDiskBlocksReadHits | blocks/s | Number of blocks read from memory |
| | RamDiskBlocksEvicted | blocks/s | Total number of blocks cleared in memory |
| | RamDiskBlocksEvictedWithoutRead | blocks/s | Total number of blocks retrieved from memory |
| | RamDiskBlocksLazyPersisted | blocks/s | Number of disk writes by lazy writer |
| | RamDiskBytesLazyPersisted | Bytes/s | Total number of bytes written to disk by lazy writer |
| RamDisk write speed | RamDiskBytesWrite | Bytes/s | Total number of bytes written to memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapCommittedM configured by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Committed size of JVM HeapMemory |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |

| | MemMaxM | MB | Maximum size of memory available to JVM runtime |
|---|---|---|---|
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | ThreadsWaiting | - | Number of threads in WAITING status |
| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
| | ThreadsTerminated | - | Number of threads in Terminated status |
| JVM logs | LogFatal | - | Number of Fatal logs |
| | LogError | - | Number of Error logs |
| | LogWarn | - | Number of Warn logs |
| | LogInfo | - | Number of Info logs |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | | | |

| | S1 | % | Percentage of used Survivor 1 memory |
|---|---|---|---|
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| Data traffic | ReceivedBytes | Bytes/s | Data receiving rate |
| | SentBytes | Bytes/s | Data sending rate |
| QPS | RpcQueueTimeNumOps | count/s | RPC call rate |
| Request processing latency | RpcQueueTimeAvgTime | ms | Average RPC latency |
| | RpcProcessingTimeAvgTime | count/s | Average RPC request processing time |
| Authentication and authorization | RpcAuthenticationFailures | count/s | Number of RPC authentication failures |
| | RpcAuthenticationSuccesses | count/s | Number of RPC authentication successes |
| | RpcAuthorizationFailures | count/s | Number of RPC authorization failures |
| | RpcAuthorizationSuccesses | count/s | Number of RPC authorization successes |
| Current connections | NumOpenConnections | - | Number of current connections |
| Length of RPC processing queue | CallQueueLength | 1 | Length of current RPC processing queue |
| CPU time | CurrentThreadSystemTime | ms | System time |
| | CurrentThreadUserTime | ms | User time |
| Start time | StartTime | s | Process start time |
| Threads | PeckThreadCount | - | Peak number of threads |
| | DaemonThreadCount | - | Number of backend threads |
| Read/Write | write | ms | Write time |

| latency | read | ms | Read time |
|---|---|---|---|
| Packet transfer QPS | DataPacketOps | count/s | Packet transfer QPS |
| Blocks | Related to disk information, such as `/data/qcloud/data/hdfs` | - | Blocks |
| Used disk capacity | Related to disk information, such as `/data/qcloud/data/hdfs` | GB | Used disk capacity |
| Free disk capacity | Related to disk information, such as `/data/qcloud/data/hdfs` | GB | Free disk capacity |
| Reserved disk capacity | Related to disk information, such as `/data/qcloud/data/hdfs` | GB | Reserved disk capacity |

## HDFS - JournalNode

| Title | Metric | Unit | Description |
|---|---|---|---|
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapCommittedM configured by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Committed size of JVM HeapMemory |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemMaxM | MB | Maximum size of memory available to JVM runtime |
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | | | |

| | ThreadsWaiting | - | Number of threads in WAITING status |
|---|---|---|---|
| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
| | ThreadsTerminated | - | Number of threads in Terminated status |
| JVM logs | LogFatal | - | Number of FATAL-level logs |
| | LogError | - | Number of ERROR-level logs |
| | LogWarn | - | Number of WARN-level logs |
| | LogInfo | - | Number of INFO-level logs |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| Data traffic | ReceivedBytes | Bytes/s | Data receiving rate |
| | SentBytes | Bytes/s | Data sending rate |
| Request processing latency | RpcQueueTimeAvgTime | ms | Average RPC latency |

| Authentication and authorization | RpcAuthenticationFailures | count/s | Number of RPC authentication failures |
|---|---|---|---|
| | RpcAuthenticationSuccesses | count/s | Number of RPC authentication successes |
| | RpcAuthorizationFailures | count/s | Number of RPC authorization failures |
| | RpcAuthorizationSuccesses | count/s | Number of RPC authorization successes |
| Current connections | NumOpenConnections | - | Number of current connections |
| Length of RPC processing queue | CallQueueLength | 1 | Length of current RPC processing queue |
| CPU time | CurrentThreadSystemTime | ms | System time |
| | CurrentThreadUserTime | ms | User time |
| Start time | StartTime | s | Process start time |
| Threads | PeckThreadCount | - | Peak number of threads |
| | DaemonThreadCount | - | Number of backend threads |

## HDFS - ZKFC

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |

| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |

## HDFS-Router

| Title | Metric | Unit | Description |
|-------|--------|------|-------------|
| ALTER TABLE request duration | HIVE.HMS.API_ALTER_TABLE | ms | Average duration of ALTER TABLE requests |
| ALTER TABLE WITH ENV CONTEXT request duration | HIVE.HMS.API_ALTER_TABLE_WITH_ENV_CONTEXT | ms | Average duration of ALTER TABLE WITH ENV CONTEXT requests |
| CREATE TABLE request duration | HIVE.HMS.API_CREATE_TABLE | ms | Average duration of CREATE TABLE requests |
| CREATE TABLE WITH ENV CONTEXT request duration | HIVE.HMS.API_CREATE_TABLE_WITH_ENV_CONTEXT | ms | Average duration of CREATE TABLE WITH ENV CONTEXT requests |
| DROP TABLE request duration | HIVE.HMS.API_DROP_TABLE | ms | Average duration of DROP TABLE requests |
| DROP TABLE WITH ENV CONTEXT request duration | HIVE.HMS.API_DROP_TABLE_WITH_ENV_CONTEXT | ms | Average duration of DROP TABLE WITH |

| | | | ENV CONTEXT requests |
|---|---|---|---|
| GET TABLE request duration | HIVE.HMS.API_GET_TABLE | ms | Average duration of GET TABLE requests |
| GET TABLES request duration | HIVE.HMS.API_GET_TABLES | ms | Average duration of GET TABLES requests |
| GET MULTI TABLE request duration | HIVE.HMS.API_GET_MULTI_TABLE | ms | Average duration of GET MULTI TABLE requests |
| GET TABLE REQ request duration | HIVE.HMS.API_GET_TABLE_REQ | ms | Average duration of GET TABLE REQ requests |
| GET DATABASE request duration | HIVE.HMS.API_GET_DATABASE | ms | Average duration of GET DATABASE requests |
| GET DATABASES request duration | HIVE.HMS.API_GET_DATABASES | ms | Average duration of GET DATABASES requests |
| GET ALL DATABASES request duration | HIVE.HMS.API_GET_ALL_DATABASES | ms | Average duration of GET ALL DATABASES requests |
| GET ALL FUNCTIONS request | HIVE.HMS.API_GET_ALL_FUNCTIONS | ms | Average duration of |

| duration | | | GET ALL FUNCTIONS requests |
|---|---|---|---|
| Current number of active CREATE TABLE requests | HIVE.HMS.ACTIVE_CALLS_API_CREATE_TABLE | - | Current number of active CREATE TABLE requests |
| Current number of active DROP TABLE requests | HIVE.HMS.ACTIVE_CALLS_API_DROP_TABLE | - | Current number of active DROP TABLE requests |
| Current number of active ALTER TABLE requests | HIVE.HMS.ACTIVE_CALLS_API_ALTER_TABLE | - | Current number of active ALTER TABLE requests |

# YARN Monitoring Metrics

Last updated：2023-05-30 11:24:00

## YARN - overview

| Title | Metric | Unit | Description |
|-------|--------|------|-------------|
| Nodes | NumActiveNMs | - | Number of live NodeManagers |
| | NumDecommissionedNMs | - | Number of decommissioned NodeManagers |
| | NumLostNMs | - | Number of lost NodeManagers |
| | NumUnhealthyNMs | - | Number of unhealthy NodeManagers |
| CPU cores | AllocatedVCores | - | Number of allocated VCores in the current queue |
| | ReservedVCores | - | Number of reserved VCores in the current queue |
| | AvailableVCores | - | Number of available VCores in the current queue |
| | PendingVCores | - | Number of pending VCores in resource requests in the current queue |
| Total applications | AppsSubmitted | - | Number of submitted jobs in the current queue |
| | AppsRunning | - | Number of running jobs in the current queue |
| | AppsPending | - | Number of pending jobs in the current queue |
| | AppsCompleted | - | Number of completed jobs in the current queue |
| | AppsKilled | - | Number of killed jobs in the current queue |
| | AppsFailed | - | Number of failed jobs in the |

| | | | current queue |
|---|---|---|---|
| | ActiveApplications | - | Number of active jobs in the current queue |
| | running_0 | - | Number of running jobs in the current queue that have run for less than 60 minutes |
| | running_60 | - | Number of running jobs in the current queue that have run for 60–300 minutes |
| | running_300 | - | Number of running jobs in the current queue that have run for 300–1,440 minutes |
| | running_1440 | - | Number of running jobs in the current queue that have run for more than 1,440 minutes |
| Memory size | AllocatedMB | MB | Amount of allocated memory in the current queue |
| | AvailableMB | MB | Amount of available memory in the current queue |
| | PendingMB | MB | Amount of pending memory in resource requests in the current queue |
| | ReservedMB | MB | Amount of reserved memory in the current queue |
| Containers | AllocatedContainers | - | Number of allocated containers in the current queue |
| | PendingContainers | - | Number of pending containers in resource requests in the current queue |
| | ReservedContainers | - | Number of reserved containers in the current queue |
| Total allocated/released containers | AggregateContainersAllocated | - | Total number of allocated containers in the current queue |
| | AggregateContainersReleased | - | Total number of released |

| | | | containers in the current queue |
|---|---|---|---|
| Users | ActiveUsers | - | Number of active users in the current queue |
| Memory | allocatedMB | MB | Amount of allocated memory in the cluster |
| | availableMB | MB | Amount of available memory in the cluster |
| | reservedMB | MB | Amount of reserved memory in the cluster |
| | totalMB | MB | Total amount of memory in the cluster |
| Applications | completed | - | Number of completed jobs in the cluster during the statistical period |
| | failed | - | Number of failed jobs in the cluster during the statistical period |
| | killed | - | Number of killed jobs in the cluster during the statistical period |
| | pending | - | Number of pending jobs in the cluster during the statistical period |
| | running | - | Number of running jobs in the cluster during the statistical period |
| | submitted | - | Number of submitted jobs in the cluster during the statistical period |
| Containers | containersAllocated | - | Number of allocated containers in the cluster |
| | containersPending | - | Number of pending containers in the cluster |
| | containersReserved | - | Number of reserved containers in the cluster |
| Memory utilization | usageRatio | % | Current memory utilization of the cluster |
| Cores | allocatedVirtualCores | - | Number of allocated CPU cores in |

| | | | the cluster |
|---|---|---|---|
| | availableVirtualCores | - | Number of available CPU cores in the cluster |
| | reservedVirtualCores | - | Number of reserved CPU cores in the cluster |
| | totalVirtualCores | - | Total number of CPU cores in the cluster |
| CPU utilization | usageRatio | % | Current CPU utilization of the cluster |
| Launched AMs | AMLaunchDelayNumOps | - | Launched AMs |
| Average time for RM to launch AM | AMLaunchDelayAvgTime | ms | Average time for RM to launch AM |
| Total registered AMs | AMRegisterDelayNumOps | - | Total registered AMs |
| Average time for AM to register with RM | AMRegisterDelayAvgTime | ms | Average time for AM to register with RM |
| Queue CPU utilization | YARN.RM.QUEUE.VCORES.RATIO | - | Utilization of CPU allocated for the current queue |
| Queue memory utilization | YARN.RM.QUEUE.MEM.RATIO | - | Utilization of memory allocated for the current queue |

## YARN - ResourceManager

| Title | Metric | Unit | Description |
|---|---|---|---|
| RPC authentications/authorizations | RpcAuthenticationFailures | - | Number of failed RPC authentications |
| | RpcAuthenticationSuccesses | - | Number of successful RPC authentications |
| | RpcAuthorizationFailures | - | Number of failed RPC authorizations |
| | RpcAuthorizationSuccesses | - | Number of successful RPC authorizations |

| Data received/sent by RPC | ReceivedBytes | bytes/s | Amount of data received by RPC |
|---|---|---|---|
| | SentBytes | bytes/s | Amount of data sent by RPC |
| RPC connections | NumOpenConnections | - | Current number of open connections |
| RPC requests | RpcProcessingTimeNumOps | - | Number of RPC requests |
| | RpcQueueTimeNumOps | - | Number of RPC requests |
| RPC queue length | CallQueueLength | - | Length of the current RPC queue |
| Average RPC processing time | RpcProcessingTimeAvgTime | s | Average RPC request processing time |
| | RpcQueueTimeAvgTime | s | Average time of RPC in the queue |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |

| | O | % | Percentage of used Old memory |
|---|---|---|---|
| | M | % | Percentage of used Metaspace memory |
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | ThreadsWaiting | - | Number of threads in WAITING status |
| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
| | ThreadsTerminated | - | Number of threads in Terminated status |
| JVM logs | LogFatal | - | Number of Fatal logs |
| | LogError | - | Number of Error logs |
| | LogWarn | - | Number of Warn logs |
| | LogInfo | - | Number of Info logs |
| JVM memory | MemNonHeapUsedM | MB | Non-heap memory size used by process |
| | MemNonHeapCommittedM | MB | Non-heap memory size committed to process |
| | MemHeapUsedM | MB | Heap memory size used by process |
| | MemHeapCommittedM | MB | Heap memory size committed to process |
| | | | |

| | MemHeapMaxM | MB | Maximum heap memory size available to process |
|---|---|---|---|
| | MemMaxM | MB | Maximum memory size available to process |
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | DaemonThreadCount | - | Number of daemon threads in the process |
| | ThreadCount | - | Number of threads in the process |
| Node status | haState | 1:Active,0:Standby | ResourceManager active/standby status |
| Active/Standby switch | switchOccurred | - | ResourceManager active/standby switch |

## YARN - JobHistoryServer

| Title | Metric | Unit | Description |
|---|---|---|---|
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | | | |

| | ThreadsWaiting | - | Number of threads in WAITING status |
|---|---|---|---|
| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
| | ThreadsTerminated | - | Number of threads in Terminated status |
| JVM logs | LogFatal | - | Number of FATAL-level logs |
| | LogError | - | Number of ERROR-level logs |
| | LogWarn | - | Number of WARN-level logs |
| | LogInfo | - | Number of INFO-level logs |
| JVM memory | MemNonHeapUsedM | MB | Non-heap memory size used by process |
| | MemNonHeapCommittedM | MB | Non-heap memory size committed to process |
| | MemHeapUsedM | MB | Heap memory size used by process |
| | MemHeapCommittedM | MB | Heap memory size committed to process |
| | MemHeapMaxM | MB | Maximum heap memory size available to process |
| | MemMaxM | MB | Maximum memory size available to process |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |

| | M | % | Percentage of used Metaspace memory |
|---|---|---|---|
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | DaemonThreadCount | - | Number of daemon threads in the process |
| | ThreadCount | - | Number of threads in the process |

## YARN - NodeManager

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |

| JVM threads | ThreadsNew | - | Number of threads in NEW status |
|---|---|---|---|
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | ThreadsWaiting | - | Number of threads in WAITING status |
| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
| | ThreadsTerminated | - | Number of threads currently in TERMINATED status |
| JVM logs | LogFatal | - | Number of FATAL-level logs |
| | LogError | - | Number of ERROR-level logs |
| | LogWarn | - | Number of WARN-level logs |
| | LogInfo | - | Number of INFO-level logs |
| JVM memory | MemNonHeapUsedM | MB | Non-heap memory size used by process |
| | MemNonHeapCommittedM | MB | Non-heap memory size committed to process |
| | MemHeapUsedM | MB | Heap memory size used by process |
| | MemHeapCommittedM | MB | Heap memory size committed to process |
| | MemHeapMaxM | MB | Maximum heap memory size available to process |
| | MemMaxM | MB | Maximum memory size available to process |
| Total containers | ContainersLaunched | - | Number of launched containers |
| | ContainersCompleted | - | Number of completed containers |
| | ContainersFailed | - | Number of failed containers |
| | ContainersKilled | - | Number of killed containers |

| | ContainersIniting | - | Number of containers being initialized |
|---|---|---|---|
| | ContainersRunning | - | Number of running containers |
| | AllocatedContainers | - | Number of containers allocated by NodeManager |
| Average container launch time | ContainerLaunchDurationAvgTime | ms | Average container launch time |
| Container launches | ContainerLaunchDurationNumOps | - | Container launches |
| CPU cores | AvailableVCores | - | Number of VCores available to NodeManager |
| | AllocatedVCores | - | Number of VCores allocated by NodeManager |
| Memory size | AllocatedGB | GB | Amount of memory allocated by NodeManager |
| | AvailableGB | GB | Amount of memory available to NodeManager |
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | DaemonThreadCount | - | Number of daemon threads in the process |
| | ThreadCount | - | Number of threads in the process |

# ZooKeeper Monitoring Metrics

Last updated：2023-12-27 14:48:59

## ZooKeeper

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |

| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
|---|---|---|---|
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| File descriptors | zk_max_file_descriptor_count | - | Maximum number of file descriptors |
| | zk_open_file_descriptor_count | - | Number of opened file descriptors |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | ThreadCount | - | Total number of threads |
| Connections | zk_num_alive_connections | - | Number of current connections |
| Latency | zk_avg_latency | ms | Average ZooKeeper processing latency |
| | zk_max_latency | ms | Maximum ZooKeeper processing latency |
| | zk_min_latency | ms | Minimum ZooKeeper processing latency |
| Zondes | zk_watch_count | - | Number of ZooKeeper watches |
| | zk_znode_count | - | Number of ZooKeeper znodes |
| | zk_ephemerals_count | - | Number of ephemeral ZooKeeper nodes |
| Data size | zk_approximate_data_size | Byte | Volume of data stored in ZooKeeper |

| Node status | zk_server_state | 1: Master. 0: Slave. 2: Standalone | ZooKeeper node type |
|---|---|---|---|
| Received/Sent packets | zk_packets_received | count/s | ZooKeeper packet receiving rate |
| | zk_packets_sent | count/s | ZooKeeper packet sending rate |
| Queuing requests | zk_outstanding_requests | - | Number of queuing requests |

# HBase Monitoring Metrics

Last updated：2023-12-27 14:49:18

## HBase - overview

| Title | Metric | Unit | Description |
|---|---|---|---|
| Cluster regions in RIT status | ritCount | - | Number of regions in transition |
| | ritCountOverThreshold | - | Number of regions that have been in transition for more than the threshold time |
| Cluster RIT time | ritOldestAge | ms | Age of the longest region in transition |
| Average number of regions per RegionServer | averageLoad | - | Average number of regions per RegionServer |
| Cluster RegionServers | numRegionServers | - | Number of live RegionServers |
| | numDeadRegionServers | - | Number of dead RegionServers |
| Data read/written from/to HMaster | receivedBytes | bytes/s | Amount of data received by cluster |
| | sentBytes | bytes/s | Amount of data sent by cluster |
| Total cluster API requests | clusterRequests | count/s | Total number of cluster requests |
| Cluster assignment manager operation | Assign_num_ops | - | Number of region assignments |
| | BulkAssign_num_ops | - | Number of bulk region assignments |
| Cluster load balancing operations | BalancerCluster_num_ops | - | Number of cluster load balancing operations |

## HBase - HMaster

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | | | |

| | GCT | s | Garbage collection time |
|---|---|---|---|
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM logs | LogFatal | - | Number of Fatal logs |
| | LogError | - | Number of Error logs |
| | LogWarn | - | Number of Warn logs |
| | LogInfo | - | Number of Info logs |
| JVM memory | MemNonHeapUsedM | MB | Non-heap memory size used by process |
| | MemNonHeapCommittedM | MB | Non-heap memory size committed to process |
| | MemHeapUsedM | MB | Heap memory size used by process |
| | MemHeapCommittedM | MB | Heap memory size committed to process |
| | MemHeapMaxM | MB | Maximum heap memory size available to process |
| | MemMaxM | MB | Maximum memory size available to process |
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | ThreadsWaiting | - | Number of threads in WAITING status |

| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
|---|---|---|---|
| | ThreadsTerminated | - | Number of threads currently in TERMINATED status |
| RPC connections | numOpenConnections | - | Number of RPC connections |
| RPC exceptions | FailedSanityCheckException | - | Number of FailedSanityCheckException exceptions |
| | NotServingRegionException | - | Number of NotServingRegionException exceptions |
| | OutOfOrderScannerNextException | - | Number of OutOfOrderScannerNextException exceptions |
| | RegionMovedException | - | Number of RegionMovedException exceptions |
| | RegionTooBusyException | - | Number of RegionTooBusyException exceptions |
| | UnknownScannerException | - | Number of UnknownScannerException exceptions |
| RPC queue requests | numCallsInPriorityQueue | - | Number of requests in the general queue |
| | numCallsInReplicationQueue | - | Number of RPC requests in the replication queue |
| Process start time | masterActiveTime | s | Master active time |
| | masterStartTime | s | Master process start time |

## HBase - RegionServer

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |

| | YGCT | s | Young GC time |
|---|---|---|---|
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM logs | LogFatal | - | Number of Fatal logs |
| | LogError | - | Number of Error logs |
| | LogWarn | - | Number of Warn logs |
| | LogInfo | - | Number of Info logs |
| JVM memory | MemNonHeapUsedM | MB | Non-heap memory size used by process |
| | MemNonHeapCommittedM | MB | Non-heap memory size committed to process |
| | MemHeapUsedM | MB | Heap memory size used by process |
| | MemHeapCommittedM | MB | Heap memory size committed to process |
| | MemHeapMaxM | MB | Maximum heap memory size available to process |
| | MemMaxM | MB | Maximum memory size available to process |
| JVM threads | ThreadsNew | - | Number of threads in NEW status |
| | ThreadsRunnable | - | Number of threads in RUNNABLE status |
| | ThreadsBlocked | - | Number of threads in BLOCKED status |
| | ThreadsWaiting | - | Number of threads in WAITING status |

| | ThreadsTimedWaiting | - | Number of threads in TIMED WAITING status |
|---|---|---|---|
| | ThreadsTerminated | - | Number of threads currently in TERMINATED status |
| Regions | regionCount | - | Number of regions |
| Region localization | percentFilesLocal | % | Percentage of HFiles on the local HDFS data node in the region |
| Region replica localization | percentFilesLocalSecondaryRegions | % | Percentage of HFiles on the local HDFS data node in the region replica |
| RPC authentications | authenticationFailures | - | Number of RPC authentication failures |
| | authenticationSuccesses | - | Number of RPC authentication successes |
| RPC connections | numOpenConnections | - | Number of RPC connections |
| RPC exceptions | FailedSanityCheckException | - | Number of FailedSanityCheckException exceptions |
| | NotServingRegionException | - | Number of NotServingRegionException exceptions |
| | OutOfOrderScannerNextException | - | Number of OutOfOrderScannerNextException exceptions |
| | RegionMovedException | - | Number of RegionMovedException exceptions |
| | RegionTooBusyException | - | Number of RegionTooBusyException exceptions |
| | UnknownScannerException | - | Number of UnknownScannerException exceptions |
| RPC handlers | numActiveHandler | - | Number of active RPC handlers |
| | numActiveWriteHandler | - | Number of active read RPC handlers |
| | numActiveReadHandler | - | Number of active write RPC handlers |

| | numActiveScanHandler | - | Number of active scan RPC handlers |
|---|---|---|---|
| | | | |
| RPC queue requests | numCallsInPriorityQueue | - | Number of requests in the priority queue |
| | numCallsInReplicationQueue | - | Number of RPC requests in the replication queue |
| | numCallsInPriorityQueue | - | Number of requests in the general queue |
| | numCallsInWriteQueue | - | Number of RPC calls in the write call queue |
| | numCallsInReadQueue | - | Number of RPC calls in the read call queue |
| | numCallsInScanQueue | - | Number of RPC calls in the scan call queue |
| WAL files | hlogFileCount | - | Number of WAL files |
| WAL file size | hlogFileSize | Byte | WAL file size |
| MemStore size | memStoreSize | MB | MemStore size |
| Stores | storeCount | - | Number of stores |
| StoreFiles | storeFileCount | - | Number of StoreFiles |
| StoreFile size | storeFileSize | MB | StoreFile size |
| Disk write rate | flushedCellsSize | bytes/s | Disk write rate |
| Average latency | Append_mean | ms | Average Append latency |
| | Replay_mean | ms | Average Replay latency |
| | Get_mean | ms | Average GET latency |
| | updatesBlockedTime | ms | Number of milliseconds updates have been blocked so the memstore can be flushed |
| RegionServer disk writes | FlushTime_num_ops | - | Number of MemStore flushes |
| | | | |

| Requests in operation queue | splitQueueLength | - | Length of the split queue |
|---|---|---|---|
| | compactionQueueLength | - | Length of the compaction queue |
| | flushQueueLength | - | Length of the region flush queue |
| Replay operations | Replay_num_ops | - | Number of Replay operations |
| Slow operations | slowAppendCount | - | Number of Append requests that took over 1s to complete |
| | slowDeleteCount | - | Number of Delete requests that took over 1s to complete |
| | slowGetCount | - | Number of Get requests that took over 1s to complete |
| | slowIncrementCount | - | Number of Increment requests that took over 1s to complete |
| | slowPutCount | - | Number of Put requests that took over 1s to complete |
| Split request | splitRequestCount | - | Number of split requested |
| | splitSuccessCount | - | Number of successfully executed splits |
| Cache blocks | blockCacheCount | - | Number of blocks in the block cache |
| | blockCacheHitCount | - | Number of block cache hits |
| | blockCacheMissCount | - | Number of block cache misses |
| Cache read hit rate | blockCacheExpressHitPercent | % | Cache read hit rate |
| Memory size used by the cache block | blockCacheSize | Byte | Memory size used by the cache block |
| Index size | staticBloomSize | Byte | Uncompressed size of static bloom filters |
| | staticIndexSize | Byte | Uncompressed size of static indexes |
| | storeFileIndexSize | Byte | Size of indexes in StoreFiles on disk |
| Received bytes | receivedBytes | bytes/s | Received bytes |

| | sentBytes | bytes/s | Sent bytes |
|---|---|---|---|
| Read and write requests | Total | count/s | Total number of requests. When there are scan requests, this value will be smaller than the sum of read and write requests |
| | Read | count/s | Number of read requests |
| | Write | count/s | Number of write requests |
| | Append_num_ops | count/s | Number of Append requests |
| | Mutate_num_ops | count/s | Number of Mutate requests |
| | Delete_num_ops | count/s | Number of Delete requests |
| | Increment_num_ops | count/s | Number of Increment requests |
| | Get_num_ops | count/s | Number of Get requests |
| | Put_num_ops | count/s | Number of Put requests |
| | ScanTime_num_ops | count/s | Scan requests (time) |
| | ScanSize_num_ops | count/s | Scan requests (size) |
| Mutations | mutationsWithoutWALCount | - | Number of mutations |
| Mutation size | mutationsWithoutWALSize | Byte | Mutation size |
| Process start time | regionServerStartTime | s | Process start time |
| Log sync | source.sizeOfLogQueue | - | Total length of synced logs |
| Sync duration | source.ageOfLastShippedOp | ms | Sync duration |
| Requests | ReadRequestCount | count/s | Read requests/s |
| | WriteRequestCount | count/s | Write requests/s |
| Requests | Read | count/s | Read requests/s |
| | Write | count/s | Write requests/s |
| Store size | memstoreSize | Byte | MemStore size |
| | storeFileSize | Byte | StoreFile size |

| Table-level request latency | getTime_99th_percentile | ms | 99th percentile of request processing latency |
|---|---|---|---|
| | scanTime_99th_percentile | ms | 99th percentile of request processing latency |
| | putTime_99th_percentile | ms | 99th percentile of request processing latency |
| | incrementTime_99th_percentile | ms | 99th percentile of request processing latency |
| | appendTime_99th_percentile | ms | 99th percentile of request processing latency |
| | deleteTime_99th_percentile | ms | 99th percentile of request processing latency |
| Request processing latency | 99th_percentile | ms | 99th percentile of request processing latency |
| | 99.9th_percentile | ms | 99.9% request processing latency |
| Request queueing latency | 99th_percentile | ms | 99th percentile of request queueing latency |
| | 99.9th_percentile | ms | 99.9% request queueing latency |
| Scan size | max | bytes | Maximum scan size |
| | mean | bytes | Average scan size |
| | min | bytes | Minimum scan size |
| Scan time | max | s | Maximum scan time |
| | mean | s | Average scan time |
| | min | s | Minimum scan time |

# Hive Monitoring Metrics

Last updated：2023-12-27 14:49:37

## Hive - HiveMetaStore

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC c |
| | FGC | - | Full GC cou |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection tir |
| | YGCT | s | Young GC t |
| Memory zone proportion | S0 | % | Percentage used Surviv memory |
| | E | % | Percentage used Eden memory |
| | CCS | % | Percentage used compressed class space memory |
| | S1 | % | Percentage used Surviv memory |
| | O | % | Percentage used Old me |
| | M | % | Percentage used Metas memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemo |

| | | | |
|---|---|---|---|
| | | | currently us JVM |
| | MemHeapCommittedM | MB | Size of HeapMemo committed t JVM |
| | MemHeapMaxM | MB | Size of HeapMemo configured t JVM |
| | MemHeapInitM | MB | Size of initia JVM HeapM |
| | MemNonHeapUsedM | MB | Size of NonHeapM currently us JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapM currently committed t JVM |
| | MemNonHeapInitM | MB | Size of initia JVM NonHeapM |
| File descriptors | OpenFileDescriptorCount | - | Number of opened file descriptors |
| | MaxFileDescriptorCount | - | Maximum number of fi descriptors |
| CPU utilization | ProcessCpuLoad | % | Process CP utilization |
| | SystemCpuLoad | % | System CPU utilization |
| Percentage of CPU | CPURate | seconds/second | Percentage CPU usage |

| | | | |
|---|---|---|---|
| usage time | | | |
| Worker threads | DaemonThreadCount | - | Number of daemon thre |
| | ThreadCount | - | Total numbe threads |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative usage time |
| Process execution duration | Uptime | s | Process execution duration |
| GC extra sleep time | ExtraSleepTime | ms/s | GC extra sle time |
| ALTER TABLE request duration | HIVE.HMS.API_ALTER_TABLE | ms | Average dur of ALTER T requests |
| ALTER TABLE WITH ENV CONTEXT request duration | HIVE.HMS.API_ALTER_TABLE_WITH_ENV_CONTEXT | ms | Average dur of ALTER T WITH ENV CONTEXT requests |
| CREATE TABLE request duration | HIVE.HMS.API_CREATE_TABLE | ms | Average dur of CREATE TABLE requ |
| CREATE TABLE WITH ENV CONTEXT request duration | HIVE.HMS.API_CREATE_TABLE_WITH_ENV_CONTEXT | ms | Average dur of CREATE TABLE WIT ENV CONT requests |
| DROP TABLE request duration | HIVE.HMS.API_DROP_TABLE | ms | Average dur of DROP TA requests |

| DROP TABLE WITH ENV CONTEXT request duration | HIVE.HMS.API_DROP_TABLE_WITH_ENV_CONTEXT | ms | Average dur of DROP TA WITH ENV CONTEXT requests |
|---|---|---|---|
| GET TABLE request duration | HIVE.HMS.API_GET_TABLE | ms | Average dur of GET TAB requests |
| GET TABLES request duration | HIVE.HMS.API_GET_TABLES | ms | Average dur of GET TAB requests |
| GET MULTI TABLE request duration | HIVE.HMS.API_GET_MULTI_TABLE | ms | Average dur of GET MUI TABLE requ |
| GET TABLE REQ request duration | HIVE.HMS.API_GET_TABLE_REQ | ms | Average dur of GET TAB REQ reques |
| GET DATABASE request duration | HIVE.HMS.API_GET_DATABASE | ms | Average dur of GET DATABASE requests |
| GET DATABASES request duration | HIVE.HMS.API_GET_DATABASES | ms | Average dur of GET DATABASE requests |
| GET ALL DATABASES request duration | HIVE.HMS.API_GET_ALL_DATABASES | ms | Average dur of GET ALL DATABASE requests |
| GET ALL FUNCTIONS request duration | HIVE.HMS.API_GET_ALL_FUNCTIONS | ms | Average dur of GET ALL FUNCTION requests |
| Current number of | HIVE.HMS.ACTIVE_CALLS_API_CREATE_TABLE | - | Current num of active |

| | | | |
|---|---|---|---|
| active CREATE TABLE requests | | | CREATE TA requests |
| Current number of active DROP TABLE requests | HIVE.HMS.ACTIVE_CALLS_API_DROP_TABLE | - | Current num of active DF TABLE requ |
| Current number of active ALTER TABLE requests | HIVE.HMS.ACTIVE_CALLS_API_ALTER_TABLE | - | Current num of active AL TABLE requ |

## Hive - HiveServer2

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |

| | S1 | % | Percentage of used Survivor 1 memory |
| --- | --- | --- | --- |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Percentage of used heap memory | MemHeapUsedRate | % | Proportion of the heap memory |

| | | | size currently used by JVM to the heap memory size configured for JVM |
|---|---|---|---|
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| Percentage of CPU usage time | CPUUsedRate | seconds/second | Percentage of CPU usage time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | ThreadCount | - | Total number of threads |
| Driver execution latency | 99th_percentile | ms | 99th percentile of driver execution latency |
| | Avg | ms | Average driver execution latency |
| Open connections | NumOpenConnections | - | Number of open connections |
| Current size of the pool of async HiveServer2 threads | PoolSize | - | Current size of the pool of async HiveServer2 threads |
| Current size of the queue | QueueSize | - | Current size of |

| for async HiveServer2 operations | | | the queue for async HiveServer2 operations |
|---|---|---|---|
| Hive operations | Closed | - | Number of closed operations |
| | Finished | - | Number of completed operations |
| | Canceled | - | Number of canceled operations |
| | Error | - | Number of erroneous operations |
| GC extra sleep time | ExtraSleepTime | ms/s | GC extra sleep time |
| Number of API requests | HIVE.H2.ACTIVE.CALLS.API | Count | Current number of serializePlan requests |
| | | Count | Current number of semanticAnalyze requests |
| | | Count | Current number of runtask requests |
| | | Count | Current number of releaseLocks requests |
| | | Count | Current number of getSplits requests |
| Duration of SQL tasks in PENDING status | HIVE.H2.SQL.OPERATION.PENDING | ms | Average duration of SQL tasks in PENDING status |

| Duration of SQL tasks in RUNNING status | HIVE.H2.SQL.OPERATION.RUNNING | ms | Average duration of SQL tasks in RUNNING status |
|---|---|---|---|
| Current number of active users | HIVE.H2.SQL.OPERATION | Count | Current number of active users |
| Query execution time | HIVE.H2.EXECUTING.QUERIES | ms | Average query execution time |
| Query submission time | HIVE.H2.SUBMITTED.QUERIES | ms | Query submission time |
| Number of submitted Hive on MR jobs | HIVE.H2.MR.TASKS | Count | Number of submitted Hive on MR jobs |
| Number of submitted Hive on Spark jobs | HIVE.H2.SPARK.TASKS | Count | Number of submitted Hive on Tez jobs |
| Number of submitted Hive on Tez jobs | HIVE.H2.TEZ.TASKS | Count | Number of submitted Hive on Spark jobs |
| Failed queries | HIVE.H2.FAILED.QUERIES.RATE | Count/min | Number of failed queries per minute |
| Number of worker threads | HIVE.H2.THREAD.COUNT | - | Number of JVM blocked threads |
| | | - | Number of JVM terminate threads |
| | | - | Number of JVM deadlock threads |
| | | - | Number of JVM new threads |
| | | - | Number of JVM runnable threads |
| | | - | Number of JVM timed_waiting threads |
| | | | |

| | | - | Number of JVM waiting threads |
|---|---|---|---|
| Number of sessions | HIVE.H2.OPEN.SESSIONS | - | Number of open sessions |
| Current number of active sessions | HIVE.H2.ACTIVE.SESSIONS | - | Number of active sessions |

## Hive - HiveWebHcat

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |

# Spark Monitoring Metrics

Last updated：2023-12-27 14:49:54

## Spark - history server

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |

# Presto Monitoring Metrics

Last updated：2023-12-27 14:50:22

## Presto - overview

| Title | Metric | Unit | Description |
|---|---|---|---|
| Nodes | Active | - | Number of active nodes |
| | Total | - | Total number of nodes |
| | Failed | - | Number of failed nodes |
| Query | RunningQueries | - | Total number of running queries |
| | QueuedQueries | - | Total number of waiting queries |
| Query frequency | FailedQueries | count/min | Total number of failed queries |
| | AbandonedQueries | count/min | Total number of abandoned queries |
| | CanceledQueries | count/min | Total number of canceled queries |
| | CompletedQueries | count/min | Total number of completed queries |
| | StartedQueries | count/min | Total number of started queries |
| Data volume input/output per minute | InputDataSizeOneMinute | GB/min | Data input rate |
| | OutputDataSizeOneMinute | GB/min | Data output rate |

## Presto - worker

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | | | |

| | FGC | - | Full GC count |
|---|---|---|---|
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Data input/output rate | InputDataSize.OneMinute.Rate | GB/min | Data input rate |
| | OutputDataSize.OneMinute.Rate | GB/min | Data output rate |
| Processes | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Number of threads |
| | | | |

| | DaemonThreadCount | - | Number of backend threads |
|---|---|---|---|
| Process execution duration | Uptime | s | Process execution duration |
| Process start time | StartTime | s | Process start time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

## Presto - coordinator

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed |

| | | | by JVM |
|---|---|---|---|
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Processes | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Number of threads |
| | DaemonThreadCount | - | Number of backend threads |
| Process execution duration | Uptime | s | Process execution duration |
| Process start time | StartTime | s | Process start time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

# Trino Monitoring Metrics

Last updated：2023-12-27 14:50:38

## Trino - Overview

| Title | Metric | Unit | Description |
|---|---|---|---|
| Nodes | Active | - | Number of active nodes |
| | Total | - | Total number of nodes |
| | Failed | - | Number of failed nodes |
| Query | RunningQueries | - | Total number of running queries |
| | QueuedQueries | - | Total number of waiting queries |
| Query frequency | FailedQueries | count/min | Total number of failed queries |
| | AbandonedQueries | count/min | Total number of abandoned queries |
| | CanceledQueries | count/min | Total number of canceled queries |
| | CompletedQueries | count/min | Total number of completed queries |
| | StartedQueries | count/min | Total number of started queries |
| Data volume input/output per minute | InputDataSizeOneMinute | GB/min | Data input rate |
| | OutputDataSizeOneMinute | GB/min | Data output rate |

**Trino - worker**

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |

| | FGC | - | Full GC count |
|---|---|---|---|
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Data input/output rate | InputDataSize.OneMinute.Rate | GB/min | Data input rate |
| | OutputDataSize.OneMinute.Rate | GB/min | Data output rate |
| Worker threads | PeakThreadCount | - | Peak number of threads |
| | | | |

| | ThreadCount | - | Number of threads |
|---|---|---|---|
| | DaemonThreadCount | - | Number of backend threads |
| Process execution duration | Uptime | s | Process execution duration |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

## Trino - coordinator

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed |

| | | | by JVM |
|---|---|---|---|
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Worker threads | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Number of threads |
| | DaemonThreadCount | - | Number of backend threads |
| Process execution duration | Uptime | s | Process execution duration |
| Process start time | StartTime | s | Process start time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

# ClickHouse Monitoring Metrics

Last updated：2023-12-27 14:50:56

**ClickHouse - metrics**

| Title | Metric | Unit | Description |
|-------|--------|------|-------------|
| Network connections | tcp | - | Number of TCP server connections |
| | http | - | Number of HTTP server connections |
| ZooKeeper event subscriptions | watches | - | Number of ZooKeeper event subscriptions |
| Ephemeral nodes held in ZooKeeper | ephemeralNode | - | Number of ephemeral nodes held in ZooKeeper |
| Active tasks in BackgroundPool | backgroundPoolTask | - | Number of active tasks in BackgroundProcessingPool |
| | backgroundSchedulePoolTask | - | Number of active tasks in BackgroundSchedulePool |
| Threads waiting for locks in Context | contextLockWait | - | Number of threads waiting for locks in Context |
| Throttled INSERT queries | delayedInserts | - | Number of throttled INSERT queries |
| Requests in fly to data sources of dictionaries of cache type | dictCacheRequests | - | Number of requests in fly to data sources of dictionaries of cache type |
| Pending files to process for asynchronous insertion into distributed tables | distributedSend | - | Number of pending files to process for asynchronous insertion into distributed tables |
| Threads | global | - | Number of threads in the global thread pool |
| | globalActive | - | Number of active threads in the global thread pool |

| | local | - | Number of threads in the local thread pool |
|---|---|---|---|
| | localActive | - | Number of active threads in the local thread pool |
| Replicas participating in leader election | leaderElection | - | Number of replicas participating in leader election |
| Replicated tables | leaderReplica | - | Number of replicated tables that are leaders |
| | readonlyReplica | - | Number of replicated tables in read-only status |
| Total allocated memory size | memoryTracking | GB | Total memory size allocated to queries being executed |
| | backgroundProcessingPool | GB | Total memory size allocated to the background processing pool |
| | backgroundSchedulePool | GB | Total memory size allocated to the background scheduling pool |
| | memoryTrackingForMerges | GB | Total memory size allocated to background merges |
| Merges being executed in the background | merge | - | Number of merges being executed in the background |
| Opened files | read | - | Number of opened readable files |
| | write | - | Number of opened writable files |
| Table changes | partMutation | - | Number of table changes |
| Query processing threads | queryThread | - | Number of query processing threads |
| Stopped or waiting queries | queryPreempted | - | Number of stopped or waiting queries |

| System calls | read | - | Number of read system calls |
|---|---|---|---|
| | write | - | Number of write system calls |
| Blocks | fetch | - | Number of blocks collected from replicas |
| | send | - | Number of blocks sent to replicas |
| | check | - | Number of blocks for consistency check |
| Server change | revision | - | Number of server changes |
| Version number | version | 1 | Version number |
| Threads waiting for read/write locks | waitingRead | - | Number of threads waiting for read/write locks to read from tables |
| | waitingWrite | - | Number of threads waiting for read/write locks to write to tables |
| | activeRead | - | Number of threads holding read locks on tables |
| | activeWrite | - | Number of threads holding write locks on tables |
| Rows in buffer tables | storageBufferRows | - | Number of rows in buffer tables |
| Bytes in buffer tables | storageBufferBytes | MB | Number of bytes in buffer tables |

## ClickHouse - events

| Title | Metric | Unit | Description |
|---|---|---|---|
| Queries | total | - | Total number of queries that may be executed |
| | select | - | Number of SELECT queries that may be executed |
| | | | |

| | insert | - | Number of INSERT queries that may be executed |
|---|---|---|---|
| Inserted rows | insertedRows | - | Number of rows inserted into all tables |
| Inserted bytes | insertedBytes | - | Number of bytes inserted into all tables |
| Total time waiting for system calls | read | µs | Total time waiting for read system calls |
| | write | µs | Total time waiting for write system calls |
| Opened files | fileOpen | - | Number of opened files |
| Reads/Writes from/to file descriptors | read | - | Number of reads from file descriptors |
| | write | - | Number of writes to file descriptors |
| Bytes read/written from/to file descriptors | read | MB | Number of bytes read from file descriptors |
| | write | MB | Number of bytes written to file descriptors |
| Total time processing threads | realtime | µs | Total time processing threads |
| | user | µs | Total time processing threads executing CPU instructions in user space |
| | system | µs | Total time processing threads executing CPU instructions in OS kernel space |
| Compiled regular expressions | regexpCreated | - | Number of compiled regular expressions |

## ClickHouse - asynchronous metrics

| Title | Metric | Unit | Description |
|---|---|---|---|
| Size of cache of marks of StorageMergeTree | markCacheBytes | MB | Size of cache of marks of StorageMergeTree |
| Cache files of marks of StorageMergeTree | markCacheFiles | - | Number of cache files of marks of StorageMergeTree |
| Maximum number of active blocks | maxPartCountForPartition | - | Number of largest active data blocks in partition |
| Databases | databaseCount | - | Number of databases |
| Data tables | tableCount | - | Number of data tables |
| Maximum replica latency | absolute | µs | Maximum replica queue latency |

| | relative | µs | Maximum difference of the absolute latency from other blocks |
|---|---|---|---|
| Size of the queue for the operations to be performed | replicasMaxQueueSize | - | Size of the queue for the operations to be performed |

# Druid Monitoring Metrics

Last updated：2023-12-27 14:51:14

**Druid - Broker**

| Metric Name | Unit | Description |
|---|---|---|
| YGC | - | Young GC count |
| FGC | - | Full GC count |
| FGCT | s | Full GC time |
| GCT | s | Garbage collection time |
| YGCT | s | Young GC time |
| S0 | % | Percentage of used Survivor 0 memory |
| E | % | Percentage of used Eden memory |
| CCS | % | Percentage of memory used by compressed class space |
| S1 | % | Percentage of used Survivor 1 memory |
| O | % | Percentage of used Old memory |
| M | % | Percentage of used Metaspace memory |
| MemHeapInitM | MB | Size of initial JVM HeapMem |
| MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| ThreadsNew | - | Number of threads in NEW state |
| ThreadsRunnable | - | Number of threads in RUNNABLE state |
| ThreadsBlocked | - | Number of threads in BLOCKED state |

| ThreadsWaiting | - | Number of threads in WAITING state |
|---|---|---|
| ThreadsTimedWaiting | - | Number of threads in TIMED_WAITING state |
| ThreadsTerminated | - | Number of threads in TERMINATED state |
| LogFatal | - | Number of FATAL logs |
| LogError | - | Number of ERROR logs |
| LogWarn | - | Number of WARN logs |
| LogInfo | - | Number of INFO logs |
| jetty.numOpenConnections | - | Number of open Jetty connections |
| segment.scan.pending | - | Number of segments in queue waiting to be scanned |
| broker.query.count | - | number of total queries |
| broker.query.success.count | - | number of queries successfully processed |
| broker.query.failed.count | - | number of failed queries |
| broker.query.interrupted.count | - | number of queries interrupted due to cancellation or timeout |
| normal.count | - | Number of queries with delay shorter than 1 second |
| abnormal.count | - | Number of queries with delay longer than or equal to 1 second |

## Druid - Coordinator

| Metric Name | Unit | Description |
|---|---|---|
| segment.assigned.count | - | Number of segments loaded into Druid cluster |
| segment.moved.count | - | Number of segments moved in Druid cluster |
| segment.dropped.count | - | Number of segments deleted in Druid cluster due to expiration |
| segment.deleted.count | - | Number of segments deleted in Druid cluster due to rule settings |
| segment.unneeded.count | - | Number of segments deleted in Druid cluster for being marked as "unused" |
| segment.cost.raw | ms | Used in cost balancing. The raw cost of hosting segments. |
| segment.cost.normalization | ms | Used in cost balancing. The normalization of hosting segments. |

| segment.cost.normalized | ms | Used in cost balancing. The normalized cost of hosting segments. |
|---|---|---|
| segment.loadQueue.size | Bytes | Size in bytes of segments to load. |
| segment.loadQueue.failed | - | Number of segments that failed to load. |
| segment.loadQueue.count | - | Number of segments to load. |
| segment.dropQueue.count | - | Number of segments to drop. |
| segment.overshadowed.count | - | Number of overshadowed segments. |
| tier.historical.count | - | Number of available historical nodes in each tier. |
| tier.replication.factor | - | Configured maximum replication factor in each tier. |
| tier.required.capacity | Bytes | Total capacity in bytes required in each tier. |
| tier.total.capacity | Bytes | Total capacity in bytes available in each tier. |
| compact.task.count | - | Number of compaction tasks |
| YGC | - | Young GC count |
| FGC | - | Full GC count |
| FGCT | s | Full GC time |
| GCT | s | Garbage collection time |
| YGCT | s | Young GC time |
| S0 | % | Percentage of used Survivor 0 memory |
| E | % | Percentage of used Eden memory |
| CCS | % | Percentage of memory used by compressed class space |
| S1 | % | Percentage of used Survivor 1 memory |
| O | % | Percentage of used Old memory |
| M | % | Percentage of used Metaspace memory |
| MemHeapInitM | MB | Size of initial JVM HeapMem |
| MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | | |

| MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
|---|---|---|
| MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| ThreadsNew | - | Number of threads in NEW state |
| ThreadsRunnable | - | Number of threads in RUNNABLE state |
| ThreadsBlocked | - | Number of threads in BLOCKED state |
| ThreadsWaiting | - | Number of threads in WAITING state |
| ThreadsTimedWaiting | - | Number of threads in TIMED_WAITING state |
| ThreadsTerminated | - | Number of threads in TERMINATED state |
| LogFatal | - | Number of FATAL logs |
| LogError | - | Number of ERROR logs |
| LogWarn | - | Number of WARN logs |
| LogInfo | - | Number of INFO logs |
| segment.size | Bytes | Total size of used segments in a data source. Emitted only for data sources to which at least one used segment belongs. |
| segment.count | - | Number of used segments belonging to a data source. Emitted only for data sources to which at least one used segment belongs. |
| segment.unavailable.count | - | Number of segments (not including replicas) left to load until segments that should be loaded in the cluster are available for queries. |
| segment.underReplicated.count | - | Number of segments (including replicas) left to load until segments that should be loaded in the cluster are available for queries. |
| jetty.numOpenConnections | - | Number of open Jetty connections |

## Druid - Historical

| Metric Name | Unit | Description |
|---|---|---|
|  |  |  |

| YGC | - | Young GC count |
|---|---|---|
| FGC | - | Full GC count |
| FGCT | s | Full GC time |
| GCT | s | Garbage collection time |
| YGCT | s | Young GC time |
| S0 | % | Percentage of used Survivor 0 memory |
| E | % | Percentage of used Eden memory |
| CCS | % | Percentage of memory used by compressed class space |
| S1 | % | Percentage of used Survivor 1 memory |
| O | % | Percentage of used Old memory |
| M | % | Percentage of used Metaspace memory |
| MemHeapInitM | MB | Size of initial JVM HeapMem |
| MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| ThreadsNew | - | Number of threads in NEW state |
| ThreadsRunnable | - | Number of threads in RUNNABLE state |
| ThreadsBlocked | - | Number of threads in BLOCKED state |
| ThreadsWaiting | - | Number of threads in WAITING state |
| ThreadsTimedWaiting | - | Number of threads in TIMED_WAITING state |
| ThreadsTerminated | - | Number of threads in TERMINATED state |
| LogFatal | - | Number of FATAL logs |

| LogError | - | Number of ERROR logs |
|---|---|---|
| LogWarn | - | Number of WARN logs |
| LogInfo | - | Number of INFO logs |
| jetty.numOpenConnections | - | Number of open Jetty connections |
| segment.scan.pending | - | Number of segments in queue waiting to be scanned |
| segment.max | Bytes | Maximum byte limit available for segments |
| segment.pendingdelete | Bytes | On-disk size in bytes of segments that are waiting to be cleared out |
| historical.query.count | - | Total number of historical queries |
| historical.query.success.count | - | Total number of historical query successes |
| historical.query.failed.count | - | Total number of historical query failures |
| historical.query.interrupted.count | - | Total number of historical query interruptions |
| normal.count | - | Number of queries with delay shorter than 1 second |
| abnormal.count | - | Number of queries with delay longer than or equal to 1 second |

## Druid - MiddleManager

| Metric Name | Unit | Description |
|---|---|---|
| YGC | - | Young GC count |
| FGC | - | Full GC count |
| FGCT | s | Full GC time |
| GCT | s | Garbage collection time |
| YGCT | s | Young GC time |
| S0 | % | Percentage of used Survivor 0 memory |
| E | % | Percentage of used Eden memory |
| CCS | % | Percentage of memory used by compressed class space |
| S1 | % | Percentage of used Survivor 1 memory |
| | | |

| O | % | Percentage of used Old memory |
|---|---|---|
| M | % | Percentage of used Metaspace memory |
| MemHeapInitM | MB | Size of initial JVM HeapMem |
| MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| ThreadsNew | - | Number of threads in NEW state |
| ThreadsRunnable | - | Number of threads in RUNNABLE state |
| ThreadsBlocked | - | Number of threads in BLOCKED state |
| ThreadsWaiting | - | Number of threads in WAITING state |
| ThreadsTimedWaiting | - | Number of threads in TIMED_WAITING state |
| ThreadsTerminated | - | Number of threads in TERMINATED state |
| LogFatal | - | Number of FATAL logs |
| LogError | - | Number of ERROR logs |
| LogWarn | - | Number of WARN logs |
| LogInfo | - | Number of INFO logs |
| jetty.numOpenConnections | - | Number of open Jetty connections |

## Druid - Overlord

| Metric Name | Unit | Description |
|---|---|---|
| YGC | - | Young GC count |
| FGC | - | Full GC count |
| FGCT | s | Full GC time |

| GCT | s | Garbage collection time |
|---|---|---|
| YGCT | s | Young GC time |
| S0 | % | Percentage of used Survivor 0 memory |
| E | % | Percentage of used Eden memory |
| CCS | % | Percentage of memory used by compressed class space |
| S1 | % | Percentage of used Survivor 1 memory |
| O | % | Percentage of used Old memory |
| M | % | Percentage of used Metaspace memory |
| MemHeapInitM | MB | Size of initial JVM HeapMem |
| MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| ThreadsNew | - | Number of threads in NEW state |
| ThreadsRunnable | - | Number of threads in RUNNABLE state |
| ThreadsBlocked | - | Number of threads in BLOCKED state |
| ThreadsWaiting | - | Number of threads in WAITING state |
| ThreadsTimedWaiting | - | Number of threads in TIMED_WAITING state |
| ThreadsTerminated | - | Number of threads in TERMINATED state |
| LogFatal | - | Number of FATAL logs |
| LogError | - | Number of ERROR logs |
| LogWarn | - | Number of WARN logs |
| LogInfo | - | Number of INFO logs |

| jetty.numOpenConnections | - | Number of open Jetty connections |
|---|---|---|

## Druid - Router

| Metric Name | Unit | Description |
|---|---|---|
| YGC | - | Young GC count |
| FGC | - | Full GC count |
| FGCT | s | Full GC time |
| GCT | s | Garbage collection time |
| YGCT | s | Young GC time |
| S0 | % | Percentage of used Survivor 0 memory |
| E | % | Percentage of used Eden memory |
| CCS | % | Percentage of memory used by compressed class space |
| S1 | % | Percentage of used Survivor 1 memory |
| O | % | Percentage of used Old memory |
| M | % | Percentage of used Metaspace memory |
| MemHeapInitM | MB | Size of initial JVM HeapMem |
| MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| ThreadsNew | - | Number of threads in NEW state |
| ThreadsRunnable | - | Number of threads in RUNNABLE state |
| ThreadsBlocked | - | Number of threads in BLOCKED state |
| ThreadsWaiting | - | Number of threads in WAITING state |

| ThreadsTimedWaiting | - | Number of threads in TIMED_WAITING state |
|---|---|---|
| ThreadsTerminated | - | Number of threads in TERMINATED state |
| LogFatal | - | Number of FATAL logs |
| LogError | - | Number of ERROR logs |
| LogWarn | - | Number of WARN logs |
| LogInfo | - | Number of INFO logs |
| jetty.numOpenConnections | - | Number of open Jetty connections |

# Kudu Monitoring Metrics

Last updated：2023-12-27 14:51:37

## Kudu - overview

| Title | Metric | Unit | Description |
|---|---|---|---|
| Tablets | TabletRunning | - | Total number of tablets currently running on all tablet servers |
| Difference in the number of tablet replicas | ClusterReplicaSkew | - | Difference between the number of replicas on the tablet server hosting the most replicas and the number of replicas on the tablet server hosting the fewest replicas |
| TServer threads | ThreadsRunning | - | Number of threads currently running on all tablet servers |
| Master threads | ThreadsRunning | - | Number of threads currently running on all masters |
| TServer logs | ErrorMessages | - | Number of ERROR-level log messages emitted in all processes |
| Master logs | ErrorMessages | - | Number of ERROR-level log messages emitted in all processes |
| | WarningMessages | - | Number of WARNING-level log messages emitted in all processes |
| Oversized write requests | OversizedWriteRequests | - | Number of oversized write requests to the system catalog tablet rejected by the master since start |

## Kudu - server

| Title | Metric | Unit | Description |
|---|---|---|---|
| Block cache hit | BlockCacheHit | - | Number of block cache hits. When confirming the cache efficiency, use the value of this metric instead of that of cache_hits |
| | BlockCacheMiss | - | Number of block cache misses. When confirming the cache efficiency, use the |

| | | | value of this metric instead of that of cache_misses |
|---|---|---|---|
| Block cache utilization | BlockCacheUsage | bytes | Memory used by block cache |
| File cache hit | FileCacheHit | - | Number of file descriptor cache hits. When confirming the cache efficiency, use the value of this metric instead of that of cache_hits |
| | FileCacheMiss | - | Number of file descriptor cache misses. When confirming the cache efficiency, use the value of this metric instead of that of cache_misses |
| File cache utilization | FileCacheUsage | - | Number of entries file cache |
| Scanner | ActiveScanners | - | Number of currently active scanners |
| | ExpiredScanners | - | Number of scanners that have expired due to inactivity since service start |
| Block manager blocks | BlockUnderManagement | - | Number of currently managed data blocks |
| | BlockOpenReading | - | Number of data blocks currently opened for read |
| | BlockOpenWriting | - | Number of data blocks currently opened for write |
| Block manager bytes | BytesUnderManagement | bytes | Number of bytes of currently managed data blocks |
| Block manager containers | ContainersUnderManagement | - | Number of log block containers |
| | FullContainersUnderManagement | - | Number of full log block containers |
| Tablet leaders | NumRaftLeaders | - | Number of tablet replicas that are Raft leaders |
| Tablet sessions | OpenClientSessions | - | Number of currently opened tablet copy client sessions on this server |
| | OpemSourceSessions | - | Number of currently opened tablet copy source sessions on this server |

| Tablets | TabletBootstrapping | - | Number of currently bootstrapping tablets |
|---------|---------------------|---|-------------------------------------------|
| | TabletFailed | - | Number of failed tablets |
| | TabletInitialized | - | Number of currently initialized tablets |
| | TabletNotInitialized | - | Number of currently uninitialized tablets |
| | TabletRunning | - | Number of currently running tablets/Number of currently running threads |
| | TabletShutdown | - | Number of currently shut down tablets |
| | TabletStopped | - | Number of currently stopped tablets |
| | TabletStopping | - | Number of currently stopping tablets |
| CPU time | CpuStime | ms | Total system CPU time of process |
| | CpuUtime | ms | Total user CPU time of process |
| Data path | DataDirsFailed | - | Number of data directories whose disks are currently in failed status |
| | DataDirsFull | - | Number of data directories whose disks are currently full |
| Thread | ThreadsRunning | - | Number of currently running threads |
| Context | InvoluntarySwitches | - | Total involuntary context switches |
| | VoluntarySwitches | - | Total voluntary context switches |
| Spinlock | SpinlockContentionTime | µs | Amount of time consumed by contention on internal spinlocks since server start |
| Log information | ErrorMessages | - | Number of ERROR-level log messages emitted by the application |
| | WarningMessages | - | Number of WARNING-level log messages emitted by the application |
| Operations in queue | TotalCount | - | Total number |
| | Min | - | Minimum number of tasks waiting in |

| | | | |
|---|---|---|---|
| | | | the queue |
| | Max | - | Maximum number of tasks waiting in the queue |
| | Mean | - | Average number of tasks waiting in the queue |
| | Percentile_99_9 | - | 99.9th percentile of the number of tasks waiting in the queue |
| Operation execution duration | TotalCount | μs | Total number of operations |
| | Min | μs | Minimum run time |
| | Max | μs | Maximum run time |
| | Mean | μs | Average run time |
| | Percentile_99_9 | μs | 99.9th percentile of the run time |
| Queuing wait time | TotalCount | μs | Total number of operations |
| | Min | μs | Minimum wait time |
| | Max | μs | Maximum wait time |
| | Mean | μs | Average wait time |
| | Percentile_99_9 | μs | 99.9th percentile of the wait time |
| Allocated bytes | AllocatedBytes | bytes | Number of bytes used by applications. This usually does not match the memory usage reported by the operating system because it does not include TCMalloc overhead or memory fragments |
| Hybrid clock error | HybridClockError | μs | Server clock maximum error; returns 2^64-1 when unable to read the base clock |
| Hybrid clock timestamp | HybridClockTimestamp | μs | Hybrid clock timestamp; returns 2^64-1 when unable to read the base clock |
| TCMalloc memory | HeapSize | bytes | Bytes of system memory reserved by TCMalloc |
| | CurrentThreadCacheBytes | bytes | A measure of some of the memory |

| | | | |
|---|---|---|---|
| | | | TCMalloc is using (for small objects) |
| | TotalThreadCacheBytes | bytes | A limit to how much memory TCMalloc dedicates for small objects |
| TCMalloc PageHeap | FreeBytes | bytes | Number of bytes of free mapped pages in the page heap |
| | UnMappedBytes | bytes | Number of bytes of free unmapped pages in the page heap |
| RPC request | ConnectionsAccepted | - | Number of incoming TCP connections made to the RPC server |
| | QueueOverflow | - | Number of RPCs dropped because the service queue was full |
| | TimesOutInQueue | - | Number of RPCs that timed out while waiting in the service queue and thus were not processed |
| RPC FetchData | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC AlterSchema | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC CreateTablet | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | | | |

| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
|---|---|---|---|
| RPC DeleteTablet | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC Quiesce | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC scan | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC ScannerKeepAlive | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC write | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |

| | Percentile_99_9 | μs | 99.9th percentile of the processing time |
|---|---|---|---|
| Write requests rejected due to queue overloading | QueueOverloadRejections | count | Number of write requests rejected due to queue overloading |
| Scan rate | ScannedFromDiskRate | bytes/s | Amount of data scanned per second |
| | ScannerReturnedRate | bytes/s | Amount of data returned per second |
| Scanner bytes | ScannedFromDisk | bytes | Total amount of data scanned from disk |
| | ScannerReturned | bytes | Total amount of returned data |
| Total row operations | RowsInserted | count | Number of rows inserted into the node |
| | RowsDeleted | count | Number of rows deleted from the node |
| | RowsUpserted | count | Number of rows upserted into the node |
| | RowsUpdated | count | Number of rows updated on the node |
| Row operation rate | RowsInsertedRate | count/s | Number of rows inserted into the node per second |
| | RowsDeletedRate | count/s | Number of rows deleted from the node per second |
| | RowsUpsertedRate | count/s | Number of rows upserted into the node per second |
| | RowsUpdatedRate | count/s | Number of rows updated on the node per second |

## Kudu - master

| Title | Metric | Unit | Description |
|---|---|---|---|
| Block cache hit | BlockCacheHit | - | Number of block cache hits. When confirming the cache efficiency, use the value of this metric instead of that of cache_hits |
| | BlockCacheMiss | - | Number of block cache misses. When confirming the cache efficiency, use the value of this metric instead of that of cache_misses |

| Block cache utilization | BlockCacheUsage | bytes | Memory used by block cache |
|---|---|---|---|
| File cache hit | FileCacheHit | - | Number of file descriptor cache hits. When confirming the cache efficiency, use the value of this metric instead of that of cache_hits |
| | FileCacheMiss | - | Number of file descriptor cache misses. When confirming the cache efficiency, use the value of this metric instead of that of cache_misses |
| File cache utilization | FileCacheUsage | - | Number of entries file cache |
| Block manager blocks | BlockUnderManagement | - | Number of currently managed data blocks |
| | BlockOpenReading | - | Number of data blocks currently opened for read |
| | BlockOpenWriting | - | Number of data blocks currently opened for write |
| Block manager bytes | BytesUnderManagement | bytes | Number of bytes of currently managed data blocks |
| Block manager containers | ContainersUnderManagement | - | Number of log block containers |
| | FullContainersUnderManagement | - | Number of full log block containers |
| CPU time | CpuStime | ms | Total system CPU time of process |
| | CpuUtime | ms | Total user CPU time of process |
| Thread | ThreadsRunning | - | Number of currently running threads |
| Data path | DataDirsFailed | - | Number of data directories whose disks are currently in failed status |
| | DataDirsFull | - | Number of data directories whose disks are currently full |
| Allocated bytes | AllocatedBytes | bytes | Number of bytes used by applications. This usually does not match the memory usage reported by the operating system |

| | | | because it does not include TCMalloc overhead or memory fragments |
|---|---|---|---|
| Log information | ErrorMessages | - | Number of ERROR-level log messages emitted by the application |
| | WarningMessages | - | Number of WARNING-level log messages emitted by the application |
| Context | InvoluntarySwitches | - | Total involuntary context switches |
| | VoluntarySwitches | - | Total voluntary context switches |
| Operations in queue | TotalCount | - | Total number |
| | Min | - | Minimum number of tasks waiting in the queue |
| | Max | - | Maximum number of tasks waiting in the queue |
| | Mean | - | Average number of tasks waiting in the queue |
| | Percentile_99_9 | - | 99.9th percentile of the number of tasks waiting in the queue |
| Queuing wait time | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum wait time |
| | Max | µs | Maximum wait time |
| | Mean | µs | Average wait time |
| | Percentile_99_9 | µs | 99.9th percentile of the wait time |
| Operation execution duration | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum run time |
| | Max | µs | Maximum run time |
| | Mean | µs | Average run time |
| | Percentile_99_9 | µs | 99.9th percentile of the run time |
| Spinlock | SpinlockContentionTime | µs | Amount of time consumed by contention on internal spinlocks since server start |

| Oversized write requests | OversizedWriteRequests | - | Number of oversized write requests to the system catalog tablet rejected since start |
|---|---|---|---|
| Hybrid clock error | HybridClockError | µs | Server clock maximum error; returns 2^64-1 when unable to read the base clock |
| Hybrid clock timestamp | HybridClockTimestamp | µs | Hybrid clock timestamp; returns 2^64-1 when unable to read the base clock |
| Difference in the number of tablet replicas | ClusterReplicaSkew | - | Difference between the number of replicas on the tablet server hosting the most replicas and the number of replicas on the tablet server hosting the fewest replicas |
| Tablet leaders | NumRaftLeaders | - | Number of tablet replicas that are Raft leaders |
| Tablet sessions | OpemSourceSessions | - | Number of currently opened tablet copy source sessions on this server |
| TCMalloc memory | HeapSize | bytes | Bytes of system memory reserved by TCMalloc |
| | CurrentThreadCacheBytes | bytes | A measure of some of the memory TCMalloc is using (for small objects) |
| | TotalThreadCacheBytes | bytes | A limit to how much memory TCMalloc dedicates for small objects |
| TCMalloc page heap | FreeBytes | bytes | Number of bytes of free mapped pages in the page heap |
| | UnMappedBytes | bytes | Number of bytes of free unmapped pages in the page heap |
| RPC request | ConnectionsAccepted | - | Number of incoming TCP connections made to the RPC server |
| | QueueOverflow | - | Number of RPCs dropped because the service queue was full |
| | TimesOutInQueue | - | Number of RPCs that timed out while waiting in the service queue and thus were not processed |

| RPC RunLeaderElection | TotalCount | µs | Total number of operations |
|---|---|---|---|
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC ConnectToMaster | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC Ping | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC TSHeartbeat | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |
| RPC FetchData | TotalCount | µs | Total number of operations |
| | Min | µs | Minimum processing time |
| | Max | µs | Maximum processing time |
| | Mean | µs | Average processing time |
| | Percentile_99_9 | µs | 99.9th percentile of the processing time |

# Alluxio Monitoring Metrics

Last updated：2023-12-27 14:51:59

## Alluxio - cluster

| Title | Metric | Unit | Description |
|---|---|---|---|
| Total data reads and writes | BytesReadAlluxio | Bytes | Total number of bytes read from Alluxio storage reported by all workers |
| | BytesReadUfsAll | Bytes | Total number of bytes read by all workers from all Alluxio UFSes |
| | BytesWrittenAlluxio | Bytes | Total number of bytes written to Alluxio storage in all workers |
| | BytesWrittenUfsAll | Bytes | Total number of bytes written to all Alluxio UFSes by all workers |
| Data read and write throughput | BytesReadAlluxioThroughput | Bytes | Data read throughput from Alluxio storage by all workers |
| | BytesReadUfsThroughput | Bytes | Read throughput from all Alluxio UFSes by all workers |
| | BytesWrittenAlluxioThroughput | Bytes | Data write throughput to Alluxio storage by all workers |
| | BytesWrittenUfsThroughput | Bytes | Write throughput to all Alluxio UFSes by all workers |
| Worker capacity on tiers | CapacityFree | Bytes | Total number of free bytes on all tiers of all workers |
| | CapacityTotal | Bytes | Total capacity on all tiers of all workers |
| | CapacityUsed | Bytes | Total number of used bytes on all tiers of all workers |
| Total workers | Workers | - | Total number of active workers inside the cluster |

## Alluxio - master

| Title | Metric | Unit | Description |
|---|---|---|---|
| | | | |

| CompleteFile operation | CompleteFileOps | - | Total number of CompleteFile operations |
|---|---|---|---|
| | FilesCompleted | - | Total number of successful CompleteFile operations |
| CreateDirectory operation | CreateDirectoryOps | - | Total number of CreateDirectory operations |
| | DirectoriesCreated | - | Total number of successful CreateDirectory operations |
| CreateFile operation | CreateFileOps | - | Total number of CreateFile operations |
| | FilesCreated | - | Total number of successful CreateFile operations |
| Delete operation | DeletePathOps | - | Total number of Delete operations |
| | PathsDeleted | - | Total number of successful Delete operations |
| FreeFile operation | FreeFileOps | - | Total number of FreeFile operations |
| | FilesFreed | - | Total number of successful FreeFile operations |
| GetFileBlockInfo operation | GetFileBlockInfoOps | - | Total number of GetFileBlockInfo operations |
| | FileBlockInfosGot | - | Total number of successful GetFileBlockInfo operations |
| GetFileInfo operation | GetFileInfoOps | - | Total number of GetFileInfo operations |
| | FileInfosGot | - | Total number of successful GetFileInfo operations |
| GetNewBlock operation | GetNewBlockOps | - | Total number of GetNewBlock operations |
| | NewBlocksGot | - | Total number of successful GetNewBlock operations |
| Mount operation | MountOps | - | Total number of Mount operations |
| | PathsMounted | - | Total number of successful Mount operations |
| Unmount operation | UnmountOps | - | Total number of Unmount operations |
| | PathsUnmounted | - | Total number of successful Unmount operations |
| Rename operation | RenamePathOps | - | Total number of Rename operations |

| | PathsRenamed | - | Total number of successful Rename operations |
|---|---|---|---|
| SetAcl operation | SetAclOps | - | Total number of SetAcl operations |
| SetAttribute operation | SetAttributeOps | - | Total number of SetAttribute operations |
| Total files | FilesPersisted | - | Total number of successfully persisted files |
| | FilesPinned | - | Total number of currently pinned files |
| Total file directories | TotalPaths | - | Total number of files and directories in the Alluxio namespace |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |

| | MemHeapInitM | MB | Size of initial JVM HeapMem |
|---|---|---|---|
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |

## Alluxio - worker

| Title | Metric | Unit | Description |
|---|---|---|---|
| Async cache request | AsyncCacheDuplicateRequests | - | Total number of duplicate async cache requests received by worker |
| | AsyncCacheRequests | - | Total number of async cache requests received by worker |
| Async cache blocks | AsyncCacheFailedBlocks | - | Total number of failed async cache blocks in worker |
| | AsyncCacheRemoteBlocks | - | Total number of blocks that need to be asynchronously cached from remote sources |
| | AsyncCacheSucceededBlocks | - | Total number of blocks that were asynchronously cached successfully in the worker |
| | AsyncCacheUfsBlocks | - | Total number of blocks that need to be asynchronously cached from local sources |
| Blocks | BlocksAccessed | - | Total number of blocks accessed in worker |
| | BlocksCached | - | Total number of blocks used for caching data in worker |
| | BlocksCancelled | - | Total number of temporary blocks canceled in worker |
| | BlocksDeleted | - | Total number of blocks deleted from worker by external requests |
| | BlocksEvicted | - | Total number of blocks evicted from worker |
| | BlocksLost | - | Total number of blocks lost in worker |
| | BlocksPromoted | - | Total number of blocks moved to a new tier in worker |
| Worker capacity on | CapacityFree | Bytes | Total free bytes on all tiers of worker |
| | CapacityTotal | Bytes | Total capacity on all tiers of worker |

| tiers | CapacityUsed | Bytes | Total used bytes on all tiers of worker |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |

# PrestoSQL Monitoring Metrics

Last updated：2023-12-27 14:52:15

**Note:**

Currently, only PrestoSQL v322, PrestoSQL v350, and later support PrestoSQL metrics.

## PrestoSQL - overview

| Title | Metric | Unit | Description |
| --- | --- | --- | --- |
| Nodes | Active | - | Number of active nodes |
| | Total | - | Total number of nodes |
| | Failed | - | Number of failed nodes |
| Query | RunningQueries | - | Total number of running queries |
| | QueuedQueries | - | Total number of waiting queries |
| Query frequency | FailedQueries | count/min | Total number of failed queries |
| | AbandonedQueries | count/min | Total number of abandoned queries |
| | CanceledQueries | count/min | Total number of canceled queries |
| | CompletedQueries | count/min | Total number of completed queries |
| | StartedQueries | count/min | Total number of started queries |
| Data volume input/output per minute | InputDataSizeOneMinute | GB/min | Data input rate |
| | OutputDataSizeOneMinute | GB/min | Data output rate |

## PrestoSQL - worker

| Title | Metric | Unit | Description |
| --- | --- | --- | --- |
| | | | |

| GC count | YGC | - | Young GC count |
|---|---|---|---|
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Data input/output rate | InputDataSize.OneMinute.Rate | GB/min | Data input rate |
| | OutputDataSize.OneMinute.Rate | GB/min | Data output rate |

| Processes | PeakThreadCount | - | Peak number of threads |
|---|---|---|---|
| | ThreadCount | - | Total number of threads |
| | DaemonThreadCount | - | Number of daemon threads |
| Process execution duration | Uptime | s | Process execution duration |
| Process start time | StartTime | s | Process start time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

## PrestoSQL - coordinator

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |

| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| Processes | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Number of threads |
| | DaemonThreadCount | - | Number of backend threads |
| Process execution duration | Uptime | s | Process execution duration |
| Process start time | StartTime | s | Process start time |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

# Impala Monitoring Metrics

Last updated：2023-12-27 14:52:35

**Note:**

Currently, only Impala v3.4.0 and later support Impala metrics.

## Impala - catalog

| Title | Metric | Unit | Description |
|---|---|---|---|
| Resident set size | RSS | bytes | Resident set size |
| JVM memory | MemHeapInitM | MB | Peak size of initial JVM HeapMemory |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| Interval between heartbeats from the daemon process to StateStore | Last | s | Latest interval between heartbeats from the daemon process to StateStore |
| | Max | s | Maximum interval between heartbeats from the daemon process to StateStore |
| | Mean | s | Average interval between |

| | | | heartbeats from the daemon process to StateStore |
|---|---|---|---|
| | Min | s | Minimum interval between heartbeats from the daemon process to StateStore |
| | Stddev | s | Standard deviation in the interval between heartbeats from the daemon process to StateStore |
| TCMalloc memory | Used | bytes | Number of bytes used by the program |
| | PageheapFreeBytes | bytes | Number of bytes of free mapped pages in the page heap |
| | PageheapUnmappedBytes | bytes | Number of bytes of free unmapped pages in the page heap |
| | PhysicalBytesReserved | bytes | Amount of physical memory used by the process |
| | TotalBytesReserved | bytes | Number of bytes of system memory reserved by TCMalloc |
| Active connections | Thrift_Server_Connections_Used | - | Number of active connections |
| Process execution duration | Uptime | s | Process execution duration |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Threads | ThreadCount | - | Total number of threads |

| | DaemonThreadCount | - | Number of daemon threads |
|---|---|---|---|
| CPU utilization | SystemCpuLoad | - | System CPU Utilization |

## Impala - StateStore

| Title | Metric | Unit | Description |
|---|---|---|---|
| Resident set size | RSS | bytes | Resident set size |
| TCMalloc memory | Used | bytes | Number of bytes used by the program |
| | PageheapFreeBytes | bytes | Number of bytes of free mapped pages in the page heap |
| | PageheapUnmappedBytes | bytes | Number of bytes of free unmapped pages in the page heap |
| | PhysicalBytesReserved | bytes | Amount of physical memory used by the process |
| | TotalBytesReserved | bytes | Number of bytes of system memory reserved by TCMalloc |
| Connections | Used | - | Number of active connections |
| Running threads | Count | - | Number of running threads |
| StateStore subscribers | Count | - | Number of StateStore subscribers |

## Impala - daemon

| Title | Metric | Unit | Description |
|---|---|---|---|
| JVM memory | MemHeapInitM | MB | Peak size of initial JVM HeapMemory |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory |

| | | | |
|---|---|---|---|
| | | | currently used by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| TCMalloc memory | Used | bytes | Number of bytes used by the program |
| | PageheapFreeBytes | bytes | Number of bytes of free mapped pages in the page heap |
| | PageheapUnmappedBytes | bytes | Number of bytes of free unmapped pages in the page heap |
| | PhysicalBytesReserved | bytes | Amount of physical memory used by the process |
| | TotalBytesReserved | bytes | Number of bytes of system memory reserved by TCMalloc |
| Threads | ThreadCount | - | Total number of threads |
| | DaemonThreadCount | - | Number of daemon threads |
| Process execution duration | Uptime | s | Process execution duration |
| File descriptors | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |

| CPU utilization | SystemCpuLoad | - | System CPU Utilization |
|---|---|---|---|
| Beeswax API client connections | Use | - | Number of active Beeswax API connections |
| | Conn_In_Use | - | Number of active Beeswax API connections to this Impala daemon |
| | TotalConns | - | Total number of active Beeswax API connections to this Impala daemon |
| | ConnSetupQueueSize | - | Number of Beeswax API connections that this Impala daemon has received and are waiting to establish |
| HS2 API client connections | Use | - | Number of active HS2 API connections |
| | Conn_In_Use | - | Number of active HS2 API connections |
| | TotalConns | - | Total number of HS2 API connections this Impala daemon has established during the lifecycle |
| | ConnSetupQueueSize | - | Number of HS2 API connections that this Impala daemon has received and are waiting to establish |
| Thread manager | RunningThreads | - | Number of running threads |
| | TotalCreatedThreads | - | Number of threads created during the lifecycle |

| Memory manager limit | Limit | Bytes | Memory beyond the memory limit (default value: -1) |
|---|---|---|---|
| Amount of memory beyond the memory limit (default value: -1) | OverLimit | Bytes | Number of threads created during the lifecycle |
| HS2 API client wait time for connection establishment | P20 | us | HS2 API client wait time for connection establishment |
| | P50 | | |
| | P70 | | |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| Beeswax API client wait time for service thread establishment | P20 | us | Beeswax API client wait time for service thread establishment |
| | P50 | | |
| | P70 | | |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| Timed-out Beeswax API connections | TimeOutCnncRequests | - | Number of timed-out Beeswax API connections |
| Time spent parsing requests from request pool (milliseconds) | Total | ms | Time spent parsing requests from request pool (milliseconds) |
| Cache misses in external data source cache class | Misses | - | Number of cache misses in external data source cache class |
| Impala backend server connection requests that timed out waiting for setup | ConnSetupQueueSize | - | Number of Impala backend server connection requests |

| | | | that timed out waiting for setup |
|---|---|---|---|
| Impala backend connection requests that timed out waiting for setup | TimeOutCnncRequests | - | Number of Impala backend connection requests that timed out waiting for setup |
| Total Impala backend client connections to this Impala daemon | TotalConnections | - | Total number of Impala backend client connections to this Impala daemon |
| Time spent by Impala backend client waiting for connection establishment | P20 | us | Time spent by Impala backend client waiting for connection establishment |
| | P50 | | |
| | P70 | | |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| | Count | | |
| | Sum | | |
| Time spent by Impala backend client waiting for service thread | P20 | us | Time spent by Impala backend client waiting for service thread |
| | P50 | | |
| | P70 | | |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| | Count | | |
| | Sum | | |
| HS2 API client wait time for service thread establishment | P20 | us | HS2 API client wait time for service thread establishment |
| | P50 | | |

| | P70 | | |
| --- | --- | --- | --- |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| | Count | | |
| | Sum | | |
| HS2 HTTP API client wait time for service thread | P20 | us | HS2 HTTP API client wait time for service thread |
| | P50 | | |
| | P70 | | |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| | Count | | |
| | Sum | | |
| DataStreamService: Rejected service queue overflows | RpcsQueueOverflow | - | DataStreamService: Number of rejected service queue overflows |
| ControlStreamService: Rejected service queue overflows | RpcsQueueOverflow | - | ControlStreamService: Number of rejected service queue overflows |
| DataStreamService: Used bytes | PeakUsageBytes | Bytes | Peak number of bytes used by Memtracker DataStreamService |
| | CurrentUsageBytes | Bytes | Current number of bytes used by Memtracker DataStreamService |
| ControlService: Used bytes | PeakUsageBytes | Bytes | Peak number of bytes used by Memtracker |

| | | | ControlService |
|---|---|---|---|
| | CurrentUsageBytes | Bytes | Current number of bytes used by Memtracker ControlService |
| Resident set size (RSS) for this process | RSS | Bytes | Resident set size (RSS) for this process |
| Total backends registered in StateStore | Total | - | Total number of backends registered in StateStore |
| Query release delay | P20 | us | Query release delay |
| | P50 | | |
| | P70 | | |
| | P90 | | |
| | P95 | | |
| | P99.9 | | |
| | Count | | |
| | Sum | | |
| Opened HDFS filed | NumFilesOpenForInsert | - | Number of opened HDFS files |
| Scan range read during the lifecycle of the process | ScanRangesTotal | - | Scan range read during the lifecycle of the process |
| Opened Beeswax sessions | NumOpenBeeswaxSessions | - | Number of opened Beeswax sessions |
| Total query fragments processed during the lifecycle of the process | NumFragments | - | Total number of query fragments processed during the lifecycle of the process |
| Total scan ranges read during the lifecycle of the process without volume metadata | ScanRangesNumMissingVolumId | - | Total number of scan ranges read during the lifecycle of the process |

| | | | without volume metadata |
|---|---|---|---|
| Hedged read attempts | HedgedReadOps | - | Number of hedged read attempts |
| Total queries processed during the lifecycle of the process | NumQueries | - | Total number of queries processed during the lifecycle of the process |
| Total rows supporting caching HS2 FETCH_FIRST | ResultSetCacheTotalNumRows | - | Total number of rows supporting caching HS2 FETCH_FIRST |
| Total queries registered on this Impala server | NumQueriesRegistered | - | Total number of queries registered on this Impala server |
| Total backend queries | NumQueriesExecuted | - | Total number of backend queries |
| Sessions terminated due to inactivity | NumSessionsExpired | - | Number of sessions terminated due to inactivity |
| Queries terminated due to inactivity | NumQueriesExpired | - | Number of queries terminated due to inactivity |
| Opened HS2 sessions | NumOpenHS2Sessions | - | Number of opened HS2 sessions |
| Tables in catalog | NumTables | - | Number of tables in catalog |
| Databases in catalog | NumDatabases | - | Number of databases in catalog |
| Bytes written to the disk by the I/O manager | BytesWritten | - | Number of bytes written to the disk by the I/O manager |
| Files opened by the I/O manager | NumOpenFiles | - | Number of files opened by the I/O manager |
| Used HDFS file handles | NumFileHandlesOutstanding | Bytes | Number of used |

| | | | HDFS file handles |
|---|---|---|---|
| Read local bytes | LocalBytesRead | Bytes | Number of local bytes read by the I/O manager |

# Ranger Monitoring Metrics

Last updated：2023-12-27 14:52:55

**Ranger - admin**

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |

| CPU utilization | ProcessCpuLoad | % | CPU utilization |
|---|---|---|---|
| File handles | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |
| Threads | ThreadCount | - | Number of threads |
| | PeakThreadCount | - | Peak number of threads |
| | DaemonThreadCount | - | Number of backend threads |

## Ranger - UserSync

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |

| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
|---|---|---|---|
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| File handles | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |
| Threads | ThreadCount | - | Number of threads |
| | PeakThreadCount | - | Peak number of threads |
| | DaemonThreadCount | - | Number of backend threads |

# COSRanger Monitoring Metrics

Last updated：2023-12-27 14:53:49

**Note:**

 Metrics such as verification failure statistics, authentication failure statistics, and authentication success statistics currently don't have specific data, which will be available in the future.

## COSRanger - COSRangerServer

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |

| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
|---|---|---|---|
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| File handles | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | ThreadCount | - | Number of threads |
| | PeakThreadCount | - | Peak number of threads |
| | DaemonThreadCount | - | Number of backend threads |
| - | Leader | - | Whether it is the COSRanger master node |
| Check statistics | PermissionAllowCnt | count | Total number of permission allows |
| | AuthDenyCnt | count | Total number of authentication failures |
| | PermissionDenyCnt | count | Total number of permission denies |
| Authentication success statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |

| | Total_1m | count | Total number of requests per minute |
|---|---|---|---|
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| Authentication failure statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| Permission deny statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| Permission allow statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests |

| | | | per minute |
|---|---|---|---|
| | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| accessStat_DELETE operation statistics | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| accessStat_LIST operation statistics | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| accessStat_READ operation statistics | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| accessStat_WRITE operation statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |

| | Total_1m | count | Total number of requests per minute |
|---|---|---|---|
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| rpc_getRangerAuthPolicy call count statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| rpc_checkPermission call count statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| rpc_getDelegationToken call count statistics | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests |

| | | | per minute |
|---|---|---|---|
| | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| rpc_renewDelegationToken call count statistics | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| rpc_cancelDelegationToken call count statistics | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| | Qps | count | Number of queries per second |
| | Total_5m | count | Total number of requests per five minutes |
| rpc_getSTS call count statistics | Total_1m | count | Total number of requests per minute |
| | Qps_5m | count | Average number of requests per five minutes |
| | Qps_1m | count | Average number of requests per minute |
| cosRpc_getSTS call duration | Cost_Avg | μs | Average duration in the current second |
| | Cost_Avg_1m | μs | Average duration per minute |

| | Cost_Avg_5m | μs | Average duration per five minutes |
|---|---|---|---|
| | Cost_Max | μs | Maximum duration in the current second |
| | Cost_Max_1m | μs | Maximum duration per minute |
| | Cost_Max_5m | μs | Maximum duration per five minutes |
| | Cost_Min | μs | Minimum duration in the current second |
| | Cost_Min_1m | μs | Minimum duration per minute |
| | Cost_Min_5m | μs | Minimum duration per five minutes |
| cosRpc_renewDelegationToken call duration | Cost_Avg | μs | Average duration in the current second |
| | Cost_Avg_1m | μs | Average duration per minute |
| | Cost_Avg_5m | μs | Average duration per five minutes |
| | Cost_Max | μs | Maximum duration in the current second |
| | Cost_Max_1m | μs | Maximum duration per minute |
| | Cost_Max_5m | μs | Maximum duration per five minutes |
| | Cost_Min | μs | Minimum duration in the current second |
| | Cost_Min_1m | μs | Minimum duration per minute |
| | Cost_Min_5m | μs | Minimum duration per five minutes |
| cosRpc_cancelDelegationToken call duration | Cost_Avg | μs | Average duration in the current second |
| | Cost_Avg_1m | μs | Average duration per minute |
| | Cost_Avg_5m | μs | Average duration per five |

| | | | |
|---|---|---|---|
| | | | minutes |
| | Cost_Max | μs | Maximum duration in the current second |
| | Cost_Max_1m | μs | Maximum duration per minute |
| | Cost_Max_5m | μs | Maximum duration per five minutes |
| | Cost_Min | μs | Minimum duration in the current second |
| | Cost_Min_1m | μs | Minimum duration per minute |
| | Cost_Min_5m | μs | Minimum duration per five minutes |
| cosRpc_getDelegationToken call duration | Cost_Avg | μs | Average duration in the current second |
| | Cost_Avg_1m | μs | Average duration per minute |
| | Cost_Avg_5m | μs | Average duration per five minutes |
| | Cost_Max | μs | Maximum duration in the current second |
| | Cost_Max_1m | μs | Maximum duration per minute |
| | Cost_Max_5m | μs | Maximum duration per five minutes |
| | Cost_Min | μs | Minimum duration in the current second |
| | Cost_Min_1m | μs | Minimum duration per minute |
| | Cost_Min_5m | μs | Minimum duration per five minutes |
| cosRpc_checkPermission call duration | Cost_Avg | μs | Average duration in the current second |
| | Cost_Avg_1m | μs | Average duration per minute |
| | Cost_Avg_5m | μs | Average duration per five minutes |

| | Cost_Max | µs | Maximum duration in the current second |
| --- | --- | --- | --- |
| | Cost_Max_1m | µs | Maximum duration per minute |
| | Cost_Max_5m | µs | Maximum duration per five minutes |
| | Cost_Min | µs | Minimum duration in the current second |
| | Cost_Min_1m | µs | Minimum duration per minute |
| | Cost_Min_5m | µs | Minimum duration per five minutes |
| cosRpc_getRangerAuthPolicy call duration | Cost_Avg | µs | Average duration in the current second |
| | Cost_Avg_1m | µs | Average duration per minute |
| | Cost_Avg_5m | µs | Average duration per five minutes |
| | Cost_Max | µs | Maximum duration in the current second |
| | Cost_Max_1m | µs | Maximum duration per minute |
| | Cost_Max_5m | µs | Maximum duration per five minutes |
| | Cost_Min | µs | Minimum duration in the current second |
| | Cost_Min_1m | µs | Minimum duration per minute |
| | Cost_Min_5m | µs | Minimum duration per five minutes |

# Doris Monitoring Metrics

Last updated：2023-12-27 14:54:08

## Doris - FE

| Title | Metric | Unit | Description |
|---|---|---|---|
| Node information | FeNodeNum | count | Total number of FE nodes |
| | BeAliveNum | count | Number of alive BE nodes |
| | BkDeadNum | count | Number of dead broker nodes |
| Connections | Num | count | Number of FE node JVM connections |
| JVM threads | Total | count | Total number of threads in the JVM of the FE node, including daemon threads and non-daemon threads |
| | Peak | count | Peak number of threads in the JVM of the FE node |
| GC count | YoungGC | count | FE node JVM Young GC count |
| | OldGC | count | FE node JVM Old GC count |
| GC time | YoungGC | s | FE node JVM Young GC time |
| | OldGC | s | FE node JVM Old GC time |
| FE query latency | Quantile75 | ms | 75th percentile of the FE query latency |
| | Quantile95 | ms | 95th percentile of the FE query latency |
| | Quantile99 | ms | 99th percentile of the FE query latency |
| | Quantile999 | ms | 99.9th percentile of the FE query latency |
| Maximum compaction score of tablet | MAX | score | Maximum compaction score of FE tablet |
| Scheduled tablets | ScheduledTablet | count | Number of scheduled tablets in FE |
| Request response | QPS | count | Number of queries per second |
| | RPS | count | Number of requests that can be processed per |

| | | | second |
|---|---|---|---|
| Query failure rate | ErrRate | % | Query error rate |
| Cache query | SqlModelHitQuery | count | Number of queries that hit the SQL model |
| | PartitionModelHitQuery | count | Number of queries that hit the partition model |
| | SqlModelQuery | count | Number of queries in the SQL cache model |
| | PartitionModelQuery | count | Number of queries in the partition cache model |
| | CachePartitionHit | count | Number of partitions that hit the cache in the query |
| | CachePartitionScan | count | Number of all partitions scanned in the query |
| Routine load rows | TotalRows | count | Number of FE routine load rows |
| | ErrorRows | count | Number of FE routine load error rows |
| Transaction status statistics | Reject | count | Number of rejected transactions on FE |
| | Begin | count | Number of transactions that have begun on FE |
| | Success | count | Number of successful transactions on FE |
| | Failed | count | Number of failed transactions on FE |
| Images | Write | count | Number of image writes on FE |
| | Push | count | Number of image pushes on FE |
| ALTER job statistics | RollupRunning | count | Number of running alter jobs of ROLLUP type |
| | SchemaChangeRunning | count | Number of running alter jobs of SCHEMA_CHANGE type |
| UNKNOWN load job statistics | UNKNOWN | count | Number of load jobs of UNKNOWN type in UNKNOWN status |
| | PENDING | count | Number of load jobs of UNKNOWN type in PENDING status |
| | ETL | count | Number of load jobs of UNKNOWN type in ETL status |
| | LOADING | count | Number of load jobs of UNKNOWN type in |

| | | | |
|---|---|---|---|
| | | | LOADING status |
| | COMMITTED | count | Number of load jobs of UNKNOWN type in COMMITTED status |
| | FINISHED | count | Number of load jobs of UNKNOWN type in FINISHED status |
| | CANCELLED | count | Number of load jobs of UNKNOWN type in CANCELLED status |
| SPARK load job statistics | UNKNOWN | count | Number of load jobs of SPARK type in UNKNOWN status |
| | PENDING | count | Number of load jobs of SPARK type in PENDING status |
| | ETL | count | Number of load jobs of SPARK type in ETL status |
| | LOADING | count | Number of load jobs of SPARK type in LOADING status |
| | COMMITTED | count | Number of load jobs of SPARK type in COMMITTED status |
| | FINISHED | count | Number of load jobs of SPARK type in FINISHED status |
| | CANCELLED | count | Number of load jobs of SPARK type in CANCELLED status |
| DELETE load job statistics | UNKNOWN | count | Number of load jobs of DELETE type in UNKNOWN status |
| | PENDING | count | Number of load jobs of DELETE type in PENDING status |
| | ETL | count | Number of load jobs of DELETE type in ETL status |
| | LOADING | count | Number of load jobs of DELETE type in LOADING status |
| | COMMITTED | count | Number of load jobs of DELETE type in COMMITTED status |
| | FINISHED | count | Number of load jobs of DELETE type in |

| | | | FINISHED status |
|---|---|---|---|
| | CANCELLED | count | Number of load jobs of DELETE type in CANCELLED status |
| INSERT load job statistics | UNKNOWN | count | Number of load jobs of INSERT type in UNKNOWN status |
| | PENDING | count | Number of load jobs of INSERT type in PENDING status |
| | ETL | count | Number of load jobs of INSERT type in ETL status |
| | LOADING | count | Number of load jobs of INSERT type in LOADING status |
| | COMMITTED | count | Number of load jobs of INSERT type in COMMITTED status |
| | FINISHED | count | Number of load jobs of INSERT type in FINISHED status |
| | CANCELLED | count | Number of load jobs of INSERT type in CANCELLED status |
| BROKER load job statistics | UNKNOWN | count | Number of load jobs of BROKER type in UNKNOWN status |
| | PENDING | count | Number of load jobs of BROKER type in PENDING status |
| | ETL | count | Number of load jobs of BROKER type in ETL status |
| | LOADING | count | Number of load jobs of BROKER type in LOADING status |
| | COMMITTED | count | Number of load jobs of BROKER type in COMMITTED status |
| | FINISHED | count | Number of load jobs of BROKER type in FINISHED status |
| | CANCELLED | count | Number of load jobs of BROKER type in CANCELLED status |
| MINI load job | UNKNOWN | count | Number of load jobs of MINI type in |

| statistics | | | UNKNOWN status |
| --- | --- | --- | --- |
| | PENDING | count | Number of load jobs of MINI type in PENDING status |
| | ETL | count | Number of load jobs of MINI type in ETL status |
| | LOADING | count | Number of load jobs of MINI type in LOADING status |
| | COMMITTED | count | Number of load jobs of MINI type in COMMITTED status |
| | FINISHED | count | Number of load jobs of MINI type in FINISHED status |
| | CANCELLED | count | Number of load jobs of MINI type in CANCELLED status |
| HADOOP load job statistics | UNKNOWN | count | Number of load jobs of HADOOP type in UNKNOWN status |
| | PENDING | count | Number of load jobs of HADOOP type in PENDING status |
| | ETL | count | Number of load jobs of HADOOP type in ETL status |
| | LOADING | count | Number of load jobs of HADOOP type in LOADING status |
| | COMMITTED | count | Number of load jobs of HADOOP type in COMMITTED status |
| | FINISHED | count | Number of load jobs of HADOOP type in FINISHED status |
| | CANCELLED | count | Number of load jobs of HADOOP type in CANCELLED status |

## Doris - BE

| Title | Metric | Unit | Description |
| --- | --- | --- | --- |
| Thrift utilization | Broker | count | Number of thrifts used by broker |
| | Backend | count | Number of thrifts used by BE |

| | Extdatasource | count | Number of thrifts used by extdatasource |
|---|---|---|---|
| | Frontend | count | Number of thrifts used by FE |
| Streaming load statistics | RequestsTotal | count | Number of streaming load requests |
| | CurrentProcessing | count | Number of existing streaming load processes |
| | PipeCount | count | Number of streaming load pipes |
| Streaming load time | Duration | ms | Streaming load duration |
| Streaming load data volume | LoadTotal | bytes | Size of data imported by streaming load |
| Fragment statistics | PlanFragment | count | Number of plan fragments |
| | Endpoint | count | Number of DataStreams |
| | RequestsTotal | count | Number of fragment requests |
| Fragment request duration | Duration | µs | Fragment request duration |
| BE memory | Total | bytes | Size of BE memory pool |
| | Allocated | bytes | Size of allocated BE memory |
| Maximum compaction score of tablet | CumulativeMax | score | Maximum cumulative compaction score of tablet |
| | BaseMax | score | Maximum base compaction score of tablet |
| Data handled by compaction | Cumulative | bytes | Amount of data handled by cumulative compaction |
| | Base | bytes | Amount of data handled by base compaction |
| Delta data handled by compaction | Cumulative | bytes | Amount of delta data handled by cumulative compaction |
| | Base | bytes | Amount of delta data handled by base compaction |
| MemPools used by compaction | CurrentConsumption | count | Sum of MemPools used by compaction (all compaction threads) |
| | | | |

| Process file handles | Used | count | Number of file handles used by BE process |
| --- | --- | --- | --- |
| | SoftLimit | count | Soft limit on file handles for BE process |
| | HardLimit | count | Hard limit on file handles for BE process |
| Running threads in process | NUM | count | Number of threads running in BE process |
| Engine request statistics | FailedBaseCompaction | count | Number of failed engine requests of base_compaction type |
| | FailedCultCompt | count | Number of failed engine requests of cumulative_compaction type |
| | TotalBaseCompaction | count | Total number of engine requests of base_compaction type |
| | TotalCultCompt | count | Total number of engine requests of cumulative_compaction type |

## Doris - BK

| Title | Metric | Unit | Description |
| --- | --- | --- | --- |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | | | |

| | M | % | Percentage of used Metaspace memory |
|---|---|---|---|
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMem |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMem |
| CPU utilization | ProcessCpuLoad | % | CPU utilization |
| File handles | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| | OpenFileDescriptorCount | - | Number of opened file descriptors |
| CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | ThreadCount | - | Number of threads |
| | PeakThreadCount | - | Peak number of threads |
| | DaemonThreadCount | - | Number of backend threads |

# Kylin Monitoring Metrics

Last updated：2023-12-27 14:54:25

**Kylin - Kylin**

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |

| File descriptors | OpenFileDescriptorCount | - | Number of opened file descriptors |
|---|---|---|---|
| | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| | SystemCpuLoad | % | System CPU Utilization |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | ThreadCount | - | Total number of threads |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |

# Zeppelin Monitoring Metrics

Last updated：2023-12-27 14:54:44

## Zeppelin - Zeppelin

| Title | Metric | Unit | Description |
|-------|--------|------|-------------|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |

# Oozie Monitoring Metrics

Last updated：2023-12-27 14:55:01

**Oozie - Oozie**

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |

# Storm Monitoring Metrics

Last updated：2023-12-27 14:55:18

**Storm - Nimbus**

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |

| File descriptors | OpenFileDescriptorCount | - | Number of opened file descriptors |
|---|---|---|---|
| | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Total number of threads |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |

## Storm - supervisor

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |

| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
|---|---|---|---|
| | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |
| File descriptors | OpenFileDescriptorCount | - | Number of opened file descriptors |
| | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Total number of threads |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |

# Livy Monitoring Metrics

Last updated：2023-12-27 14:55:36

## Livy - LivyServer

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |

| File descriptors | OpenFileDescriptorCount | - | Number of opened file descriptors |
|---|---|---|---|
| | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Total number of threads |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |

# Kyuubi Monitoring Metrics

Last updated：2023-12-27 14:55:56

## Kyuubi - KyuubiServer

| Title | Metric | Unit | Description |
|---|---|---|---|
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | FGCT | s | Full GC time |
| | GCT | s | Garbage collection time |
| | YGCT | s | Young GC time |
| Memory zone proportion | S0 | % | Percentage of used Survivor 0 memory |
| | E | % | Percentage of used Eden memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| JVM memory | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory committed by JVM |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory committed by JVM |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |

| File descriptors | OpenFileDescriptorCount | - | Number of opened file descriptors |
|---|---|---|---|
| | MaxFileDescriptorCount | - | Maximum number of file descriptors |
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| Worker threads | DaemonThreadCount | - | Number of daemon threads |
| | PeakThreadCount | - | Peak number of threads |
| | ThreadCount | - | Total number of threads |
| Cumulative CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| Process execution duration | Uptime | s | Process execution duration |

# StarRocks Monitoring Metrics

Last updated：2023-12-27 15:00:50

## StarRocks - BE

| Title | Metric | Unit | Description |
|---|---|---|---|
| Delta data handled by compaction | Cumulative | rowsets | Amount of delta data handled by cumulative compaction |
| | Base | rowsets | Amount of delta data handled by base compaction |
| | Update | rowsets | Amount of delta data handled by update compaction |
| Data handled by compaction | Cumulative | bytes | Amount of data handled by cumulative compaction |
| | Base | bytes | Amount of data handled by base compaction |
| | Update | bytes | Amount of data handled by update compaction |
| Maximum compaction score of tablet | CumulativeMax | score | Maximum cumulative compaction score of tablet |
| | BaseMax | score | Maximum base compaction score of tablet |
| Engine request failure statistics (1) | base_compaction | count | Number of failed engine requests of base_compaction type |
| | clone | count | Number of failed engine requests of clone type |
| | create_rollup | count | Number of failed engine requests of create_rollup type |
| | create_tablet | count | Number of failed engine requests of create_tablet type |
| | cumulative_compaction | count | Number of failed engine requests of |

| | | | cumulative_compaction type |
|---|---|---|---|
| | delete | count | Number of failed engine requests of delete type |
| | finish_task | count | Number of failed engine requests of finish_task type |
| Engine request failure statistics (2) | publish | count | Number of failed engine requests of publish type |
| | report_all_tablets | count | Number of failed engine requests of report_all_tablets type |
| | report_disk | count | Number of failed engine requests of report_disk type |
| | report_tablet | count | Number of failed engine requests of report_tablet type |
| | report_task | count | Number of failed engine requests of report_task type |
| | schema_change | count | Number of failed engine requests of schema_change type |
| Engine request statistics (1) | base_compaction | count | Number of failed engine requests of base_compaction type |
| | clone | count | Number of failed engine requests of clone type |
| | create_rollup | count | Number of failed engine requests of create_rollup type |
| | create_tablet | count | Number of failed engine requests of create_tablet type |
| | cumulative_compaction | count | Number of failed engine requests of cumulative_compaction type |
| | delete | count | Number of failed engine requests of delete type |
| | drop_tablet | count | Number of failed engine requests of drop_tablet type |
| | finish_task | count | Number of failed engine requests of |

| | | | finish_task type |
|---|---|---|---|
| Engine request statistics (2) | publish | count | Number of failed engine requests of publish type |
| | report_all_tablets | count | Number of failed engine requests of report_all_tablets type |
| | report_disk | count | Number of failed engine requests of report_disk type |
| | report_tablet | count | Number of failed engine requests of report_tablet type |
| | report_task | count | Number of failed engine requests of report_task type |
| | schema_change | count | Number of failed engine requests of schema_change type |
| | storage_migrate | count | Number of failed engine requests of storage_migrate type |
| Fragment statistics | PlanFragment | count | Number of plan fragments |
| | Endpoint | count | Number of DataStreams |
| Fragment request duration | Duration | µs | Fragment request duration |
| Transaction request statistics | begin | count | Number of transaction requests of begin type |
| | commit | count | Number of transaction requests of commit type |
| | exec | count | Number of transaction requests of exec type |
| | rollback | count | Number of transaction requests of rollback type |
| Data imported by streaming load | LoadTotal | bytes | Amount of data imported by streaming load |
| Streaming load statistics | CurrentProcessing | count | Number of existing streaming load processes |
| | PipeCount | count | Number of streaming load pipes |

| Streaming load duration | Duration | ms | Streaming load duration |
|---|---|---|---|
| BE memory | Total | bytes | Size of BE memory pool |
| | Allocated | bytes | Size of allocated BE memory |
| Process file handles | Used | count | Number of file handles used by BE process |
| | SoftLimit | count | Soft limit on file handles for BE process |
| | HardLimit | count | Hard limit on file handles for BE process |
| Running threads in process | Thread | count | Number of threads running in BE process |
| Used thrifts | Broker | count | Number of thrifts used by broker |
| | Backend | count | Number of thrifts used by BE |
| | Frontend | count | Number of thrifts used by FE |
| Tablet write statistics | Writer | count | Statistics of tablet writes on BE |

# StarRocks - FE

| Title | Metric | Unit | Description |
|---|---|---|---|
| ALTER job statistics | RollupRunning | count | Number of running alter jobs of rollup type |
| | SchemaChangeRunning | count | Number of running alter jobs of schema_change type |
| Image statistics | Write | count | Number of image writes on FE |
| | Push | count | Number of image pushes on FE |
| Scheduled tablets | ScheduledTablet | count | Number of scheduled tablets on FE |
| Transaction status statistics | Reject | count | Number of rejected transactions on FE |
| | Begin | count | Number of started transactions on FE |
| | Success | count | Number of successful transactions on FE |

| | Failed | count | Number of failed transactions on FE |
|---|---|---|---|
| Heap memory in JVM | max | bytes | Maximum amount of heap memory |
| | committed | bytes | Amount of committed heap memory |
| | used | bytes | Amount of used heap memory |
| Non-heap memory in JVM | committed | bytes | Amount of committed non-heap memory |
| | used | bytes | Amount of used non-heap memory |
| Old memory in JVM | used | bytes | Amount of used old memory |
| | peak_used | bytes | Maximum amount of used old memory |
| | max | bytes | Maximum amount of old memory |
| Young memory in JVM | used | bytes | Amount of used young memory |
| | peak_used | bytes | Maximum amount of used young memory |
| | max | bytes | Maximum amount of young memory |
| Routine load queue size | report queue | count | Size of FE report queue |
| Routine load rows | TotalRows | count | Number of FE routine load rows |
| | ErrorRows | count | Number of FE routine load error rows |
| Routine load size | Receive | bytes | Size of FE routine load |
| Maximum compaction score of tablet | MAX | score | Maximum compaction score of FE tablet |
| EditLog write latency | Quantile75 | ms | 75th percentile of the FE EditLog write latency |
| | Quantile95 | ms | 95th percentile of the FE EditLog write latency |
| | Quantile98 | ms | 98th percentile of the FE EditLog write latency |
| | Quantile99 | ms | 99th percentile of the FE EditLog write latency |
| | | | |

| | Quantile999 | ms | 99.9th percentile of the FE EditLog write latency |
|---|---|---|---|
| GC count | YoungGC | count | FE node JVM Young GC count |
| | OldGC | count | FE node JVM Old GC count |
| GC time | YoungGC | Sec | FE node JVM Young GC time |
| | OldGC | Sec | FE node JVM Old GC time |
| JVM threads | Total | count | Total number of threads in the JVM of the FE node |
| | Peak | count | Peak number of threads in the JVM of the FE node |
| BROKER load job statistics | UNKNOWN | count | Number of load jobs of BROKER type in UNKNOWN status |
| | PENDING | count | Number of load jobs of BROKER type in PENDING status |
| | ETL | count | Number of load jobs of BROKER type in ETL status |
| | LOADING | count | Number of load jobs of BROKER type in LOADING status |
| | COMMITTED | count | Number of load jobs of BROKER type in COMMITTED status |
| | FINISHED | count | Number of load jobs of BROKER type in FINISHED status |
| | CANCELLED | count | Number of load jobs of BROKER type in CANCELLED status |
| DELETE load job statistics | UNKNOWN | count | Number of load jobs of DELETE type in UNKNOWN status |
| | PENDING | count | Number of load jobs of DELETE type in PENDING status |
| | ETL | count | Number of load jobs of DELETE type in ETL status |
| | LOADING | count | Number of load jobs of DELETE type in LOADING status |

| | COMMITTED | count | Number of load jobs of DELETE type in COMMITTED status |
|---|---|---|---|
| | FINISHED | count | Number of load jobs of DELETE type in FINISHED status |
| | CANCELLED | count | Number of load jobs of DELETE type in CANCELLED status |
| HADOOP load job statistics | UNKNOWN | count | Number of load jobs of HADOOP type in UNKNOWN status |
| | PENDING | count | Number of load jobs of HADOOP type in PENDING status |
| | ETL | count | Number of load jobs of HADOOP type in ETL status |
| | LOADING | count | Number of load jobs of HADOOP type in LOADING status |
| | COMMITTED | count | Number of load jobs of HADOOP type in COMMITTED status |
| | FINISHED | count | Number of load jobs of HADOOP type in FINISHED status |
| | CANCELLED | count | Number of load jobs of HADOOP type in CANCELLED status |
| INSERT load job statistics | UNKNOWN | count | Number of load jobs of INSERT type in UNKNOWN status |
| | PENDING | count | Number of load jobs of INSERT type in PENDING status |
| | ETL | count | Number of load jobs of INSERT type in ETL status |
| | LOADING | count | Number of load jobs of INSERT type in LOADING status |
| | COMMITTED | count | Number of load jobs of INSERT type in COMMITTED status |
| | FINISHED | count | Number of load jobs of INSERT type in FINISHED status |

| | CANCELLED | count | Number of load jobs of INSERT type in CANCELLED status |
|---|---|---|---|
| Routine load job statistics | NEED_SCHEDULE | count | Number of routine load jobs in NEED_SCHEDULE status |
| | RUNNING | count | Number of routine load jobs in RUNNING status |
| | PAUSED | count | Number of routine load jobs in PAUSED status |
| | STOPPED | count | Number of routine load jobs in STOPPED status |
| | CANCELLED | count | Number of routine load jobs in CANCELLED status |
| SPARK load job statistics | UNKNOWN | count | Number of load jobs of SPARK type in UNKNOWN status |
| | PENDING | count | Number of load jobs of SPARK type in PENDING status |
| | ETL | count | Number of load jobs of SPARK type in ETL status |
| | LOADING | count | Number of load jobs of SPARK type in LOADING status |
| | COMMITTED | count | Number of load jobs of SPARK type in COMMITTED status |
| | FINISHED | count | Number of load jobs of SPARK type in FINISHED status |
| | CANCELLED | count | Number of load jobs of SPARK type in CANCELLED status |
| FE MASTER | FE Master | count | Whether it is the FE master. Valid values: 1: Master; 0: Follower. |
| Node information | FeNodeNum | count | Total number of FE nodes |
| | BeNodeNum | count | Total number of BE nodes |
| | BeAliveNum | count | Number of live BE nodes |
| | BeDecommissionedNum | count | Number of live BE nodes |

| | BkDeadNum | count | Number of dead broker nodes |
|---|---|---|---|
| Request response | QPS | count/s | Number of queries per second |
| | RPS | count/s | Number of requests that can be processed per second |
| FE query statistics | total | count | Total number of FE queries |
| | err | count | Total number of FE query errors |
| | timeout | count | Number of timed-out FE queries |
| | success | count | Total number of successful FE queries |
| | slow | count | Total number of slow FE queries |
| Query failure rate | ErrRate | % | Query error rate |
| FE query latency | Quantile75 | ms | 75th percentile of the FE query latency |
| | Quantile95 | ms | 95th percentile of the FE query latency |
| | Quantile99 | ms | 99th percentile of the FE query latency |
| | Quantile999 | ms | 99.9th percentile of the FE query latency |
| Connections | Num | count | Number of FE node connections |

# StarRocks - Broker

| Title | Metric | Unit | Description |
|---|---|---|---|
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | GCT | s | Garbage collection time |
| | FGCT | s | Full GC time |
| | | | |

| | YGCT | s | Young GC time |
|---|---|---|---|
| Memory zone proportion | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S0 | % | Percentage of used Survivor 0 memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | E | % | Percentage of used Eden memory |
| JVM memory | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |
| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| File handles | OpenFileDescriptorCount | count | Number of opened file descriptors |
| | MaxFileDescriptorCount | count | Maximum number of file descriptors |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | PeakThreadCount | count | Peak number of threads |
| | ThreadCount | count | Total number of threads |
| | DaemonThreadCount | count | Number of Daemon threads |

# Kafka Monitoring Metrics

Last updated：2023-12-27 15:01:37

| Title | Metric | Unit | Description |
|---|---|---|---|
| CPU utilization | ProcessCpuLoad | % | Process CPU utilization |
| CPU usage time | ProcessCpuTime | ms | Cumulative CPU usage time |
| GC count | YGC | - | Young GC count |
| | FGC | - | Full GC count |
| GC time | GCT | s | Garbage collection time |
| | FGCT | s | Full GC time |
| | YGCT | s | Young GC time |
| Memory zone proportion | O | % | Percentage of used Old memory |
| | M | % | Percentage of used Metaspace memory |
| | CCS | % | Percentage of used compressed class space memory |
| | S0 | % | Percentage of used Survivor 0 memory |
| | S1 | % | Percentage of used Survivor 1 memory |
| | E | % | Percentage of used Eden memory |
| JVM memory | MemHeapInitM | MB | Size of initial JVM HeapMemory |
| | MemNonHeapInitM | MB | Size of initial JVM NonHeapMemory |
| | MemHeapMaxM | MB | Size of HeapMemory configured by JVM |
| | MemHeapCommittedM | MB | Size of HeapMemory currently committed by JVM |

| | MemHeapUsedM | MB | Size of HeapMemory currently used by JVM |
|---|---|---|---|
| | MemNonHeapCommittedM | MB | Size of NonHeapMemory currently committed by JVM |
| | MemNonHeapUsedM | MB | Size of NonHeapMemory currently used by JVM |
| File handles | OpenFileDescriptorCount | count/s | Number of opened file descriptors |
| | MaxFileDescriptorCount | count/s | Maximum number of file descriptors |
| Process execution duration | Uptime | s | Process execution duration |
| Worker threads | PeakThreadCount | count/s | Peak number of threads |
| | ThreadCount | count/s | Total number of threads |
| | DaemonThreadCount | count/s | Number of Daemon threads |
| Production traffic of Broker | OneMinuteRate | bytes/s | Rate of message production by the Broker per minute |
| Consumption traffic of Broker | OneMinuteRate | bytes/s | Rate of message consumption by the Broker per minute |
| Rejected consumption traffic | OneMinuteRate | bytes/s | Rate of Topic request rejection per minute |
| Failed Fetch requests | OneMinuteRate | count/s | Rate of Fetch request failures per minute |
| Failed Produce requests | OneMinuteRate | count/s | Rate of Produce request failures per minute |
| Produced messages | OneMinuteRate | count/s | Rate of message production per minute |
| Read traffic from other Brokers | OneMinuteRate | bytes/s | Amount of traffic read from other Brokers per minute |
| Read traffic to other Brokers | OneMinuteRate | bytes/s | Amount of traffic read to other Brokers per minute |
| Fetch requests | OneMinuteRate | count/s | Rate of Fetch requests per |

| | | | minute |
|---|---|---|---|
| Produce requests | OneMinuteRate | count/s | Rate of Produce requests per minute |
| ControllerBroker | IsControllerBroker | - | The metric value is 1 on the Broker where Controller is located and is 0 on other Brokers. |
| LeaderElection rate | OneMinuteRate | count/s | Rate of LeaderElections per minute |
| LeaderElection latencies | 99thPercentile | ms | The 99th percentile of LeaderElection latencies |
| | 999thPercentile | | The 99.9th percentile of LeaderElection latencies |
| | Mean | | Mean value of LeaderElection latencies |
| UncleanLeaderElections rate | OneMinuteRate | count/s | Rate of UncleanLeaderElections per minute |
| GlobalPartition count | GlobalPartitionCount | count/s | Number of global partitions observed by the controller |
| OfflinePartitions count | OfflinePartitionCount | count/s | Number of offline partitions observed by the controller |
| GlobalTopic count | GlobalTopicCount | count/s | Number of global topics observed by the controller |
| Offline log directory count | OfflineLogDirectory | count/s | Number of offline log directories |
| LogFlush rate | OneMinuteRate | calls/s | Rate of message log flush per minute |
| LogFlush latencies | 99thPercentile | ms | The 99th percentile of LogFlush latencies |
| | 999thPercentile | | The 99.9th percentile of LogFlush latencies |
| | Mean | | Mean value of LogFlush latencies |

| Average network processor idleness rate | NetworkProcessorAvgIdlePercent | - | Average idleness rate of threads in a network thread pool |
|---|---|---|---|
| ISR expansion rate | OneMinuteRate | count/s | In-sync replica (ISR) expansion rate per minute |
| ISR shrinkage rate | OneMinuteRate | count/s | ISR shrinkage rate per minute |
| Replicas count | LeaderReplicaCount | count/s | Number of offline replicas |
| | OfflineReplicaCount | count/s | Number of leader replicas |
| Partitions count | PartitionCount | count | Number of partitions |
| | UnderMinIsrPartitionCount | | Number of partitions under the minimum ISR count |
| | UnderReplicatedPartitions | | Number of UnderReplicatedPartitions |
| FetchConsumer request latencies | 99thPercentile | ms | The 99.9th percentile of FetchConsumer request latencies |
| | 999thPercentile | | The 99.9th percentile of FetchConsumer request latencies |
| | Mean | | Mean value of FetchConsumer request latencies |
| FetchFollower request latencies | 99thPercentile | ms | The 99.9th percentile of FetchFollower request latencies |
| | 999thPercentile | | The 99.9th percentile of FetchFollower request latencies |
| | Mean | | Mean value of FetchFollower request latencies |
| Produce request latency | 99thPercentile | ms | The 99.9th percentile of Produce request latencies |
| | 999thPercentile | | The 99.9th percentile of Produce request latencies |
| | Mean | | Mean value of Produce request latencies |

| Size of the request queue | RequestQueueSize | size | Size of the request queue |
|---|---|---|---|
| Purgatory size | Fetch | size | Number of requests waiting in fetch purgatory |
| | Produce | - | Number of requests waiting in producer purgatory |
| Average request handler idleness rate | OneMinuteRate | - | Idleness rate of request handlers per minute |
| ZooKeeper request latencies | 99thPercentile | ms | The 99th percentile of ZooKeeper request latencies |
| | 999thPercentile | | The 99.9th percentile of ZooKeeper request latencies |
| | Mean | | Mean value of ZooKeeper request latencies |

# Alarm Configurations

Last updated：2023-12-27 15:01:54

## Overview

Elastic MapReduce (EMR) has been connected to Cloud Monitor (CM). You can configure alarm policies for EMR nodes and service monitoring metrics in the CM console.

**Note:**

EMR has been connected to CM's default alarms. CM will automatically create a default alarm policy. For information about metrics/events or alarm rules supported by the EMR default policy, see Default Alarm Policy.

You can manually create an alarm policy and set it as the default alarm policy. After the default policy is set, newly purchased instances will be automatically associated with the default policy without requiring manual addition.

## Directions

1. Log in to the Cloud Monitor console and click **Alarm Configuration** > Alarm Policy on the left sidebar.
2. On the **Alarm Policy** page, click **Create**.
3. In the pop-up window, configure the basic information, alarm policy, and notification template as instructed below.

| Configuration Type | Configuration Item | Description |
|---|---|---|
| Basic information | Policy name | Custom policy name |
| | Remarks | Custom policy remarks |
| | Monitoring type | Supported cloud service monitoring type |
| | Policy type | Select the desired policy type for monitoring Tencent Cloud services. |
| | Project | You can filter to find alarm policies under a project in the alarm policy list. |
| Alarm rule configuration | Alarm object | If you select an instance ID, the alarm policy will be associated with the selected instance. If you select an instance group, the alarm policy will be associated with the selected instance group. If you select all objects, the alarm policy will be associated all instances that the current account has permission on. |

| | Trigger condition | Manual configuration (metric alarm) | Trigger condition: consists of metric, comparison, threshold, statistical period, and the number of consecutive periods. You can expand the trigger condition to view the metric trend, and based on which, set a proper threshold. |
|---|---|---|---|
| | | Manual configuration (event alarm) | Create an event alarm policy to get notifications in case of service resources or underlying infrastructure exceptions. |
| | | Select template | Select a configured template in the drop-down list. For more information, see Configuring Trigger Condition Template. |
| | Configure alarm notification (optional) | Notification template | You can use the preset notification template (alarms will be notified to the root account administrator via SMS and email. This template is selected by default.) or customize one. Up to three notification templates can be bound to each alarm policy. For more information about notification template configuration, see Creating Notification Template. |

4. Click **Complete**.

# Alarm Records

Last updated：2023-12-27 15:02:12

## Feature Overview

The alarm records feature allows you to view alarms in the past six months. On the **Alarm Records** page, you can also quickly access alarm policies by policy type and subscribe to them.

## Directions

1. Log in to the EMR console and click the **ID/Name** of the target cluster in the cluster list to enter the cluster details page.
2. Select **Cluster Monitoring > Alarm Records**. On the displayed page, you can perform fuzzy search for alarms by alarm content, filter by time range and alarm status, and sort by start time.
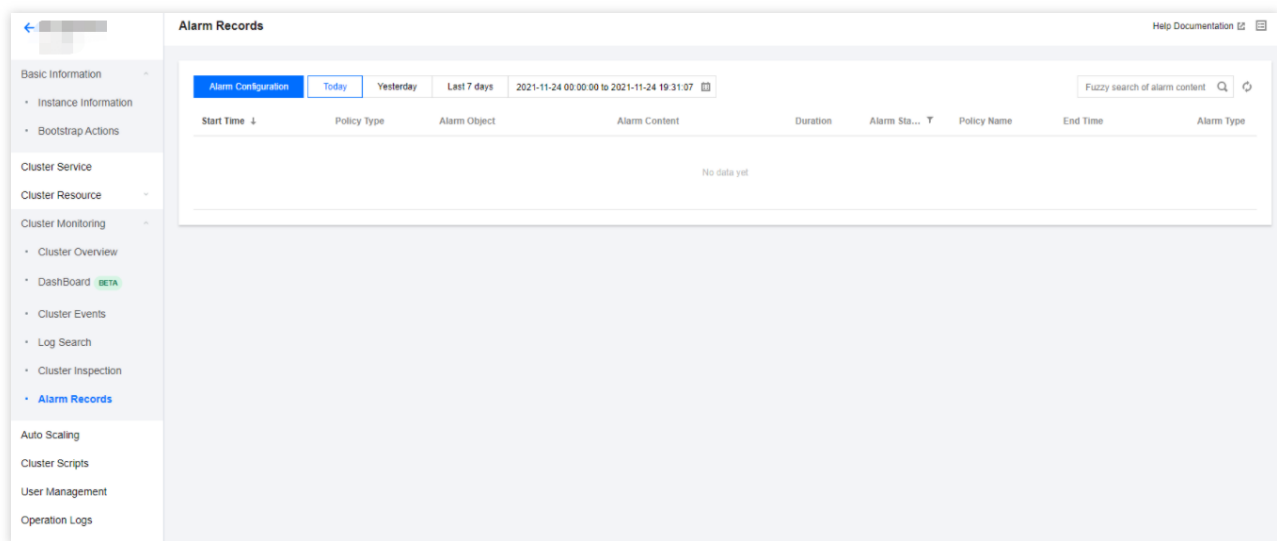


3. Click **Alarm Configuration** to go to the alarm configuration page in the Cloud Monitor console to configure alarm policies. For details, see Alarm Configurations.