

# **TDMQ for RabbitMQ**

## **Operation Guide**

### **Product Documentation**



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## Operation Guide

### Cluster Management

- Creating Cluster

- Viewing Cluster

- Upgrading Cluster

- Deleting Cluster

- Accessing Native Console

- Adding Routing Policy

- Public Network Bandwidth Management

- Connecting to Prometheus

### Node Management

### Vhost Management

### Exchange Management

### Queue Management

### Binding

### User and Permission Management

### Policy List

- Policy Management

- Default Image Policy

### Smart Inspection

### Change Records

### Monitoring and Alarms

### Plugin Management

### Message Query

### Access Management

### Tag Management

- Managing Resource with Tag

- Editing Tag

### Migration to Cloud

- Migrating RabbitMQ to Cloud

- Step 1. Purchasing a TDMQ Instance

- Step 2. Migrating Metadata to Cloud

- Step 3. Migrating Data to Cloud

# Operation Guide

## Cluster Management

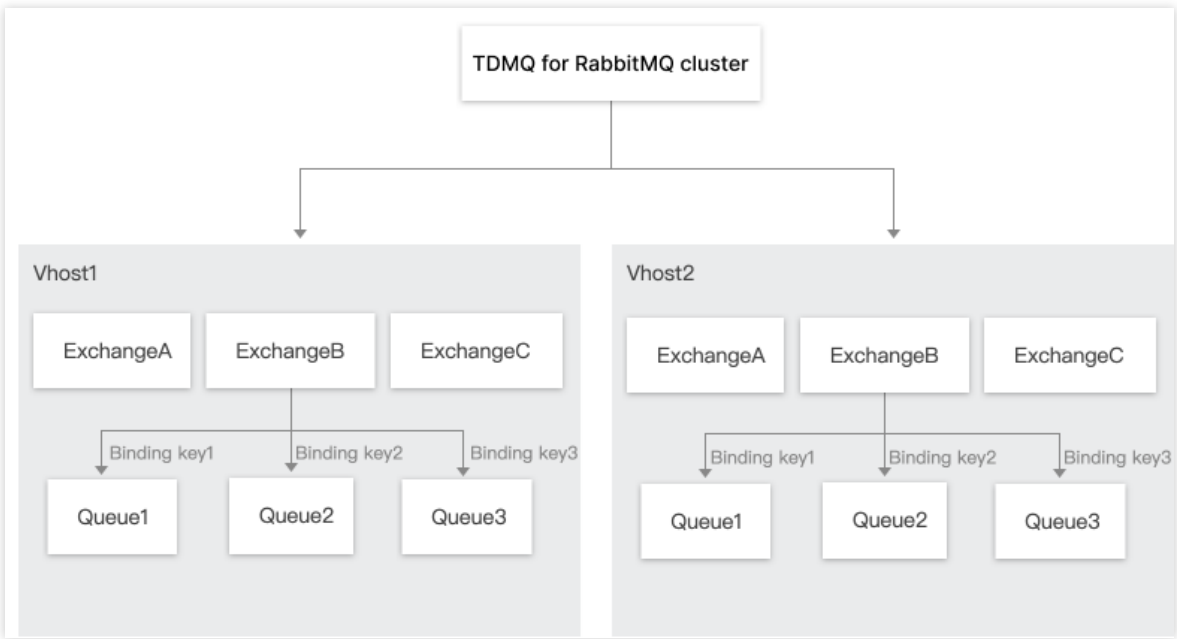
### Creating Cluster

Last updated : 2024-06-26 15:56:25

#### Overview

Cluster is a resource dimension in TDMQ for RabbitMQ. For different clusters, their vhosts, exchanges, and queues are completely isolated from each other. It is a common practice to use respective dedicated clusters for development, test, and production environments.

This document describes how to create an exclusive cluster in the TDMQ for RabbitMQ console.



#### Directions

1. Log in to [RabbitMQ Console](#).
2. Choose **Cluster Management** > **Cluster List** in the left sidebar, click **Create Cluster** to proceed to the purchase page.
3. On the purchase page, select the target instance specification.

| Parameter    | Required | Description                       |
|--------------|----------|-----------------------------------|
| Cluster Type | Yes      | Select <b>Exclusive cluster</b> . |

|                                   |     |   |
|-----------------------------------|-----|---|
| Billing Mode                      | Yes | TDMQ for RabbitMQ exclusive cluster adopts the <b>monthly subscription</b> billing mode.  |
| Region                            | Yes | Select a region close to resources of the deployed client. Cloud products in different regions are not interconnected over private networks and the region cannot be changed after you purchase the service. Proceed with caution.  |
| RabbitMQ Version                  | Yes | Supports versions 3.8.30 and 3.11.8.  |
| AZ                                | Yes | Select an AZ based on your business needs. Cross-AZ deployment is supported.  |
| Node Specification                | Yes | Select an appropriate node specification based on your business needs.  |
| Node Count                        | Yes | Select an appropriate number of nodes based on your business needs.   |
| Single-Node Storage Specification | Yes | Message storage is billed separately, and messages can be retained without limit in the purchased storage capacity. The actual storage usage can be estimated by multiplying the message production traffic by the retention period.  |
| VPC                               | Yes | Bind the domain name of the new cluster's access point to the selected VPC.   |
| Mirrored Queue                    | No  | We recommend that you enable mirrored queue to ensure the availability. After you enable it, a policy will be generated by default in the policy list in the instance details. You can delete it or customize a new policy to overwrite it. For more information, see <a href="#">Policy List</a> . |
| Cluster Name                      | No  | Enter Cluster Name, which should contain 3-64 characters, can only include digits, letters, hyphens (-), and underscores (_).   |
| Image Queue                       | No  | It is recommended to enable the image queue to ensure availability. Once enabled, a default policy is generated under Instance Details-Policy List, which can be deleted, or overwritten by a Custom New Policy. See Policy List.   |
| Tag                               | No  | Tags are used to categorize and manage resources. For more information, see <a href="#">Tag Overview</a> .  |

4. Select **I have read and agree to TDMQ for RabbitMQ Terms of Service** and click **Buy Now**.

5. On the order payment page, click **Pay** and wait 3–5 minutes. Then, you can see the created cluster on the **Cluster** page.

Create ClusterEdit Resource Tag

Search by keyword

Cluster ...

Version

Status

AZ

Cluster Specification

Billing Mode

Resource Tag

Descrip...

Operati...

|                          |  |        |        |                  |  |   |                       |  |
|--------------------------|--|--------|--------|------------------|--|---|-----------------------|--|
| <input type="checkbox"/> | <a href="#">amqp-25mpxb9xhuanhuan_test</a> | 3.8.30 | Normal | Guangzhou Zone 3 | Experience   1 node<br>Model 2-core, 4 GB MEM<br>Storage 200GB | Monthly subscription<br>Expire at 2023-08-27 15:35:15 | <a href="#">Renew</a> | <a href="#">Upgrade</a><br><a href="#">Renew</a><br><a href="#">Delete</a> |
|--------------------------|--|--------|--------|------------------|--|---|-----------------------|--|

Total items: 1

20 / page

1

/ 1 page

# Viewing Cluster

Last updated : 2024-08-16 17:30:07

## Overview

This document describes how to view the configuration information and health status of a cluster in the TDMQ for RabbitMQ console.

## Directions

1. log in to [RabbitMQ Console](#).
2. In the left navigation bar, select **Cluster** > **Cluster**, after selecting the region, click the ID of the target cluster, on the **Basic Info** page, you can view information such as the cluster's resource overview, health status, basic information, connection method, and Web console access address.

Cluster /

Upgrade

Renew

Termin

Basic Info

Monitoring

Node

Vhost

User and Permission

Smart Inspection

Change Record

Plugin Management

Cluster Overview

Unfold

Vhosts

1

Queue Count

2

Exchange Count

8

Messages Produced per Sec

0

Messages consumed per second

0

Currently Heaped Messages

0

Basic Info

Name

Model

2-core, 4 GB MEM

ID

Node Count

3

Status

Normal

Storage

600GB

Region

North China(Beijing)

Public Network Bandwidth

3Mbps

Adjust Configuration

AZ

Beijing Zone 3

Beijing Zone 5

Billing Mode

Monthly subscription

Description

-

Type

Exclusive Edition

Resource Tag

No tag

Creation Time

2024-08-13 17:47:37

Version

3.8.30

Expiration Time

2024-09-13 17:58:48

Other Information

Client Access

Network Information

Web Console Access Address

Monitor Instance with Prometheus

Client Access

Add a routing policy

| Access Type | Access Policy | Public Network Ba... | Network  | Operation |
|-------------|---------------|----------------------|--|-----------|
| VPC Network | -             | -                    | <div>vpc-8jg39v8g</div> <div>subnet-8ahra8ep</div> <div>amqp://10.52.3.16:5672</div> | Delete    |

## Health Status Description

TDMQ for RabbitMQ has an inspection program in place for each cluster, which checks the cluster's node status, disk utilization, memory utilization, and other metrics and displays different health status when these metrics exceed certain thresholds as detailed below:

| Metric                   | Threshold (N) | Status Description |
|--------------------------|---------------|--------------------|
| Node not started         | -             | Abnormal           |
| Abnormal node connection | -             | Abnormal           |
|                          |               |                    |



|                            |   |          |
|----------------------------|---|----------|
| Available disk space       | $N \geq 100 \text{ GB}$                 | Healthy  |
|                            | $50 \text{ MB} \leq N < 100 \text{ GB}$ | Alarmed  |
|                            | $N < 50 \text{ MB}$                     | Abnormal |
| Memory utilization         | $N \leq 60\%$                           | Healthy  |
|                            | $60\% < N \leq 90\%$                    | Alarmed  |
|                            | $N > 90\%$                              | Abnormal |
| Socket utilization         | $N \leq 90\%$                           | Healthy  |
|                            | $N > 90\%$                              | Abnormal |
| Erlang process utilization | $N \leq 90\%$                           | Healthy  |
|                            | $N > 90\%$                              | Abnormal |
| FD utilization             | $N \leq 90\%$                           | Healthy  |
|                            | $N > 90\%$                              | Abnormal |

# Upgrading Cluster

Last updated : 2024-06-26 15:56:25

## Overview

If the current cluster specifications cannot meet your business needs, you can increase the node specification, node quantity, and single-node storage specification in the console.

### Note:

Currently, you can only upgrade but not downgrade the node specification, node count, and node storage specification.

## Directions

1. Log in to the [RabbitMQ console](#).
2. On the left sidebar, select **Cluster** > **Cluster**. Then, click **Upgrade** in the **Operation** column on the cluster list page.
3. Select the target node specification and click **Confirm the Adjustment**.

**Target Node Specification:** When the node specification is changed, resources will be changed accordingly, and the open-source console needs to be restarted. We recommend that you configure multi-node mirrored queues before upgrading the configurations to avoid service interruption and data loss during the upgrade.

**Target Node Count:** You can select 3, 5, or 7 nodes.

**Single-Node Storage Specification:** The new single-node storage specification takes effect for all nodes in the cluster.

**Upgrade Time:** You can select **Execute** or **Custom time** (upgrade at night is recommended to reduce the impact on your business).

### Adjust Configurations

Current Configuration

| Node Specification  | Storage Specification | Expiration Time     |
|---------------------|-----------------------|---------------------|
| Experience   1-node | 200 GB                | 2023-08-27 15:35:15 |

Target Node Specification

Experience

Base

Base II

Standard

Standard II

Advanced I

Advanced II

When the node specification is changed, resources will be changed accordingly, and the open-source console needs to be restarted. We recommend that you configure multi-node mirrored queues before upgrading the configurations to avoid service interruption and data loss during the upgrade.

Target Node Count

—

1

+

Single-Node Storage Specification

—

200

+

G

The adjusted storage specification of a single node applies to all nodes in the cluster.

Configuration Adjustment Time

☒ Execute

☐ Custom time (any time within the next 24 hours)

2023-07-27 16:01:39

Configuration Upgrade Fees

0USD/month

Confirm the Adjustment

Cancel

# Deleting Cluster

Last updated : 2024-06-26 15:56:25

## Overview

You can delete a TDMQ for RabbitMQ cluster if you no longer need it.

The lifecycle of a TDMQ for RabbitMQ cluster is the process the cluster goes through from start to release. By properly managing the cluster lifecycle, you can ensure that the applications running in the cluster can provide services efficiently and economically. A cluster has the following status:

| Status           | Status Attribute    | Status Description  |
|------------------|---------------------|---|
| Creating         | Intermediate status | The cluster is being created.   |
| Running          | Stable status       | The cluster is running normally, indicating that your node status, disk utilization, and other metrics are within the normal range.   |
| Deleting         | Intermediate status | The cluster is being deleted in the console or via APIs.  |
| Isolated         | Intermediate status | The cluster is in the 7-day isolation period after overdue payment. Production and consumption cannot be performed in a cluster in the isolated status, but the data and configurations saved in the cluster will not be deleted. |
| Failed to create | Intermediate status | The cluster is purchased with fees successfully deducted in the console or via APIs but fails to be assigned. In this case, <a href="#">contact us</a> for assistance.  |
| Deletion failed  | Stable status       | TDMQ for RabbitMQ fails to release resources after the cluster is manually deleted or isn't renewed within 7 days after expiration.   |

## Directions

### Manual deletion

You can manually delete a monthly subscribed cluster before it expires by following the steps below:

1. Log in to the [RabbitMQ console](#).
2. On the left sidebar, select **Cluster** > **Cluster**. Then, click **More** > **Terminate** in the **Operation** column on the cluster list page.

3. In the Delete Confirmation pop-up window, delete the cluster after double confirmation. .

### Terminate Instance

1 Instance Termination Details

>

2 Note

You have selected 1 instance. [View Details](#) ▲

| No. | ID            | Instance Name | Operation         |
|-----|---------------|---------------|-------------------|
| 1   | amqp-gawpazk2 | test-huanhuan | Can be terminated |

Instance Termination Details

| Instance ID   | Model | Node Count | Storage |
|---------------|-------|------------|---------|
| amqp-gawpazk2 | 2C4G  | 3          | 600GB   |

Next

Cancel

**Note:**

After a cluster is deleted, all its configurations will be cleared and cannot be recovered. Therefore, proceed with caution.

### Automatic deletion upon expiration or overdue payment

Monthly subscribed instances can be retained in the TDMQ for RabbitMQ console for up to 7 calendar days after they expire or have overdue payments. You can continue to use them if you renew them within 7 days after expiration. For details, see [Payment Overdue](#).

If the TDMQ for RabbitMQ instance is not renewed within 7 days (inclusive) after expiration, its resources will be released at midnight on the 8th day, and data will be cleared and cannot be restored.

**Note:**

Production and consumption cannot be performed in a cluster in the isolated status, but the data and configurations saved in the cluster will not be deleted.

For clusters in the 7-day isolation status, you can go to the console to renew them by clicking **Renew** in the **Operation** column on the cluster list page. Successfully renewed clusters can go back to the running status and be normally used.

# Accessing Native Console

Last updated : 2024-06-26 15:56:25

## Overview

This document describes how to access the open-source RabbitMQ console in the TDMQ for RabbitMQ console.

## Directions

1. Log in to the [RabbitMQ console](#).
2. Select **Cluster** > **Cluster** on the left sidebar, select a region, and click the ID of the target cluster to enter the **Basic Info** page of the cluster.
3. In the **Web Console Access Address** module, you can perform the following operations:

Enable and disable the public or private network access address.

Log in to the open-source RabbitMQ console by using the **Public Network Access Address**, username, and password.

Click **Modify** next to the public network access policy to set the console access allowlist.

You can enter multiple IPs or IP ranges and separate them by comma.

All users are denied access to the console by default.

### Note:

Only /28-/32 subnet masks can be configured for IP ranges. If you want to use the /24 mask, [submit a ticket](#) for assistance.

### Edit Public Network Access Policy

Access Policy Type

☒ Allowlist

IP

You can enter multiple IPs or IP ranges and separate them by comma.

Submit

Close

# Adding Routing Policy

Last updated : 2024-06-26 15:56:25

## Overview

This document describes how to configure route access rules in the TDMQ for RabbitMQ console.

## Directions

### Note:

You can create up to five routes (including only one public network route) for a cluster. If you need more, [submit a ticket](#) for assistance.

VPC Network

Public Domain Name Access

**Operation scenario:** When purchasing a cluster, if you select VPC and choose a corresponding VPC environment (such as VPC A), then TDMQ for RabbitMQ services (such as data production and consumption) can be accessed only from VPC A. If you subsequently find that you need to access the TDMQ for RabbitMQ services in VPC A from other VPCs (such as VPC B), you can select an appropriate routing policy for VPC by configuring the access mode.

### Directions:

1. Log in to the [RabbitMQ console](#).
2. Select **Cluster** > **Cluster** on the left sidebar, select a region, and click the ID of the target cluster to enter the **Basic Info** page of the cluster.
3. In the client access module, click **Add a routing policy** in the top-right corner.
4. In the pop-up window, select **VPC Network** as the route type, choose whether to enable SSL/TLS transfer encryption, and select the VPC and subnet.

### Add a routing policy

Route Type

VPC Network

Enable SSL/TLS

☒

VPC

vpc-fs6qq7yn | clue-test | 10.0.0.0/16

If no suitable VPCs are available, you can [create one](#).

Subnet

subnet-8ah6a7rs | test | 10.0.2.0/24

If no suitable subnets are available, you can [create one](#).

IP

Enter the IP address (optional)

If no IP address is specified, one will be automatically assigned.

Submit

Close

**Note:**

If you select VPC access, you can specify the IP to keep it unchanged when changing the access mode.

If you want to enable SSL/TLS transfer encryption, click [here](#) to download the certificate first.

5. Click **Submit**.

**Operation scenario:** If your consumer or producer is located in a self-built data center or another cloud, you can produce and consume data in TDMQ for RabbitMQ through public network access.

**Directions:**

1. Log in to the [RabbitMQ console](#).
2. Select **Cluster** > **Cluster** on the left sidebar, select a region, and click the ID of the target cluster to enter the **Basic Info** page of the cluster.
3. In the client access module, click **Add a routing policy** in the top-right corner.
4. In the pop-up window, select **Public domain name access** as the route type.



**Add a routing policy**

Route Type

Public domain name access ▼

Public Network Bandwidth

3Mbps

TDMQ for RabbitMQ provides public network bandwidth of 3 Mbps free of charge by default.

**Submit**

Close

5. Click **Submit**.

**Note:**

When using a public network domain name for access, you can click the **Edit** icon next to the access policy to set the public network access allowlist.

**Edit Public Network Access Policy**

Access Policy Type

☒ Allowlist

IP

127.0.0.1

You can enter multiple IPs or IP ranges and separate them by comma.

**Submit**

Close

# Public Network Bandwidth Management

Last updated : 2024-06-26 15:56:25

## Overview

TDMQ for RabbitMQ uses private network transmission by default. To access it through the public network, you need to open an independent public network route. For more information, see [Add Routing Policy](#). Free public network bandwidth of 3 Mbps is provided by default.

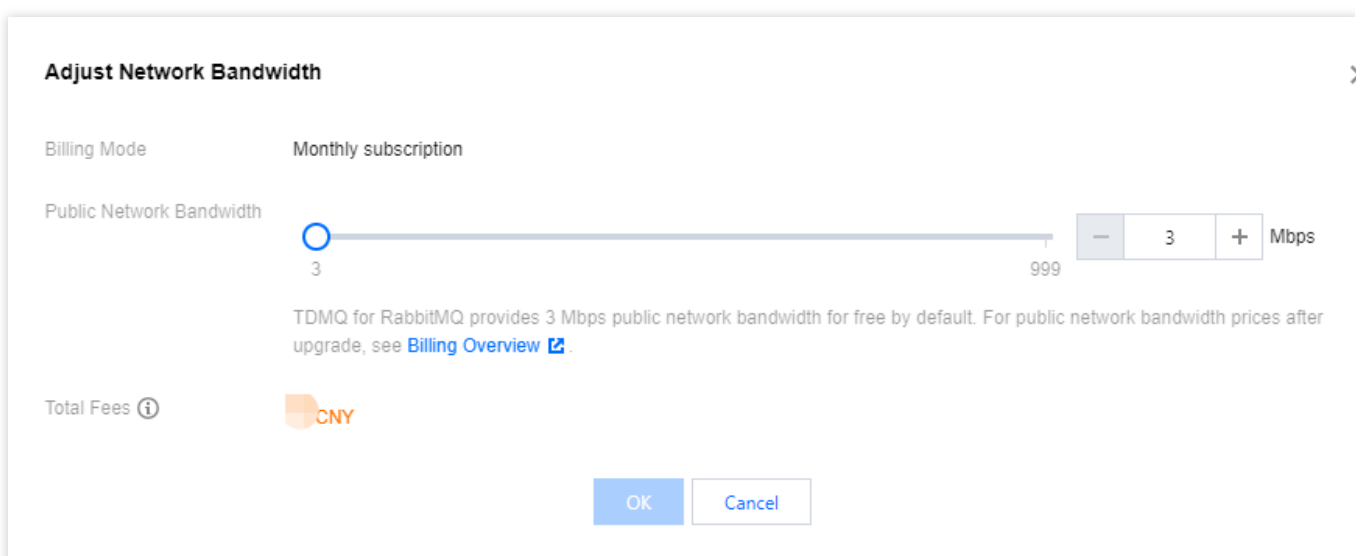
If you require higher bandwidth, TDMQ for RabbitMQ supports public network bandwidth upgrade. You can pay additional fees to purchase it. For detailed pricing, see [Billing Overview](#).

This document describes how to adjust the public network bandwidth configuration and delete public network bandwidth in the TDMQ for RabbitMQ console.

## Directions

### Adjusting Public Network Bandwidth Configuration

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, select **Cluster** > **Cluster**, select a region, and click the ID of the target cluster to enter the basic cluster information page.
3. In the **Basic Information** module, click **Adjust Configuration** next to the Public Network Bandwidth.



**Adjust Network Bandwidth**

Billing Mode: Monthly subscription

Public Network Bandwidth: 3 Mbps (Slider range: 3 to 999 Mbps)

TDMQ for RabbitMQ provides 3 Mbps public network bandwidth for free by default. For public network bandwidth prices after upgrade, see [Billing Overview](#).

Total Fees ①: CNY

OK Cancel

4. In the pop-up window, modify the public network bandwidth and click **OK**. The public network bandwidth configuration is adjusted.

## Deleting Public Network Routes

**Note:**

You can delete public network routes only when the public network bandwidth is 3 Mbps (without purchasing additional public network bandwidth).

On the Basic Information page of the cluster, in the **Client Access** module, click **Delete** next to the Public Network Bandwidth Operation column you want to delete.

Client Access ⓘ

Add a route

| Access Type               | Access Policy   | Public Networ...                                       | Network  | Operation              |
|---------------------------|---|--|--|------------------------|
| VPC Network               | -   | -  | <div>VPC ID: [redacted] </div> <div>Subnet ID: [redacted] </div> | <a href="#">Delete</a> |
| Public domain name access | <div>Allowlist: [redacted] <a href="#">Modify</a></div> | <div>3 Mbps <a href="#">Adjust Configuration</a></div> | <div>Public IP: [redacted] </div>                                | <a href="#">Delete</a> |

Last updated : 2024-06-26 15:56:25

TDMQ for RabbitMQ clusters currently provide Prometheus to scrape the metric information of nodes, including basic monitoring metrics such as queue, channel, and connection, as well as the metrics exposed by the broker's JMX.

1. Log in to the [RabbitMQ console](#).
2. Select **Cluster** > **Cluster** on the left sidebar, select a region, and click the ID of the target cluster to enter the **Basic Info** page of the cluster.
3. Click **Obtain Monitoring Target** in the top-right corner of the **Other Information** > **Monitor Instance with Prometheus** module and select the VPC and subnet.

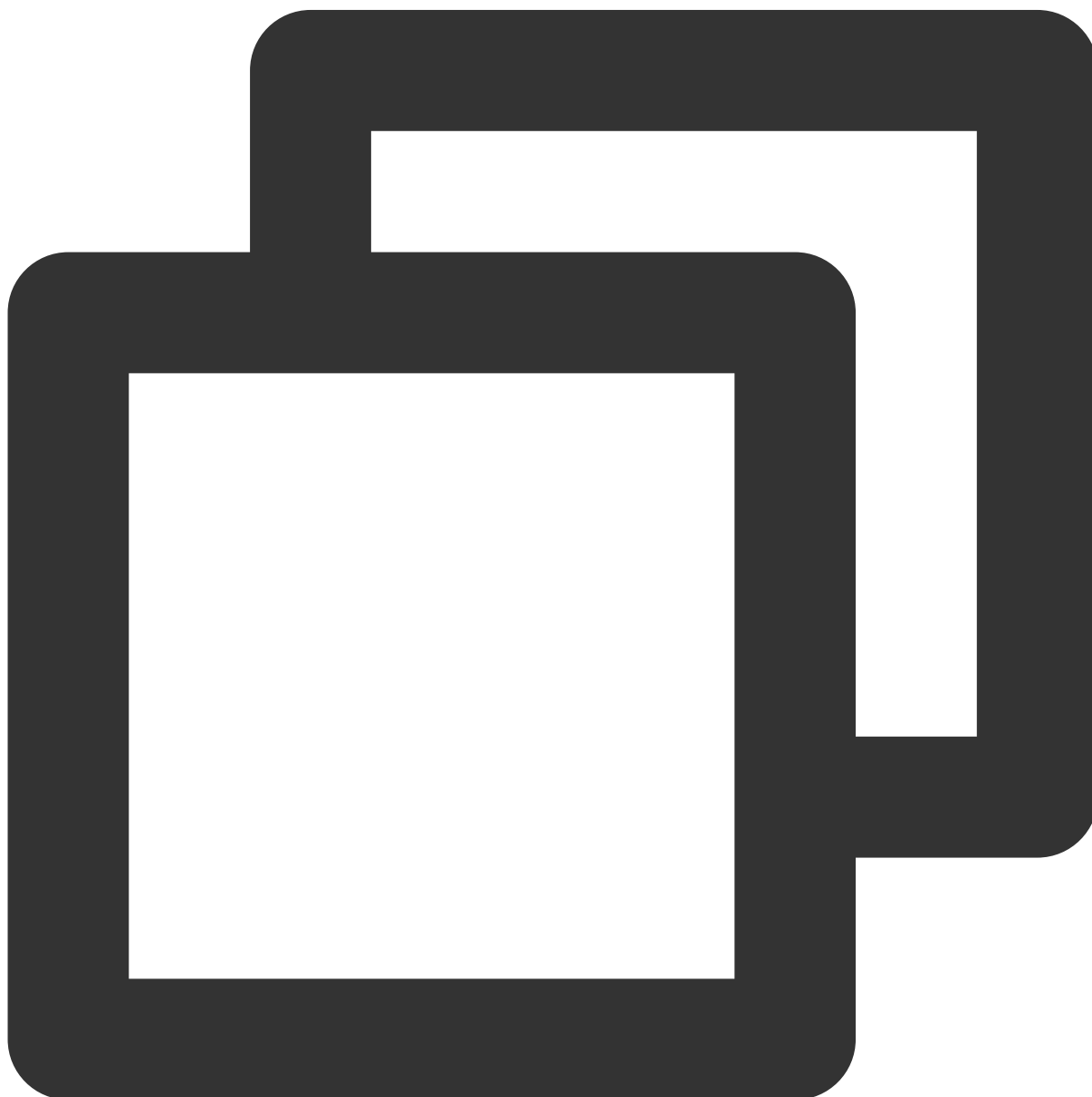
4. Click **Submit** to get the set of monitoring targets.

**Monitor Instance with Prometheus**

| jmx exporter   | node exporter   | Operation              |
|----------------|---|------------------------|
| 10.0.2.14:7001 | rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local:15692 | <a href="#">Delete</a> |
| 10.0.2.14:7002 | rabbit@rabbitmq-broker-1.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local:15692 |                        |
| 10.0.2.14:7003 | rabbit@rabbitmq-broker-2.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local:15692 |                        |

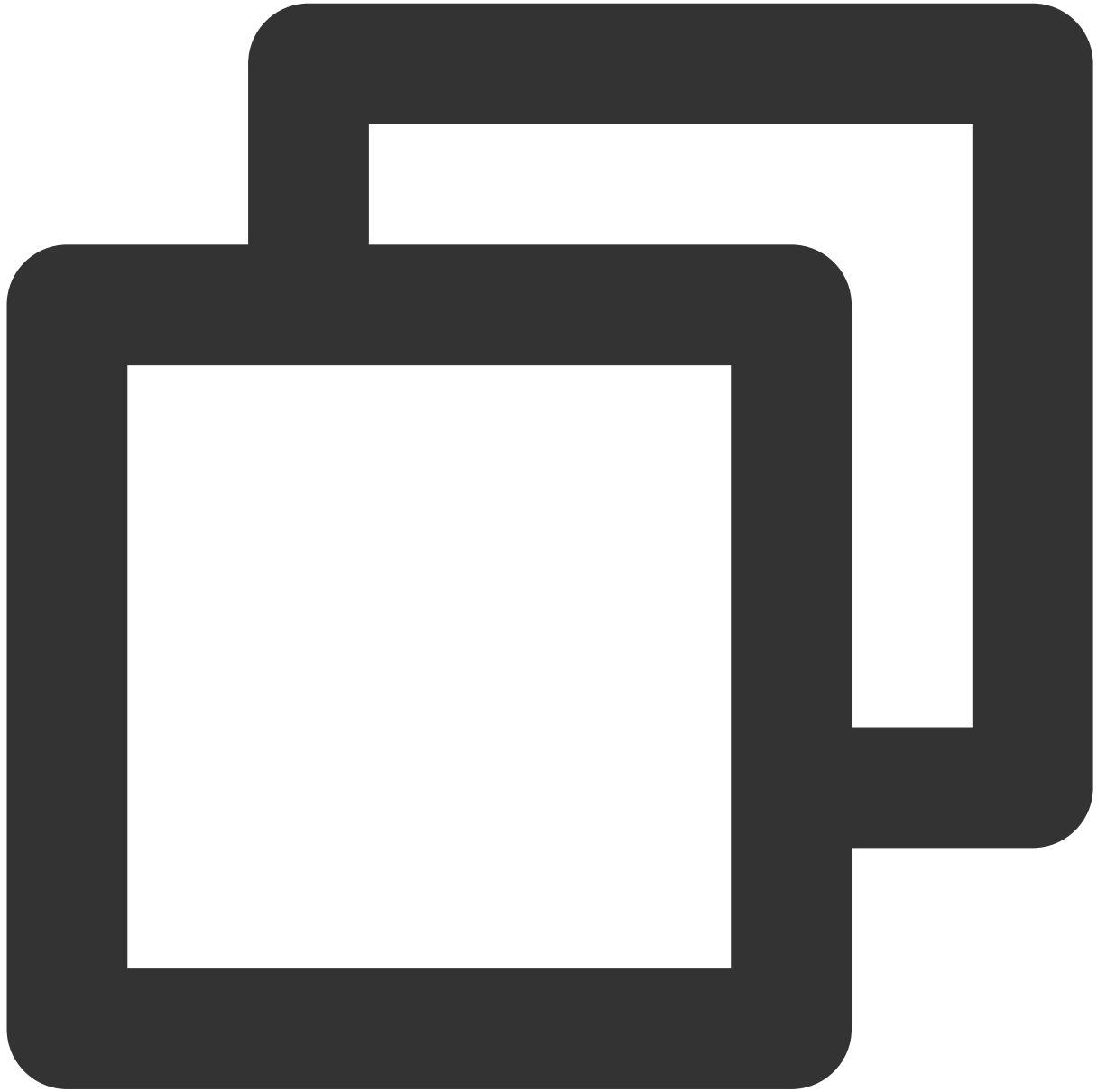
5. Download [Prometheus](#) and configure the monitoring scrape address.

5.1 Enter the directory of the Prometheus package and run the following command to decompress it.



```
tar -vxf prometheus-2.30.3.linux-amd64.tar.gz
```

5.2 Modify the `prometheus.yml` configuration file by adding the `jmx_exporter` and `node_exporter` scrape tasks.



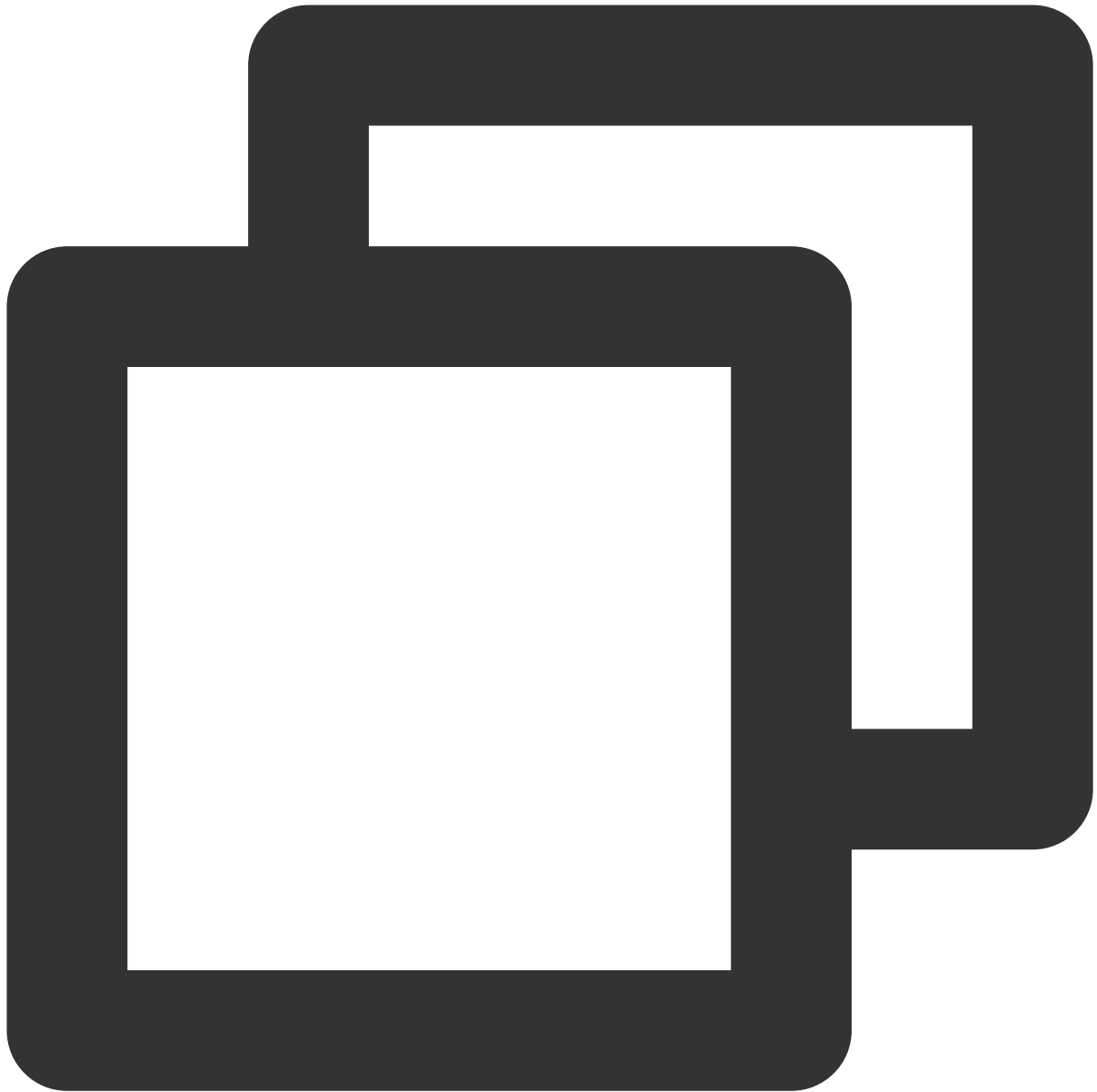
```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from
  - job_name: "prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
```

```
- targets: ["localhost:9090"]

- job_name: "broker-jmx-exporter"
  scrape_interval: 5s
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.x.x.0:60001', '10.x.x.0:60003', '10.x.x.0:60005']
      labels:
        application: 'broker-jmx'
- job_name: "broker-node-exporter"
  scrape_interval: 10s
  metrics_path: '/metrics'
  static_configs:
    - targets: ['10.x.x.0:60002', '10.x.x.0:60004', '10.x.x.0:60006']
      labels:
        application: 'broker-node'
```

Here, `broker-jmx-exporter` is the tag configured for the `jmx` metric of the broker scraped by Prometheus, `Targets` contains the information of the mapped port, `broker-node-exporter` is the tag configured for the basic metrics of the node of the scraped broker, and `scrape_interval` is the frequency of scraping metric data.

5.3 Start Prometheus.

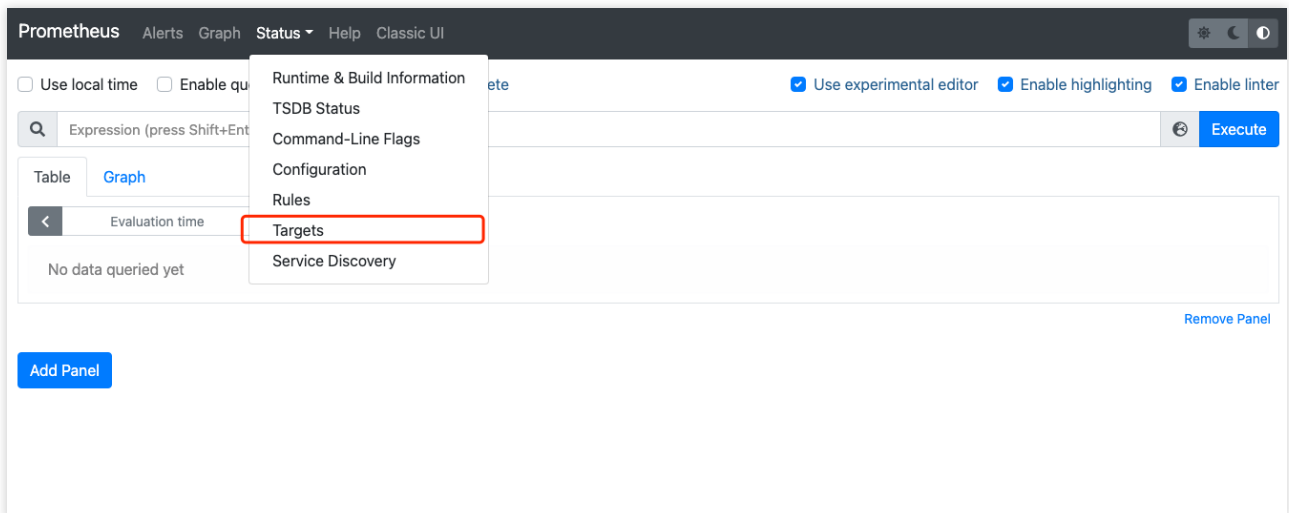


```
./prometheus --config.file=prometheus.yml --web.enable-lifecycle
```

5.4 Open the UI provided by Prometheus to check whether the connected targets are normal by entering

`http://localhost:9090` in the browser for example.





5.5 Confirm that all targets are in **UP** status.

| Prometheus Alerts Graph Status ▾ Help Classic UI                              |       |  |             |                 |  |
|---|-------|--|-------------|-----------------|--|
| Targets   |       |  |             |                 |  |
| <a href="#">All</a> <a href="#">Unhealthy</a> <a href="#">Collapse All</a>    |       |  |             |                 |  |
| <b>broker-jmx-exporter (2/2 up)</b> <a href="#">show less</a>                 |       |  |             |                 |  |
| Endpoint  | State | Labels   | Last Scrape | Scrape Duration |  |
| <a href="http://10.0.1.176:60003/metrics">http://10.0.1.176:60003/metrics</a> | UP    | <a href="#">application="broker-jmx"</a> <a href="#">instance="10.0.1.176:60003"</a> <a href="#">job="broker-jmx-exporter"</a>   | 7.171s ago  | 0.819ms         |  |
| <a href="http://10.0.1.176:60001/metrics">http://10.0.1.176:60001/metrics</a> | UP    | <a href="#">application="broker-jmx"</a> <a href="#">instance="10.0.1.176:60001"</a> <a href="#">job="broker-jmx-exporter"</a>   | 5.206s ago  | 1.464ms         |  |
| <b>broker-node-exporter (1/1 up)</b> <a href="#">show less</a>                |       |  |             |                 |  |
| Endpoint  | State | Labels   | Last Scrape | Scrape Duration |  |
| <a href="http://10.0.1.176:60002/metrics">http://10.0.1.176:60002/metrics</a> | UP    | <a href="#">application="broker-node"</a> <a href="#">instance="10.0.1.176:60002"</a> <a href="#">job="broker-node-exporter"</a> | 4.241s ago  | 8.333ms         |  |

If the targets are in **DOWN** status, check whether the network is accessible or check the **Error** option at the end of the status bar for the cause.

5.6 Query the monitoring metric data.

Click **Graph**, enter a metric name such as `node_memory_MemAvailable_bytes`, and click **Execute** to view the monitoring data.

Prometheus Alerts Graph Status Help Classic UI

Targets

All Unhealthy Collapse All

broker-jmx-exporter (2/2 up) show less

| Endpoint                        | State | Labels   | Last Scrape | Scrape Duration |
|---------------------------------|-------|--|-------------|-----------------|
| http://10.0.1.176:60003/metrics | UP    | application="broker-jmx" instance="10.0.1.176:60003" job="broker-jmx-exporter" | 7.171s ago  | 0.819ms         |
| http://10.0.1.176:60001/metrics | UP    | application="broker-jmx" instance="10.0.1.176:60001" job="broker-jmx-exporter" | 5.206s ago  | 1.464ms         |

broker-node-exporter (1/1 up) show less

| Endpoint                        | State | Labels   | Last Scrape | Scrape Duration |
|---------------------------------|-------|--|-------------|-----------------|
| http://10.0.1.176:60002/metrics | UP    | application="broker-node" instance="10.0.1.176:60002" job="broker-node-exporter" | 4.241s ago  | 8.333ms         |

# Node Management

Last updated : 2024-06-26 15:56:25

## Overview

The node management page displays the list of all nodes in the current cluster and node status metrics. This document describes how to view the node list in the TDMQ for RabbitMQ console.

## Prerequisites

You have created a cluster as instructed in [Cluster Management](#).

## Directions

### Viewing a node

1. Log in to the [RabbitMQ console](#).
2. Select **Cluster** > **Cluster** on the left sidebar, select a region, and click the ID of the target cluster to enter the **Basic Info** page of the cluster.
3. In the **Basic Information** tab, by selecting the top **Node** Tab, you can view the node information of the current cluster.

| Basic Info  | Monitoring | Node             | Vhost            | User and Permission | Smart Inspection                | Change Record | Plugin Management |
|---|------------|------------------|------------------|---------------------|---------------------------------|---------------|-------------------|
| <input type="text" value="Search by name"/>                                       |            |                  |                  |                     |                                 |               |                   |
| Node  | Status     | Disk Utilization | Erlang Processes | Memory              | Operation                       |               |                   |
| rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local | Online     | 0.000%           | 451              | 0                   | <a href="#">View monitoring</a> |               |                   |
| rabbit@rabbitmq-broker-1.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local | Online     | 0.000%           | 450              | 0                   | <a href="#">View monitoring</a> |               |                   |
| rabbit@rabbitmq-broker-2.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local | Online     | 0.000%           | 451              | 0                   | <a href="#">View monitoring</a> |               |                   |
| Total items: 3  |            |                  |                  | 20 / page           |                                 | 1             | / 1 page          |

### Note:

On the node list page, click **View monitoring** in the **Operation** column to view the detailed monitoring information of the corresponding node.

# Vhost Management

Last updated : 2024-08-16 17:34:45

## Overview

Virtual host (vhost) is a resource management concept in TDMQ for RabbitMQ. It is used for logical isolation.

Exchanges and queues of different vhosts are isolated from each other.

Generally, different business scenarios can be isolated by vhost and configured with dedicated settings, such as message retention period.

This document describes how to create multiple vhosts in TDMQ for RabbitMQ so as to use the same TDMQ for RabbitMQ cluster in different scenarios.

### Note :

Exchange and queue names must be unique in the same vhost.

## Prerequisites

You have created a cluster as instructed in [Cluster Management](#).

## Directions

### Creating vhost

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, choose **Cluster Management > Vhost**, click **Create** to enter the Create Vhost page.
3. In the **Create Vhost** window, configure the vhost attributes:

Vhost Name: Enter the vhost name, which cannot be modified after creation and can contain 1–64 letters, digits, "-", and "\_".

Trace Plugin: The [rabbitmq\\_tracing](#) plugin can track messages in RabbitMQ. After it is enabled, you can use the message query feature in the console.

Remarks: Enter the vhost remarks.

4. Click **Submit**.

### Create Vhost

Vhost Name \*

Please enter the vhost name

It should contain 1 to 64 characters, including letters, digits, ".", "-", and "\_".

Trace Plugin ⓘ

☐

Image Queue

☒ Enable Image Queue

It is recommended to enable the Image Queue to ensure availability.  
After enabling, a default policy [🔗](#) will be generated under Instance Details - Policy List. This policy only applies to the default Vhost /, and can be deleted or overridden by a custom new policy.

Vhost Description

Please enter the description

Submit

Close

Next steps: You can create an exchange and queue in the vhost to produce and consume messages.

## Viewing a vhost

On the vhost list page, click the ID of the target vhost to enter the **Basic Info** page, which is divided into two modules: Overview

Queue Count: The number of queues under the current vhost.

Exchange Count: The number of exchanges under the current vhost.

Channel Count: The number of channels under the current vhost.

User Count: The number of users under the current vhost.

Number of heaped messages: The number of the heaped messages under the current Vhost.

Production rate: The production rate of the current Vhost.

Consumption rate: The consumption rate of the current Vhost.

(AMQP default vhost) ▾

Default virtual host[系统自动批量同步]

**Basic Info** Exchange Queue Routing Policy

|             |                |          |       |                 |                  |                 |
|-------------|----------------|----------|-------|-----------------|------------------|-----------------|
| Queue Count | Exchange Count | Channels | Users | Heaped Messages | Production Speed | Messaging Speed |
| 3           | 8              | 6        | 1     | 0               | 0 /s             | 0 /s            |

**Connection List**

| Connection Name | Client IP | Status  | SSL/TLS | Protocol   | Channels |
|-----------------|-----------|---------|---------|------------|----------|
| [REDACTED]      | 10.0.0.67 | running | Close   | AMQP 0-9-1 | 1        |
| [REDACTED]      | 10.0.0.67 | running | Close   | AMQP 0-9-1 | 1        |
| [REDACTED]      | 10.0.0.67 | running | Close   | AMQP 0-9-1 | 1        |
| [REDACTED]      | 10.0.0.67 | running | Close   | AMQP 0-9-1 | 1        |
| [REDACTED]      | 10.0.0.67 | running | Close   | AMQP 0-9-1 | 1        |
| [REDACTED]      | 10.0.0.67 | running | Close   | AMQP 0-9-1 | 1        |

Total items: 6 10 ▾ / page 1 / 1 p

### Connection List

The information of connections under the current vhost and the number of channels of each connection.

## Modifying a vhost

The trace plugin can be toggled on/off on the vhost list page.

The Trace plugin can be enabled and disabled on the Vhost list page. It is recommended to use the Trace plugin for small-traffic verification or troubleshooting scenarios, and it is not advised to enable it when the cluster's TPS exceeds 10,000. Please read [Message Query](#).

**Vhost** Guangzhou ▾ Current Cluster [REDACTED]

[Create \(1/20\)](#)

| Vhost Name           | Status | Heaped Messages ▴ ▾ | Messaging Speed ▴ ▾                 | Trace Plugin ⓘ                      | Image Queue | Description                  | Creation/Update Time                               | Operation  |
|----------------------|--------|---------------------|-------------------------------------|-------------------------------------|-------------|------------------------------|--|--|
| (AMQP default vhost) | Normal | 0                   | Production 0/s<br>Consumption 0.4/s | <input checked="" type="checkbox"/> | Enable      | Default virtual host[系统自...] | 2024-06-05 14:36:08.012<br>2024-06-14 16:07:51.727 | <a href="#">View Use</a><br><a href="#">Delete</a> |

Total items: 1 20 ▾ / page 1 / 1 p

1. On the **Vhost** list page, click **Edit** in the **Operation** column of the target vhost to enter the editing page.
2. Modify the description and click **Submit**.

## Deleting a vhost

You can modify the description of a vhost in the following steps:

1. On the **Vhost** list page, click **Delete** in the **Operation** column.
2. In the deletion confirmation pop-up window, click **OK**.

**Note:**

---

After a vhost is deleted, all the configurations under it will be cleared and cannot be recovered.



# Exchange Management

Last updated : 2024-06-26 15:56:25

## Overview

A producer sends a message to an exchange, which subsequently routes the message to one or more queues based on its attributes or content (or discards it). Then, a consumer pulls it from one of these queues and consumes it. This document describes how to create, delete, and query an exchange in the TDMQ for RabbitMQ console.

## Prerequisites

You have created a vhost as instructed in [Vhost Management](#).

## Directions

### Creating an exchange

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, choose **Cluster Management > Exchange** tab, choose a Vhost, and click **Create** to enter the Create Exchange page.
3. In the **Create Exchange** window, enter the following information:

### Create Exchange

Current Vhost

(AMQP default vhost)

Exchange Name \*

Please enter the name

This field is required. Please enter 1-64 letters, digits, or symbols (".", "-", or "\_").

Route Type \*

direct

For route type descriptions, see [Route Type](#)

Durable

☒

If this option is enabled, the exchange will still exist after the service is restarted; if it is disabled, the exchange will disappear after the service restart and needs to be created again.

AutoDelete

☐

If this option is enabled, the exchange will be automatically deleted when the last queue bound to it is deleted.

Internal

☐

If this option is enabled, this exchange cannot be directly used by producers but bound with other exchanges.

Exchange Description

Please enter the description

Up to 128 characters

Advanced Settings ▶

Submit

Close

**Exchange Name:** Enter the exchange name, which cannot be modified after creation and can contain 1–64 letters, digits, hyphens, and underscores.

**Route Type:** Select a route type (direct, fanout, topic, or header), which cannot be changed after creation. For more information on route types, see [Exchange](#).

**Direct:** A direct exchange will route messages to the queue with the binding key exactly matching the routing key.

**Fanout:** A fanout exchange will route messages to all queues bound to it.

**Topic:** A topic exchange supports multi-condition match and fuzzy match; that is, it will route messages to the queues bound to it by using routing key pattern match and string comparison.

**Header:** A header exchange has nothing to do with routing key and matches messages by the `Headers` attribute.

When binding a queue to a header exchange, declare a map key-value pair to implement the binding. When a message is sent to RabbitMQ, the `Headers` attribute of the message will be obtained to match the key-value pair specified during exchange binding, and the message will be routed to the queue only if there is a full match.

Durable: If this option is set to `true` , the exchange will still exist after the service is restarted; if it is set to `false` , the exchange will disappear after the service is restarted and needs to be created again.

AutoDelete: If this option is set to `true` , when the last queue bound to the exchange is deleted, the exchange will be deleted automatically.

Internal: If this option is set to `true` , the exchange cannot be directly used by producers but can only be bound to other exchanges.

Exchange Description: Enter the exchange description of up to 128 characters.

Add Alternate Exchange: This parameter is optional, and no alternate exchange is used by default. Messages that are sent to the primary exchange but cannot be routed will be sent to the alternate exchange specified here.

4. Click **Submit**, and you can see the created exchange in the exchange list.

## Editing an exchange

1. In the exchange list, click **Edit** in the **Operation** column of the target exchange.
2. In the pop-up window, you can edit the exchange description.
3. Click **Submit**.

## Deleting an exchange

1. In the exchange list, click **Delete** in the **Operation** column of the target exchange.
2. In the pop-up window, click **Delete**.

### Note:

After an exchange is deleted, all the configurations under it will be cleared and cannot be recovered.

# Queue Management

Last updated : 2024-06-26 15:56:25

## Overview

A queue is used to store messages. Each message will be put into one or more queues. Producers produce messages and deliver them to queues, and consumers pull messages from queues for consumption.

Multiple consumers can subscribe to the same queue. In this case, messages in the queue will be evenly distributed to such consumers for processing, rather than making each consumer receive and process all the messages.

This document describes how to create, delete, and query a queue in the TDMQ for RabbitMQ console.

## Prerequisites

You have created a vhost as instructed in [Vhost Management > Creating a vhost](#).

## Directions

### Creating a queue

#### Note:

The creation of regular queues or quorum queues is supported. You can click the following tabs to view the creation methods of different types of queues.

Regular queue

Quorum queue

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, select **Cluster > Queue** tab, choose a Vhost, then click **Create** to enter the Create Queue page.
3. Enter the basic information of the queue.

### Create Queue ✕

1 Basic Info >

2 Common Parameters >

3 Other Advanced Options

Current Vhost

(AMQP default vhost)

Queue Name \*

Please enter the name

This field is required. Please enter 1-64 letters, digits, or symbols (".", "-", or "\_").

Type

Regular queue ▾

Durable

☒

Node

rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc. ▾

AutoDelete

☐

The queue will be immediately deleted after the last consumer unsubscribes from it.

Queue Description

Please enter the description

Up to 128 characters

Next

Close

Queue Name: Enter the queue name, which cannot be modified after creation and can contain 3-64 letters, digits, hyphens, and underscores.

Type: Regular queue.

Durable: Set whether the queue performs persistence.

Node: Select the node where the queue resides.

AutoDelete: After this feature is enabled, the queue will be deleted immediately after its last consumer unsubscribes from it.

Queue Remarks: Enter the queue remarks of up to 128 characters.

4. Click **Next** to configure common parameters.

### Create Queue

✓ Basic Info

2 Common Parameters

3 Other Advanced Options

Message TTL

ms

Messages in the queue will be discarded or delivered to the dead letter exchange after the specified time period elapses.

Auto Expire

x ms

The queue will be deleted if it is not used or accessed in the specified time period.

Max Length

The maximum number of messages in a queue

Max Length Bytes

Byte

The maximum message storage capacity of a queue. When the limit is reached, messages will be processed in the method specified in "Overflow Behavior" field.

Overflow Behavior

Please select

▼

Dead Letter Exchange

(AMQP default exchange)

▼

A message will be delivered to the dead letter exchange if it is not acknowledged within the TTL.

Previous

Next

Close

**Message TTL:** Messages in the queue will be discarded/sent to the dead letter exchange after the specified time elapses.

**Auto expire:** The queue will be deleted if it is not used (accessed) within the specified time.

**Max length:** The maximum number of messages that the queue can contain.

**Max length bytes:** The upper limit of the queue's storage capacity. If it is reached, messages will be processed according to `Overflow behaviour`.

**Overflow behaviour:** When the queue capacity reaches the upper limit, the message at the beginning of the queue will be discarded.

Dead Letter Exchange: A message will be delivered to the dead letter exchange if it is not acknowledged within the TTL.

5. Click **Next** to configure other advanced options.

**Create Queue** ✕

**Basic Info** > **Common Parameters** > **3 Other Advanced Options**

**Single Active Consumer** ☐

If this option is enabled, ensure that only one consumer consumes messages from the queue at a time.

**Maximum Priority**

This field specifies the highest priority value of messages in this queue.

**Lazy Mode** ☐

If this option is enabled, the queue will preferably store pushed messages on the disk to reduce the memory usage.

**Master Locator**

This field specifies at which node the master resides if the mirrored queue is configured.

[Previous](#) [Submit](#) [Close](#)

Single active consumer: If this option enabled, it is necessary to ensure that there must be only one consumer consuming from the queue at a time.

Maximum priority: Configure the maximum priority of messages in this queue.

Lazy mode: If this option is enabled, the queue will preferably store pushed messages on the disk to reduce the memory usage.

Master locator: Configure the allocation method of the node where the master is located if the mirrored queue is configured.

min-masters: The node hosting the minimum number of masters will be selected as the node where the master of the current queue is located if the mirrored queue is configured.

client-local: The node the client that declares the queue is connected to will be selected as the node where the master of the current queue is located if the mirrored queue is configured.

random: A random node will be selected as the node where the master of the current queue is located if the mirrored queue is configured.

6. Click **Submit**.

1. Log in to the [RabbitMQ console](#).

2. In the left navigation bar, Select **Cluster > \*\*\*\*Queue** tab, choose a vhost, and click **Create** to enter the **Create Queue** page.

3. Enter the basic information of the queue.

**Create Queue** [X]

1 Basic Info > 2 Common Parameters > 3 Other Advanced Options

Current Vhost (AMQP default vhost)

Queue Name \*   
This field is required. Please enter 1-64 letters, digits, or symbols (".", "-", or "\_").

Type

Node

Queue Description   
Up to 128 characters

Next Close

Queue Name: Enter the queue name, which cannot be modified after creation and can contain 3–64 letters, digits, hyphens, and underscores.

Type: Quorum queue.



Node: Select the node where the queue resides.

Queue Remarks: Enter the queue remarks of up to 128 characters.

4. Click **Next** to configure common parameters.

### Create Queue

✓ Basic Info

2 Common Parameters

3 Other Advanced Options

Auto Expire

ms

The queue will be deleted if it is not used or accessed in the specified time period.

Max Length

The maximum number of messages in a queue

Max Length Bytes

Byte

The maximum message storage capacity of a queue. When the limit is reached, messages will be processed in the method specified in "Overflow Behavior" field.

Delivery Limit

This field specifies the number of retries allowed upon message delivery failure.

Overflow Behavior

Please select

Dead Letter Policy

at-most-once

Dead Letter Exchange

(AMQP default exchange)

A message will be delivered to the dead letter exchange if it is not acknowledged within the TTL.

Previous

Next

Close

Auto expire: The queue will be deleted if it is not used (accessed) within the specified time.

Max length: The maximum number of messages that the queue can contain.

Max length bytes: The upper limit of the queue's storage capacity. If it is reached, messages will be processed according to `Overflow behaviour`.

Overflow behaviour: When the queue capacity reaches the upper limit, the message at the beginning of the queue will be discarded.

Dead Letter Exchange: A message will be delivered to the dead letter exchange if it is not acknowledged within the TTL.

5. Click **Next** to configure other advanced options.

**Create Queue**

Basic Info > Common Parameters > **3 Other Advanced Options**

Single Active Consumer ☐

If this option is enabled, ensure that only one consumer consumes messages from the queue at a time.

Max In Memory Length

The maximum number of messages in a Quorum queue

Max In Memory Bytes  Byte

The maximum message storage capacity of a Quorum queue in bytes

Initial cluster size

Leader locator

[Previous](#) [Submit](#) [Close](#)

Single active consumer: If this option enabled, it is necessary to ensure that there must be only one consumer consuming from the queue at a time.

Max in memory length: The maximum number of messages in the quorum queue memory.

Max in memory bytes: The maximum total message size (in bytes) in the quorum queue.

6. Click **Submit**.

## Viewing queue details

In the **Queue** list, click the ID of the target queue to view its details.

You can view:

Basic Info: Queue type, online consumer, dead letter exchange, and AutoDelete status. Click **More Advanced Options** to view all parameter settings of the queue.

Consumer list: Information of consumers subscribed to this queue.

## Viewing binding

Binding relationship: The routing relationships bound to this Queue.

The screenshot displays the Tencent Cloud TDMQ console interface. On the left, a 'Queue' list is shown with columns for Queue Name, Type, and Online Consumers. A 'Create (3/1000)' button is at the top. The main panel on the right shows 'Queue Details' for a specific queue, including its name, type (Regular queue), node, creation time, and online consumer count. Below this, the 'Consumer List' is displayed with columns for Client IP and Consumer Tag. The list shows three consumers, all with the tag 'tdmq\_trace\_handler'. A search bar and pagination controls are also visible.

| Queue Name | Type       | Online C... |
|------------|------------|-------------|
| [Redacted] | D          | 3           |
| [Redacted] | [Redacted] | 0           |
| [Redacted] | D          | 3           |

Total items: 3

### Queue Details

Queue Name: [Redacted] Message TTL: -ms  
Queue Type: Regular queue Overflow Behavior: -  
Node: rabbit@rabbitmq-broker-2.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local Policy: [View Applied Policies](#)  
Creation Time: 2024-06-14 16:08:38 Auto Expire: -  
Online Consumer: 3 Dead Letter Policy: -  
[View More Advanced Options >](#)

### Consumer List

Binding Relationship

Enter a keyword

| Client IP  | Consumer Tag       |
|------------|--------------------|
| [Redacted] | tdmq_trace_handler |
| [Redacted] | tdmq_trace_handler |
| [Redacted] | tdmq_trace_handler |

Total items: 3 20 / page 1 / 1 page

## Editing a queue

1. In the queue list, click **Edit** in the **Operation** column of the target queue.
2. In the pop-up window, edit the queue information.
3. Click **Submit**.

## Deleting a queue

1. In the queue list, click **Delete** in the **Operation** column of the target queue.
2. In the pop-up window, click **Delete**.

### Note:

After a queue is deleted, all the configurations under it will be cleared and cannot be recovered.

# Binding

Last updated : 2024-06-26 15:56:25

## Overview

This document describes how to establish or cancel a binding between an exchange and a queue in the TDMQ console.

## Prerequisites

You have created an exchange as instructed in [Exchange Management](#).

You have created a queue as instructed in [Queue Management](#).

## Directions

### Creating a binding

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, choose **Cluster Management > Vhost** , click on the target Vhost's ID, then click **Create** to enter the Basic Info page.
3. On the top of the page, select **Routing Relationship** tab, click **Create** , and set the Source Exchange, Binding Key, Binding Target Type and Binding Target.

### Create Binding

Current Vhost

(AMQP default vhost)

Source Exchange \*

amq.direct

Binding Key \*

It can only contain 1-255 letters, digits, and symbols (-\_@#\*).

Binding Target Type

Exchange

Queue

Binding Target \*

tdmq\_trace\_handle

Submit

Close

4. Click **Submit**.

## Unbinding

1. In the binding list, click **Unbind** in the **Operation** column of the target binding.
2. In the pop-up window, click **Delete**.

### Note:

Once deleted, the route will no longer be available and cannot be recovered.

# User and Permission Management

Last updated : 2024-06-26 15:56:25

## Glossary

User is the smallest unit for dividing permissions within a TDMQ for RabbitMQ cluster. You can grant users the permissions of configuration and read/write under different vhosts.

User password is an authentication method. You can add a username and password in a client to access TDMQ for RabbitMQ clusters for message production/consumption.

Permission refers to your operation permission for exchanges and queues under a vhost, including configuration and read/write. The configuration permission controls declaring and deleting exchanges and queues. The read/write permission control reading messages from queues, sending messages to exchanges, and binding queues to exchanges.

## Use Limits

There can be up to 20 users in a cluster.

## Use Cases

You need to securely use TDMQ for RabbitMQ to produce/consume messages.

You need to set production/consumption permissions of different vhosts for different users.

For example, your company has departments A and B, and department A's system produces transaction data and department B's system performs transaction data analysis and display. In line with the principle of least privilege, two users can be created to grant department A only the permission to produce messages to the transaction system vhost and grant department B only the permission to consume messages. This helps greatly avoid problems caused by unclear division of permissions, such as data disorder and dirty business data.

## Directions

### Adding a user

Every cluster has a user named "admin" by default. You can configure permissions for this default user or create new users as needed.

1. Log in to the [RabbitMQ console](#).
2. Select **Cluster** > **Cluster** on the left sidebar, select a region, and click the ID of the target cluster to enter the **Basic Info** page of the cluster.
3. Select the **User and Permission** tab at the top of the page and click **Create User** on the **User Management** tab.
4. On the **Create User** page, enter the username, password, and description:

Username: Cannot be empty, should not only contain (.), must be 1-64 characters long, and can only include letters, digits, periods (.), hyphens (-), and underscores (\_).

Password: Cannot be empty, must be 8-64 characters long, and must include at least two of the following: lowercase letters, uppercase letters, digits, special characters [()~!@#\$%^&\*\_{ }|:;,'.?/] .

Role: Select the user role.

| Role          | Permission Description   |
|---------------|--|
| none          | Unable to log in to the Web console, typically applies to normal producers and consumers.  |
| management    | Can log in to the Web console;<br>Can view Vhosts under their name, as well as queues, exchanges, and bindings within;<br>Can view and close channels and connections under their name.                  |
| policymaker   | On the basis of all Management rights:<br>Can view, modify, and delete policies and parameters of Vhosts under their name.   |
| monitoring    | On the basis of all Management rights:<br>Can view all Vhost, connection, and channel lists;<br>Can view node-related information (such as disk usage, memory usage, number of processes.).              |
| administrator | Super Administrator, on the basis of all rights in Policymaker and Monitoring:<br>Can create and delete Vhost;<br>Can view, create, and delete users and permissions;<br>Close other users' connections. |

Description (optional): Enter a user description.

5. Click **Submit**.

### Create User

Username \*

Please enter the name

It cannot be empty, and you cannot only enter ".". It should contain 1 to 64 characters, including letters, digits, ".", "-", and "\_".

Password \*

Please enter user password

This field is required and must contain 8-64 characters in at least two of the following types: lowercase letters, uppercase letters, digits, and symbols (()<sup>^</sup>~!@#%&\*\_{|}[]:;','./).

Please keep your password properly and remember it.

Confirm Password \*

Please enter the user password again

This field is required and must contain 8-64 characters in at least two of the following types: lowercase letters, uppercase letters, digits, and symbols (()<sup>^</sup>~!@#%&\*\_{|}[]:;','./).

Role

administrator

For permission description for different roles, see [Documentation](#).

Description

Please enter the description

Submit

Close

## Configuring a permission

1. On the **User and Permission** tab, select the **Permission List** tab and click **Configure Permission**.

2. On the **Configure Permission** page, select the target vhost and user and set permission rules.

Permission rules can match resources through **regex**. For example, if you select **Configuration** and enter "test.\*" in the input box, then the user will be granted the permission to configure all resources with a name starting with "test-" under the current vhost.



### Configure Permission

Vhost \*

(AMQP default vhost)

No vhost is available? Please go to the [Vhost](#) tab to create one.

Username \*

admin

Permission ⓘ

☐ Configuration

If this option is selected, the defai

☐ read

If this option is selected, the defai

☐ write

If this option is selected, the defai

For more permission type information, see [here](#)

Submit

Close

3. Click **Submit**.

4. Add the username and password to the client parameters. For directions on how to add the token parameters to the client code, see [Spring Boot Starter](#) (the parameters in this document are the username and password).

5. Check whether the permission is effective. You can run the configured client to access the exchange and queue resources in the vhost and produce/consume messages according to the configured permission. Check whether a no permission error is reported, and if not, the permission has been configured successfully.

## Deleting a permission

Before deleting a permission, make sure that the current business no longer uses the user to produce/consume messages; otherwise, a client exception may occur due to the failure to produce/consume messages.

1. In the **User and Permission** tab, click **Delete** in the **Operation** column of the target user.
2. In the pop-up window, click **OK**.

# Policy List

## Policy Management

Last updated : 2024-06-26 16:03:21

## Overview

In RabbitMQ message queue, besides forced attributes such as durable and Exclusive, when creating a Queue or an Exchange, you can configure some optional attributes to obtain different features, such as x-message-ttl, x-expires, and x-max-length.

However, once the attribute parameters set for a Queue or an Exchange via RabbitMQ client are successfully set, they cannot be changed unless the original Queue or Exchange is deleted and a new one is created.

A policy is a special use of runtime parameters that supports dynamically modifying some attribute parameters. Policies are aimed at the Vhost level, and a policy can match one or more Queues or Exchanges, facilitating batch management. This solves the problem that Exchanges and Queues created by RabbitMQ clients cannot be modified, and greatly enhances the flexibility of the application.

## Directions

### Creating a Policy

When you create a cluster, if mirrored queue is enabled, there will be a default policy under the **Policy** tab in the console, which can be deleted. You can create a new policy.

1. Log in to the [RabbitMQ Console](#).
2. In the left sidebar, choose **Cluster Management** > **Vhost**, choose the region, then click the target Vhost's ID to enter the Basic Info page.
3. Click the **Policy** tab at the top, click Create Policy, and fill in the Basic Settings of the policy.  
Policy Name: 1-64 characters, can only include digits, letters, hyphens (-), and underscores (\_).  
Match Mode: A regular expression used to match the relevant Queue or Exchange. For example, ^test.\* will match all Queues or Exchanges that start with test.  
Application Scope: Used to specify the current Policy's effective range.  
Exchanges and queues: Indicates that it applies to all Queues or Exchanges matching the Pattern.  
Exchanges: Indicates that it applies to all Exchanges matching the Pattern.

Queues: Indicates that it applies to all Queues matching the Pattern.

Priority: Defines the priority of the Policy. If multiple policies apply to the same Queue or Exchange, the Policy with the highest priority number will be effective.

**Create policy**

1 Basic Settings > 2 Policy Configuration

Current Vhost (AMQP default vhost)

Policy Name \*   
This field is required and can contain 1-64 digits, letters, hyphens, and underscores.

Match Mode \*   
For match mode configuration rules, see [here](#).

Application Scope

Priority   
The greater the number, the higher the priority.

**Next** Cancel

4. Click Next, to set the Policy Configuration.

Mirror Mode: The mode of the mirror queue, valid values are all/exactly/nodes.

all: Indicates that mirror occurs on all nodes in the cluster.

exactly: Indicates that mirror occurs on a specified number of nodes, as specified by a mirror parameter.

nodes: Indicates that mirror occurs on specified nodes, with the node names specified by a mirror parameter.

Message Sync Method: The method of message synchronization in mirror queues, valid values are automatic and manual.

Primary Node Exit Processing: Whether to allow the election of an unsynchronized mirror as master when the primary node gracefully exits.

Primary Node Failure Processing: Whether to allow the election of an unsynchronized mirror as master in the event of a primary node failure. To ensure availability, it is recommended to Allow All Mirrors to be elected.

### Create policy

✓ Basic Settings

>

2 Policy Configuration

Mirror Mode

all

Mirror Parameter

Message Sync Method

automatic

Method of syncing messages in the mirrored queue

Master Node Exit Processing

☐ Allow all mirrors to be elected

☒ Only elect synced mirror

This specifies whether to allow electing an unsynced mirror the new master when the master node gracefully exits.

Master Node Failure Processing

☒ Allow all mirrors to be elected

☐ Only elect synced mirror

This specifies whether to allow electing an unsynced mirror the new master when the master node fails.  
To ensure the availability, we recommend that you select "Allow all mirrors to be elected".

Previous

Complete

5. Click Complete to finish creating the policy.

## Editing a Policy

1. In the policy list, click **Edit** in the Operation column of the target policy.
2. In the pop-up window, edit the policy information.
3. Click **Submit** to complete the modification.

## Deleting a Policy

1. In the policy list, find the policy you want to delete, and click **Delete** in the Operation column.

---

2. In the pop-up prompt box, click **Delete** to complete the deletion.

# Default Image Policy

Last updated : 2024-06-26 16:04:16

## Policy Description

To enhance the reliability and fault tolerance of the RabbitMQ message queue cluster, users are offered the option to enable the Default Image Queue when they create a RabbitMQ cluster or a Vhost, provided the cluster has at least three nodes. This image queue can duplicate messages across multiple nodes within the RabbitMQ cluster, ensuring that messages in the queue are not lost in the event of a node failure.

Below is a detailed description of the default image queue policy parameters provided by RabbitMQ for users:

| Parameter Name        | Configuration Parameters | Parameter Description  |
|-----------------------|--------------------------|--|
| Name                  | pay-mirror-policy        | Policy name, used to identify and reference the policy.  |
| Pattern               | *                        | The match mode of the policy, which uses regular expression syntax. The dot (.) indicates matching any character, and asterisk (*) indicates matching the preceding character zero or more times, so * indicates a queue that matches any name.  |
| Apply to              | Queues                   | The application object of the policy, set to Queues to indicate that the policy applies to queues.   |
| Priority              | 0                        | The priority of the policy. If a queue matches multiple policies, then the policy with the highest priority will be applied. 0 indicates the lowest priority.  |
| ha-mode               | exactly                  | The copy mode of the image queues.<br>exactly: The messages of the queue will be copied to a specified number of nodes<br>all: The messages of the queue will be copied to all nodes.<br>Choosing exactly allows for reduced network and storage overhead while ensuring availability, thus improving performance. |
| ha-params             | 3                        | The copy parameters of image queues. When ha-mode is set to exactly, the number of copied nodes must be specified here. The default setting is 3, which can still maintain a good performance level even if expanded to 5 nodes in the future.   |
| ha-promote-on-failure | always                   | The promotion policy for image queues in the event of a node failure. always means that regardless of the cause of the node  |

|                        |             |   |
|------------------------|-------------|---|
|                        |             | failure, the image queue will be promoted to the main queue. when-synced means that the image queue will only be promoted to the main queue once it has resynchronized after the node failure. The default setting is always, to ensure service availability under any failure conditions.  |
| ha-promote-on-shutdown | when-synced | <p>The promotion policy for image queues when a node is shut down normally.</p> <p>always: Regardless of the reason for the node shutdown, the image queue will be promoted to the main queue.</p> <p>when-synced: The image queue will only be promoted to the main queue once it has resynchronized after the node shutdown.</p> <p>The default setting is when-synced, to avoid unnecessary promotion operations.</p>                            |
| ha-sync-mode           | manual      | <p>Synchronization mode of image queues.</p> <p>automatic: Indicates that the image queue will automatically sync with the main queue when a node is enabled or reconnected to the cluster.</p> <p>manual: Indicates that the synchronization operation needs to be manually triggered to sync the image queue with the main queue.</p> <p>The default setting is manual, to avoid auto-sync impacting cluster performance during message heap.</p> |

By configuring the default image queue policy, it is possible to optimize performance and resource utilization while ensuring the reliability of the RabbitMQ cluster. Users can further adjust these parameters according to their own requirements and scenario, or delete and create a new policy.

## Directions

### Enabling Default Image Queue

1. To enable image queues when you create a cluster: Purchase page for cluster > **Other Configurations** > **Enable Mirrored Queue**.
- 2.

**Other Configurations**

Cluster Name

Please enter cluster name.

The cluster name is required and can only contain 3-64 digits, letters, hyphens (-) and underscores (\_).

Mirrored Queue

☒ Enable mirrored queue

We recommend that you enable mirrored queue to ensure availability.  
When this feature is enabled, a default policy is automatically created in Instance Details-Policy List, applicable only to the default Vhost. You can delete it or customize a new policy to override the default pc

Tag①

Tag Key

Tag Value

Delete

+ Add

Paste

3. To enable image queues when you create a Vhost: **Cluster List** > **Vhost** > **Create Vhost** > **Enable Image Queue**.

**Create Vhost**

Vhost Name \*

Please enter the vhost name

It should contain 1 to 64 characters, including letters, digits, ".", "-", and "\_".

Trace Plugin ⓘ

☐

Image Queue

☒ Enable Image Queue

It is recommended to enable the Image Queue to ensure availability.  
After enabling, a default policy [🔗](#) will be generated under Instance Details - Policy List. This policy only applies to the default Vhost /, and can be deleted or overridden by a custom new policy.

Vhost Description

Please enter the description

Submit

Close

## Limit



RabbitMQ Message Queue only allows clusters with **three or more** nodes to enable the Default Image Queue, mainly to ensure the high availability and fault tolerance of the cluster. In a cluster with three or more nodes, image queues can copy messages across multiple nodes. This not only shares the load across each node, improving performance, but also ensures the normal operation of the service in case of a node failure. At the same time, it provides greater flexibility, allowing us to configure image queue parameters according to actual needs. Therefore, this limit is in place to provide a more stable and reliable service.

# Smart Inspection

Last updated : 2024-06-26 15:56:25

## Overview

As monitoring metrics increase, there are higher requirements on understanding Ops metrics. TDMQ for RabbitMQ provides smart inspection, which can actively troubleshoot cluster issues and potential risks, provide solution based on expert experience, and automatically summarize health check results into reports.

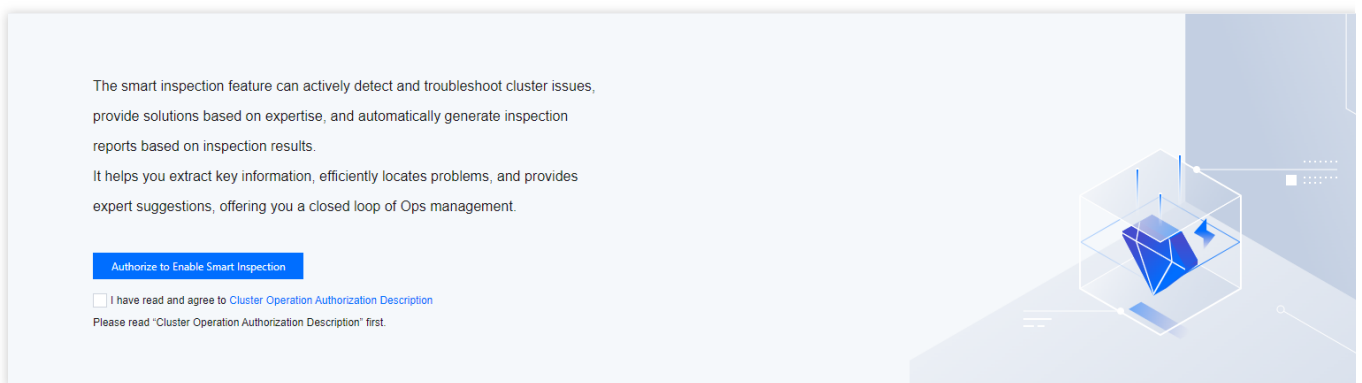
Smart inspection helps you extract key information, efficiently locate issues, offer professional suggestions, and close Ops experience issues.

## Enabling Smart Inspection

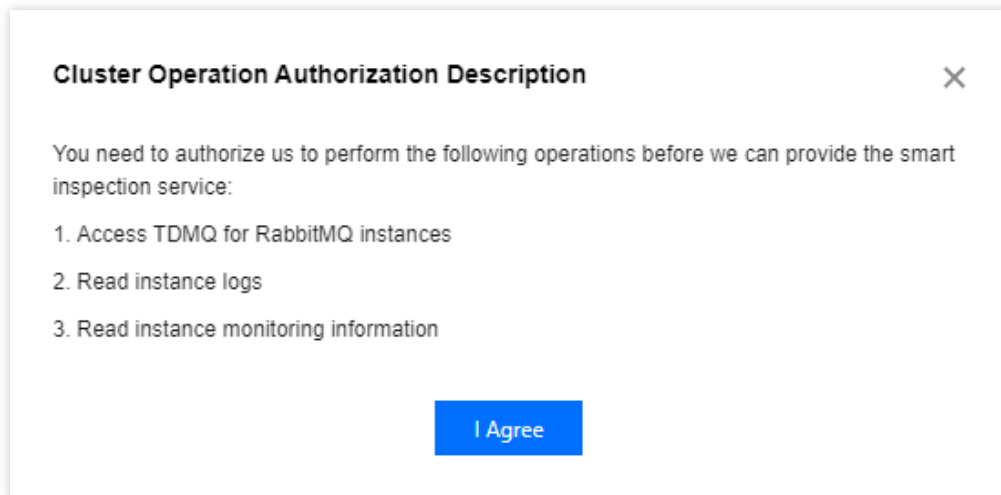
Smart inspection needs to access basic information, logs, and monitoring data of your cluster. Therefore, you need to authorize access when the smart inspection service is enabled for the first time.

### Directions

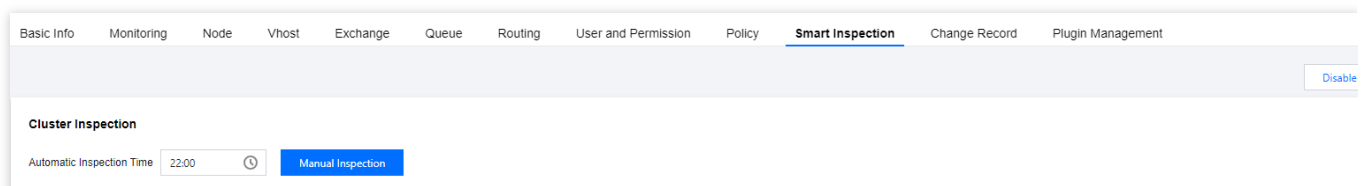
1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, select **Cluster** > **Cluster**, select a region, and click the ID of the target cluster to enter the cluster detail page.
3. At the top of the cluster detail page, select the **Smart Inspection** tab to enter the smart inspection page.



4. On the initialization page, you can select *Cluster Operation Authorization Description* and read related explanations and prompts.



5. After confirming it, check the box **I have read and agree to Cluster Operation Authorization Description**. Then, click **Authorize to Enable Smart Inspection** to enable smart inspection.



6. Click **Manual Inspection** to start inspection. You can set **Automatic Inspection Time** to a business off-peak period. In this case, TDMQ for RabbitMQ conducts smart inspection at the designated time every day.

## Viewing Inspection Results

The inspection results summarize and display the inspection results and change trends of the cluster for you to view the recent health status of the cluster. The health status of a cluster includes high risk, low risk, and secure.

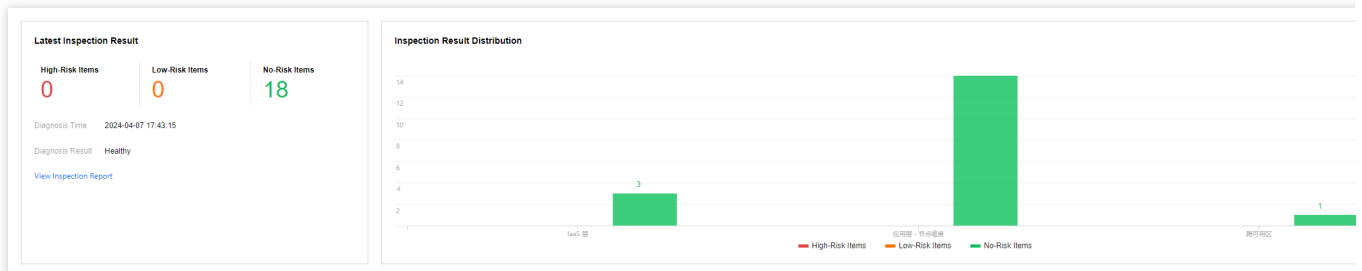
High risk: The cluster has encountered severe issues or hidden dangers, which affect cluster availability and require immediate attention to prevent data loss, cluster failure, or other issues.

Low risk: The cluster has significant issues or hidden dangers that might affect cluster availability. It is recommended that you address these issues as soon as possible.

Secure: The cluster is healthy.

### Directions

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, choose **RabbitMQ > Cluster**, select a region, and click the ID of the target cluster to enter the Cluster detail page.
3. At the top of the Cluster detail page, select the **Smart Inspection** tab to enter the smart inspection page.
4. On the smart inspection result page, the latest inspection results and result distribution of the cluster are displayed.



5. Click **View Inspection Report** to view the detailed inspection report, including the inspection time, resource ID, inspection conclusion, inspection items, and detailed inspection results.

6. Click the **Download** icon in the upper left corner of the report to download the inspection report.

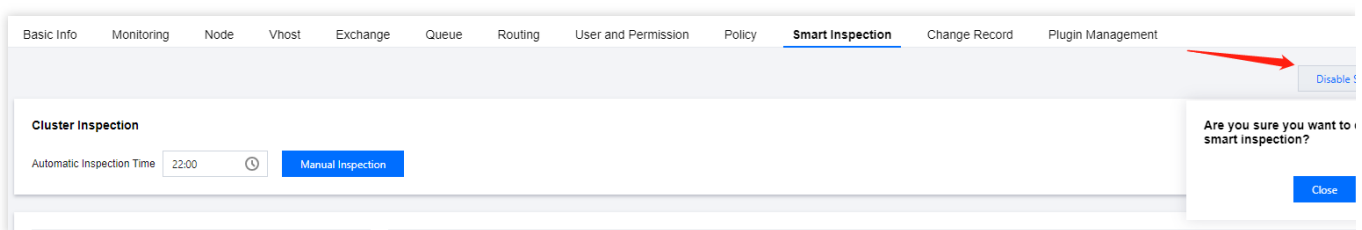
7. Click the directory on the right to view and download inspection reports in the latest 30 days.

## Disabling Smart Inspection

When you no longer need the smart inspection feature, you can disable the service. After it is disabled, the system no longer performs scheduled cluster inspection or generates new inspection reports.

### Directions

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, select **RabbitMQ > Cluster**, select a region, and click the ID of the target cluster to enter the cluster detail page.
3. At the top of the cluster detail page, select the **Smart Inspection** tab to enter the smart inspection page, and click **Disable Smart Inspection**.



4. Click **Close** to disable the smart inspection service.

# Change Records







Last updated : 2024-06-26 16:06:00

Change records in TDMQ for RabbitMQ manage, store, analyze, and display the change event data uniformly, making it easier for you to view and analyze. You can view details within the Change Record module.

This document explains how to view the details of change records on the TDMQ for RabbitMQ console.

## Viewing Change Records

1. Log in to the [RabbitMQ Console](#).
2. In the left sidebar, choose **cluster**, choose the appropriate region, then click the ID of the cluster you want to view, and enter the Cluster Details page.
3. On the top of the Cluster Details page, select the **Change Record** tab to enter the Change Record page.
4. Set the time period (supports Last 7 Days, Last 30 Days, and Custom Time Range), and you can view change events within the corresponding time period.

| Basic Info  | Monitoring          | Node                 | Vhost              | User and Permission          | Smart Inspection | Change Record    | Plugin Management |
|---|---------------------|----------------------|--------------------|------------------------------|------------------|------------------|-------------------|
| <div>Last 7 daysLast 30 days2024-06-07 17:34:56 ~ 2024-06-14 17:34:56📅</div>        |                     |                      |                    |                              |                  |                  |                   |
| Cluster Name  | Time                | Name                 | Event              | Operation                    |                  |                  |                   |
|  | 2024-06-14 16:07:51 | -                    | permission.created | <a href="#">View Details</a> |                  |                  |                   |
|  | 2024-06-14 16:07:51 | -                    | permission.created | <a href="#">View Details</a> |                  |                  |                   |
|  | 2024-06-14 16:07:51 | default-trace-policy | policy.set         | <a href="#">View Details</a> |                  |                  |                   |
|  | 2024-06-14 16:08:38 | tdmq_trace_handle    | queue.created      | <a href="#">View Details</a> |                  |                  |                   |
|  | 2024-06-14 16:08:38 | -                    | binding.created    | <a href="#">View Details</a> |                  |                  |                   |
|  | 2024-06-14 16:08:38 | -                    | binding.created    | <a href="#">View Details</a> |                  |                  |                   |
| Total items: 6  |                     |                      |                    |                              | 20 / page        | <div>1 / 1</div> |                   |

5. Click the **View Details** in the Operation Column, and you can view the event details on the right sidebar.

Cluster / [redacted]

Basic InfoMonitoringNodeVhostUser and Permissions

Last 7 daysLast 30 days2024-06-07 17:34:56 ~ 2024-06-14 17:34:56

| Cluster Name | Time                |
|--------------|---------------------|
| [redacted]   | 2024-06-14 16:07:51 |
| [redacted]   | 2024-06-14 16:07:51 |
| [redacted]   | 2024-06-14 16:07:51 |
| [redacted]   | 2024-06-14 16:08:38 |
| [redacted]   | 2024-06-14 16:08:38 |
| [redacted]   | 2024-06-14 16:08:38 |

Total items: 6

Event Details

Details

Cluster Name

[redacted]

Node Name

rabbit@rabbitmq-broker-2.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local

Time

2024-06-14 16:07:51

Name

-

Event

permission.created

User

admin

Message Body

Headers

{vhost=/, timestamp\_in\_ms=1718352471705, read=., configure=., user\_who\_performed\_action=admin, write=., user=admin}

# Monitoring and Alarms

Last updated : 2024-06-26 15:56:25

## Overview

A TDMQ for RabbitMQ exclusive cluster can monitor resources created under your account, including clusters and nodes. You can analyze the cluster usage based on the monitoring data and handle possible risks promptly. You can also set alarm rules for monitoring metrics to receive alarm messages when metrics are abnormal. This helps you deal with risks in time and ensure the stable operations of your system.

## Monitoring Metrics

You can check monitoring metrics of a TDMQ for RabbitMQ exclusive cluster from four dimensions: cluster, node, vhost, and queue. The following table describes supported monitoring metrics.

- Cluster
- Node
- Vhost
- Queue

| Type              | Monitoring Metric                                | Unit    | Description   |
|-------------------|--|---------|---|
| Basic Information | Connection Count                                 | Count   | Number of opened connections  |
|                   | Channel Count                                    | Count   | Number of opened channels   |
|                   | Queue Count                                      | Count   | Total number of available queues  |
|                   | Consumer Count                                   | Count   | Total number of online consumers  |
|                   | Message Backlog                                  | Count   | Total number of messages in the Ready state (accumulated but not delivered) |
|                   | Inbound Public Network Bandwidth                 | Mbps    | Inbound public network bandwidth  |
|                   | Outbound Public Network Bandwidth                | Mbps    | Outbound public network bandwidth   |
|                   | Recommended TPS Limit (Production + Consumption) | Count/s | Recommended TPS limit for the cluster when the mirror queue is not enabled  |

|                            |                                 |         |   |
|----------------------------|---------------------------------|---------|---|
|                            | Total Opened Channels           | Count   | Total number of opened channels   |
| Production and Consumption | Production Confirmation Rate    | Count/s | Rate that the broker confirms to return a packet after the client successfully generates a message      |
|                            | Messages Produced per Second    | Count/s | Rate that the client generates messages   |
|                            | Unacknowledged Message Count    | Count   | Total number of messages that are delivered to but are not acknowledged by the consumer                 |
|                            | Consumption Acknowledgment Rate | Count/s | Rate that the consumer acknowledges messages  |
|                            | Messages Consumed per Second    | Count/s | Number of messages consumed per second, including scenarios in which autoAck = false and autoAck = true |
|                            | Redelivery Rate                 | Count/s | Rate for redelivering messages to the consumer in a channel   |
|                            | Message Discard Rate            | Count/s | Rate for dropping messages that are sent to an exchange with mandatory = false due to no matching route |
|                            | Delayed Message Count           | Count   | Delayed Message Count in the current cluster.   |

| Type              | Monitoring Metric | Unit  | Description   |
|-------------------|-------------------|-------|---|
| Basic Information | Connection Count  | Count | Number of open connections  |
|                   | Channel Count     | Count | Number of opened channels   |
|                   | Queue Count       | Count | Total number of available queues  |
|                   | Consumer Count    | Count | Total number of online consumers  |
|                   | Message Backlog   | Count | Total number of messages in the Ready state (accumulated but not delivered) |
|                   | CPU Usage         | %     | CPU usage of a node   |
|                   | Memory Usage      | %     | Memory usage of a node  |
|                   | Disk Usage        | %     | Disk usage of a node  |
|                   |                   |       |   |



|                            |                                 |         |   |
|----------------------------|---------------------------------|---------|---|
| Production and Consumption | Production Confirmation Rate    | Count/s | Rate that the broker confirms to return a packet after the client successfully generates a message      |
|                            | Messages Produced per Second    | Count/s | Rate that the client generates messages   |
|                            | Unacknowledged Message Count    | Count   | Total number of messages that are delivered to but are not acknowledged by the consumer                 |
|                            | Consumption Acknowledgment Rate | Count/s | Rate that the consumer acknowledges messages  |
|                            | Messages Consumed per Second    | Count/s | Number of messages consumed per second, including scenarios in which autoAck = false and autoAck = true |
|                            | Redelivery Rate                 | Count/s | Rate for redelivering messages to the consumer in a channel   |
|                            | Message Discard Rate            | Count/s | Rate for dropping messages that are sent to an exchange with mandatory = false due to no matching route |

| Type                       | Monitoring Metric            | Unit    | Description  |
|----------------------------|------------------------------|---------|--|
| Basic Information          | Consumer Count               | Count   | Total number of online consumers   |
|                            | Message Backlog              | Count   | Total number of messages in the Ready state (accumulated but not delivered)<br>++                  |
| Production and Consumption | Production Confirmation Rate | Count/s | Rate that the broker confirms to return a packet after the client successfully generates a message |
|                            | Messages Produced per Second | Count/s | Rate that the client generates messages  |
|                            | Unacknowledged Message Count | Count   | Total number of messages that are delivered to but are not acknowledged by the consumer            |
|                            | Consumption Acknowledgment   | Count/s | Rate that the consumer acknowledges messages   |




|  |                              |         |   |
|--|------------------------------|---------|---|
|  | Rate                         |         |   |
|  | Messages Consumed per Second | Count/s | Number of messages consumed per second, including scenarios in which autoAck = false and autoAck = true |
|  | Redelivery Rate              | Count/s | Rate for redelivering messages to the consumer in a channel   |
|  | Message Discard Rate         | Count/s | Rate for dropping messages that are sent to an exchange with mandatory = false due to no matching route |

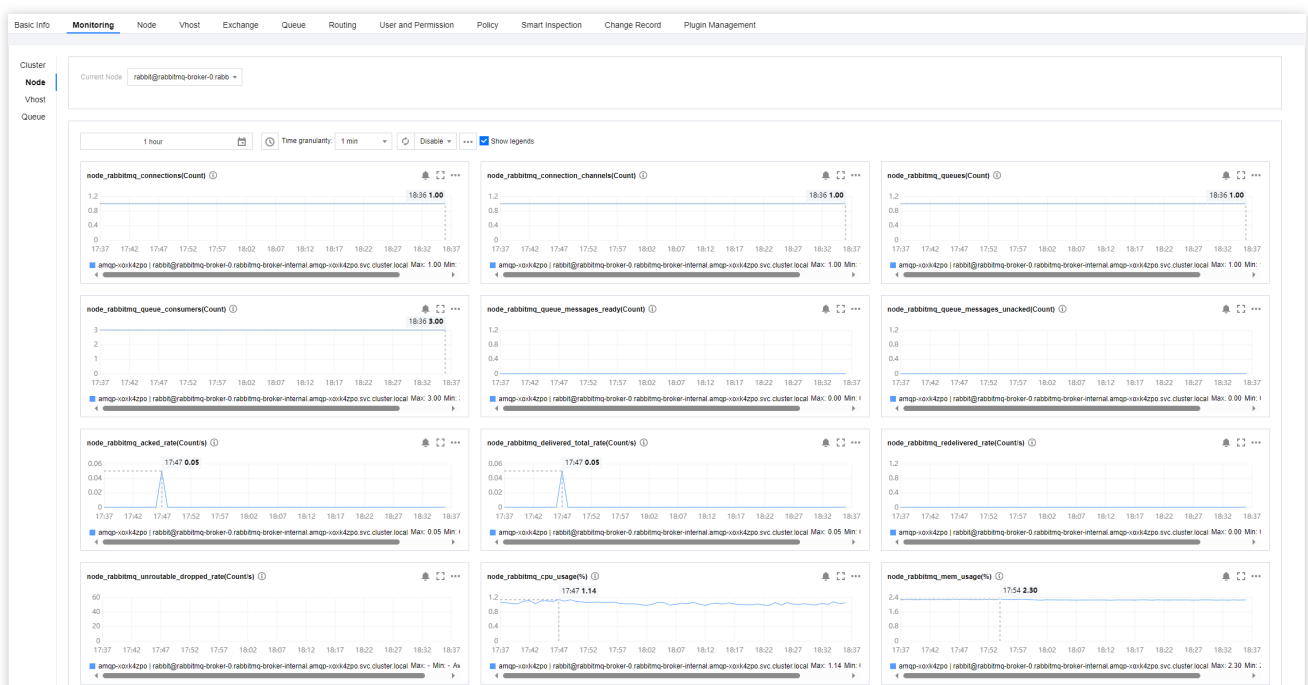
| Type                       | Monitoring Metric               | Unit    | Description   |
|----------------------------|---------------------------------|---------|---|
| Basic Information          | Consumer Count                  | Count   | Total number of online consumers  |
|                            | Message Backlog                 | Count   | Total number of messages in the Ready state (accumulated but not delivered)             |
| Production and Consumption | Unacknowledged Message Count    | Count   | Total number of messages that are delivered to but are not acknowledged by the consumer |
|                            | Consumption Acknowledgment Rate | Count/s | Rate that the consumer acknowledges messages  |
|                            | Redelivery Rate                 | Count/s | Rate for redelivering messages to the consumer in a channel                             |

## Viewing Monitoring Data

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, select **Cluster > Cluster**, select a region, and click the ID of the target cluster to enter the cluster details page.
3. At the top of the cluster details page, select the **Monitoring** tab to enter the monitoring page.
4. Select the **Resource** tab, select the resource you want to view, and set the time range to view monitoring data.

| Icon | Note |
|------|------|
|      |      |

|   |  |
|---|--|
|  | Click it to view the monitoring metrics on a YoY basis. YoY, MoM, and custom date are supported.   |
|  | Click it to refresh and obtain the latest monitoring data. Monitoring data can be refreshed at intervals of 30 seconds, 5 minutes, 30 minutes, and 1 hour. |
|  | Click it copy the chart to the dashboard. For more information about the dashboard, see <a href="#">Dashboard</a> .  |
| <input checked="" type="checkbox"/> Show legends                                  | After it is selected, legend information can be displayed on the chart.  |



## Configuring Alarm Rules

### Creating an Alarm Rule

You can configure alarm rules for monitoring metrics. When a monitoring metric reaches the set alarm threshold, Cloud Monitor will notify you of exceptions in time via email or SMS.

1. On the [Monitoring](#) page of the cluster, click the alarm icon below to enter the [CM console](#) and configure an alarm policy.
2. On the alarm configuration page, select a policy type and instance, and set the alarm rule and notification template.

**Policy Type:** Select **TDMQ/RabbitMQ**.

**Alarm Object:** Select the RabbitMQ resource for which to configure the alarm policy.

Trigger Condition: You can select **Select template** or **Configure manually**. The latter is selected by default. For more information on manual configuration, see the description below. For more information on how to create a template, see [Creating a trigger condition template](#).

**Note:**

Metric: For example, if you select 1 minute as the statistical period for the "connections" metric, then if the average production duration exceeds the threshold for N consecutive data points, an alarm will be triggered.

Alarm Frequency: For example, "Alarm once every 30 minutes" means that there will be only one alarm triggered every 30 minutes if a metric exceeds the threshold in several consecutive statistical periods. Another alarm will be triggered only if the metric exceeds the threshold again in the next 30 minutes.

**Notification Template:** You can select an existing notification template or create one to set the alarm recipient objects and receiving channels.

3. Click **Complete**.

**Note:**

For more information on alarms, see [Creating Alarm Policy](#).

## Creating a trigger condition template

1. Log in to the [CM console](#).
2. On the left sidebar, click **Trigger Condition Template** to enter the **Template** list page.
3. Click **Create** on the **Trigger Condition Template** page.
4. On the template creation page, configure the policy type.

**Policy Type:** Select **TDMQ/RabbitMQ**.

**Use preset trigger condition:** Select this option and the system recommended alarm policy will be displayed.

5. After it is confirmed that everything is correct, click **Save**.
6. Return to alarm policy creation page and click **Refresh**. The alarm policy template just configured will be displayed.

## Alarm Configuration Suggestions

This section describes some key metrics and their alarm configuration suggestions while using TDMQ for RabbitMQ.

| Metric         | Dimension | Suggested Alarm Configuration  | Description   |
|----------------|-----------|--|---|
| Disk Usage (%) | Node      | Statistical period of 1 minute, > 80%, and for 5 consecutive data points.<br>Alarm once every 30 minutes | High disk usage may result in insufficient disk space on the node to accommodate messages assigned to it. As a result, messages cannot be written to the disk. You are advised to promptly clear data or scale out the cluster when the average disk usage exceeds 80%. |
|                |           |  |   |

|                          |      |  |  |
|--------------------------|------|--|--|
| Memory Usage (%)         | Node | Statistical period of 1 minute, > 50%, and for 5 consecutive data points. Alarm once every 30 minutes  | High memory usage blocks message production. You are advised to accelerate consumption, apply flow control to production, or scale out the cluster when the memory usage exceeds 50%.  |
| CPU Usage (%)            | Node | Statistical period of 1 minute, > 70%, and for 5 consecutive data points. Alarm once every 30 minutes  | High CPU usage affects the message production speed. You are advised to scale out the cluster when the CPU usage exceeds 70%.  |
| Message backlog (Count)  | Node | Statistical period of 5 minutes, > expected message backlog for the business, and for 5 consecutive data points. Alarm once every 30 minutes   | An excessive accumulation of messages causes a rapid disk usage increase of the broker node. As a result, other messages cannot be received, and scale-out is needed.  |
| Node Survival Status (%) | Node | The monitoring configuration item of granularity is collected every one minute; the value of the node survival status is 1; lasting for 3 data points will trigger an alarm, with an alarm triggered once every 15 minutes | Abnormal Node Survival Status or Downtime can lead to message loss, especially when persistence or image queues are not enabled. It also increases the load on the remaining nodes, causing a decline in cluster performance. It is recommended to check the cause in conjunction with other indicators and alarm information. |

# Plugin Management

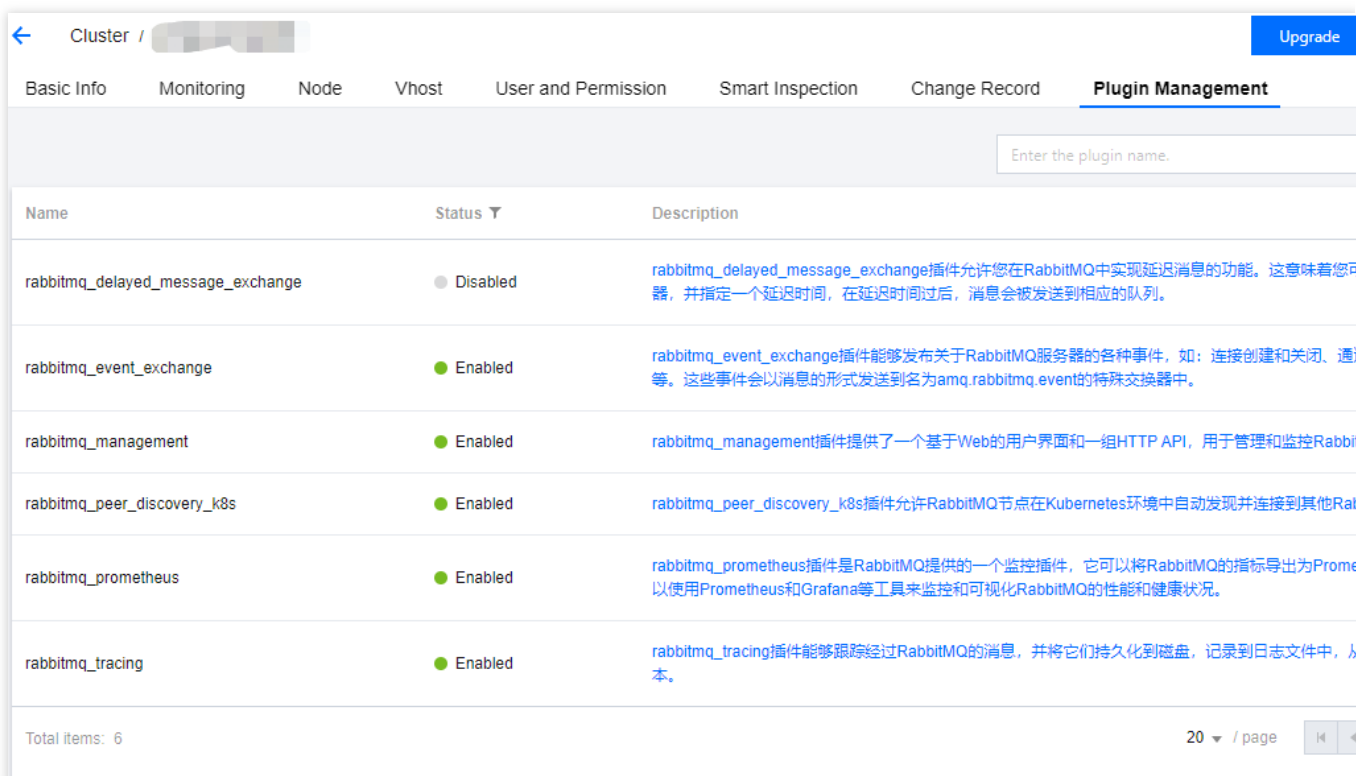
Last updated : 2024-06-26 15:56:25

The **Plugin Management** module allows you to view and manage plugins supported by a TDMQ for RabbitMQ cluster.

This document describes how to view supported plugins in the TDMQ for RabbitMQ console.

## Viewing Plug-ins

1. Log in to the [RabbitMQ console](#).
2. In the left sidebar, select **Cluster**, select an appropriate region, and click the ID of the target cluster to enter the cluster detail page.
3. At the top of the cluster detail page, select the **Plugin Management** tab to enter the plugin management page.



| Cluster / [Cluster ID]            |                  |  | Upgrade           |
|-----------------------------------|------------------|--|-------------------|
| Basic Info                        | Monitoring       | Node   | Vhost             |
| User and Permission               | Smart Inspection | Change Record  | Plugin Management |
| Enter the plugin name.            |                  |  |                   |
| Name                              | Status ▼         | Description  |                   |
| rabbitmq_delayed_message_exchange | Disabled         | rabbitmq_delayed_message_exchange插件允许您在RabbitMQ中实现延迟消息的功能。这意味着您可以在指定一个延迟时间后，将消息发送到相应的队列。                                 |                   |
| rabbitmq_event_exchange           | Enabled          | rabbitmq_event_exchange插件能够发布关于RabbitMQ服务器的各种事件，如：连接创建和关闭、通道创建和关闭，以及队列创建和删除。这些事件会以消息的形式发送到名为amq.rabbitmq.event的特殊交换器中。   |                   |
| rabbitmq_management               | Enabled          | rabbitmq_management插件提供了一个基于Web的用户界面和一组HTTP API，用于管理和监控RabbitMQ。   |                   |
| rabbitmq_peer_discovery_k8s       | Enabled          | rabbitmq_peer_discovery_k8s插件允许RabbitMQ节点在Kubernetes环境中自动发现并连接到其他RabbitMQ节点。   |                   |
| rabbitmq_prometheus               | Enabled          | rabbitmq_prometheus插件是RabbitMQ提供的一个监控插件，它可以将RabbitMQ的指标导出为Prometheus格式，以便使用Prometheus和Grafana等工具来监控和可视化RabbitMQ的性能和健康状况。 |                   |
| rabbitmq_tracing                  | Enabled          | rabbitmq_tracing插件能够跟踪经过RabbitMQ的消息，并将它们持久化到磁盘，记录到日志文件中，以便进行故障排查。  |                   |
| Total items: 6                    |                  |  | 20 / page         |

The following table describes plugins that are enabled by default after a TDMQ for RabbitMQ cluster is created.

| Plugin                  | Description  |
|-------------------------|--|
| rabbitmq_event_exchange | The rabbitmq_event_exchange plugin can publish various events about the RabbitMQ server, such as connection creation and closure, channel creation and closure, and queue creation and deletion. These events are sent to a special exchange named <b>amq.rabbitmq.event</b> using messages. |

|                             |  |
|-----------------------------|--|
| rabbitmq_management         | The rabbitmq_management plugin provides web user interfaces and a set of HTTP APIs for managing and monitoring the RabbitMQ server.  |
| rabbitmq_peer_discovery_k8s | The rabbitmq_peer_discovery_k8s plugin allows RabbitMQ nodes to automatically discover and connect to other RabbitMQ nodes in a Kubernetes environment to form a cluster.  |
| rabbitmq_prometheus         | The rabbitmq_prometheus plugin is a monitoring plugin provided by RabbitMQ. It can export RabbitMQ metrics in a format understandable by Prometheus. Therefore, you can use tools, such as Prometheus and Grafana to monitor and visualize the RabbitMQ performance and health status. |
| rabbitmq_tracing            | The rabbitmq_tracing plugin can trace messages passing through RabbitMQ and persist them to a disk or record them in log files, thereby reducing time in problem identification and debugging.   |

**Note:**

TDMQ for RabbitMQ has the rabbitmq\_delayed\_message\_exchange delayed message plugin **Disabled by Default**, due to the following risks and limitations of the plugin:

1. The current plugin's design is not suitable for scenarios with a high volume of delayed messages (unscheduled messages reaching hundreds of thousands or even millions). Please carefully evaluate the message throughput in a production environment to avoid unexpected long delay, message loss, and other issues.
2. Delayed messages only have one persistent copy on each node. If a node cannot function properly (for example, heaped messages cause continuous OOM leading to restarts and inability to recover), the delayed messages on that node cannot be consumed by the consumer side.
3. The delayed switch does not support setting **mandatory**. Producers cannot be notified about messages that could not be routed through **basic.return** events. Thus, before they send delayed messages, it is crucial to ensure that the corresponding switches, queues, and routing relationships exist.

For the original text on risks and limitations, please see the [Official Usage Restriction Statement](#) of the RabbitMQ Delayed Message Plugin.

Additionally, you can see the documentation [TDMQ for RabbitMQ > Development Guide > Delayed Messages](#) for two implementations of delayed messages described therein.

# Message Query

Last updated : 2024-06-26 15:56:25

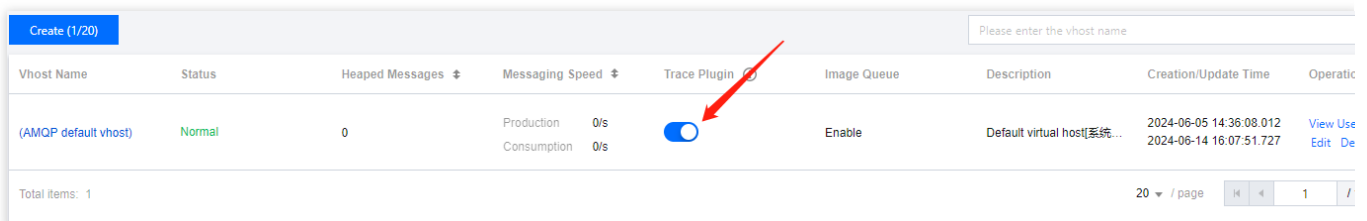
## Overview

If there are abnormal or missing messages, you can use the message query feature in the TDMQ for RabbitMQ console to analyze and locate the problem promptly.

This document describes how to query messages in the TDMQ for RabbitMQ console.

## Prerequisites

You have toggled on **Trace Plugin** for the vhost where you want to query messages.



| Create (1/20)               |        |                 |                 |     |                                     |             |                      |                         |
|-----------------------------|--------|-----------------|-----------------|-----|-------------------------------------|-------------|----------------------|-------------------------|
| Please enter the vhost name |        |                 |                 |     |                                     |             |                      |                         |
| Vhost Name                  | Status | Heaped Messages | Messaging Speed |     | Trace Plugin                        | Image Queue | Description          | Creation/Update Time    |
| (AMQP default vhost)        | Normal | 0               | Production      | 0/s | <input checked="" type="checkbox"/> | Enable      | Default virtual host | 2024-06-05 14:36:08.012 |
|                             |        |                 | Consumption     | 0/s |                                     |             |                      | 2024-06-14 16:07:51.727 |
| Total items: 1              |        |                 |                 |     |                                     |             |                      | 20 / page               |

## Use Limits

Overview of the implementation principle of message query: After the Trace Plugin of the VHost is enabled, the service component will consume the trace messages of the corresponding RabbitMQ cluster. Through a series of processing, it can achieve the feature of querying message traces on the console.

Based on the principles mentioned, message trace relies on service components consuming trace messages. Since service components are underlying public services, they cannot guarantee that trace messages from high-traffic RabbitMQ clusters can be consumed in a timely manner; heap of trace messages can cause issues such as high memory load within the cluster, affecting the stability of the RabbitMQ cluster.

Therefore, it is not recommended to enable the Trace Plugin in production environments, especially in overall clusters (including all VHosts), where the scenario involves sending **TPS over 10,000**. The Trace Plugin is advised to be used in small-traffic verification/troubleshooting scenarios.

## Directions

1. Log in to the [RabbitMQ console](#).



2. On the left sidebar, click **Message Query** and select the region and time range for the query.
3. Select the cluster, vhost, and queue to be queried. You can enter the routing key, user, message headers, and message body to narrow down the query.
4. Click **Query**, and all results will be displayed in the list below.

Time Range

Last 30 minutes

Last hour

Last 6 hours

Last 24 hours

Last 3 days

2024-06-14 17:13:33 ~ 2024-06-14 17:43:33

Current Cluster

test-huanhuan(amqp-gawpazk2)

Vhost

(AMQP default vhost)

Queue

tdmq\_trace\_handle

Routing Key

User

headers

Fuzzy query is supported

body

Fuzzy query is supported

Query

| Message Creation Time | Exchange Name | Routing Key | Operation |
|-----------------------|---------------|-------------|-----------|
| No data yet           |               |             |           |

Total items: 0

20 / page

1 / 1 page

**Note :**

To ensure the stability of the cluster, the console restricts the quantity and dimensions of message queries. Users can query messages under specific queues, with up to 10,000 results returned. These results are based on filtering by queue and Routing Key (adding a Routing Key filter is optional). Users can also add filters for user, headers, and body to further refine their search, but the query will be conducted within the aforesaid limit of up to 10,000 messages. Therefore, it is recommended to use the message query feature in small-traffic verification/troubleshooting scenarios.

5. Click **View Message Details** in the **Operation** column of the target message to view its details and content (message body).

Message Query

Guangzhou

Time Range

Last 30 minutes

Last hour

Last 6 hours

Current Cluster

Vhost

(AMQP default vhost)

Queue

Routing Key

User

headers

Fuzzy query is supported

body

Fuzzy query is supported

Query

| Message Creation Time | Exchange Name |
|-----------------------|---------------|
| 2024-06-14 16:45:09   | -             |
| 2024-06-14 16:45:09   | -             |
| 2024-06-14 16:45:09   | -             |
| 2024-06-14 16:45:09   | -             |
| 2024-06-14 16:45:14   | -             |
| 2024-06-14 16:45:14   | -             |
| 2024-06-14 16:45:14   | -             |
| 2024-06-14 16:45:14   | -             |
| 2024-06-14 16:45:14   | -             |
| 2024-06-14 16:45:14   | -             |

Message Details

Details

Cluster

VirtualHost

(AMQP default vhost)

Queue

Exchange

-

Message Type

deliver

Creation Time

2024-06-14 16:45:09

User

admin

Connection Information

rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local.1717568912.6370

Channel

1

Delivery Rule

{"deliveryTag":147,"exchange":"amq.rabbitmq.trace","redeliver":false,"routingKey":"deliver.aliveness-test"}

Node

rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local

Message Body

Headers

{exchange\_name=, node=rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local, vhost=, channel=1, connection=<rabbit@rabbitmq-broker-0.rabbitmq-broker-internal.amqp-gawpazk2.svc.cluster.local.1717568912.6370.418>, routing\_keys=[aliveness-test], user=admin, properties={}, redelivered=0}

Body

test\_message

©2013-2022 Tencent Cloud. All rights reserved.

Page 74 of 96

# Access Management

Last updated : 2024-01-03 11:42:22

## Basic CAM Concepts

A root account authorizes sub-accounts by binding policies. The policy settings can be specific to the level of **API, Resource, User/User Group, Allow/Deny, and Condition**.

### Account

**Root account:** it owns all Tencent Cloud resources and can access any of its resources.

**Sub-account:** it includes sub-users and collaborators.

**Sub-user:** it is created and fully owned by a root account.

**Collaborator:** it has the identity of a root account. After it is added as a collaborator of the current root account, it becomes one of the sub-accounts of the current root account and can switch back to its root account identity.

**Identity credential:** it includes login credentials and access certificates. **Login credential** refers to a user's login name and password. **Access certificate** refers to Tencent Cloud API keys ( `SecretId` and `SecretKey` ).

### Resource and permission

**Resource:** it is an object manipulated in Tencent Cloud services. TDMQ for RabbitMQ resources include clusters, vhosts, exchanges, queues, and bindings.

**Permission:** it is an authorization that allows or forbids users to perform certain operations. By default, **a root account has full access to all resources under it**, while **a sub-account does not have access to any resources under its root account**.

**Policy:** it is a syntax rule that defines and describes one or more permissions. The **root account** performs authorization by **associating policies** with users/user groups.

[View CAM documentation >>](#)

## Relevant Documents

| Document Description                 | Link                                 |
|--------------------------------------|--------------------------------------|
| Relationship between policy and user | <a href="#">Policy</a>               |
| Basic policy structure               | <a href="#">Policy Syntax</a>        |
| CAM-Enabled products                 | <a href="#">CAM-Enabled Products</a> |

## List of APIs Supporting Resource-Level Authorization

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

APIs supporting resource-level authorization include:

| API                  | Description                                   | Resource Type | Six-Segment Example of Resource                |
|----------------------|---|---------------|--|
| DeleteAMQPCluster    | Deletes AMQP cluster                          | cluster       | qcs::tdmq:\${region}:uin/\${uin}:cluster/\${cl |
| ModifyAMQPCluster    | Modifies AMQP cluster                         | cluster       | qcs::tdmq:\${region}:uin/\${uin}:cluster/\${cl |
| CreateAMQPVHost      | Creates AMQP vhost                            | cluster       | qcs::tdmq:\${region}:uin/\${uin}:cluster/\${cl |
| DescribeAMQPClusters | Queries the list of AMQP clusters             | cluster       | qcs::tdmq:\${region}:uin/\${uin}:cluster/\${cl |
| DescribeAMQPCluster  | Gets the information of specific AMQP cluster | cluster       | qcs::tdmq:\${region}:uin/\${uin}:cluster/\${cl |
| CreateAMQPExchange   | Creates AMQP exchange                         | vhost         | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu  |
| ModifyAMQPVHost      | Modifies AMQP vhost                           | vhost         | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu  |
| DeleteAMQPVHost      | Deletes AMQP vhost                            | vhost         | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu  |
| CreateAMQPQueue      | Creates AMQP                                  | vhost         | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu  |

|                              |   |          |  |
|------------------------------|---|----------|--|
|                              | queue   |          |  |
| CreateAMQPRouteRelation      | Creates AMQP binding                          | vhost    | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu: |
| DescribeAMQPVHostConnections | Queries the list of AMQP vhost connections    | vhost    | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu: |
| DescribeAMQPVHosts           | Queries the list of AMQP vhosts               | vhost    | qcs::tdmq:\${region}:uin/\${uin}:vHost/\${clu: |
| DeleteAMQPExchange           | Deletes AMQP exchange                         | exchange | qcs::tdmq:\${region}:uin/\${uin}:exchange/\$   |
| ModifyAMQPExchange           | Modifies AMQP exchange                        | exchange | qcs::tdmq:\${region}:uin/\${uin}:exchange/\$   |
| DescribeAMQPExchanges        | Queries the list of AMQP exchanges            | exchange | qcs::tdmq:\${region}:uin/\${uin}:exchange/\$   |
| DeleteAMQPQueue              | Deletes AMQP queue                            | queue    | qcs::tdmq:\${region}:uin/\${uin}:queue/\${clu  |
| DescribeAMQPQueueConsumers   | Gets the list of consumers in specified queue | queue    | qcs::tdmq:\${region}:uin/\${uin}:queue/\${clu  |
| ModifyAMQPQueue              | Modifies AMQP queue                           | queue    | qcs::tdmq:\${region}:uin/\${uin}:queue/\${clu  |
| DescribeAMQPQueues           | Queries the list of                           | queue    | qcs::tdmq:\${region}:uin/\${uin}:queue/\${clu  |

|                            |                                   |               |   |
|----------------------------|-----------------------------------|---------------|---|
|                            | AMQP queues                       |               |   |
| DescribeAMQPRouteRelations | Queries the list of AMQP bindings | routeRelation | qcs::tdmq:\${region}:uin/\${uin}:routeRelatic |
| DeleteAMQPRouteRelation    | Deletes AMQP binding              | routeRelation | qcs::tdmq:\${region}:uin/\${uin}:routeRelatic |

## List of APIs Not Supporting Resource-Level Authorization

| API                     | Description          | Six-Segment Resource |
|-------------------------|----------------------|----------------------|
| CreateAMQPCluster       | Creates AMQP cluster | *                    |
| DescribeAMQPCreateQuota | Gets user quota      | *                    |

## Authorization Scheme Examples

### Full access policy

Grant a sub-user full access to the TDMQ for RabbitMQ service (for creating, managing, etc.).

1. Log in to the [CAM console](#).
2. Click **Policy** on the left sidebar.
3. In the policy list, click **Create Custom Policy**.
4. In the **Select Policy Creation Method** pop-up window, select **Create by Policy Generator**.
5. On the **Edit Policy** page, click **Import Policy Syntax** in the top-right corner.
6. On the **Import Policy Syntax** page, search for **TDMQ**, select **QcloudTDMQFullAccess** in the search results, and click **OK**.
7. On the **Edit Policy** page, click **Next**, enter the policy name and description, and select the user/user group you want to associate.
8. Click **Complete**.

### Read-Only access policy

The following uses granting the read-only permission of a cluster as an example.

1. Log in to the [CAM console](#).

- Click **Policy** on the left sidebar.
- In the policy list, click **Create Custom Policy**.
- In the **Select Policy Creation Method** pop-up window, select **Create by Policy Generator** and enter the policy information.

←

Create by Policy Generator

1 Edit Policy >

2 Associate Users/User Groups

Visual Policy Generator

JSON

▼ Tencent Distributed Message Queue(All actions)

Effect \*

☒ Allow
 ☐ Deny

Service \*

Tencent Distributed Message Queue (tdmq)

Action \*

Collapse

Select actions

☒ All actions (tdmq:\*) [Show More](#)

Action Type

☒ Read (16 selected) [Show More](#)
☒ Write (62 selected) [Show More](#)
☒ List (23 selected) [Show More](#)

Resource \*

Select resources.

Condition

☐ Source IP ⓘ
 [Add other conditions.](#)

+ Add Permissions

Next

Characters: 128(up to 6,144)

| Parameter | Description         |
|-----------|---------------------|
| Effect    | Select <b>Allow</b> |
| Service   | Select <b>TDMQ</b>  |
|           |                     |

|           |   |
|-----------|---|
| Action    | Select <b>Read operation</b>  |
| Resource  | Select <b>Specific resources</b> and click <b>Add six-segment resource description</b><br>Region: select the resource region<br>Account: it is automatically populated<br>Resource Prefix: clusterId<br>Enter the ID of the cluster you want to authorize |
| Condition | Allow access to specified operations only when the request is from the specified IP range   |

5. Click **Next**, enter the policy name and description, and select the user/user group you want to associate.

6. Click **Complete**.



# Tag Management

## Managing Resource with Tag

Last updated : 2024-01-03 11:42:22

### Overview

**Tag** is a key-value pair provided by Tencent Cloud to identify a resource in the cloud. It can help you easily categorize and manage TDMQ for RabbitMQ resources in many dimensions such as business, purpose, and owner.

**Note:**

Tencent Cloud will not use the tags you set, and they are only used for your management of TDMQ for RabbitMQ resources.

### Use Limits

You need to pay attention to the following use limits of tags:

|           |   |
|-----------|---|
| Limit     | Description   |
| Quantity  | One Tencent Cloud resource can have up to 50 tags.  |
| Tag key   | You cannot place <code>qcloud</code> , <code>tencent</code> , or <code>project</code> at the beginning of a tag key as they are reserved by the system.<br>A tag key can contain up to 255 digits, letters, and special symbols ( <code>+=.@-</code> ). |
| Tag value | It can contain up to 127 digits, letters, and special symbols ( <code>+=.@-</code> ) or be an empty string.   |

### Directions and Use Cases

**Use case**

A company has 6 TDMQ for RabbitMQ clusters, with the department, business scope, and owner information as described below:

|                   |            |                |       |
|-------------------|------------|----------------|-------|
| Cluster ID        | Department | Business Scope | Owner |
| amqp-78383dp8p8w1 | Ecommerce  | Marketing      | John  |
| amqp-78383dp8p8w2 | Ecommerce  | Marketing      | Harry |
|                   |            |                |       |

|                   |               |                 |       |
|-------------------|---------------|-----------------|-------|
| amqp-78383dp8p8w3 | Gaming        | Game A          | Jane  |
| amqp-78383dp8p8w4 | Gaming        | Game B          | Harry |
| amqp-78383dp8p8w5 | Entertainment | Post-production | Harry |
| amqp-78383dp8p8w6 | Entertainment | Post-production | John  |

You can add the following three tags to the `amqp-78383dp8p8w1` cluster:


| Tag Key  | Tag Value |
|----------|-----------|
| dept     | ecommerce |
| business | mkt       |
| owner    | zhangsan  |

Similarly, you can also set appropriate tags for other resources based on their department, business scope, and owner information.

## Setting tag in TDMQ for RabbitMQ console

After designing the tag keys and values as detailed above, you can log in to the TDMQ for RabbitMQ console to set tags.

1. Log in to the [TDMQ for RabbitMQ console](#).
2. On the **Cluster Management** page, select the target region and cluster and click **Edit Resource Tag** at the top of the page.

| Create Cluster (1/10)  |                           | Edit Resource Tag         |             | Search by keyword |
|--|---------------------------|---------------------------|-------------|-------------------|
| <input type="checkbox"/> Cluster ID/Name   | Exchange Count            | Queue Count               | Description |                   |
| <input type="checkbox"/> amqp- <br>test | Used: 1<br>Capacity: 1000 | Used: 1<br>Capacity: 1000 | test        |                   |

3. Set tags in the **Edit Tag** pop-up window.

For example, add three tags for the `amqp-78383dp8p8w1` cluster.

### Edit Tags

The tag is used to manage resources by category from different dimensions. If the existing tag does not meet your requirements, please go to [Manage Tags](#)

1 resource selected

|          |           |   |
|----------|-----------|---|
| business | mkt       | × |
| dept     | ecommerce | × |
| owner    | zhangsan  | × |

[+ Add](#)

[OK](#) [Cancel](#)

**Note:**

If existing tags cannot meet your needs, go to [Tag Management](#) to create more.

4. Click **OK**, and you will be prompted that the tags have been modified successfully. You can view the tags bound to a cluster in its **Resource Tag** column.

Create Cluster (1/10)

Edit Resource Tag

Search by keyw

| <input checked="" type="checkbox"/> | Cluster ID/Name           | Exchange Count            | Queue Count               | Description |
|-------------------------------------|---------------------------|---------------------------|---------------------------|-------------|
| <input checked="" type="checkbox"/> | amqp-namzjkam8xdr<br>test | Used: 1<br>Capacity: 1000 | Used: 1<br>Capacity: 1000 | test        |

**Filtering resource by tag key**

You can filter out clusters bound to a specific tag in the following steps:


1. Select **Tag** in the search box at the top-right corner of the page.
2. In the window that pops up, select the tag you want to search for and click **OK**.

For example, if you select Tag: owner:zhangsan , you can filter out clusters bound to the tag key owner:zhangsan .

Create Cluster (1/10)

Edit Resource Tag

Tag: owner : zhangsan

| <input checked="" type="checkbox"/> Cluster ID/Name   | Exchange Count            | Queue Count               | Description |
|---|---------------------------|---------------------------|-------------|
| 1 result found for "Tag:owner : zhangsan" <a href="#">Back to list</a>  |                           |                           |             |
| <input checked="" type="checkbox"/> amqp- <br>test | Used: 1<br>Capacity: 1000 | Used: 1<br>Capacity: 1000 | test        |

# Editing Tag

Last updated : 2024-06-26 15:56:25

## Overview

This document describes how to edit resource tags.

## Use Limits

For the use limits of tags, see [Managing Resource with Tag - Use Limits](#).

## Prerequisites

You have logged in to the [RabbitMQ console](#).

## Directions

1. On the **Cluster Management** page, select the target region and cluster and click **Edit Resource Tag** at the top of the page.

| Create Cluster (1/10)               |                   | Edit Resource Tag         |                           |             | Search by keyword |   |
|-------------------------------------|-------------------|---------------------------|---------------------------|-------------|-------------------|---|
| <input checked="" type="checkbox"/> | Cluster ID/Name   | Exchange Count            | Queue Count               | Description | Resource Tag      | Operation   |
| <input checked="" type="checkbox"/> | amqp-namz<br>test | Used: 0<br>Capacity: 1000 | Used: 0<br>Capacity: 1000 | test        |                   | <a href="#">Access Address</a> <a href="#">Edit</a> <a href="#">D</a> |

### Note:

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

2. In the **Edit Tag** pop-up window, add, modify, or delete tags as needed.

## Use Cases

For directions on how to use tags, see [Managing Resource with Tag](#).

# Migration to Cloud

## Migrating RabbitMQ to Cloud

Last updated : 2024-08-07 14:24:45

### Overview

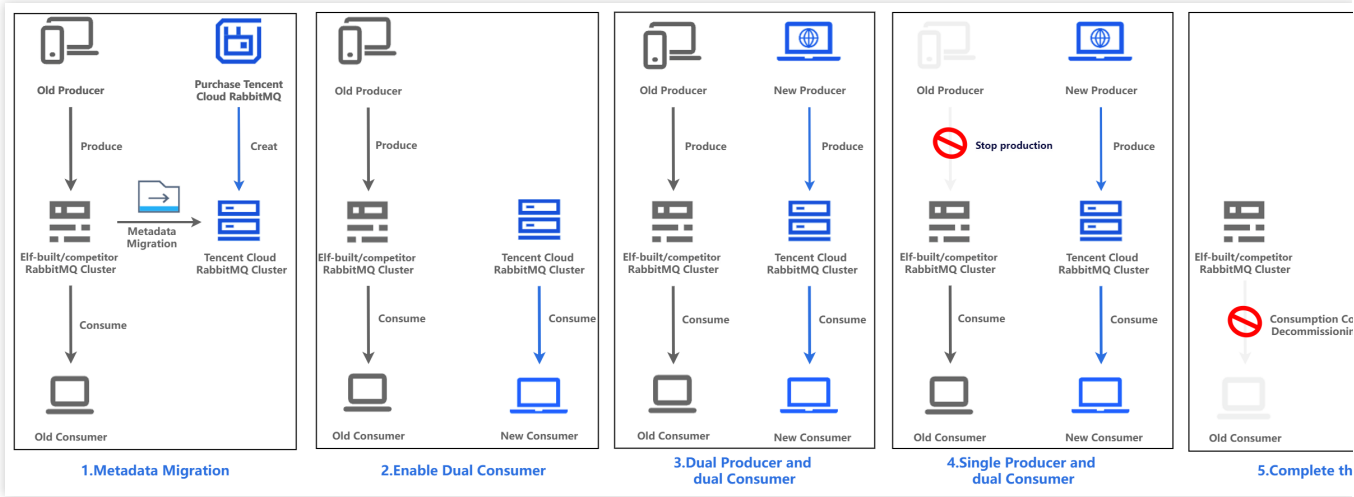
This document provides an overall introduction to a feasible scheme for migrating self-built/competitor RabbitMQ clusters to TDMQ for RabbitMQ clusters.

### Scheme Description: Dual-Producer Dual-Consumer

This scheme is simple, clear, and easy to operate, with no data heap, ensuring timely message consumption.

Directions are as follows:

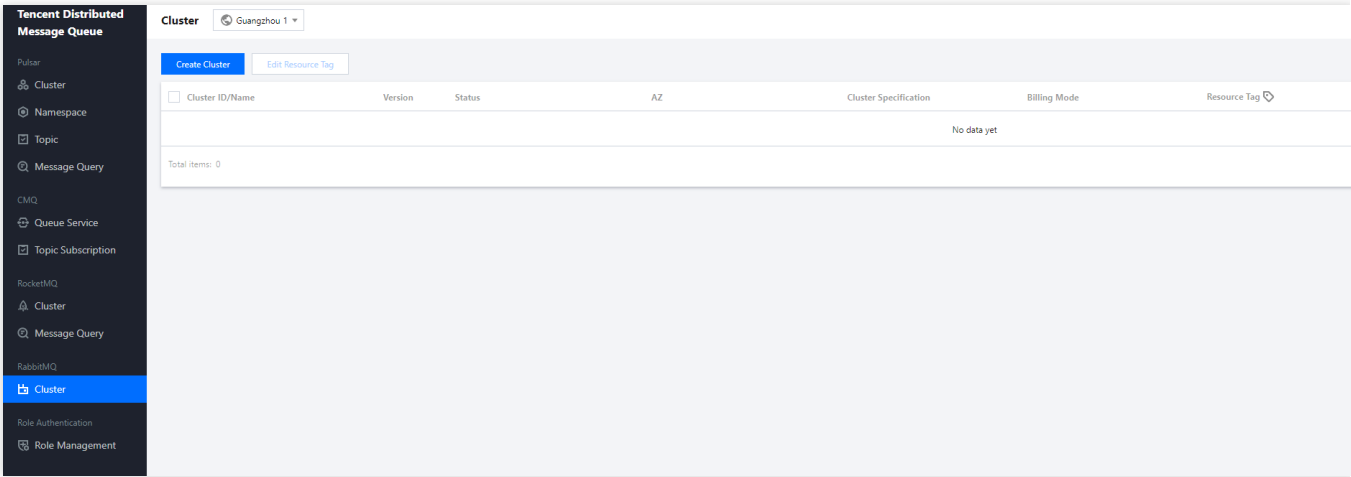
1. Purchase a RabbitMQ cluster on Tencent Cloud. In the initial status, producer and consumer operations are performed on the old cluster. Once the new cluster is ready, complete the metadata migration of RabbitMQ.
2. Deploy a new consumer side to the new cluster and the dual-producer mode is enabled.
3. Deploy a new producer side and gradually switch the traffic to the new cluster to enable the dual-producer dual-consumer mode.
4. Stop the production traffic of the old cluster, temporarily retain the consumer side of the old cluster, and enter the single-producer dual-consumer mode.
5. After the data on the old cluster is consumed and confirming that there are no heaped messages, disable the old consumption service to complete the migration process.



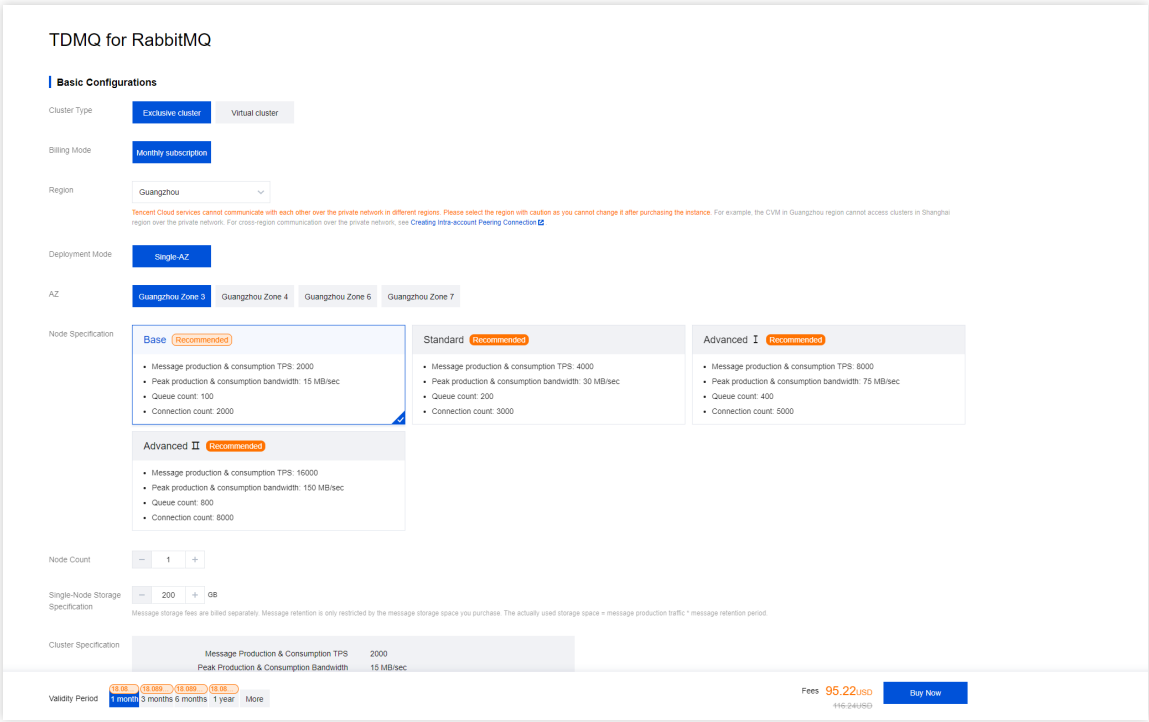
# Step 1. Purchasing a TDMQ Instance

Last updated : 2024-01-03 11:42:22

- 1. Log in to the [TDMQ console](#).
- 2. Select **RabbitMQ > Cluster** on the left sidebar and click **Create Cluster** to enter the purchase page.



- 3. On the purchase page, select the target instance specification.



| Parameter | Required | Description |
|-----------|----------|-------------|
|-----------|----------|-------------|



|                                   |     |  |
|-----------------------------------|-----|--|
| Cluster Type                      | Yes | Select <b>Exclusive cluster</b> .  |
| Billing Mode                      | Yes | TDMQ for RabbitMQ exclusive cluster adopts the <b>monthly subscription</b> billing mode.   |
| Region                            | Yes | Select a region close to resources of the deployed client. Cloud products in different regions are not interconnected over private networks and the region cannot be changed after you purchase the service. Proceed with caution.   |
| AZ                                | Yes | Select an AZ based on your business needs. Cross-AZ deployment is supported.   |
| Node Specification                | Yes | Select an appropriate node specification based on your business needs.   |
| Node Count                        | Yes | Select an appropriate number of nodes based on your business needs.  |
| Single-Node Storage Specification | Yes | Message storage is billed separately, and messages can be retained without limit in the purchased storage capacity. The actual storage usage can be estimated by multiplying the message production traffic by the retention period. |
| VPC                               | Yes | Bind the domain name of the new cluster's access point to the selected VPC.  |
| Cluster Name                      | Yes | Enter the cluster name, which can contain 3–64 digits, letters, hyphens, and underscores.  |
| Tag                               | No  | Tags are used to categorize and manage resources. For more information, see <a href="#">Tag Overview</a> .   |

4. Select **I have read and agree to TDMQ for RabbitMQ Terms of Service** and click **Buy Now**.

5. On the order payment page, click **Pay** and wait 3–5 minutes. Then, you can see the created cluster on the **Cluster** page.

# Step 2. Migrating Metadata to Cloud

Last updated : 2024-04-23 11:24:16

## Overview

This document describes how to migrate metadata from an open-source RabbitMQ cluster to a TDMQ for RabbitMQ cluster.

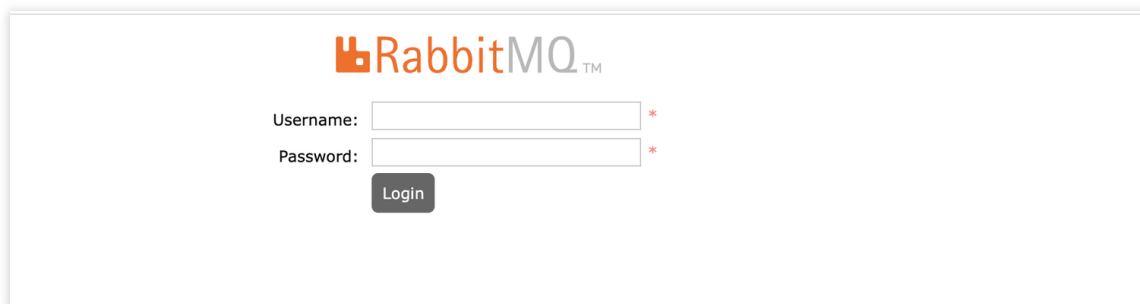
## Prerequisites

You have exported the metadata file from open-source RabbitMQ.

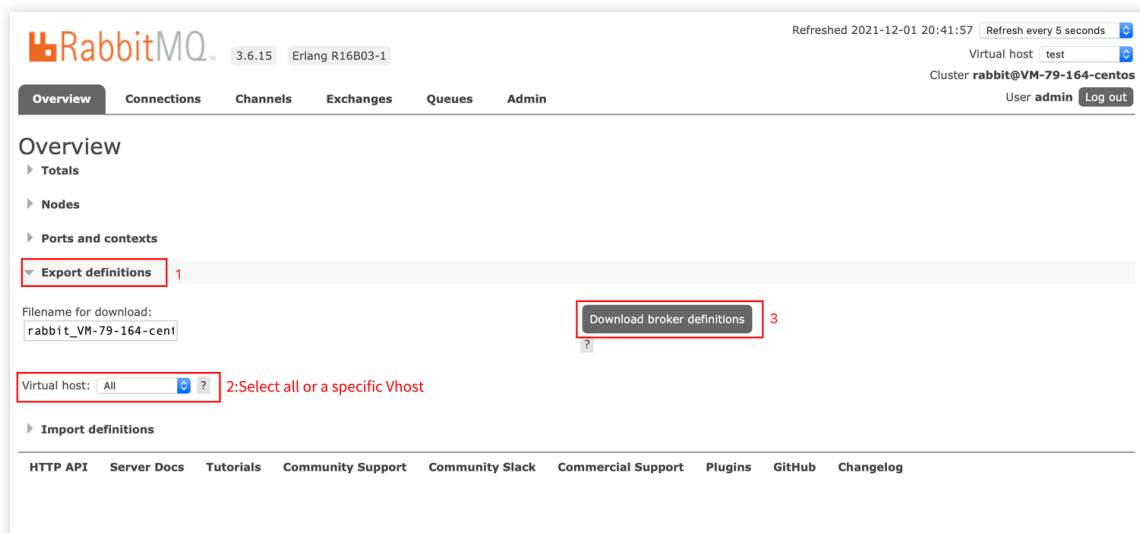
## Directions

### Exporting the metadata from a self-built RabbitMQ cluster

1. Open a browser and log in to the self-built open-source RabbitMQ console.



2. On the **Overview** tab page, click **Export definitions**, and enter the filename for download. Select All or a specific vhost for the Virtual host field and click **Download broker definitions** on the right to export the metadata file of all Vhosts or the specified vhost.



3. View the content of the exported metadata.



## Importing Metadata into a Tencent Cloud RabbitMQ Cluster

1. Log in to the Tencent Cloud console, and click **Migrate to Cloud**.
2. On the **Cloud Migration Task List** page, click **Create Task**.

←

Create Migration Task

Target cluster \*

No data yet

↻

If there is no desired cluster, you can [create one](#).

Task Type

All

Specified Vhost

Import all metadata from the open-source RabbitMQ into the TDMQ RabbitMQ cluster.

Metadata File \*

Select Local File

Importing cluster name or admin password may cause some functions to be unavailable. It is recommended to remove such metadata fields.

Create Task

Close

The following are the brief steps for import. See [Migrating RabbitMQ to Cloud](#) for detailed operations.

- Export Metadata**
  - Log in to the open-source RabbitMQ console.
  - At the bottom of the Overview tab, click Export definitions, select All or a specified Vhost name from the Virtual host list, and click Download broker definitions.
  - ALL: Export metadata for all Vhosts.
  - Vhost Name: Export metadata of the specified Vhost.
- Import JSON File**

Create a corresponding migration task on the console and upload the JSON file generated by the tool. Confirm whether the metadata is migrated based on cluster dimensions or imported into the specified Vhost.
- Check Data**

On the preview page for importing data, check if the imported data is correct. If there are errors, modify the metadata format of the source cluster according to the error message.
- View Migration Result**

In the migration task list, you can check the progress and results of migration tasks. Click [View Details](#) to see the details of the imported data.

**Target cluster:** Select the target TDMQ for RabbitMQ cluster to which metadata is imported.

### Task Type

**All:** Import all metadata from the open-source RabbitMQ cluster into the TDMQ for RabbitMQ cluster.

**Specified Vhost:** Import specified Vhost metadata from the open-source RabbitMQ cluster into the specified Vhost of the TDMQ for RabbitMQ cluster.

**Metadata File:** Select a local metadata file.

### Note:

Importing the cluster name or admin password may disable some features. It is recommended that you remove such metadata fields.

3. Check the data to import on the preview page to ensure its accuracy. If there are any errors, modify the source cluster metadata format based on the error message.

4. Click **Create Task**. A task record is generated on the **Cloud Migration Task List** page.

5. Click **View Details** to view details of this migration task.

Task Details

|                   |                                |
|-------------------|--------------------------------|
| Target Cluster ID | a [redacted] (test [redacted]) |
| Import Mode       | All                            |
| Task Status       | Succeeded                      |
| Creation Time     | 2024-04-07 17:46:29            |

## Step 3. Migrating Data to Cloud

Last updated : 2024-08-07 14:25:27

### Overview

This document mainly introduces how to use the dual-producer dual-consumer scheme to switch the services of a self-built RabbitMQ cluster to TDMQ for RabbitMQ.

## Scheme: Dual-Producer Dual-Consumer Mode

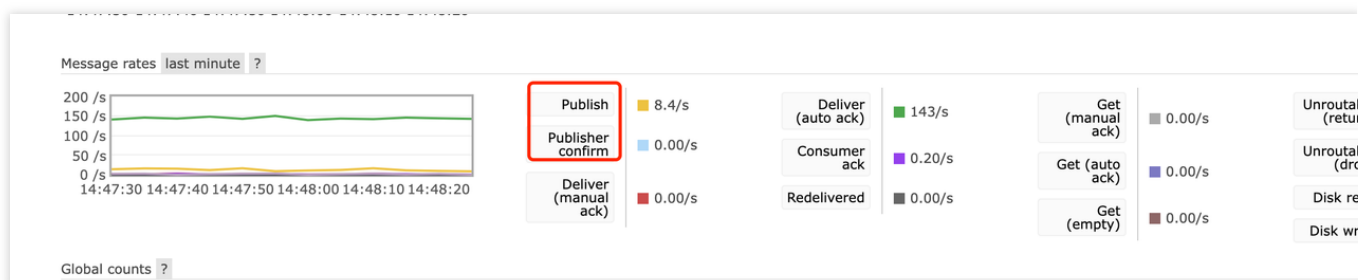
### Prerequisites

1. Purchased a RabbitMQ Instance on Cloud
2. Migrated the metadata of the self-built RabbitMQ cluster to Tencent Cloud RabbitMQ.

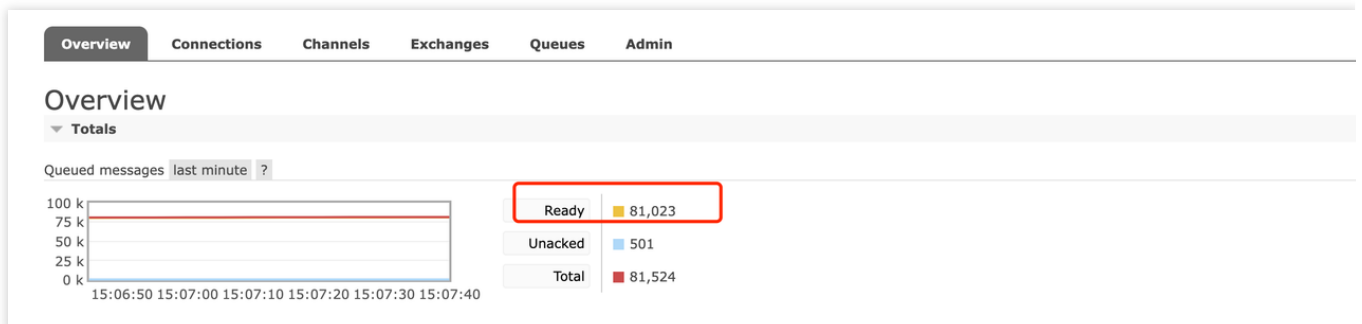
### Directions

1. Switch the access information for some nodes in the consumer cluster and transfer these consumers to the new RabbitMQ cluster. These consumers will consume messages from the new RabbitMQ cluster, while the remaining consumers will continue to consume messages from the old RabbitMQ cluster.
2. Switch the access information for some nodes in the producer cluster and transfer these producers to the new RabbitMQ. These producers will send messages to the new RabbitMQ cluster, while the remaining producers will continue to send messages to the old RabbitMQ cluster. To prevent duplicate or lost messages, you can implement idempotent logic for message consumption in advance.
3. Transfer the remaining producers to the new RabbitMQ cluster. Then, all messages will be sent to the new RabbitMQ cluster.

**Tips 1:** You can confirm that the production traffic of the self-built RabbitMQ cluster has stopped in the community management console.

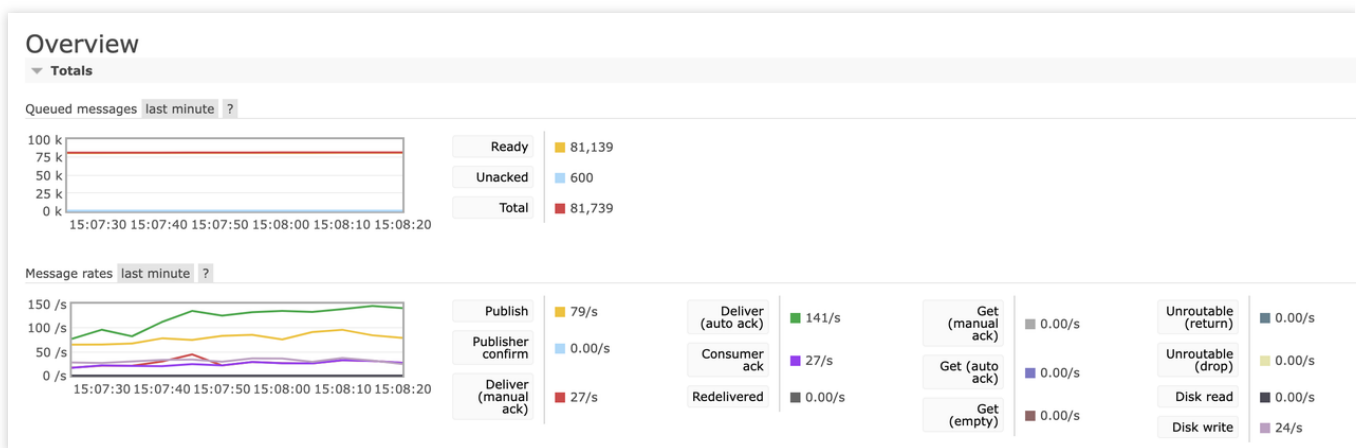


**Tips 2:** You can confirm that the heaped messages in the self-built RabbitMQ cluster are decreasing in the community management console.



4. Check if there are any unconsumed messages heaped in the old RabbitMQ cluster. After you confirm that there are no heaped or unhandled messages in the old RabbitMQ cluster, transfer the remaining consumers to the new RabbitMQ cluster to complete the migration of the entire data flow.

**Tips:** Confirm the message production and consumption in the cloud RabbitMQ cluster, and ensure there is no message heap.



#### Note:

Follow the above steps strictly. If you switch producers first and then switch consumers, message loss may occur. Before you switch the remaining consumers, ensure that all messages in the old RabbitMQ cluster have been consumed to avoid any missed consumption.

## Possible Issues

### Order Issue

Due to the cluster switch, the order of messages cannot be guaranteed during the switch process. There may be partial disorder during the switch.

### Message Duplication

In theory, Message will not duplicate, but in extreme cases it can occur. For example, during the switch process, a consumer has consumed a message but has not sent an ACK to the server (the old RocketMQ cluster). This can

cause the message to enter the retry queue, leading to duplicate consumption. Implementing idempotent logic for the messages can avoid this issue.

## **Consumption Delay**

During the consumer switch process, the reallocation of partitions requires rebalance between queues and consumer clients, which may cause short consumption delays. No additional operations are needed in this situation, and operations will resume once the switch is completed.