

# RabbitMQ

RabbitMQ????????????????????????????RabbitMQ?????Erlang?????????????????????????????  
????????????????????????????????????

- [Learning](#)
- [Benchmark](#)
- [Development with RabbitMQ](#)
  - [Bash](#)
  - [Python](#)
  - [PHP](#)
  - [dotNet](#)
  - [AMQP Client](#)
- [RabbitMQ Standalone](#)
- [Install on RedHat](#)
- [Management UI](#)
- [Plugins](#)
- [Monitoring & Management](#)
- [RabbitMQ Cluster](#)
- [FAQ](#)

# Learning

## Introduction

- [Message Queue ???\( RabbitMQ ??\) | ????????? \(godleon.github.io\)](#)
- [RabbitMQ ??????????](#)

??????????

- [Kafka](#)
  - Video: [Taichung.py 2024-01 Meetup01, ??????????kafka ??? - YouTube](#)

## Installation

- [Installing on RPM-based Linux \(RedHat Enterprise Linux, CentOS, Fedora, openSUSE\)](#)
- [Production Checklist](#)
- [RabbitMQ Checklist For Production Environments - A Complete Guide](#)

## Cluster & Load Balancing

- [RabbitMQ ????????? cluster ?????](#)
- [VMWARE] [VMware Tanzu RabbitMQ for Kubernetes Overview](#)
- [HUAWEI] [Distributed Message Service for RabbitMQ](#)

## Development

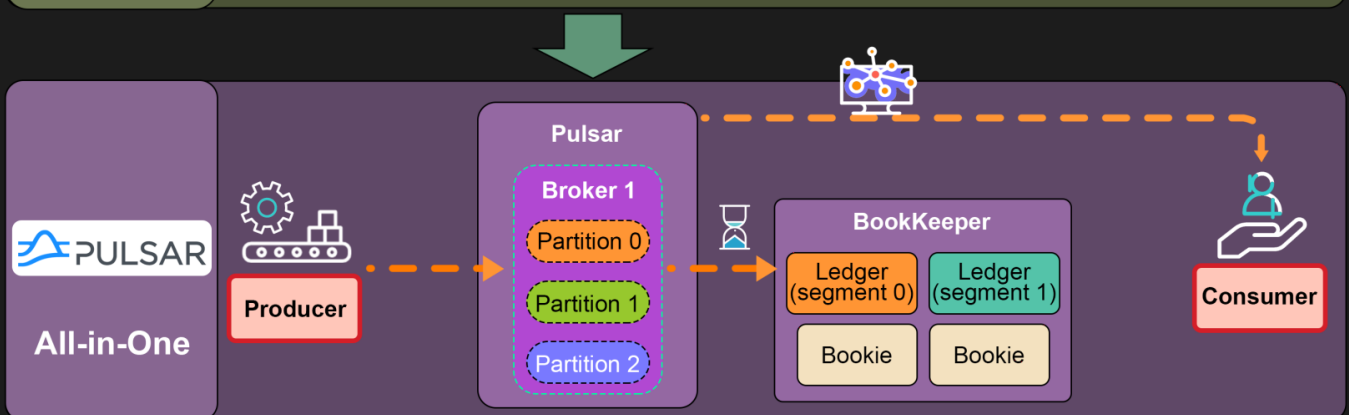
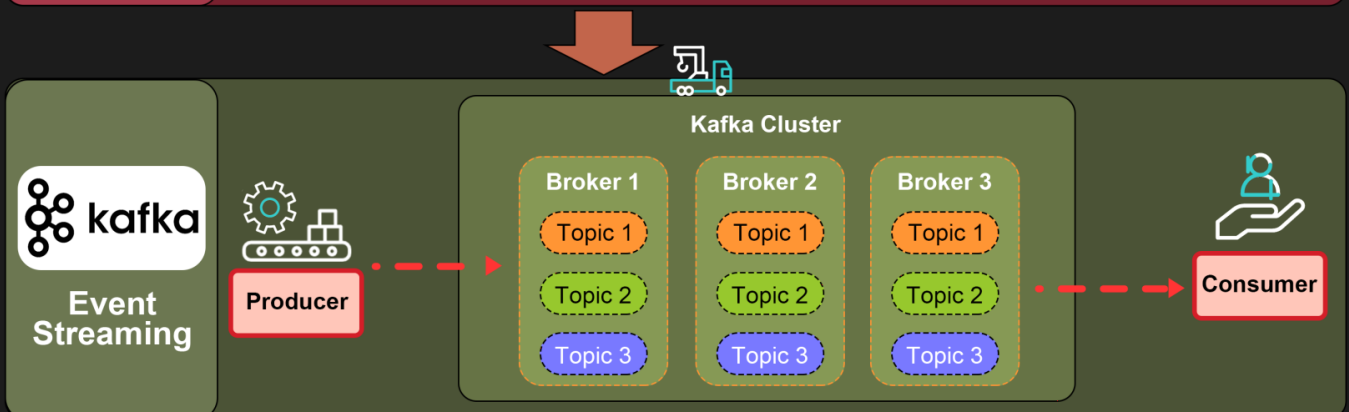
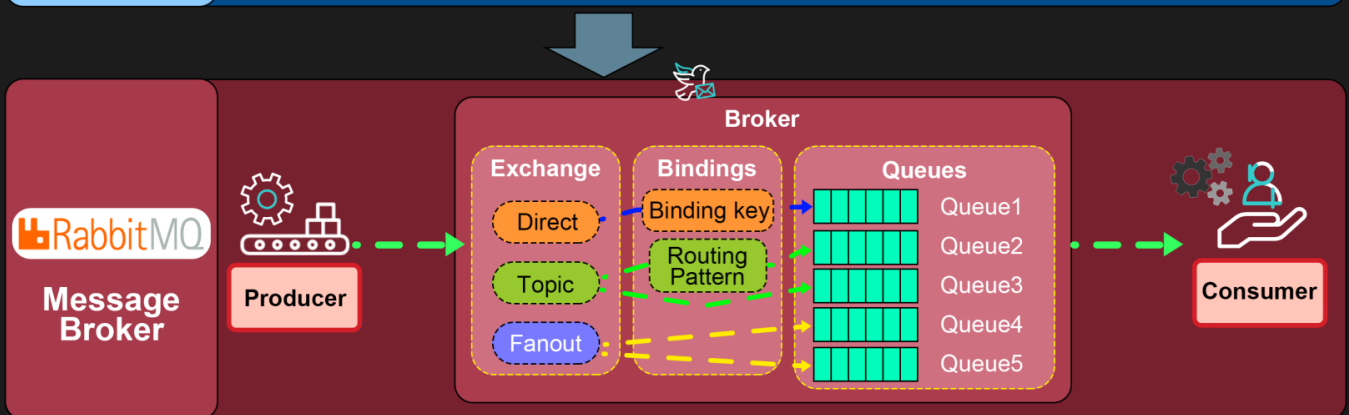
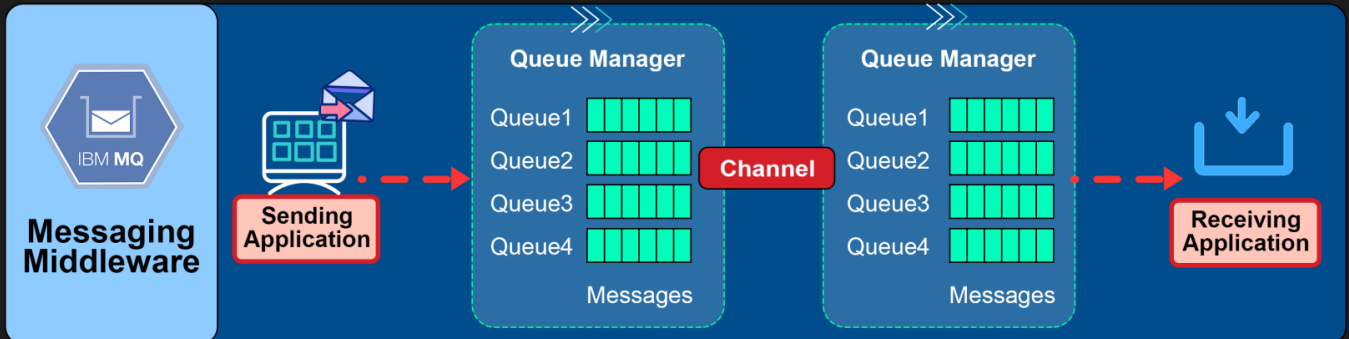
- [RabbitMQ ??](#)
- [Amazon MQ for RabbitMQ best practices](#)
- [Red Hat AMQ Clients](#)

## Commercial RabbitMQ

- [IBM Messages for RabbitMQ](#)
- [VMware Tanzu RabbitMQ](#)

# How do Message Queues Evolve ?

 [blog.bytebytego.com](https://blog.bytebytego.com)



# Benchmark

- [Java Tools](#)
  - [Load testing and performance measurements in RabbitMQ - CloudAMQP](#)
- [amqpc](#)

## Java Tools

Install: [RabbitMQ PerfTest](#)

### Usage

- --time: Seconds, ??????
- --uri: MQ Server ?????
- --producers: Producer ??
- --consumers: Consumer ??
- --size: bytes, Message ? size
- --rate: msg/s, Producer/Consumer ? Msg Rate
- --consumer-rate: msg/s, Consumer ? Msg Rate

```
#> alternatives --list
libnssckbi.so.x86_64  auto  /usr/lib64/pkcs11/p11-kit-trust.so
ld  auto  /usr/bin/ld.bfd
mta  auto  /usr/sbin/sendmail.postfix
ksh  auto  /bin/ksh93
java  manual /usr/lib/jvm/java-11-openjdk-11.0.8.10-1.el7.x86_64/bin/java
jre_openjdk  auto  /usr/lib/jvm/java-11-openjdk-11.0.8.10-1.el7.x86_64
jre_11  auto  /usr/lib/jvm/java-11-openjdk-11.0.8.10-1.el7.x86_64
jre_11_openjdk  auto  /usr/lib/jvm/jre-11-openjdk-11.0.8.10-1.el7.x86_64

#> export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-11.0.8.10-1.el7.x86_64
#> cd rabbitmq-perf-test-2.18.0
#> bin/runjava com.rabbitmq.perf.PerfTest --help
```

### Scenario #1

- Condition:
  - Queue: 200

- Producer: 200
- Consumer: 200
- Message Size: 20KB
- Message Rate: 20/s
- Ack Mode: Manual
- Throughput:
  - Publish: 4000/s
  - Consumer: 4000/s
- Resource:
  - CPU (2 Core): 110 %
  - Memory (4 GB): 8.8 %

```
bin/runjava com.rabbitmq.perf.PerfTest --uri amqp://<username>:<password>@<rabbitmq-server-
ip>:<port>/<vhost> \
--queue-pattern 'perf-test-%d' \
--queue-pattern-from 1 \
--queue-pattern-to 200 \
--producers 200 \
--consumers 200 \
--size 20000 \
--rate 20
```

## Scenario #2 for the Cluster with 3 nodes

### Queue Type: Classic

```
bin/runjava com.rabbitmq.perf.PerfTest \
--uris "amqp://<username>:<password>@<rabbitmq-server1-
ip>:<port>/<vhost>,amqp://<username>:<password>@<rabbitmq-server2-
ip>:<port>/<vhost>,amqp://<username>:<password>@<rabbitmq-server3-ip>:<port>/<vhost>" \
--queue-pattern 'perf-test-%d' \
--queue-pattern-from 1 \
--queue-pattern-to 200 \
--producers 200 \
--consumers 200 \
--size 20000 --rate 20
```

### Queue Type: Quorum

```
bin/runjava com.rabbitmq.perf.PerfTest \
--uris "amqp://<username>:<password>@<rabbitmq-server1-
ip>:<port>/<vhost>,amqp://<username>:<password>@<rabbitmq-server2-
```

```
ip>:<port>/<vhost>,amqp://<username>:<password>@<rabbitmq-server3-ip>:<port>/<vhost>" \
--queue-args x-queue-type=quorum \
--auto-delete false \
--flag persistent \
--queue-pattern 'perf-test-quorum-%d' \
--queue-pattern-from 1 \
--queue-pattern-to 200 \
--producers 200 \
--consumers 200 \
--size 20000 \
--rate 6
```

???????

- ??????: [How to Run Benchmarks | RabbitMQ - Blog](#)
- [13 Common RabbitMQ Mistakes and How to Avoid Them - CloudAMQP](#)
- [Cluster Sizing and Other Considerations | RabbitMQ - Blog](#)

# Development with RabbitMQ

# Bash

## curl

```
curl -u login:pass -i -H "content-type:application/json" -X POST
http://localhost:15672/api/exchanges/%2Fvhost/exchange/publish \
-d '{"properties":{ }, "routing_key":"","payload":"you message","payload_encoding":"string"}'
```

## rabbitmqadmin

`rabbitmqadmin` uses HTTP API authentication mechanism (basic HTTP authentication).

- [Command Line Tools](#)
- [Management Command Line Tool](#)

## Install

```
# Install the dependency
yum install python3

# Download the script from the RabbitMQ Server
wget http://<rabbitmq-server-hostname>:15672/cli/rabbitmqadmin

#
chmod 0755 rabbitmqadmin
mv rabbitmqadmin /usr/local/bin
rabbitmqadmin -h

# Verify the connection via HTTP API
rabbitmqadmin -H <rabbitmq-server-hostname> -u <username> -p <password> list vhosts
```

## Basic Operation

```
## 创建 Linux 队列
# 创建 queue
rabbitmqadmin -H <rabbitmq-server-hostname> -u <username> -p <password> -V <vhost-name> declare
```



```
queue name=my-testq durable=true
```

```
# 发布
```

```
rabbitmqadmin -H <rabbitmq-server-hostname> -u <username> -p <password> -V <vhost-name> publish  
exchange=amq.default routing_key=my-testq payload="This is Alang"
```

```
# 消费
```

```
rabbitmqadmin -H <rabbitmq-server-hostname> -u <username> -p <password> -V <vhost-name> get  
queue=my-testq ackmode=ack_requeue_false
```

```
# 消费, 以 tsv 格式
```

```
rabbitmqadmin -H <rabbitmq-server-hostname> -u <username> -p <password> -V <vhost-name> -f tsv get  
queue=my-testq ackmode=ack_requeue_false
```

```
# 消费 5 条
```

```
rabbitmqadmin -H <rabbitmq-server-hostname> -u <username> -p <password> -V <vhost-name> get  
queue=my-testq count=5 ackmode=ack_requeue_false
```

```
while read -r line; do
```

```
    echo $line | rabbitmqadmin publish exchange=amq.default routing_key=my_queue ;
```

```
done < messages
```

```
rabbitmqadmin publish exchange=amq.default routing_key=test payload="hello, world"
```

```
# With parallel
```

```
cat messages | parallel -j 100 \
```

```
    ./rabbitmqadmin -H $RABBITMQ_HOST \
```

```
        -u $RABBITMQ_USERNAME \
```

```
        -p $RABBITMQ_PASSWORD \
```

```
        publish exchange=amq.default \
```

```
        routing_key=myqueue \
```

```
        payload="{ }"
```

# Python

## Tutorials

- [python rabbitmq Code Example \(codegrepper.com\)](#)
- [Part 2.3: Getting started with RabbitMQ and Python - CloudAMQP](#)
- [??Python?????RabbitMQ????? | ????? \(codertw.com\)](#)
- [Python ?? RabbitMQ ????????? Python ???\\_Python\\_AlwaysBeta\\_InfoQ????](#)
- [RedHat AMQ Python Client](#)

## RedHat AMQ Python Client

Install via RHN

```
[root@dotnetdev ~]# dnf repolist all | grep amq-client
```

amq-clients-2-for-rhel-8-x86_64-debug-rpms	disabled
amq-clients-2-for-rhel-8-x86_64-rpms	disabled
amq-clients-2-for-rhel-8-x86_64-source-rpms	disabled
amq-clients-2.9-for-rhel-8-x86_64-debug-rpms	disabled
amq-clients-2.9-for-rhel-8-x86_64-rpms	disabled
amq-clients-2.9-for-rhel-8-x86_64-source-rpms	disabled

```
[root@dotnetdev ~]# subscription-manager repos --enable=amq-clients-2-for-rhel-8-x86_64-rpms
Repository 'amq-clients-2-for-rhel-8-x86_64-rpms' is enabled for this system.
```

```
[root@dotnetdev ~]# dnf repolist
```

Updating Subscription Management repositories.

repo id	repo name
amq-clients-2-for-rhel-8-x86_64-rpms	Red Hat AMQ Clients 2 for RHEL 8 x86_64 (RPMs)
rhel-8-for-x86_64-appstream-rpms	Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
rhel-8-for-x86_64-baseos-rpms	Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)

```
[root@dotnetdev ~]# yum install python3-qpid-proton python-qpid-proton-docs
```

Development with RabbitMQ

# PHP

## Tutorials

- [how to call a model from rabbitmq php consumer's callback in codeigniter? \(google.com\)](#)

Development with RabbitMQ

# dotNet

## Tutorials

- [GitHub] [AMQP.Net Lite](#)
- [RedHat AMQ .Net Client](#)
- [C# ??? RabbitMQ Cluster - ?? RabbitMQ .Net Client ? EasyNetQ](#)
- [GitHub] [EasyNetQ](#)

## Case: Console RabbitMQ Client

[ConsoleRabbitMQ.zip](#)

Package Required:

- RabbitMQ.Client

Target Framework:

- net6.0

Platform Supported:

- Windows 64
- RedHat Linux 8 64 (net6 runtime is required)

# AMQP Client

## Terms & Concepts

### Queue

- prefetch: Consumer ???? consumer ???? PUSH ?????????????????? consumer ? PUSH ?????????(?????)????? consumer ?????????????????????? Unacked ???? Unacked ?????????????????? ??????consumer ???? prefetch ?????????? consumer ?????
- Lazy Queue: Queue ??????(?)????????? queue????????????????????

### Consume Message

- Consumer ?????????? Push (??) ? Poll (?) ?????
  - [????RabbitMQ\(?\)——?????QOS?C#?? - ????? - ??? \(cnblogs.com\)](#)

### Publish Message

- ??? nodes ? Cluster ????????????????????? nodes????????
- Publish Confirm:

## TTL (Time to Live)

?? TTL?? RabbitMQ ? TTL ???? Message ? Queue ???

### Message TTL

- ?? : ??????????????????
- ?? Message ????? Queues ????? Queue ? TTL ?????????????????????? TTL ? Queue???????????? ?
- ??? : Millisecond (60 seconds = 60000)

### ??? : Policy

```
rabbitmqctl set_policy TTL ".*" '{"message-ttl":60000}' --apply-to queues
```

### ??? : Queue ??

- x-message-ttl: 60000

## Sample codes in C#

```
var args = new Dictionary<string, object>();
args.Add("x-message-ttl", 60000);
model.QueueDeclare("myqueue", false, false, false, args);
```

## Queue TTL

- ??: ???????? Queue?? Queue ??????
- ???: Millisecond (30 mins = 1800000)

## ???: Policy

```
rabbitmqctl set_policy expiry ".*" '{"expires":1800000}' --apply-to queues
```

## ???: Queue ??

- x-expires: 1800000

## Sample codes in Java

```
Map<String, Object> args = new HashMap<String, Object>();
args.put("x-expires", 1800000);
channel.queueDeclare("myqueue", false, false, false, args);
```

## rabbitmqadmin

- [Management Command Line Tool — RabbitMQ](#)

## Usage

```
# Publish a message
rabbitmqadmin -H <rabbitmq-server-ip> -u <user-name> -p <secret> -V <virtual-server> publish
exchange=amq.default routing_key=my-testq payload="This is Alang"

# Consume/Get a message
rabbitmqadmin -H <rabbitmq-server-ip> -u <user-name> -p <secret> -V <virtual-host> get queue=my-testq
ackmode=ack_requeue_false
```

## amqp-tools

A CLI tool is built-in Ubuntu.

## Install

```
sudo apt update
sudo apt install amqp-tools
```

## Usage

```
# Declare a queue
amqp-declare-queue --url="amqp://<user-name>:<secret>@<rabbitmq-server-ip>:<rabbitmq-server-
port>/<virtual-server>" -d -q "my-testq"

# Publish a message
amqp-publish --url="amqp://<user-name>:<secret>@<rabbitmq-server-ip>:<rabbitmq-server-port>/<virtual-
server>" --routing-key="my-testq" -b "Hello,World"

# Get the messages (Poll mode)
amqp-get --url="amqp://<user-name>:<secret>@<rabbitmq-server-ip>:<rabbitmq-server-port>/<virtual-
server>" --queue="my-testq"

# Get the messages (Push mode)
amqp-consume --url="amqp://<user-name>:<secret>@<rabbitmq-server-ip>:<rabbitmq-server-port>/<virtual-
server>" --queue="my-testq" -p 2 ./show.sh
```

## show.sh:

```
#!/usr/bin/env bash
read line
echo "Message: $line"
sleep 1
```

# RabbitMQ Standalone

## Opened Port if firewall is used

- [Configuration](#)

### Ports Required:

- 4369: epmd, a peer discovery service used by RabbitMQ nodes and CLI tools
- 5672, 5671: used by AMQP 0-9-1 and 1.0 clients without and with TLS
- 25672: used for inter-node and CLI tools communication (Erlang distribution server port) and is allocated from a dynamic range (limited to a single port by default, computed as AMQP port + 20000). Unless external connections on these ports are really necessary (e.g. the cluster uses federation or CLI tools are used on machines outside the subnet), these ports should not be publicly exposed. See networking guide for details.
- 35672-35682: used by CLI tools (Erlang distribution client ports) for communication with nodes and is allocated from a dynamic range (computed as server distribution port + 10000 through server distribution port + 10010). See networking guide for details.
- 15672: HTTP API clients, management UI and rabbitmqadmin (only if the management plugin is enabled)
- 61613, 61614: STOMP clients without and with TLS (only if the STOMP plugin is enabled)
- 1883, 8883: MQTT clients without and with TLS, if the MQTT plugin is enabled
- 15674: STOMP-over-WebSockets clients (only if the Web STOMP plugin is enabled)
- 15675: MQTT-over-WebSockets clients (only if the Web MQTT plugin is enabled)
- 15692: Prometheus metrics (only if the Prometheus plugin is enabled)

## Listener Port

### Change the default port 5672

Edit: `/etc/rabbitmq/rabbitmq.conf`

```
## Networking
## =====
##
## Related doc guide: https://rabbitmq.com/networking.html.
##
```



```
## By default, RabbitMQ will listen on all interfaces, using
## the standard (reserved) AMQP 0-9-1 and 1.0 port.
##
# listeners.tcp.default = 5672
listeners.tcp.default = 15690
```

## Restart the RabbitMQ Service

```
# Using systemctl
systemctl stop rabbitmq-server
systemctl start rabbitmq-server

# Alternatively, using rabbitmqctl
rabbitmqctl stop_app
rabbitmqctl start_app
```

# Users and Permissions

## Default User Access

The broker creates a user `guest` with password `guest`. Unconfigured clients will in general use these credentials. **By default, these credentials can only be used when connecting to the broker as localhost** so you will need to take action before connecting from any other machine.

See the documentation on [access control](#) for information on how to create more users and delete the guest user.

## Adding/Listing/Deleting Users

```
## Adding Users
# will prompt for password, only use this option interactively
rabbitmqctl add_user "username"

# Password is provided via standard input.
# Note that certain characters such as $, &, &, #, and so on must be escaped to avoid
# special interpretation by the shell.
echo '2a55f70a841f18b97c3a7db939b7adc9e34a0f1b' | rabbitmqctl add_user 'username'

Password is provided as a command line argument.
# Note that certain characters such as $, &, &, #, and so on must be escaped to avoid
```

```
# special interpretation by the shell.
rabbitmqctl add_user 'username' '2a55f70a841f18b97c3a7db939b7adc9e34a0f1b'

## Listing User
rabbitmqctl list_users
rabbitmqctl list_users --formatter=json

## Deleting a user
rabbitmqctl delete_user 'username'

## Verifying a user
rabbitmqctl authenticate_user "a-username" "a-password"
```

## Granting Permissions to a User

```
# First ".*" for configure permission on every entity
# Second ".*" for write permission on every entity
# Third ".*" for read permission on every entity
rabbitmqctl set_permissions -p "custom-vhost" "username" ".*" ".*" ".*"

# tag the user with "administrator" for full management UI and HTTP API access
rabbitmqctl set_user_tags username administrator

## Verifying the permission
# => Listing permissions for vhost "/" ...
# => user    configure  write  read
# => user2   .*  .*  .*
# => guest   .*  .*  .*
# => temp-user .*  .*  .*
rabbitmqctl list_permissions --vhost /
rabbitmqctl list_permissions --vhost gw1
```

## Clearing Permissions of a User in a Virtual Host

```
# Revokes permissions in a virtual host
rabbitmqctl clear_permissions -p "custom-vhost" "username"
```

## Operations on Multiple Virtual Hosts

```
# Assumes a Linux shell.
# Grants a user permissions to all virtual hosts.
for v in $(rabbitmqctl list_vhosts --silent); do rabbitmqctl set_permissions -p $v "a-user" ".*" ".*" ".*"; done
```

## Virtual Hosts

### Creating a Virtual Host

```
## Using CLI Tools
rabbitmqctl add_vhost qa1

## Using HTTP API
curl -u username:pa$sw0rD -X PUT http://rabbitmq.local:15672/api/vhosts/vh1
```

### Deleting a Virtual Host

```
## Using CLI Tools
rabbitmqctl delete_vhost qa1

## Using HTTP API
curl -u username:pa$sw0rD -X DELETE http://rabbitmq.local:15672/api/vhosts/vh1
```

## Kernel Limits

RabbitMQ nodes are most commonly affected by the maximum open file handle limit. Default limit value on most Linux distributions is usually 1024, which is very low for a messaging broker (or generally, any data service). See [Production Checklist](#) for recommended values.

## Open File Limit

With systemd (Modern Linux Distributions)

?: `/etc/systemd/system/rabbitmq-server.service.d/limits.conf`

```
[Service]
LimitNOFILE=64000
```

Restart the service

```
systemctl daemon-reload
systemctl stop rabbitmq-server
systemctl start rabbitmq-server
```

## Verify the change

```
#> ps -ef | grep rabbitmq
```

```
rabbitmq 460668 460654 0 11:43 ?      00:00:00 erl_child_setup 64000 <=====
```

```
#> rabbitmqctl status
```

File Descriptors

Total: 236, limit: 63903

Sockets: 230, limit: 57510

## TLS Connection

- [Troubleshooting TLS-enabled Connections — RabbitMQ](#)
- [TLS Support — RabbitMQ](#)

## Data Directory

- [File and Directory Locations — RabbitMQ](#)

### Environment Variables

Name	Description
RABBITMQ_BASE	Note: Windows only. This base directory contains sub-directories for the RabbitMQ server's database and log files. Alternatively, set RABBITMQ_MNESIA_BASE and RABBITMQ_LOG_BASE individually.
RABBITMQ_MNESIA_BASE	This base directory contains sub-directories for the RabbitMQ server's node database, message store and cluster state files, one for each node, unless RABBITMQ_MNESIA_DIR is set explicitly. It is important that effective RabbitMQ user has sufficient permissions to read, write and create files and subdirectories in this directory at any time. This variable is typically not overridden. Usually RABBITMQ_MNESIA_DIR is overridden instead.

RABBITMQ_MNESIA_DIR	The directory where this RabbitMQ node's data is stored. This s a schema database, message stores, cluster member information and other persistent node state.

Default Locations for Linux

Name	Location
RABBITMQ_MNESIA_BASE	\${install_prefix}/var/lib/rabbitmq/mnesia
RABBITMQ_MNESIA_DIR	\$RABBITMQ_MNESIA_BASE/\$RABBITMQ_NODENAME

Default Locations for Windows

Name	Location
RABBITMQ_BASE	%APPDATA%\RabbitMQ
RABBITMQ_MNESIA_BASE	%RABBITMQ_BASE%\db
RABBITMQ_MNESIA_DIR	%RABBITMQ_MNESIA_BASE%\%RABBITMQ_NODENAME%-mnesia

# Erlang Cookie

- [How CLI Tools Authenticate to Nodes \(and Nodes to Each Other\): the Erlang Cookie](#)

RabbitMQ nodes ? CLI tools ?? cookie ????????cookie ??????

- /var/lib/rabbitmq/.erlang.cookie (used by the server)?
- \$HOME/.erlang.cookie (used by the CLI tools)?

????

```
rabbitmq-diagnostics erlang_cookie_sources
```

????

- cookie ???? rabbitmq ????????
- ??? node ??????? cookie ??
- cookie ??????? Cluster ????????
- ?? cookie ??????? RabbitMQ ????Erlang VM ???????????

# Q & A

## How to Find Config File Location

### 1. Check the Log File

```
node      : rabbit@example
home dir   : /var/lib/rabbitmq
config file(s) : /etc/rabbitmq/advanced.config
              : /etc/rabbitmq/rabbitmq.conf
```

### 2. The comand `rabbitmq-diagnostics` or `rabbitmqctl`

```
rabbitmq-diagnostics status
rabbitmq-diagnostics status -n [node name]
rabbitmqctl status
```

??????????

Platform	Default Configuration File Directory	Example Configuration File Paths
Generic binary package	<code>\$RABBITMQ_HOME/etc/rabbitmq/</code>	<code>\$RABBITMQ_HOME/etc/rabbitmq/rabbitmq.conf</code> , <code>\$RABBITMQ_HOME/etc/rabbitmq/advanced.config</code>
Debian and Ubuntu	<code>/etc/rabbitmq/</code>	<code>/etc/rabbitmq/rabbitmq.conf</code> , <code>/etc/rabbitmq/advanced.config</code>
RPM-based Linux	<code>/etc/rabbitmq/</code>	<code>/etc/rabbitmq/rabbitmq.conf</code> , <code>/etc/rabbitmq/advanced.config</code>
Windows	<code>%APPDATA%\RabbitMQ\</code>	<code>%APPDATA%\RabbitMQ\rabbitmq.conf</code> , <code>%APPDATA%\RabbitMQ\advanced.config</code>
MacOS Homebrew Formula	<code>\${install_prefix}/etc/rabbitmq/</code> , and the Homebrew cellar prefix is usually <code>/usr/local</code>	<code>\${install_prefix}/etc/rabbitmq/rabbitmq.conf</code> , <code>\${install_prefix}/etc/rabbitmq/advanced.config</code>

# Install on RedHat

## Reference

- [Installing on RPM-based Linux \(RedHat Enterprise Linux, CentOS, Fedora, openSUSE\) — RabbitMQ](#)

## Using PackageCloud Yum Repository

### Quick-Install

Prerequisites: Internet connection

```
## Uses a PackageCloud-provided Yum repository setup script.  
## Always verify what is downloaded before piping it to a privileged shell!  
curl -s https://packagecloud.io/install/repositories/rabbitmq/rabbitmq-server/script.rpm.sh | sudo bash
```

### Manual-Install

Import the signing keys required

```
## primary RabbitMQ signing key  
rpm --import https://github.com/rabbitmq/signing-keys/releases/download/2.0/rabbitmq-release-signing-key.asc  
## modern Erlang repository  
rpm --import https://packagecloud.io/rabbitmq/erlang/gpgkey  
## RabbitMQ server repository  
rpm --import https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey
```

Add Yum Repositories for RabbitMQ and Modern Erlang

```
# In /etc/yum.repos.d/rabbitmq.repo  
  
##  
## Zero dependency Erlang  
##  
  
[rabbitmq_erlang]
```

```
name=rabbitmq_erlang
baseurl=https://packagecloud.io/rabbitmq/erlang/el/8/$basearch
repo_gpgcheck=1
gpgcheck=1
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/erlang/gpgkey
    https://github.com/rabbitmq/signing-keys/releases/download/2.0/rabbitmq-release-signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

#### [rabbitmq\_erlang-source]

```
name=rabbitmq_erlang-source
baseurl=https://packagecloud.io/rabbitmq/erlang/el/8/SRPMS
repo_gpgcheck=1
gpgcheck=0
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/erlang/gpgkey
    https://github.com/rabbitmq/signing-keys/releases/download/2.0/rabbitmq-release-signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

##

## RabbitMQ server

##

#### [rabbitmq\_server]

```
name=rabbitmq_server
baseurl=https://packagecloud.io/rabbitmq/rabbitmq-server/el/8/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
# PackageCloud's repository key and RabbitMQ package signing key
gpgkey=https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey
    https://github.com/rabbitmq/signing-keys/releases/download/2.0/rabbitmq-release-signing-key.asc
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```



```
metadata_expire=300

[rabbitmq_server-source]
name=rabbitmq_server-source
baseurl=https://packagecloud.io/rabbitmq/rabbitmq-server/el/8/SRPMS
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey
sslverify=1
sslcert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
```

## Install Packages with Yum

```
# Update Yum package metadata
yum update
yum -q makecache --disablerepo='*' --enablerepo='rabbitmq_erlang' --enablerepo='rabbitmq_server'

## install these dependencies from standard OS repositories
yum install socat logrotate

## install RabbitMQ and zero dependency Erlang from the above repositories,
## ignoring any versions provided by the standard repositories
yum install --repo rabbitmq_erlang --repo rabbitmq_server erlang rabbitmq-server
```

## Run RabbitMQ Server

```
systemctl status rabbitmq-server
systemctl stop rabbitmq-server
systemctl start rabbitmq-server
systemctl enable rabbitmq-server
```

### Check the status

Run: `rabbitmqctl status` or `rabbitmq-diagnostics status`

```
#> rabbitmq-diagnostics status

Status of node rabbit@tpeeaprmq98 ...
```

## Runtime

OS PID: 549246

OS: Linux

Uptime (seconds): 504837

Is under maintenance?: false

RabbitMQ version: 3.10.7

Node name: rabbit@tpeeaprmq98

Erlang configuration: Erlang/OTP 25 [erts-13.0.4] [source] [64-bit] [smp:2:2] [ds:2:2:10] [async-threads:1] [jit:ns]

Crypto library: OpenSSL 1.1.1k FIPS 25 Mar 2021

Erlang processes: 412 used, 1048576 limit

Scheduler run queue: 1

Cluster heartbeat timeout (net\_ticktime): 60

## Plugins

Enabled plugin file: /etc/rabbitmq/enabled\_plugins

Enabled plugins:

- \* rabbitmq\_mqtt
- \* rabbitmq\_management
- \* amqp\_client
- \* rabbitmq\_web\_dispatch
- \* cowboy
- \* cowlib
- \* rabbitmq\_management\_agent

## Data directory

Node data directory: /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq98

Raft data directory: /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq98/quorum/rabbit@tpeeaprmq98

## Config files

## Log file(s)

- \* /var/log/rabbitmq/rabbit@tpeeaprmq98.log
- \* /var/log/rabbitmq/rabbit@tpeeaprmq98\_upgrade.log

\* <stdout>

## Alarms

(none)

## Memory

Total memory used: 0.1408 gb

Calculation strategy: rss

Memory high watermark setting: 0.4 of available memory, computed to: 1.526 gb

reserved\_unallocated: 0.0882 gb (62.67 %)

code: 0.0351 gb (24.97 %)

other\_proc: 0.0201 gb (14.28 %)

other\_system: 0.0139 gb (9.85 %)

other\_ets: 0.0032 gb (2.26 %)

plugins: 0.0018 gb (1.29 %)

atom: 0.0014 gb (1.01 %)

binary: 0.0009 gb (0.62 %)

metrics: 0.0006 gb (0.4 %)

mgmt\_db: 0.0003 gb (0.22 %)

mnesia: 0.0001 gb (0.07 %)

msg\_index: 0.0001 gb (0.07 %)

queue\_procs: 0.0 gb (0.03 %)

quorum\_ets: 0.0 gb (0.02 %)

connection\_other: 0.0 gb (0.01 %)

quorum\_queue\_dlx\_procs: 0.0 gb (0.0 %)

stream\_queue\_procs: 0.0 gb (0.0 %)

stream\_queue\_replica\_reader\_procs: 0.0 gb (0.0 %)

allocated\_unused: 0.0 gb (0.0 %)

connection\_channels: 0.0 gb (0.0 %)

connection\_readers: 0.0 gb (0.0 %)

connection\_writers: 0.0 gb (0.0 %)

queue\_slave\_procs: 0.0 gb (0.0 %)

quorum\_queue\_procs: 0.0 gb (0.0 %)

stream\_queue\_coordinator\_procs: 0.0 gb (0.0 %)

## File Descriptors

Total: 6, limit: 63903

Sockets: 0, limit: 57510

#### Free Disk Space

Low free disk space watermark: 0.05 gb

Free disk space: 5.4187 gb

#### Totals

Connection count: 0

Queue count: 2

Virtual host count: 3

#### Listeners

Interface: [::], port: 15672, protocol: http, purpose: HTTP API

Interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication

Interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

Interface: [::], port: 1883, protocol: mqtt, purpose: MQTT

## FAQ

### GPG Key Error

“ Errors during downloading metadata for repository 'rabbitmq\_server':  
- Curl error (35): SSL connect error for https://github.com/rabbitmq/signing-keys/releases/download/2.0/rabbitmq-release-signing-key.asc [OpenSSL SSL\_connect: SSL\_ERROR\_SYSCALL in connection to github.com:443 ]

Solution: ?????????????? github.com????????? GPG key ?????????????? `/etc/pki/rpm-gpg`  
???? `/etc/yum.repos.d/rabbitmq.repo`

```
# PackageCloud's repository key and RabbitMQ package signing key
```

```
gpgkey=https://packagecloud.io/rabbitmq/rabbitmq-server/gpgkey
```

```
file:///etc/pki/rpm-gpg/rabbitmq-release-signing-key.asc
```

```
# https://github.com/rabbitmq/signing-keys/releases/download/2.0/rabbitmq-release-signing-key.asc
```



# Management UI

The RabbitMQ management plugin provides an HTTP-based API for management and monitoring of RabbitMQ nodes and clusters, along with a browser-based UI and a command line tool, [rabbitmqadmin](https://www.rabbitmq.com/rabbitmqadmin).

- <https://www.rabbitmq.com/management.html>

## Getting Started

```
rabbitmq-plugins enable rabbitmq_management
```

Node restart is not required after plugin activation.

Auto Enable the plugin: <https://www.rabbitmq.com/plugins.html#enabled-plugins-file>

## Management UI Access

The management UI can be accessed using a Web browser at `http://{node-hostname}:15672/`.

## Access and Permissions

```
# create a user
rabbitmqctl add_user sysadmin ThisIsPassword

# tag the user with "administrator" for full management UI and HTTP API access
rabbitmqctl set_user_tags sysadmin administrator
```

# Plugins

<https://www.rabbitmq.com/plugins.html>

A list of plugins available locally

```
rabbitmq-plugins list
```

Enable/Disable the plugin

```
rabbitmq-plugins enable rabbitmq_management  
rabbitmq-plugins disable rabbitmq_management
```

## Different Ways to Enable Plugins

“ NOTE: ??? 3.10.7 ?????????????????? enable/disable ??????????????????

```
[root@tpeeaprmq98 ~]# rabbitmq-plugins directories -s  
Plugin archives directory: /usr/lib/rabbitmq/plugins:/usr/lib/rabbitmq/lib/rabbitmq_server-3.10.7/plugins  
Plugin expansion directory: /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq98-plugins-expand  
Enabled plugins file: /etc/rabbitmq/enabled_plugins
```

Edit the file: `/etc/rabbitmq/enabled_plugins`

```
[rabbitmq_management,rabbitmq_amqp1_0,rabbitmq_mqtt].
```

????

```
systemctl stop rabbitmq-server  
systemctl start rabbitmq-server
```

# Monitoring & Management

## Web UI

### Overview

- Ready: ????? Queue ????
- Unacked: ????????? Ack ????
- Total: Ready + Unacked

## CLI

### rabbitmq-diagnostics

#### Online Resource Utilization

```
rabbitmq-diagnostics observer
```

#### RabbitMQ Version

```
[root@tpeeaprmq98 ~]# rabbitmq-diagnostics server_version
Asking node rabbit@tpeeaprmq98 for its RabbitMQ version...
3.10.7
```

#### Check the listener ports

```
rabbitmq-diagnostics -s listeners
```

## rabbitmqctl

### List the queues

```
rabbitmqctl -p <vhost-name> list_queues name state durable arguments policy
```

```
rabbitmqctl -qs -p <vhost-name> list_queues name > queue_names.lst
```

### User Management



- ?? tag ??: [Management Plugin — RabbitMQ](#)

```
# List all users
rabbitmqctl list_users

# Create a new user
rabbitmqctl add_user "eapuser"
rabbitmqctl add_vhost "eap_server"
rabbitmqctl set_permissions -p "eap_server" "eapuser" ".*" ".*" ".*"
rabbitmqctl set_user_tags eapuser monitoring # for web login only

# Remove tag 'monitoring' from user if needed
rabbitmqctl set_user_tags eapuser ""

# Tag the user with "administrator" for full management UI and HTTP API access
rabbitmqctl set_user_tags username administrator

## Verifying the permission
# => Listing permissions for vhost "/" ...
# => user    configure write read
# => user2    .* .* .*
# => guest    .* .* .*
# => temp-user .* .* .*
rabbitmqctl list_permissions --vhost /
rabbitmqctl list_permissions --vhost gw1

# Revoke user access
rabbitmqctl delete_user 'username'
```

## Connections

```
rabbitmqctl list_connections
rabbitmqctl list_connections user,peer_host,peer_port,channels,state
```

## Recreate the virtual host

```
rabbitmqctl delete_vhost <my-vhost-name>
rabbitmqctl add_vhost <my-vhost-name>
```

## Reset the RabbitMQ Node



The broker drops all virtual hosts, queues, exchanges, and non-administrative users.

```
rabbitmqctl stop_app  
rabbitmqctl reset  
rabbitmqctl start_app
```

## Force Reset the RabbitMQ Node

“ ?? Cluster ?????????????????? Node?

```
rabbitmqctl stop_app  
rabbitmqctl force_reset  
rabbitmqctl start_app
```

## Suspend all listeners and prevent new client connections

“ ?? listener ???????? `ss -ltpn` ????? listener port ?????

????????????????????????????

```
# For current node  
rabbitmqctl suspend_listeners  
  
# suspends listeners on node rabbit@node2.cluster.rabbitmq.svc: it won't accept any new client connections  
rabbitmqctl suspend_listeners -n rabbit@node2.cluster.rabbitmq.svc  
  
# For current node, to resume all listeners on a node and make it accept new client connections again  
rabbitmqctl resume_listeners  
  
# resumes listeners on node rabbit@node2.cluster.rabbitmq.svc: it will accept new client connections again  
rabbitmqctl resume_listeners -n rabbit@node2.cluster.rabbitmq.svc
```

# rabbitmqadmin

## Basic Operation

```
# List queues
rabbitmqadmin list queues

rabbitmqadmin -H <RabbitMQ-Server-IP> -u <username> -p <password> -V <vhost-name> list queues


# Add a queue with optional parameters
rabbitmqadmin declare queue name=<my-new-queue> durable=true auto_delete=true
```

## Remove multiple queues

```
rabbitmqadmin -f tsv -q list queues name > q.txt

while read -r name; do rabbitmqadmin -q delete queue name="${name}"; done < q.txt
```

## Connections

```
rabbitmqadmin -H <RabbitMQ-Server-IP> -u <username> -p <password> -V <vhost-name> list connections
name


# Close multiple connections without any channels
rabbitmqadmin -f tsv -q connections name channels | awk -F "\t" '($2 < 1) {print $1}' | tee conn_noChannels.lst
while read -r conn;do rabbitmqadmin close connection name="${conn}"; done < conn_noChannels.lst
```

# Monitoring

## Health-Check

```
rabbitmq-diagnostics check_running
rabbitmq-diagnostics ping
```

## Queue State

```
rabbitmqctl list_queues name state
```

## Cluster Status

```
rabbitmqctl cluster_status
```

# Monitor with Prometheus

- [Monitoring with Prometheus & Grafana — RabbitMQ](#)
- [First steps | Prometheus](#)

## RabbitMQ Configuration

```
# Enable the plugin rabbitmq_prometheus
rabbitmq-plugins enable rabbitmq_prometheus

# To confirm that RabbitMQ now exposes metrics in Prometheus format
curl -s localhost:15692/metrics | head -n 10
```

## Prometheus Configuration

/etc/hosts:

```
<node1.ip.addr> rmq01
<node2.ip.addr> rmq02
<node3.ip.addr> rmq03
```

prometheus.yml:

```
scrape_configs:
  - job_name: rabbitmq

    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 15s

    static_configs:
      - targets: ['rmq01:15692', 'rmq02:15692', 'rmq03:15692']
```

## Grafana Configuration

- [RabbitMQ Overview | Grafana Labs](#)
- [Dashboard: RabbitMQ-Overview](#)

Dashboard: RabbitMQ-Overview

- Download: <https://grafana.com/grafana/dashboards/10991-rabbitmq-overview/>
- Plugins
  - Stat (built-in)
  - Table (built-in)

- Time series (built-in)

# RabbitMQ Cluster

- [Clustering Guide — RabbitMQ](#)
- [RabbitMQ Learning III: RabbitMQ Clustering and Load Balancing](#)
- [How to Set Up the RabbitMQ Cluster on Ubuntu/Debian Linux](#)
- [Clustering Guide \(vmware.com\)](#)
- [RabbitMQ ???? \(with HAProxy\)](#)
- [RabbitMQ?????? | guaosi???](#)

For Windows only

- [??? Windwos ??? RabbitMQ Cluster](#)

## A few things to RabbitMQ Cluster

- Classic Queues ????? Cluster ???? node???????????? node ???????? Queues  
?????? nodes?????? [Quorum Queues](#)?
- Cluster ???? node ?????? peer????????????????????
- Cluster ??? node ???? cookie ??????cookie ???? `/var/lib/rabbitmq/.erlang.cookie` ?
- ?? Cluster ?? ? Node ????????????? 3?5?7???????????????? node ???????? Cluster  
????????????????
- ?? Node ????????????? 1 ? Node ????????????????????? 2 ?Node  
????????????????????
- Cluster ???????? LAN?????? WAN???????? Cluster ??? Node  
???????????????????? Node ???????? 60 ??Cluster ???? Network Partition ??????  
split-brain (??) ?? ????????? Cluster ?????????????????[Clustering and Network  
Partitions](#)
- ?? Cluster ?????????????????Queues?Exchanges?Routing Key????????????????
- Node Plugin ????????

????

1. tpeeaprmq98 (node01)
2. tpeeaprmq981 (node02)
3. tpeeaprmq982 (node03)

/etc/hosts:

10.14.2.51	tpeeaprmq98
10.4.1.33	tpeeaprmq981
10.4.1.34	tpeeaprmq982

## ?? RabbitMQ

????????? RabbitMQ ??????

## ?? Cookie ???

?? node ?????? Erlang cookie ?????

???? RabbitMQ ???cookie ????????????????????????????????? 0600 ???? Cluster ???  
node ?????? cookit ??

```
scp /var/lib/rabbitmq/.erlang.cookie root@tpeeaprmq981:/var/lib/rabbitmq/  
scp /var/lib/rabbitmq/.erlang.cookie root@tpeeaprmq982:/var/lib/rabbitmq/
```

cookie ???

“ ?????? /var/lib/rabbitmq/.erlang.cookie ?

rabbitmq-diagnostics erlang\_cookie\_sources

## ?? Cluster

Detach the service of all nodes

???? Cluster ????? node ?????? Cluster?

```
# On Node01  
rabbitmq-server -detached  
  
# On Node02  
rabbitmq-server -detached  
  
# On Node03
```

## Verify the cluster status

```
[root@tpeeaprmq98 ~]# rabbitmqctl cluster_status
```

Cluster status of node rabbit@tpeeaprmq98 ...

Basics

Cluster name: rabbit@tpeeaprmq98

Disk Nodes

rabbit@tpeeaprmq98

Running Nodes

rabbit@tpeeaprmq98

Versions

rabbit@tpeeaprmq98: RabbitMQ 3.10.7 on Erlang 25.0.4

Maintenance status

Node: rabbit@tpeeaprmq98, status: not under maintenance

Alarms

(none)

Network Partitions

(none)

Listeners

Node: rabbit@tpeeaprmq98, interface: [::], port: 15672, protocol: http, purpose: HTTP API

Node: rabbit@tpeeaprmq98, interface: [::], port: 1883, protocol: mqtt, purpose: MQTT

Node: rabbit@tpeeaprmq98, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication



Node: rabbit@tpeeaprmq98, interface: [::], port: 15690, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

#### Feature flags

Flag: classic\_mirrored\_queue\_version, state: enabled

Flag: drop\_unroutable\_metric, state: disabled

Flag: empty\_basic\_get\_metric, state: disabled

Flag: implicit\_default\_bindings, state: enabled

Flag: maintenance\_mode\_status, state: enabled

Flag: quorum\_queue, state: enabled

Flag: stream\_queue, state: enabled

Flag: user\_limits, state: enabled

Flag: virtual\_host\_metadata, state: enabled

```
[root@tpeeaprmq981 ~]# rabbitmqctl cluster_status
```

Cluster status of node rabbit@tpeeaprmq981 ...

#### Basics

Cluster name: rabbit@tpeeaprmq981

#### Disk Nodes

rabbit@tpeeaprmq981

#### Running Nodes

rabbit@tpeeaprmq981

#### Versions

rabbit@tpeeaprmq981: RabbitMQ 3.10.7 on Erlang 25.0.4

#### Maintenance status

Node: rabbit@tpeeaprmq981, status: not under maintenance

#### Alarms

(none)

## Network Partitions

(none)

## Listeners

Node: rabbit@tpeeaprmq981, interface: [::], port: 15672, protocol: http, purpose: HTTP API

Node: rabbit@tpeeaprmq981, interface: [::], port: 1883, protocol: mqtt, purpose: MQTT

Node: rabbit@tpeeaprmq981, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication

Node: rabbit@tpeeaprmq981, interface: [::], port: 15690, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

## Feature flags

Flag: classic\_mirrored\_queue\_version, state: enabled

Flag: drop\_unroutable\_metric, state: enabled

Flag: empty\_basic\_get\_metric, state: enabled

Flag: implicit\_default\_bindings, state: enabled

Flag: maintenance\_mode\_status, state: enabled

Flag: quorum\_queue, state: enabled

Flag: stream\_queue, state: enabled

Flag: user\_limits, state: enabled

Flag: virtual\_host\_metadata, state: enabled

```
[root@tpeeaprmq982 ~]# rabbitmqctl cluster_status
```

Cluster status of node rabbit@tpeeaprmq982 ...

## Basics

Cluster name: rabbit@tpeeaprmq982

## Disk Nodes

rabbit@tpeeaprmq982

## Running Nodes

rabbit@tpeeaprmq982

## Versions

rabbit@tpeeaprmq982: RabbitMQ 3.10.7 on Erlang 25.0.4

#### Maintenance status

Node: rabbit@tpeeaprmq982, status: not under maintenance

#### Alarms

(none)

#### Network Partitions

(none)

#### Listeners

Node: rabbit@tpeeaprmq982, interface: [::], port: 15672, protocol: http, purpose: HTTP API

Node: rabbit@tpeeaprmq982, interface: [::], port: 1883, protocol: mqtt, purpose: MQTT

Node: rabbit@tpeeaprmq982, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication

Node: rabbit@tpeeaprmq982, interface: [::], port: 15690, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

#### Feature flags

Flag: classic\_mirrored\_queue\_version, state: enabled

Flag: drop\_unroutable\_metric, state: enabled

Flag: empty\_basic\_get\_metric, state: enabled

Flag: implicit\_default\_bindings, state: enabled

Flag: maintenance\_mode\_status, state: enabled

Flag: quorum\_queue, state: enabled

Flag: stream\_queue, state: enabled

Flag: user\_limits, state: enabled

Flag: virtual\_host\_metadata, state: enabled

## Creating a Cluster

? node02 ? node03 ??? node01?

# On Node02

rabbitmqctl stop\_app

```
rabbitmqctl reset
rabbitmqctl join_cluster rabbit@tpeeaprmq98
rabbitmqctl start_app
```

```
# On Node03
rabbitmqctl stop_app
rabbitmqctl reset
rabbitmqctl join_cluster rabbit@tpeeaprmq98
rabbitmqctl start_app
```

?? cluster ??????????????????

```
[root@tpeeaprmq981 ~]# rabbitmqctl join_cluster rabbit@tpeeaprmq98
Clustering node rabbit@tpeeaprmq981 with rabbit@tpeeaprmq98
```

```
15:10:18.438 [warning] Feature flags: the previous instance of this node must have failed to write the
`feature_flags` file at `/var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-feature_flags`:
```

```
15:10:18.438 [warning] Feature flags: - list of previously disabled feature flags now marked as such:
[:maintenance_mode_status]
```

```
15:10:18.561 [warning] Feature flags: the previous instance of this node must have failed to write the
`feature_flags` file at `/var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-feature_flags`:
```

```
15:10:18.561 [warning] Feature flags: - list of previously enabled feature flags now marked as such:
[:maintenance_mode_status]
```

```
15:10:18.598 [error] Failed to create a tracked connection table for node :rabbit@tpeeaprmq981:
{:node_not_running, :rabbit@tpeeaprmq981}
```

```
15:10:18.598 [error] Failed to create a per-vhost tracked connection table for node :rabbit@tpeeaprmq981:
{:node_not_running, :rabbit@tpeeaprmq981}
```

```
15:10:18.599 [error] Failed to create a per-user tracked connection table for node :rabbit@tpeeaprmq981:
{:node_not_running, :rabbit@tpeeaprmq981}
```

Verify the cluster status

?? node ? Cluster ?????????????? Cluster name ?????????? node name?

```
[root@tpeeaprmq98 ~]# rabbitmqctl cluster_status
```

```
Cluster status of node rabbit@tpeeaprmq98 ...
```

#### Basics

```
Cluster name: rabbit@tpeeaprmq98
```

#### Disk Nodes

```
rabbit@tpeeaprmq98
```

```
rabbit@tpeeaprmq981
```

```
rabbit@tpeeaprmq982
```

#### Running Nodes

```
rabbit@tpeeaprmq98
```

```
rabbit@tpeeaprmq981
```

```
rabbit@tpeeaprmq982
```

#### Versions

```
rabbit@tpeeaprmq98: RabbitMQ 3.10.7 on Erlang 25.0.4
```

```
rabbit@tpeeaprmq981: RabbitMQ 3.10.7 on Erlang 25.0.4
```

```
rabbit@tpeeaprmq982: RabbitMQ 3.10.7 on Erlang 25.0.4
```

#### Maintenance status

```
Node: rabbit@tpeeaprmq98, status: not under maintenance
```

```
Node: rabbit@tpeeaprmq981, status: not under maintenance
```

```
Node: rabbit@tpeeaprmq982, status: not under maintenance
```

#### Alarms

```
(none)
```

#### Network Partitions

```
(none)
```

#### Listeners

Node: rabbit@tpeeaprmq98, interface: [::], port: 15672, protocol: http, purpose: HTTP API  
Node: rabbit@tpeeaprmq98, interface: [::], port: 1883, protocol: mqtt, purpose: MQTT  
Node: rabbit@tpeeaprmq98, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication  
Node: rabbit@tpeeaprmq98, interface: [::], port: 15690, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0  
Node: rabbit@tpeeaprmq981, interface: [::], port: 15672, protocol: http, purpose: HTTP API  
Node: rabbit@tpeeaprmq981, interface: [::], port: 1883, protocol: mqtt, purpose: MQTT  
Node: rabbit@tpeeaprmq981, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication  
Node: rabbit@tpeeaprmq981, interface: [::], port: 15690, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0  
Node: rabbit@tpeeaprmq982, interface: [::], port: 15672, protocol: http, purpose: HTTP API  
Node: rabbit@tpeeaprmq982, interface: [::], port: 1883, protocol: mqtt, purpose: MQTT  
Node: rabbit@tpeeaprmq982, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication  
Node: rabbit@tpeeaprmq982, interface: [::], port: 15690, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

#### Feature flags

Flag: classic\_mirrored\_queue\_version, state: enabled  
Flag: drop\_unroutable\_metric, state: enabled  
Flag: empty\_basic\_get\_metric, state: enabled  
Flag: implicit\_default\_bindings, state: enabled  
Flag: maintenance\_mode\_status, state: enabled  
Flag: quorum\_queue, state: enabled  
Flag: stream\_queue, state: enabled  
Flag: user\_limits, state: enabled  
Flag: virtual\_host\_metadata, state: enabled

## Node ??

?? Cluster???? Cluster ??????????

```
systemctl disable rabbitmq-server
```

## ?? Node

```
# Recommend using systemd  
systemctl stop rabbitmq-server  
systemctl start rabbitmq-server
```

```
# Using rabbitmqctl + systemd
rabbitmqctl stop
systemctl start rabbitmq-server
```

```
# Using rabbitmqctl
# Stop the node
rabbitmqctl stop
# Start the node
rabbitmq-server -detached
# Verify if the node is awaiting schema table sync
rabbitmq-diagnostics check_running

# Forcing node boot
rabbitmqctl force_boot
```

## ??? Node

“ NOTE: ??? Cluster ????????Queues?Exchange ?????????????? Node?

```
# Find out the path of Erlang Cookie file
rabbitmq-diagnostics erlang_cookie_sources
# Copy Cookie from one node of the cluster
scp /var/lib/rabbitmq/.erlang.cookie root@<new-node>:/var/lib/rabbitmq/

# Join a new node into the cluster rabbit@tpeeaprmq98
rabbitmqctl stop_app
rabbitmqctl reset
rabbitmqctl join_cluster <cluster-name>
rabbitmqctl start_app
# Alternatively, you can join it as RAM node by following command
rabbitmqctl join_cluster <cluster-name> --ram
```

## ?? Node

NOTE: Node ?? Cluster ??? Node ?????Queues?Exchange  
????????????

```
# [ ] node
# [ ] node [ ]
rabbitmqctl stop_app
rabbitmqctl reset
rabbitmqctl start_app
rabbitmqctl cluster_status

# [ ] node
# [ ] Cluster [ ] node [ ]
rabbitmqctl forget_cluster_node <node-name>
```

## Rebalance the queues across node

??? quorum queue ??? node ?????????????????? node????????????? **Leader node**?????  
**Follower node?**

Queue ????? Leader node ????? Leader node ????? Follower nodes  
???? Leader node?

??? quorum queue ? Leader node ????? node ?????????????????? node?

- After restarting a node
- After joining a node

“ NOTE: ?? queue ??? Leader node ??? Web-UI ???

queue ? Leader Node ? Channel Node ??????????????

```
rabbitmq-queues rebalance all
rabbitmq-queues rebalance "all" --vhost-pattern "itp_server" --queue-pattern ".*"
```





# FAQ

Q: ???? node ??

“ Application rabbit exited with reason:  
{could\_not\_write\_file,"/var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq982/cluste  
r\_nodes.config",enospc},{rabbit,start,[normal,[]]}

Solution:

???????????????? /var/lib/rabbitmq/mnesia ?? node ??????

```
[root@tpeeaprmq981 ~]# df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  1.8G   0 1.8G   0% /dev
tmpfs                     1.8G  4.0K 1.8G   1% /dev/shm
tmpfs                     1.8G  24M 1.8G   2% /run
tmpfs                     1.8G   0 1.8G   0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv 9.0G  4.0G 5.1G  45% /
/dev/sda2                 1014M 344M 671M  34% /boot
/dev/sda1                  599M  5.8M 594M   1% /boot/efi
/dev/mapper/rootvg-mqdata 5.0G  5.0G 20K 100% /var/lib/rabbitmq
/dev/mapper/rootvg-homelv 507M  30M 478M   6% /home
/dev/mapper/rootvg-worktmp 507M  46M 462M   9% /worktmp
/dev/mapper/rootvg-optlv  2.0G 997M 1.1G  49% /opt
tmpfs                     364M   0 364M   0% /run/user/0
[root@tpeeaprmq981 ~]#
[root@tpeeaprmq981 ~]#
[root@tpeeaprmq981 ~]# du -csh /var/lib/rabbitmq/mnesia/*
204K  /var/lib/rabbitmq/mnesia/rabbit@rmq981
4.0K  /var/lib/rabbitmq/mnesia/rabbit@rmq981-feature_flags
0     /var/lib/rabbitmq/mnesia/rabbit@rmq981-plugins-expand
300K  /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq98
5.0G  /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981
4.0K  /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-feature_flags
```

```

0    /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-plugins-expand
4.0K /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq98-feature_flags
0    /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq98-plugins-expand
5.0G total

[root@tpeeaprmq981 ~]# rm -rf /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981
[root@tpeeaprmq981 ~]# rm /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-feature_flags
rm: remove regular file '/var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-feature_flags'? y
[root@tpeeaprmq981 ~]#
[root@tpeeaprmq981 ~]# rm /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-plugins-expand
rm: cannot remove '/var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-plugins-expand': Is a directory
[root@tpeeaprmq981 ~]# rm -rf /var/lib/rabbitmq/mnesia/rabbit@tpeeaprmq981-plugins-expand

```

## Q: ???? Cluster

“ [error] Node rabbit@tpeeaprmq98 thinks it's clustered with node rabbit@tpeeaprmq982, but rabbit@tpeeaprmq982 disagrees

Solution:

? node rabbit@tpeeaprmq98 ?? rabbitmqctl cluster\_status ?????? node rabbit@tpeeaprmq982 ????????????

```

# On the node rabbit@tpeeaprmq98
rabbitmqctl forget_cluster_node rabbit@tpeeaprmq982

```

## Q: Network partition detected

Web UI ????????

### Overview

#### Network partition detected

Mnesia reports that this RabbitMQ cluster has experienced a network partition. There is a risk of losing data. Please read [RabbitMQ documentation about network partitions and the possible solutions](#).

The nature of the partition is as follows:

Node	Was partitioned from
rabbit@tpeeaprmq98	rabbit@tpeeaprmq982
rabbit@tpeeaprmq981	rabbit@tpeeaprmq982
rabbit@tpeeaprmq982	rabbit@tpeeaprmq98
	rabbit@tpeeaprmq981

While running in this partitioned state, changes (such as queue or exchange declaration and binding) which take place in one partition will not be visible to other partition(s). Other behaviour is not guaranteed.

Node ?? rabbitmqctl cluster\_status ?? Network Partitions

## Network Partitions

Node rabbit@tpeeaprmq98 cannot communicate with rabbit@tpeeaprmq982

Node rabbit@tpeeaprmq981 cannot communicate with rabbit@tpeeaprmq982

???tpeeaprmq982 ?????????????????? node ??????????Cluster ??? Network Partition  
??(aka split-brain ???)???? quorum queue ????????????????

??????? Network Partition ????

- ????: [Clustering and Network Partitions — RabbitMQ](#)

??????? Node tpeeaprmq982 ???

rabbitmqctl stop

systemctl start rabbitmq-server

