

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>

Revision #34

Created 4 August 2022 05:30:23 by Admin

Updated 6 October 2022 14:44:17 by Admin