

---

# **OpenStack-Ansible Documentation: rabbitmq\_server role**

*Release 18.1.0.dev309*

**OpenStack-Ansible Contributors**

**Mar 03, 2025**



# CONTENTS

- 1 Table of Contents** **3**
- 1.1 Scenario - Configuring RabbitMQ . . . . . 3
- 2 Default variables** **5**
- 3 Dependencies** **13**
- 4 Example playbook** **15**



This Ansible role deploys RabbitMQ. When multiple hosts are present in the `rabbitmq_all` inventory group, a cluster is created.



## TABLE OF CONTENTS

### 1.1 Scenario - Configuring RabbitMQ

RabbitMQ provides the messaging broker for various OpenStack services. The OpenStack-Ansible project configures a plain text listener on port 5672 and a SSL/TLS encrypted listener on port 5671.

Customize your RabbitMQ deployment in `/etc/openstack_deploy/user_variables.yml`.

#### 1.1.1 Add a TLS encrypted listener to RabbitMQ

The OpenStack-Ansible project provides the ability to secure RabbitMQ communications with self-signed or user-provided SSL certificates. Refer to Securing services with SSL certificates in the OSA Install Guide for available configuration options.

#### 1.1.2 Enable encrypted connections to RabbitMQ

The control of SSL communication between various OpenStack services and RabbitMQ is via the Ansible variable `rabbitmq_use_ssl`:

```
rabbitmq_use_ssl: true
```

Setting this variable to `true` adjusts the RabbitMQ port to 5671 (the default SSL/TLS listener port) and enables SSL connectivity between each OpenStack service and RabbitMQ.

Setting this variable to `false` disables SSL encryption between OpenStack services and RabbitMQ and configures all services to use the plain text port, 5672.

#### 1.1.3 Add custom configuration to RabbitMQ

To add custom configuration, e.g. to configure installed plugins, can be accomplished by defining a variable `rabbitmq_additional_config` and add you configuration as a block to this variable. Make sure to not add redundant configuration

```
rabbitmq_additional_config:  
  prometheus.tcp.port: 15680  
  prometheus.tcp.ip: 0.0.0.0
```

To clone or view the source code for this repository, visit the role repository for `rabbitmq_server`.





## DEFAULT VARIABLES

```
## APT Cache Options
cache_timeout: 600

# Set the package install state for distribution packages
# Options are 'present' and 'latest'
rabbitmq_package_state: "{{ package_state | default('latest') }}"

# Defined versions of RabbitMQ and Erlang
rabbitmq_package_version: "4.0.3-1"
rabbitmq_erlang_package_version: "26.2.*-1"

# Inventory group containing the hosts for the cluster
rabbitmq_host_group: "rabbitmq_all"

# The local address used for the rabbitmq cluster node
rabbitmq_node_address: "{{ ansible_host }}"

rabbit_system_user_name: rabbitmq
rabbit_system_group_name: rabbitmq

# Allow role to adjust /etc/hosts file
rabbitmq_manage_hosts_entries: true

# Hosts file entries
rabbitmq_hosts_entries: >-
  {{ groups[rabbitmq_host_group] | map('extract', hostvars) | list |
    json_query(
      "[].{address: rabbitmq_node_address || ansible_host , hostnames:
↪[ansible_facts.hostname, ansible_facts.fqdn] }"
    )
  }}

rabbitmq_primary_cluster_node: "{{ hostvars[groups[rabbitmq_host_group][0]][
↪'ansible_facts']['hostname'] }}"

# Upgrading the RabbitMQ package requires shutting down the cluster. This
↪variable makes upgrading
# the version an explicit action.
```

(continues on next page)

(continued from previous page)

```
rabbitmq_upgrade: false

# If the user does not want to upgrade but needs to rerun the playbooks for
↳any reason the
# upgrade/version state can be ignored by setting `rabbitmq_ignore_version_
↳state=true`
rabbitmq_ignore_version_state: false

rabbitmq_package_url: ""
rabbitmq_package_sha256: ""
rabbitmq_package_path: ""

# Mappings from Ansible reported architecture to distro release architecture
rabbitmq_architecture_mapping:
  x86_64: amd64
  ppc64le: ppc64el
  s390x: s390x
  armv7l: armhf
  aarch64: arm64

# Set the gpg keys needed to be imported
# This should be a list of dicts, with each dict
# giving a set of arguments to the applicable
# package module. The following is an example for
# systems using the apt package manager.
# rabbitmq_gpg_keys:
#   - id: '0xC2E73424D59097AB'
#     keyserver: 'hkp://keyserver.ubuntu.com:80'
#     validate_certs: no
rabbitmq_gpg_keys: "{{ _rabbitmq_gpg_keys | default([]) }}"

# Set the URL for the RabbitMQ repository
rabbitmq_repo_url: "{{ _rabbitmq_repo_url | default(null) }}"

# Set the repo information for the RabbitMQ repository
rabbitmq_repo: "{{ _rabbitmq_repo | default({}) }}"

# Set the URL for the Erlang repository
rabbitmq_erlang_repo_url: "{{ _rabbitmq_erlang_repo_url | default(null) }}"

# Set the repo information for the Erlang repository
rabbitmq_erlang_repo: "{{ _rabbitmq_erlang_repo | default({}) }}"

# Choose file, distro, external_repo for rabbitmq_install_method.
rabbitmq_install_method: "{{ _rabbitmq_install_method }}"
rabbitmq_erlang_install_method: "{{ _rabbitmq_erlang_install_method |
↳default(rabbitmq_install_method) }}"

# Name of the rabbitmq cluster
```

(continues on next page)

(continued from previous page)

```
rabbitmq_cluster_name: rabbitmq_cluster1

# Specify a partition recovery strategy (autoheal | pause_minority | ignore)
rabbitmq_cluster_partition_handling: pause_minority

# Rabbitmq open file limits
rabbitmq_ulimit: 65536

# Configure rabbitmq plugins
# This should be a comma-separated list of plugin names.
# Any plugin not listed will be disabled automatically.
# rabbitmq_plugins:
#   - name: rabbitmq_management,rabbitmq_prometheus
#     state: enabled
rabbitmq_plugins:
  - name: rabbitmq_management
    state: enabled

# Storage location for SSL certificate authority
rabbitmq_pki_dir: "{{ openstack_pki_dir | default('/etc/pki/rabbitmq-ca') }}"

# Delegated host for operating the certificate authority
rabbitmq_pki_setup_host: "{{ openstack_pki_setup_host | default('localhost') }}
↪}"

# Create a certificate authority if one does not already exist
rabbitmq_pki_create_ca: "{{ openstack_pki_authorities is not defined | bool }}
↪"
rabbitmq_pki_regen_ca: ""
rabbitmq_pki_authorities:
  - name: "RabbitMQRoot"
    country: "GB"
    state_or_province_name: "England"
    organization_name: "Example Corporation"
    organizational_unit_name: "IT Security"
    cn: "RabbitMQ Root CA"
    provider: selfsigned
    basic_constraints: "CA:TRUE"
    key_usage:
      - digitalSignature
      - cRLSign
      - keyCertSign
    not_after: "+3650d"
  - name: "RabbitMQIntermediate"
    country: "GB"
    state_or_province_name: "England"
    organization_name: "Example Corporation"
    organizational_unit_name: "IT Security"
    cn: "RabbitMQ Intermediate CA"
```

(continues on next page)

(continued from previous page)

```

provider: ownca
basic_constraints: "CA:TRUE,pathlen:0"
key_usage:
  - digitalSignature
  - cRLSign
  - keyCertSign
not_after: "+3650d"
signed_by: "RabbitMQRoot"

# Installation details for certificate authorities
rabbitmq_pki_install_ca:
  - name: "RabbitMQRoot"
    condition: "{{ rabbitmq_pki_create_ca }}"

# Rabbitmq server certificate
rabbitmq_pki_keys_path: "{{ rabbitmq_pki_dir ~ '/certs/private/' }}"
rabbitmq_pki_certs_path: "{{ rabbitmq_pki_dir ~ '/certs/certs/' }}"
rabbitmq_pki_intermediate_cert_name: "{{ openstack_pki_service_intermediate_
↪cert_name | default('RabbitMQIntermediate') }}"
rabbitmq_pki_intermediate_cert_path: >-
  {{ rabbitmq_pki_dir ~ '/roots/' ~ rabbitmq_pki_intermediate_cert_name ~ '/'
↪certs/' ~ rabbitmq_pki_intermediate_cert_name ~ '.crt' }}
rabbitmq_pki_regen_cert: ""
rabbitmq_pki_certificates:
  - name: "rabbitmq-{{ ansible_facts['hostname'] }}"
    provider: ownca
    cn: "{{ ansible_facts['hostname'] }}"
    san: "{{ 'DNS:' ~ ansible_facts['hostname'] ~ ',IP:' ~ rabbitmq_node_
↪address ~ ',DNS:' ~ ansible_facts['fqdn'] }}"
    signed_by: "{{ rabbitmq_pki_intermediate_cert_name }}"

# RabbitMQ destination files for SSL certificates
rabbitmq_ssl_cert: /etc/rabbitmq/rabbitmq.pem
rabbitmq_ssl_key: /etc/rabbitmq/rabbitmq.key
rabbitmq_ssl_ca_cert: /etc/rabbitmq/rabbitmq-ca.pem

# Installation details for SSL certificates
rabbitmq_pki_install_certificates:
  - src: "{{ rabbitmq_user_ssl_cert | default(rabbitmq_pki_certs_path ~
↪'rabbitmq_' ~ ansible_facts['hostname'] ~ '-chain.crt') }}"
    dest: "{{ rabbitmq_ssl_cert }}"
    owner: "rabbitmq"
    group: "rabbitmq"
    mode: "0644"
  - src: "{{ rabbitmq_user_ssl_key | default(rabbitmq_pki_keys_path ~
↪'rabbitmq_' ~ ansible_facts['hostname'] ~ '.key.pem') }}"
    dest: "{{ rabbitmq_ssl_key }}"
    owner: "rabbitmq"
    group: "rabbitmq"

```

(continues on next page)

(continued from previous page)

```

    mode: "0600"
  - src: "{{ rabbitmq_user_ssl_ca_cert | default(rabbitmq_pki_intermediate_
    ↪cert_path) }}"
    dest: "{{ rabbitmq_ssl_ca_cert }}"
    owner: "rabbitmq"
    group: "rabbitmq"
    mode: "0644"

# Define user-provided SSL certificates in:
# /etc/openstack_deploy/user_variables.yml
# rabbitmq_user_ssl_cert: <path to cert on ansible deployment host>
# rabbitmq_user_ssl_key: <path to cert on ansible deployment host>
# rabbitmq_user_ssl_ca_cert: <path to cert on ansible deployment host>

# These are highly recommended for TLSv1.2 but cannot be used
# with TLSv1.3. If TLSv1.3 is enabled, these lines will not be
# inserted into the config
rabbitmq_ssl_client_renegotiation: false
rabbitmq_ssl_secure_renegotiate: true

# Supported TLS protocol versions
rabbitmq_ssl_tls_versions:
  - "tlsv1.2"

# Mutual TLS control
rabbitmq_ssl_verify: "verify_none"
rabbitmq_ssl_fail_if_no_peer_cert: false

# Recommended ciphers taken from https://www.rabbitmq.com/ssl.html
rabbitmq_ssl_ciphers:
  - "ECDHE-ECDSA-AES256-GCM-SHA384"
  - "ECDHE-RSA-AES256-GCM-SHA384"
  - "ECDH-ECDSA-AES256-GCM-SHA384"
  - "ECDH-RSA-AES256-GCM-SHA384"
  - "DHE-RSA-AES256-GCM-SHA384"
  - "DHE-DSS-AES256-GCM-SHA384"
  - "ECDHE-ECDSA-AES128-GCM-SHA256"
  - "ECDHE-RSA-AES128-GCM-SHA256"
  - "ECDH-ECDSA-AES128-GCM-SHA256"
  - "ECDH-RSA-AES128-GCM-SHA256"
  - "DHE-RSA-AES128-GCM-SHA256"
  - "DHE-DSS-AES128-GCM-SHA256"

# RabbitMQ erlang VM parameters
rabbitmq_async_threads: 128
rabbitmq_process_limit: 1048576

# Limit memory consumption of the erlang VM
rabbitmq_memory_high_watermark: 0.2

```

(continues on next page)

(continued from previous page)

```
rabbitmq_env_use_longname: false

# Extra arguments passed to Erlang on startup
# rabbitmq_erlang_extra_args: "+sbwt none +sbwtcpu none +sbwt dio none +stbt_
↪nnts"
rabbitmq_erlang_extra_args: ""

# RabbitMQ collect statistics interval
rabbitmq_collect_statistics_interval: 5000

# RabbitMQ Management service bind address
rabbitmq_management_bind_address: 0.0.0.0
rabbitmq_management_bind_tcp_port: 15672
rabbitmq_management_bind_tls_port: 15671
rabbitmq_management_ssl: true

# RabbitMQ Management rates mode
rabbitmq_management_rates_mode: basic

# Precompile RabbitMQ with HiPE
rabbitmq_hipe_compile: false

# Disable non-TLS listeners
rabbitmq_disable_non_tls_listeners: false

# RabbitMQ logging options
# See https://www.rabbitmq.com/logging.html for the logging options
rabbitmq_log:
  journald: true
  file: false

# RabbitMQ policies
# Used to tune performance characteristics of OpenStack messaging
#
# Example override that uses HA queues only for telemetry and sets message
# expiry for RPC messages
#
# rabbitmq_policies:
#   - name: "heat_rpc_expire"
#     pattern: '^heat-engine-listener\|.'
#     tags: "expires=3600000"
#     priority: 1
#   - name: "results_expire"
#     pattern: '^results\|.'
#     tags: "expires=3600000"
#     priority: 1
#   - name: "tasks_expire"
#     pattern: '^results\|.'
```

(continues on next page)

(continued from previous page)

```
#   tags: "expires=36000000"
#   priority: 1
# - name: "ha-notif"
#   pattern: '(event|metering|notifications)\.'
#   tags: "ha-sync-mode=automatic"
#   priority: 0
#   state:present
# If policy needs to be removed, provide `state: absent`
# - name: "HA"
#   pattern: '^(?!(amq\.)|(.*_fanout_)|(reply_)).*'
#   tags: "ha-mode=all"
#   state: absent
#
rabbitmq_policies: []
rabbitmq_apply_openstack_policies: false
rabbitmq_openstack_policies:
  - name: CQv2
    pattern: ".*"
    priority: 0
    tags:
      queue-version: 2

rabbitmq_port_bindings:
  ssl_listeners:
    "0.0.0.0": 5671
  tcp_listeners:
    "0.0.0.0": 5672

rabbitmq_additional_config: {}

rabbitmq_init_overrides:
  Service:
    LimitNOFILE: "{{ rabbitmq_ulimit }}"
    Restart: on-failure
    RestartSec: 2

# Mnesia configuration
# The Mnesia dump_log_write_threshold option controls
# how often the dumping occurs
# Increase this value can increase the performances,
# reducing the IO.
# Increase it in case of:
# Mnesia is overloaded: {dump_log,write_threshold}.
# The default value is 100
mnesia_dump_log_write_threshold: 300
```





## DEPENDENCIES

This role needs pip  $\geq 7.1$  installed on the target host.

To use this role, define the following variables:

```
# RabbitMQ cluster shared secret  
rabbitmq_cookie_token: secrete
```



## EXAMPLE PLAYBOOK

```
---  
- name: Install RabbitMQ server  
  hosts: rabbitmq_all  
  user: root  
  serial:  
    - 1  
    - 100%  
  roles:  
    - role: rabbitmq_server  
      tags:  
        - rabbitmq-server  
  
  vars:  
    rabbitmq_cookie_token: secreta
```