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>
rabbltmqctl 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 queus
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-queu> 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)