RabbitMQ CLI

Message broker admin

CONCEPTS

> Run rabbitmqctl as the same OS user as the broker: it authenticates with the Erlang cookie, and a wrong cookie means a refused connection.

Publisher sends messages
↓ routing key
Exchange routes by type
↓ binding
Queue holds messages
↓ delivery + ack
Consumer processes them

NODE AND STATUS

`rabbitmqctl status`
Node health, versions, memory, listeners

`rabbitmqctl -n` *`<node>`* *`<command>`*
Target a specific node

`rabbitmqctl stop_app`
Stop RabbitMQ, keep the Erlang VM

`rabbitmqctl start_app`
Start RabbitMQ again

QUEUES

`rabbitmqctl list_queues`
List queues with name and depth

`rabbitmqctl list_queues -p` *`<vhost>`*
List queues in one vhost

`rabbitmqctl list_queues name messages_ready messages_unacknowledged`
Ready vs unacked counts

`rabbitmqctl list_queues name type state`
Queue type and state

`rabbitmqctl purge_queue` *`<queue>`*
Drop all ready messages

`rabbitmqctl delete_queue` *`<queue>`*
Delete a queue

EXCHANGES AND BINDINGS

`rabbitmqctl list_exchanges`
List exchanges with name and type

`rabbitmqctl list_bindings`
List all bindings

CONNECTIONS AND CHANNELS

`rabbitmqctl list_connections`
List client connections

USERS, VHOSTS, PERMISSIONS

`rabbitmqctl list_users`
List users and their tags

`rabbitmqctl add_user` *`<user>`* *`<password>`*
Create a user

`rabbitmqctl set_user_tags` *`<user>`* **`administrator`**
Grant management UI admin

`rabbitmqctl add_vhost` *`<vhost>`*
Create a virtual host

`rabbitmqctl set_permissions -p` *`<vhost>`* *`<user>`* `".*" ".*" ".*"`
Grant configure, write, read

POLICIES

`rabbitmqctl set_policy` *`<name>`* "*`<pattern>`*" '*`<json>`*'
Apply settings to matching queues

CLUSTERING

`rabbitmqctl cluster_status`
Members, alarms, partitions

PLUGINS AND RABBITMQADMIN

`rabbitmq-plugins list`
Show plugins and their state

`rabbitmq-plugins enable` *`<plugin>`*
Enable a plugin