**Introduction:**

**https://github.com/vishymails/rabbitmq**

**vishymails@gmail.com**

**7892279196**

**RabbitMQ**: built on Erlang programming language

56MB – Protocol ecommendation – **AMQP**

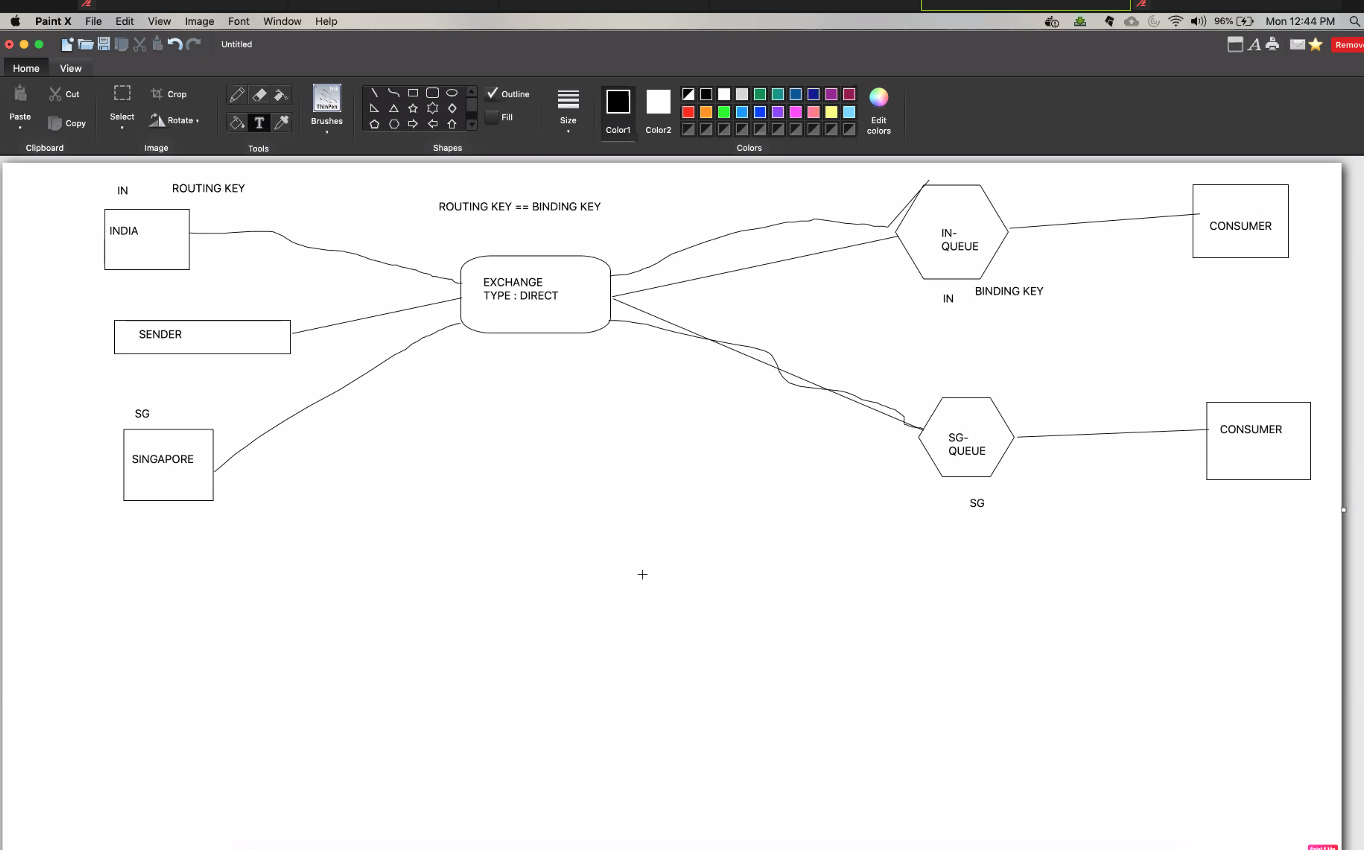
JP Morgan’s and imatics are main contributors for AMQP

**Message broker:** Acts as intermediary

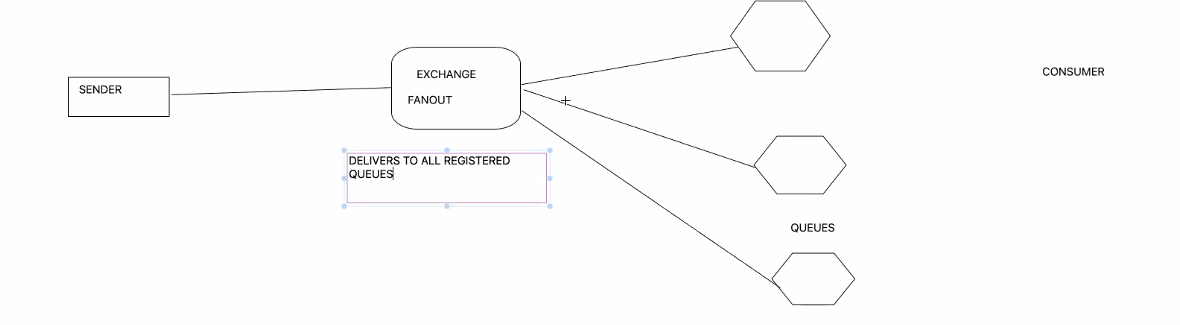
Main objects of Rabbitmq are: Exchange, Binding, queue

**Exchanges**

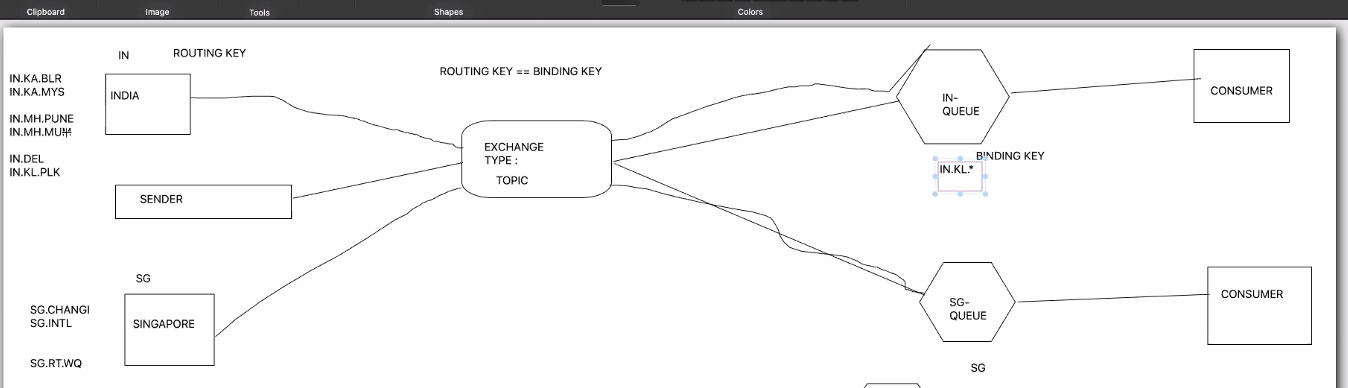
**Direct**: binding key should match with routing key



**FAN OUT:** No binding and routing key required. It send message to All registered queues

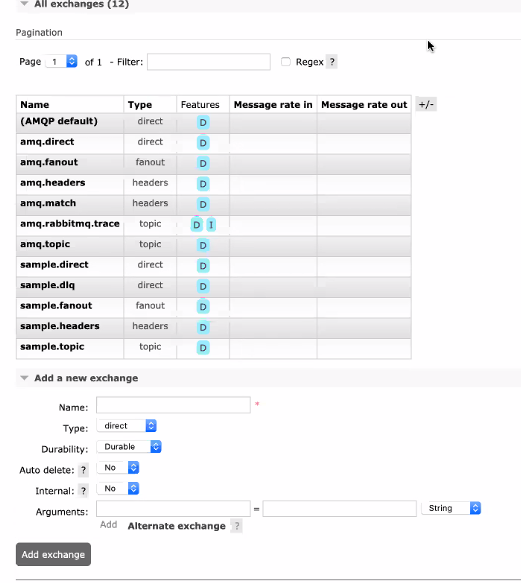


**Topic:** It will match keys with wildcard matching (Topic matching expressions)



**Header Exchange:** It uses message header attributes or variables.

**Adding Exchanges:**



Exchange Options:

**Durability** – durable and transient

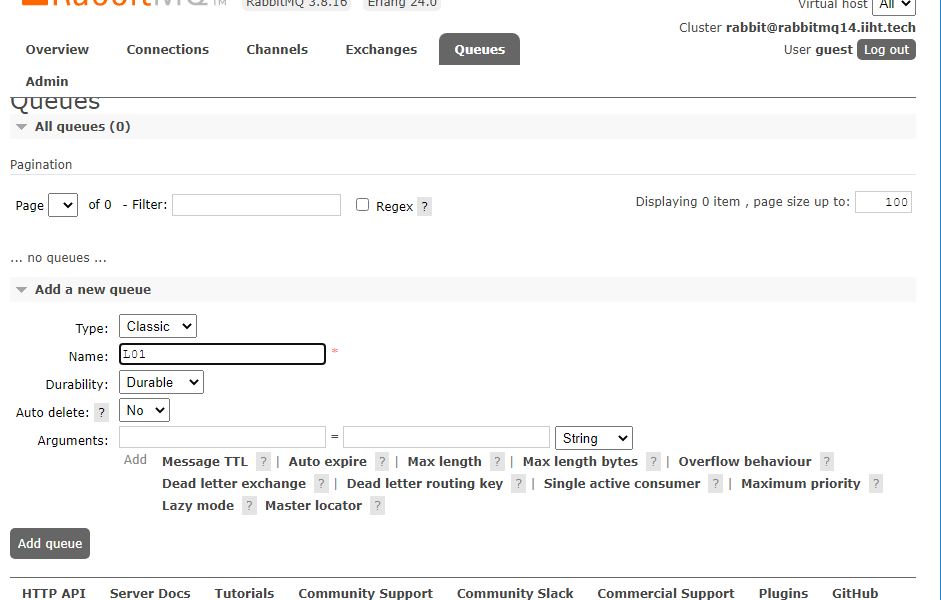
Durable – stays after restart and

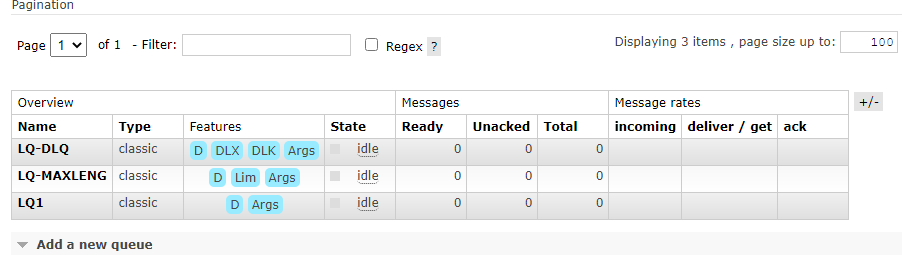
Transient - gets deleted after restart

**Arguments:**

Alternate exchange – the alternative for the exchange

**Adding Queue:**





**Message TTL:** time to live,after disconnection of sessions the time period to destroy the queue.

**Auto expire:** TTL should be more than expire time

**Max length:** Max no of elements queue can hold

**Max length bytes:**

**Overflow**: When message reaches max-length, secondary property (drop-head, reject-publish)

**Dead letter-exchange**:

**Dead letter-routing key:**

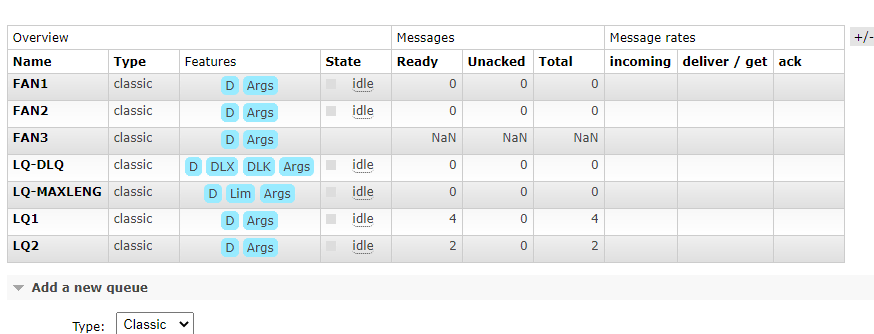
**Binding:** is a connection between queue and exchange

Auto delete

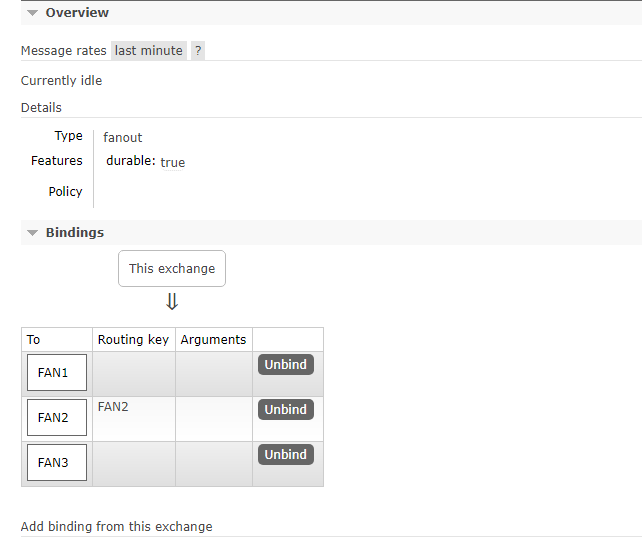
**Fan out:**

To demonstrate create 3 queues (FAN1, FNN2, FAN3) and add bindings to sample.fan

In fanout routing key is not required (optional). But we can still create routing key



In fanout routing key is not required (optional). But we can still create routing key(Ex: FAN2)

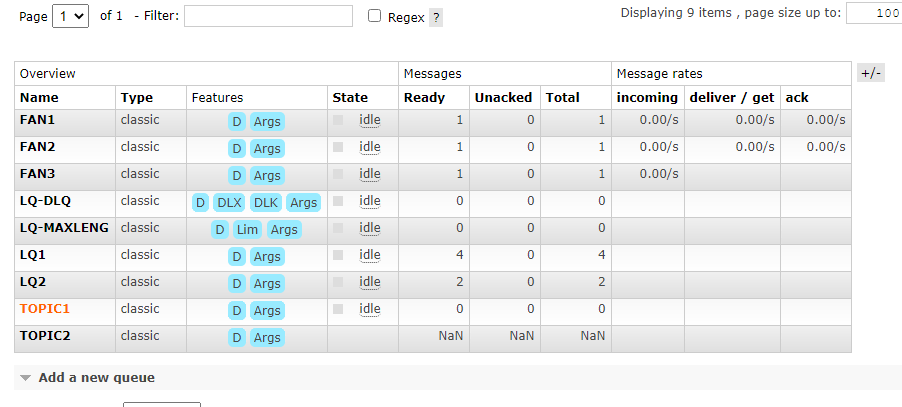


When we publish message from Exchange all queues will receive the same message (FAN1, FAN2, FAN3)

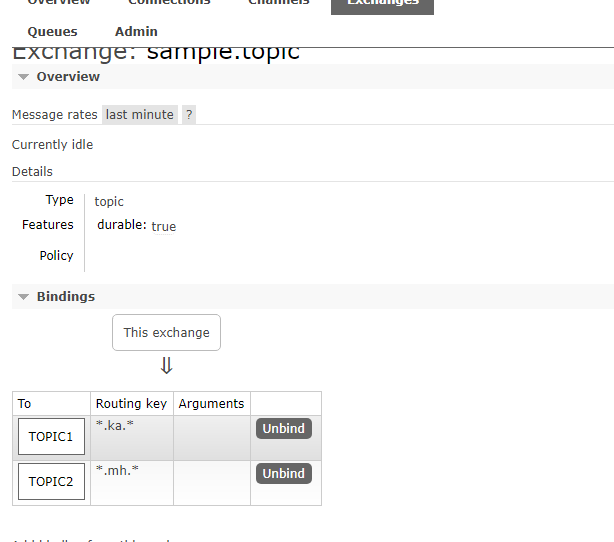


**Topic Exchange:**

Add 2 queues TOPIC1, TOPIC2

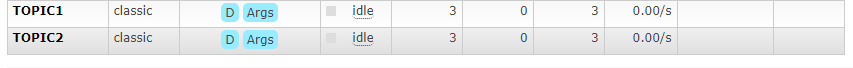


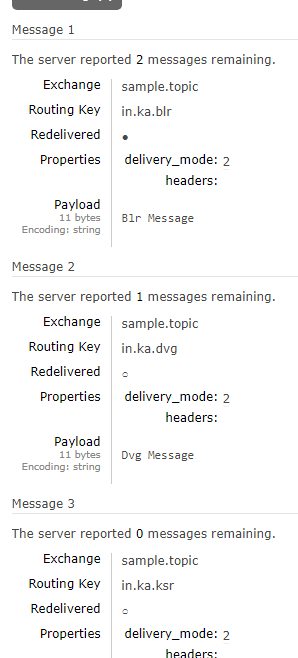
Bindings in topic exchange sample.topic , use route keys with wild cards



Then publish messages with route key

After publish we can see message inside



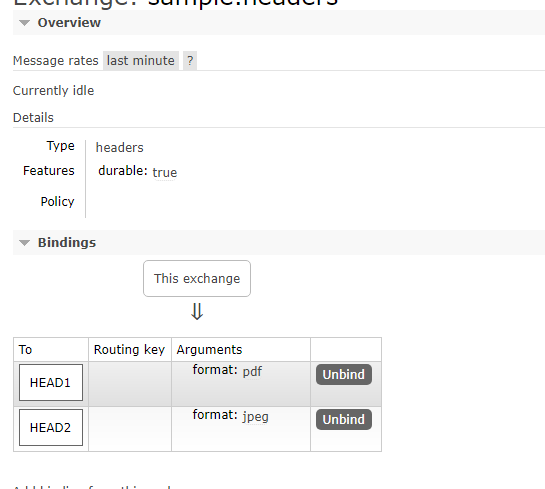


**Header Exchange:**

Create 2 queues



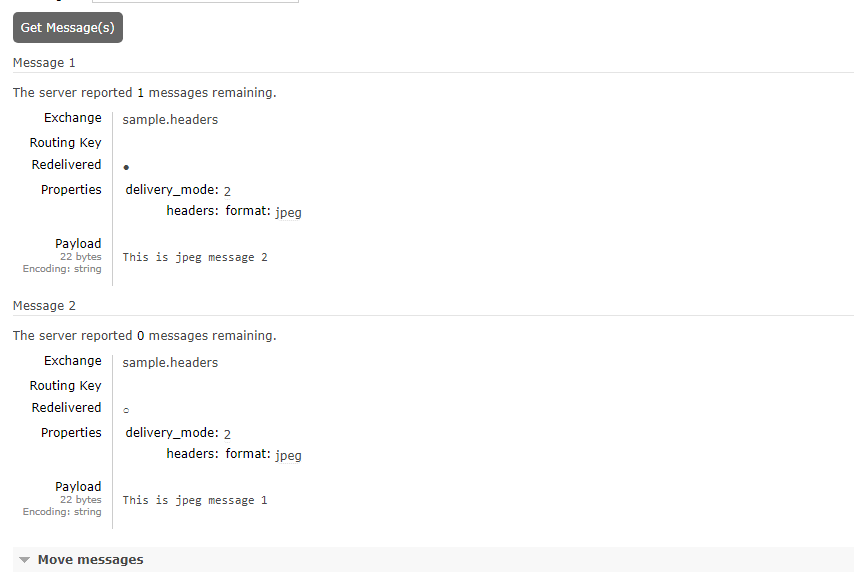
Create 2 bindings under sample.header exchange



Publish messages







**Security:**

* Server level security - Admins
* Application level Security – Developers

Server level security:

* ACL: Access control list
* UCI: Use caller identity

RabbitMQ provides tags to handle security

1. ADMIN – Super Admin
2. POLICYMAKER –They can access management, and manages the policies for virtual host
3. MONITOR – They can access management, node related functions. Always deals with lively data
4. MANAGEMENT – No access to objects
5. IMPERSONATOR – Forge the user id with the tag, On behalf of some other

Comma-separated list of tags to apply to the user. Currently supported by the management plugin:

management

User can access the management plugin

policymaker

User can access the management plugin and manage policies and parameters for the vhosts they have access to.

monitoring

User can access the management plugin and see all connections and channels as well as node-related information.

administrator

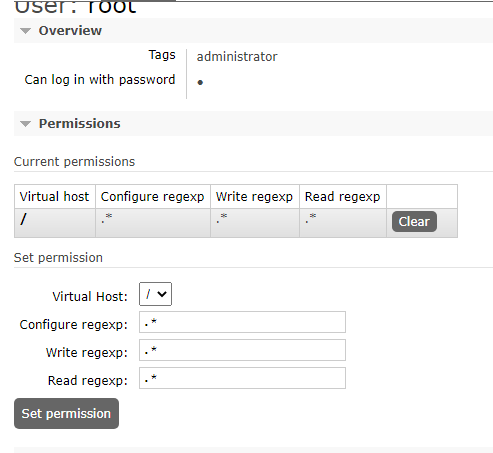
User can do everything monitoring can do, manage users, vhosts and permissions, close other user's connections, and manage policies and parameters for all vhosts.

Note that you can set any tag here; the links for the above four tags are just for convenience

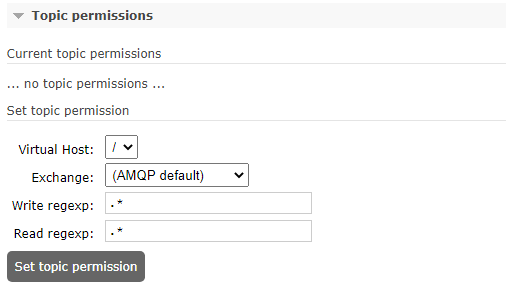
Policies means set of rules

**Virtual host:** Its collection of resources, we can logically group resources ( Exchange, queues, connections, bindings)

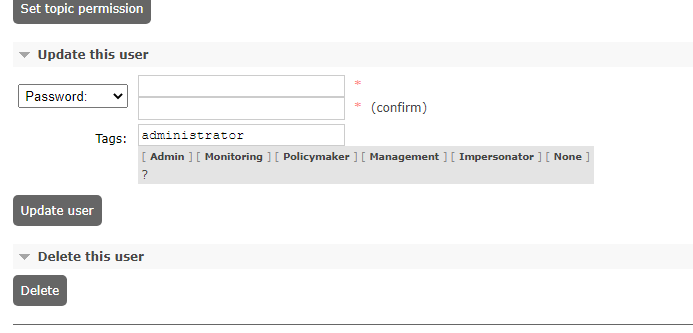
**Can access virtual hosts:** We can enable by using Set permissions



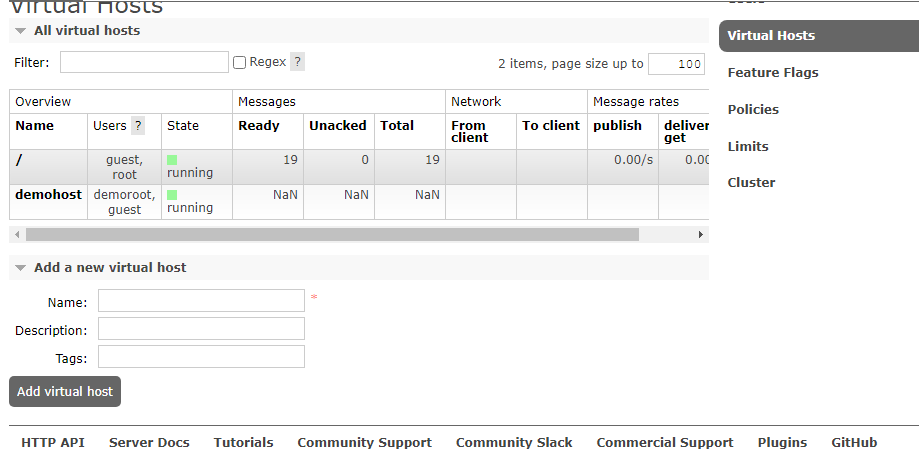
Updating access for topic exchange:



**Update this user permissions and password:**

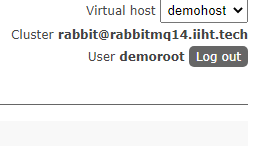


**Virtual Host:**

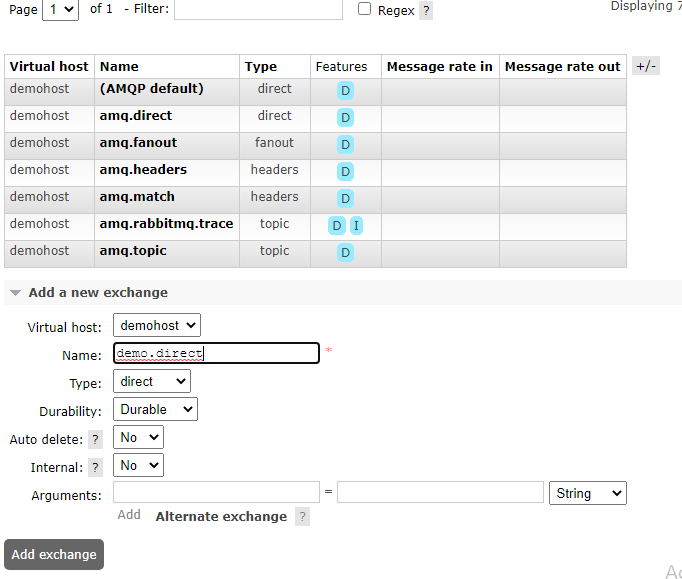


**Login from different user**

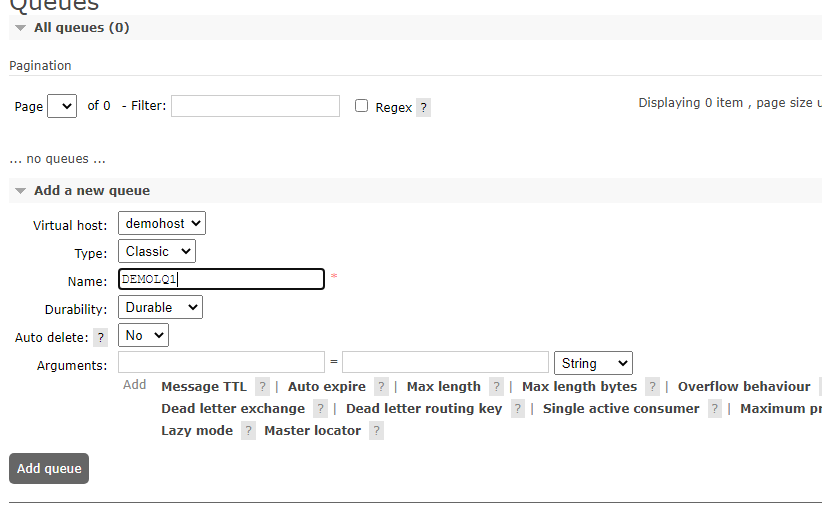
**Selection:**



Creation of exchange under demohost – Virtual host



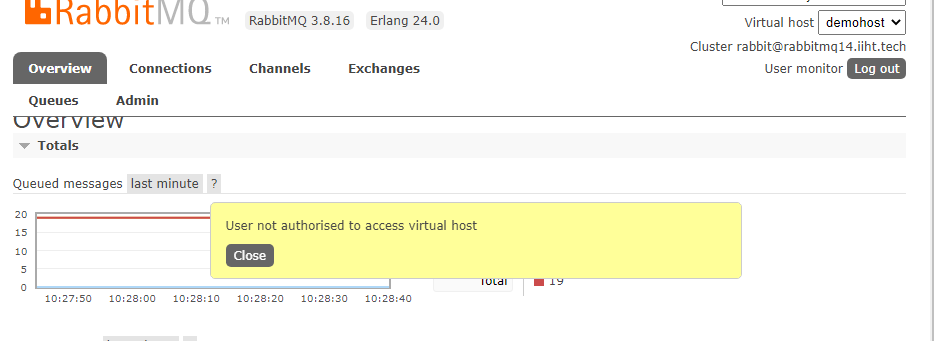
**Creating queue under demohost –virtual host**



**--------------------------**

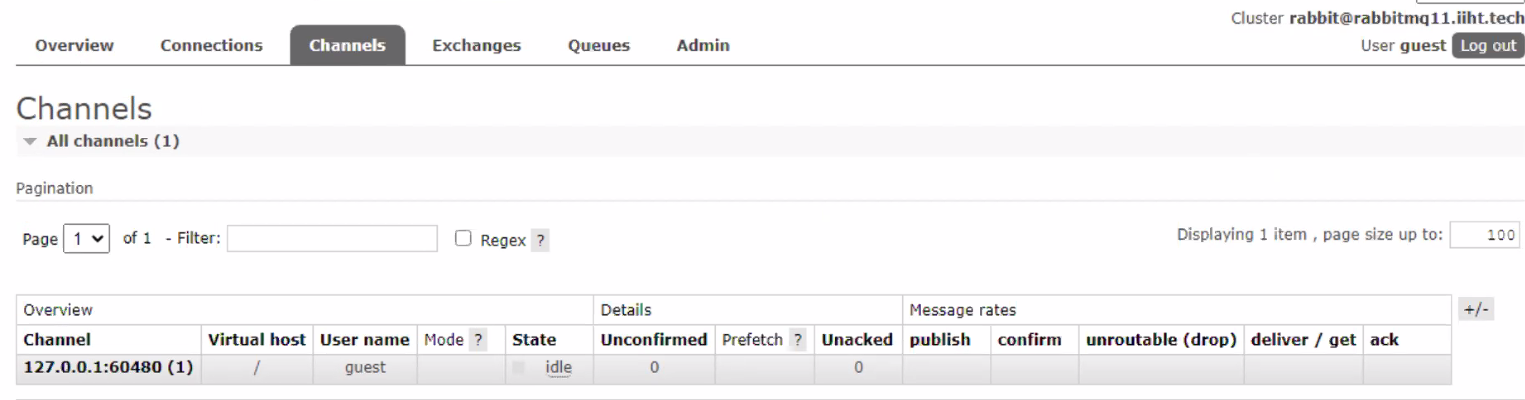
**Monitoring:**

They will not have access to any objects



**-------------------------------**

**Channels:**



**Mode:**

Channel guarantee mode. Can be one of the following, or neither:

C – confirm

Channel will send streaming publish confirmations.

T – transactional

Channel is transactional.

**Prefetch**: Details about Channel limits (Consumer count and Global count)

**Unacknowledged :**

**Publish:**

**Confirm:**

**Diagnostics:**

rabbitmq-diagnostics check\_alarms

rabbitmq-diagnostics check\_running

rabbitmq-diagnostics check\_virtual\_hosts --timeout 60

rabbitmq-diagnostics discover\_peers --timeout 60

rabbitmq-diagnostics erlang\_cookie\_hash –q

rabbitmq-diagnostics erlang\_version\_hash –q

rabbitmq-diagnostics is\_booting

rabbitmq-diagnostics is\_running

rabbitmq-diagnostics listeners

rabbitmq-diagnostics memory\_leakages

rabbitmq-diagnostics alarms

10 rabbitmq-diagnostics check\_alarams

11 rabbitmq-diagnostics check\_alarms

12 rabbitmq-diagnostics check\_port\_connectivity

13 rabbitmq-diagnostics check\_port\_listener 5672

14 rabbitmq-diagnostics check\_port\_listener mqtt

15 rabbitmq-diagnostics check\_running

16 rabbitmq-diagnostics check\_virtual\_hosts --timeout 60

17 rabbitmq-diagnostics discover\_peers --timeout 60

18 rabbitmq-diagnostics erlang\_cookie\_hash -q

19 rabbitmq-diagnostics erlang\_version -q

20 rabbitmq-diagnostics is\_booting

21 rabbitmq-diagnostics is\_running

22 rabbitmq-diagnostics listeners

23 rabbitmq-diagnostics log\_tail -number 100

24 rabbitmq-diagnostics log\_tail --number 100

25 rabbitmq-diagnostics log\_tail\_stream --duration 60

26 rabbitmq-diagnostics maybe\_stuck -q

27 rabbitmq-diagnostics memory\_breakdown --unit gigabytes

28 rabbitmq-diagnostics observer --interval 10

29 rabbitmq-diagnostics runtime\_thread\_stats --sample-interval 10

30 rabbitmq-diagnostics server\_version -q

31 rabbitmq-diagnostics tls\_versions -q

32 rabbitmq-diagnostics log\_location -a

**-----------------------------------**

**Command Line Administratrion:**

1. Rabbitmqctl
2. Rabbitmq-queues
3. Rabbitmq-plugins
4. Rabbitmq-upgrade
5. Rabbitmq-diagnostics

* Rabbitmqadmin
* Rabbitmq-collect-env

We need to install python to run rabbitmqadmin

To download rabbitmqadmin, we need to run http://localhost:15672/cli/rabbitmqadmin

https://www.python.org/downloads/release/python-395/

**Shovels:**

Way to transfer messages from one queue to another

Shovel has 2 types- Static and dynamic

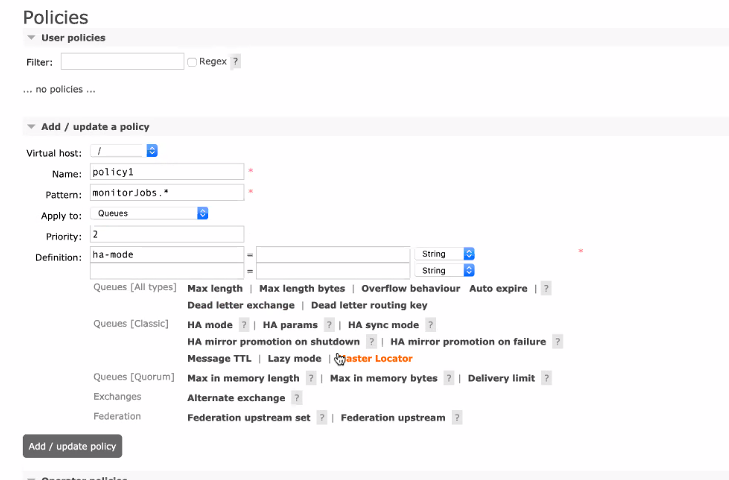
Dynamic shovel can be managed under Shovel management of Admin section

We have to enable plugin to activate this option by running below command:

**rabbitmq-plugins enable rabbitmq\_shovel rabbitmq\_shove**

**Policy:**

**Set of rules to apply on objects(Queues, Exchange**



**------------------**

**Programs:**

package com.ibm.rabbitmq;

import com.rabbitmq.client.Channel;

import com.rabbitmq.client.Connection;

import com.rabbitmq.client.ConnectionFactory;

public class Send {

private final static String QUEUE\_NAME = "Santhosh";

public static void main(String[] argv) throws Exception {

ConnectionFactory factory = new ConnectionFactory();

factory.setHost("localhost");

try (Connection connection = factory.newConnection();

Channel channel = connection.createChannel()) {

channel.queueDeclare(QUEUE\_NAME, false, false, false, null);

String message = "Hello World!";

channel.basicPublish("", QUEUE\_NAME, null, message.getBytes("UTF-8"));

System.out.println(" [x] Sent '" + message + "'");

//Thread.sleep(30000000);

}

}

}

package com.ibm.rabbitmq;

import com.rabbitmq.client.Channel;

import com.rabbitmq.client.Connection;

import com.rabbitmq.client.ConnectionFactory;

import com.rabbitmq.client.DeliverCallback;

public class Recv {

private final static String QUEUE\_NAME = "hello";

public static void main(String[] argv) throws Exception {

ConnectionFactory factory = new ConnectionFactory();

factory.setHost("localhost");

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE\_NAME, false, false, false, null);

System.out.println(" [\*] Waiting for messages. To exit press CTRL+C");

DeliverCallback deliverCallback = (consumerTag, delivery) -> {

String message = new String(delivery.getBody(), "UTF-8");

System.out.println(" [x] Received '" + message + "'");

};

channel.basicConsume(QUEUE\_NAME, true, deliverCallback, consumerTag -> { });

}

}

Command line Queries:

rabbitmqctl -help

Usage

rabbitmqctl [--node <node>] [--timeout <timeout>] [--longnames] [--quiet] <command> [<command options>]

Available commands:

Help:

autocomplete Provides command name autocomplete variants

help Displays usage information for a command

version Displays CLI tools version

Nodes:

await\_startup Waits for the RabbitMQ application to start on the target node

reset Instructs a RabbitMQ node to leave the cluster and return to its virgin state

rotate\_logs Instructs the RabbitMQ node to perform internal log rotation

shutdown Stops RabbitMQ and its runtime (Erlang VM). Monitors progress for local nodes. Does not require a PID file path.

start\_app Starts the RabbitMQ application but leaves the runtime (Erlang VM) running

stop Stops RabbitMQ and its runtime (Erlang VM). Requires a local node pid file path to monitor progress.

stop\_app Stops the RabbitMQ application, leaving the runtime (Erlang VM) running

wait Waits for RabbitMQ node startup by monitoring a local PID file. See also 'rabbitmqctl await\_online\_nodes'

Cluster:

await\_online\_nodes Waits for <count> nodes to join the cluster

change\_cluster\_node\_type Changes the type of the cluster node

cluster\_status Displays all the nodes in the cluster grouped by node type, together with the currently running nodes

force\_boot Forces node to start even if it cannot contact or rejoin any of its previously known peers

force\_reset Forcefully returns a RabbitMQ node to its virgin state

forget\_cluster\_node Removes a node from the cluster

join\_cluster Instructs the node to become a member of the cluster that the specified node is in

rename\_cluster\_node Renames cluster nodes in the local database

update\_cluster\_nodes Instructs a cluster member node to sync the list of known cluster members from <seed\_node>

Replication:

cancel\_sync\_queue Instructs a synchronising mirrored queue to stop synchronising itself

sync\_queue Instructs a mirrored queue with unsynchronised mirrors (follower replicas) to synchronise them

Users:

add\_user Creates a new user in the internal database. This user will have no permissions for any virtual hosts by default.

authenticate\_user Attempts to authenticate a user. Exits with a non-zero code if authentication fails.

change\_password Changes the user password

clear\_password Clears (resets) password and disables password login for a user

clear\_user\_limits Clears user connection/channel limits

delete\_user Removes a user from the internal database. Has no effect on users provided by external backends such as LDAP

list\_user\_limits Displays configured user limits

list\_users List user names and tags

set\_user\_limits Sets user limits

set\_user\_tags Sets user tags

Access Control:

clear\_permissions Revokes user permissions for a vhost

clear\_topic\_permissions Clears user topic permissions for a vhost or exchange

list\_permissions Lists user permissions in a virtual host

list\_topic\_permissions Lists topic permissions in a virtual host

list\_user\_permissions Lists permissions of a user across all virtual hosts

list\_user\_topic\_permissions Lists user topic permissions

list\_vhosts Lists virtual hosts

set\_permissions Sets user permissions for a vhost

set\_topic\_permissions Sets user topic permissions for an exchange

Monitoring, observability and health checks:

list\_bindings Lists all bindings on a vhost

list\_channels Lists all channels in the node

list\_ciphers Lists cipher suites supported by encoding commands

list\_connections Lists AMQP 0.9.1 connections for the node

list\_consumers Lists all consumers for a vhost

list\_exchanges Lists exchanges

list\_hashes Lists hash functions supported by encoding commands

list\_node\_auth\_attempt\_stats Lists authentication attempts on the target node

list\_queues Lists queues and their properties

list\_unresponsive\_queues Tests queues to respond within timeout. Lists those which did not respond

ping Checks that the node OS process is up, registered with EPMD and CLI tools can authenticate with it

report Generate a server status report containing a concatenation of all server status information for support purposes

schema\_info Lists schema database tables and their properties

status Displays status of a node

Parameters:

clear\_global\_parameter Clears a global runtime parameter

clear\_parameter Clears a runtime parameter.

list\_global\_parameters Lists global runtime parameters

list\_parameters Lists runtime parameters for a virtual host

set\_global\_parameter Sets a runtime parameter.

set\_parameter Sets a runtime parameter.

Policies:

clear\_operator\_policy Clears an operator policy

clear\_policy Clears (removes) a policy

list\_operator\_policies Lists operator policy overrides for a virtual host

list\_policies Lists all policies in a virtual host

set\_operator\_policy Sets an operator policy that overrides a subset of arguments in user policies

set\_policy Sets or updates a policy

Virtual hosts:

add\_vhost Creates a virtual host

clear\_vhost\_limits Clears virtual host limits

delete\_vhost Deletes a virtual host

list\_vhost\_limits Displays configured virtual host limits

restart\_vhost Restarts a failed vhost data stores and queues

set\_vhost\_limits Sets virtual host limits

trace\_off

trace\_on

Configuration and Environment:

decode Decrypts an encrypted configuration value

encode Encrypts a sensitive configuration value

environment Displays the name and value of each variable in the application environment for each running application

set\_cluster\_name Sets the cluster name

set\_disk\_free\_limit Sets the disk\_free\_limit setting

set\_log\_level Sets log level in the running node

set\_vm\_memory\_high\_watermark Sets the vm\_memory\_high\_watermark setting

Definitions:

export\_definitions Exports definitions in JSON or compressed Erlang Term Format.

import\_definitions Imports definitions in JSON or compressed Erlang Term Format.

Feature flags:

enable\_feature\_flag Enables a feature flag or all supported feature flags on the target node

list\_feature\_flags Lists feature flags

Operations:

close\_all\_connections Instructs the broker to close all connections for the specified vhost or entire RabbitMQ node

close\_all\_user\_connections Instructs the broker to close all connections of the specified user

close\_connection Instructs the broker to close the connection associated with the Erlang process id

eval Evaluates a snippet of Erlang code on the target node

eval\_file Evaluates a file that contains a snippet of Erlang code on the target node

exec Evaluates a snippet of Elixir code on the CLI node

force\_gc Makes all Erlang processes on the target node perform/schedule a full sweep garbage collection

resume\_listeners Resumes client connection listeners making them accept client connections again

suspend\_listeners Suspends client connection listeners so that no new client connections are accepted

Queues:

delete\_queue Deletes a queue

purge\_queue Purges a queue (removes all messages in it)

Deprecated:

hipe\_compile DEPRECATED. This command is a no-op. HiPE is no longer supported by modern Erlang versions

node\_health\_check DEPRECATED. Performs intrusive, opinionated health checks on a fully booted node. See https://www.rabbitmq.com/monitoring.html#health-checks instead

Use 'rabbitmqctl help <command>' to learn more about a specific command

# rabbitmq-diagnostics status --help

Usage

rabbitmq-diagnostics [--node <node>] [--longnames] [--quiet] status [--unit <unit>] [--timeout <timeout>]

Displays status of a node.

Arguments and Options

--unit <bytes | mb | gb>

byte multiple (bytes, megabytes, gigabytes) to use

--formatter <json | erlang>

alternative formatter (JSON, Erlang terms)

Relevant Doc Guides

\* https://rabbitmq.com/monitoring.html

General Options

The following options are accepted by most or all commands.

short | long | description

-----------------|---------------|--------------------------------

-? | --help | displays command help

-n <node> | --node <node> | connect to node <node>

-l | --longnames | use long host names

-t | --timeout <n> | for commands that support it, operation timeout in seconds

-q | --quiet | suppress informational messages

-s | --silent | suppress informational messages

| and table header row

-p | --vhost | for commands that are scoped to a virtual host,

| | virtual host to use

| --formatter | alternative result formatter to use

| if supported: json, pretty\_table, table, csv, erlang

not all commands support all (or any) alternative formatters.

# rabbitmq-diagnostics help status

Usage

rabbitmq-diagnostics [--node <node>] [--longnames] [--quiet] status [--unit <unit>] [--timeout <timeout>]

Displays status of a node.

Arguments and Options

--unit <bytes | mb | gb>

byte multiple (bytes, megabytes, gigabytes) to use

--formatter <json | erlang>

alternative formatter (JSON, Erlang terms)

Relevant Doc Guides

\* https://rabbitmq.com/monitoring.html

General Options

The following options are accepted by most or all commands.

short | long | description

-----------------|---------------|--------------------------------

-? | --help | displays command help

-n <node> | --node <node> | connect to node <node>

-l | --longnames | use long host names

-t | --timeout <n> | for commands that support it, operation timeout in seconds

-q | --quiet | suppress informational messages

-s | --silent | suppress informational messages

| and table header row

-p | --vhost | for commands that are scoped to a virtual host,

| | virtual host to use

| --formatter | alternative result formatter to use

| if supported: json, pretty\_table, table, csv, erlang

not all commands support all (or any) alternative formatters.

# rabbitmqctl add\_user bvr bvr

Adding user "bvr" ...

Done. Don't forget to grant the user permissions to some virtual hosts! See 'rabbitmqctl help set\_permissions' to learn more.

# rabbitmqctl authenticate\_user bvr bvr

Authenticating user "bvr" ...

Success

#

#

#

# rabbitmqctl change\_password bvr bvr123

Changing password for user "bvr" ...

# rabbitmqctl authenticate\_user bvr bvr

Authenticating user "bvr" ...

Error:

Error: failed to authenticate user "bvr"

user 'bvr' - invalid credentials

# rabbitmqctl list\_users

Listing users ...

user tags

monitor [monitoring]

guest [administrator]

bvr []

demoroot [administrator]

root [administrator]

# rabbitmqctl delete\_user root

Deleting user "root" ...

# rabbitmqctl set\_user\_tags bvr administrator

Setting tags for user "bvr" to [administrator] ...

#

#

#

# history

sh: 22: history: not found

#

# rabbitmqctl clear\_permissions -p demohost bvr

Clearing permissions for user "bvr" in vhost "demohost" ...

# rabbitmqctl list\_permissions -p demohost

Listing permissions for vhost "demohost" ...

user configure write read

guest .\* .\* .\*

demoroot .\* .\* .\*

# rabbitmqctl list\_user\_permissions bvr

Listing permissions for user "bvr" ...

# rabbitmqctl list\_vhosts

Listing vhosts ...

name

demohost

/

# rabbitmqctl set\_permissions -p demohost bvr "^bvr-.\*" ".\*" ".\*"

Setting permissions for user "bvr" in vhost "demohost" ...

# rabbitmqctl list\_topic\_user\_permissions bvr

Command 'list\_topic\_user\_permissions' not found.

Did you mean 'list\_user\_topic\_permissions'?

# rabbitmqctl list\_user\_topic\_permisssions bvr

Command 'list\_user\_topic\_permisssions' not found.

Did you mean 'list\_user\_topic\_permissions'?

# rabbitmqctl list\_user\_topic\_permissions bvr

Listing topic permissions for user "bvr" ...

# rabbitmqctl set\_topic\_permissions -p demohost bvr amq.topic "^bvr-.\*" "^bvr-.\*"

Setting topic permissions on "amq.topic" for user "bvr" in vhost "demohost" ...

#

# rabbitmqctl list\_bindings -p / sample.direct LQ1

Error (argument validation): Info key(s) LQ1,sample.direct are not supported

Arguments given:

list\_bindings -p / sample.direct LQ1

Usage

rabbitmqctl [--node <node>] [--longnames] [--quiet] list\_bindings [--vhost <vhost>] [--no-table-headers] [<column> ...] [--timeout <timeout>]

# rabbitmqctl list\_bindings --vhost /

Listing bindings for vhost /...

source\_name source\_kind destination\_name destination\_kind routing\_key arguments

exchange TOPIC1 queue TOPIC1 []

exchange TOPIC2 queue TOPIC2 []

exchange LQ-DLQ queue LQ-DLQ []

exchange LQ1 queue LQ1 []

exchange LQ-MAXLENG queue LQ-MAXLENG []

exchange HEAD2 queue HEAD2 []

exchange FAN1 queue FAN1 []

exchange LQ2 queue LQ2 []

exchange FAN2 queue FAN2 []

exchange FAN3 queue FAN3 []

exchange HEAD1 queue HEAD1 []

sample.direct exchange LQ1 queue samplekey1 []

sample.direct exchange LQ2 queue samplekey2 []

sample.fanout exchange FAN1 queue []

sample.fanout exchange FAN3 queue []

sample.fanout exchange FAN2 queue FAN2 []

sample.headers exchange HEAD1 queue [{"format","pdf"}]

sample.headers exchange HEAD2 queue [{"format","img"}]

sample.headers exchange HEAD2 queue [{"format","png"}]

sample.topic exchange TOPIC1 queue \*.ka.\* []

sample.topic exchange TOPIC2 queue \*.mh.\* []

# rabbitmqctl list\_exchanges --vhost /

Listing exchanges for vhost / ...

name type

sample.dlq direct

sample.headers headers

amq.direct direct

sample.direct direct

sample.topic topic

amq.match headers

amq.fanout fanout

amq.rabbitmq.trace topic

direct

amq.topic topic

amq.headers headers

sample.fanout fanout

#

#

# rabbitmqctl list\_hashes

Listing supported hash algorithms ...

[sha,sha224,sha256,sha384,sha512,sha3\_224,sha3\_256,sha3\_384,sha3\_512,blake2b,

blake2s,md4,md5,ripemd160]

#

#

#

# rabbitmqctl list\_queues --vhost /

Timeout: 60.0 seconds ...

Listing queues for vhost / ...

name messages

TOPIC1 3

TOPIC2 2

LQ-DLQ 0

LQ1 5

LQ-MAXLENG 0

HEAD2 1

FAN1 2

LQ2 2

FAN2 2

FAN3 2

HEAD1 1

# rabbitmqctl list\_unresponsive\_queues

Listing unresponsive queues for vhost / ...

#

#

# rabbitmqctl ping -n rabbit@my-rabbit

Will ping rabbit@my-rabbit. This only checks if the OS process is running and registered with epmd. Timeout: 60000 ms.

Ping succeeded

#

#

#

#

# rabbitmqctl schema\_info

Asking node rabbit@my-rabbit to report its schema...

name cookie active\_replicas user\_properties

rabbit\_user\_permission {{1621837054927439800,-576460752303422847,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_topic\_trie\_edge {{1621837055015949900,-576460752303421983,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_queue {{1621837055077832700,-576460752303421311,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

rabbit\_semi\_durable\_route {{1621837054983202700,-576460752303422367,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

tracked\_connection\_table\_per\_user\_on\_node\_rabbit@my-rabbit {{1621837056402273300,-576460752303422973,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

tracked\_connection\_on\_node\_rabbit@my-rabbit {{1621837056378454000,-576460752303423069,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

tracked\_channel\_on\_node\_rabbit@my-rabbit {{1621837056358826300,-576460752303423261,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

schema {{1621837054848203900,-576460752303423071,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_exchange\_serial {{1621837055052882600,-576460752303421599,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_route {{1621837054994926600,-576460752303422271,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

rabbit\_exchange {{1621837055039143600,-576460752303421695,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

rabbit\_vhost {{1621837054944111100,-576460752303422655,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

mirrored\_sup\_childspec {{1621837055091776500,-576460752303421119,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_listener {{1621837054952012400,-576460752303422559,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

gm\_group {{1621837055084524700,-576460752303421215,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

rabbit\_topic\_permission {{1621837054935387000,-576460752303422751,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_topic\_trie\_binding {{1621837055022714600,-576460752303421887,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_reverse\_route {{1621837055002373300,-576460752303422175,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_durable\_route {{1621837054962924800,-576460752303422463,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

tracked\_connection\_per\_vhost\_on\_node\_rabbit@my-rabbit {{1621837056385605700,-576460752303423390,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_user {{1621837054918347000,-576460752303422943,1}, rabbit@my-rabbit}[rabbit@my-rabbit] []

rabbit\_topic\_trie\_node {{1621837055009264100,-576460752303422079,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_durable\_queue {{1621837055068298200,-576460752303421407,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_durable\_exchange {{1621837055029237300,-576460752303421791,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

tracked\_channel\_table\_per\_user\_on\_node\_rabbit@my-rabbit {{1621837056369456000,-576460752303423165,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_runtime\_parameters {{1621837055059342500,-576460752303421503,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

rabbit\_node\_maintenance\_states {{1621837055525257200,-576460752303423485,1}, rabbit@my-rabbit} [rabbit@my-rabbit] []

# rabbitmqctl schema+info name active\_replicas

Command 'schema+info' not found.

Did you mean 'schema\_info'?

# rabbitmqctl schema\_info name active\_replicas

Asking node rabbit@my-rabbit to report its schema...

name active\_replicas

rabbit\_user\_permission [rabbit@my-rabbit]

rabbit\_topic\_trie\_edge [rabbit@my-rabbit]

rabbit\_queue [rabbit@my-rabbit]

rabbit\_semi\_durable\_route [rabbit@my-rabbit]

tracked\_connection\_table\_per\_user\_on\_node\_rabbit@my-rabbit [rabbit@my-rabbit]

tracked\_connection\_on\_node\_rabbit@my-rabbit [rabbit@my-rabbit]

tracked\_channel\_on\_node\_rabbit@my-rabbit [rabbit@my-rabbit]

schema [rabbit@my-rabbit]

rabbit\_exchange\_serial [rabbit@my-rabbit]

rabbit\_route [rabbit@my-rabbit]

rabbit\_exchange [rabbit@my-rabbit]

rabbit\_vhost [rabbit@my-rabbit]

mirrored\_sup\_childspec [rabbit@my-rabbit]

rabbit\_listener [rabbit@my-rabbit]

gm\_group [rabbit@my-rabbit]

rabbit\_topic\_permission [rabbit@my-rabbit]

rabbit\_topic\_trie\_binding [rabbit@my-rabbit]

rabbit\_reverse\_route [rabbit@my-rabbit]

rabbit\_durable\_route [rabbit@my-rabbit]

tracked\_connection\_per\_vhost\_on\_node\_rabbit@my-rabbit [rabbit@my-rabbit]

rabbit\_user [rabbit@my-rabbit]

rabbit\_topic\_trie\_node [rabbit@my-rabbit]

rabbit\_durable\_queue [rabbit@my-rabbit]

rabbit\_durable\_exchange [rabbit@my-rabbit]

tracked\_channel\_table\_per\_user\_on\_node\_rabbit@my-rabbit [rabbit@my-rabbit]

rabbit\_runtime\_parameters [rabbit@my-rabbit]

rabbit\_node\_maintenance\_states [rabbit@my-rabbit]

#

#

#

# rabbitmqctl list\_parameters

Listing runtime parameters for vhost "/" ...

# rabbitmqctl list\_policies

Listing policies for vhost "/" ...

# rabbitmqctl add\_vhost IBM

Adding vhost "IBM" ...

# rabbitmqctl clear\_vhost\_limits -p IBM

Clearing vhost "IBM" limits ...

Error:

Parameter does not exist

# rabbitmqctl delete\_vhost IBM

Deleting vhost "IBM" ...

# rabbitmqctl restart\_vhost demohost

Error (argument validation): too many arguments.

Arguments given:

restart\_vhost demohost

Usage

rabbitmqctl [--node <node>] [--longnames] [--quiet] restart\_vhost [--vhost <vhost>] [--timeout <timeout>]

# rabbitmqctl restart\_vhost --vhost demohost

Trying to restart vhost 'demohost' on node 'rabbit@my-rabbit' ...

Vhost 'demohost' is already running on node 'rabbit@my-rabbit'

#

#

#

#

# rabbitmqctl add\_vhost ibm

Adding vhost "ibm" ...

#

#

# rabbitmqctl set\_vhost\_limits -p ibm '{"max-connections": 50}'

Setting vhost limits to "{"max-connections": 50}" for vhost "ibm" ...

# rabbitmqctl set\_vhost\_limits -p ibm '{"max-queues": 256}'

Setting vhost limits to "{"max-queues": 256}" for vhost "ibm" ...

# rabbitmqctl set\_vhost\_limits -p ibm '{"max-connections": -1}'

Setting vhost limits to "{"max-connections": -1}" for vhost "ibm" ...

#

#

# rabbitmqadmin declare exchange --vhost=ibm name=ibmhost1 type=direct

\*\*\* Access refused: /api/exchanges/ibm/ibmhost1

# chmod +x rabbitmqadmin

chmod: cannot access 'rabbitmqadmin': No such file or directory

**Command for clusters:**

C:\Users\rabbitmq20>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Users\rabbitmq20>docker run -d --hostname my-rabbit --name some-rabbit -p 8080:15672 rabbitmq:3-management

Unable to find image 'rabbitmq:3-management' locally

3-management: Pulling from library/rabbitmq

4bbfd2c87b75: Pull complete

d2e110be24e1: Pull complete

889a7173dcfe: Pull complete

913ba9f31013: Pull complete

019520f99983: Pull complete

2256ef99ff86: Pull complete

b356943af928: Pull complete

b614c3545a3f: Pull complete

ea2a4aecbe6b: Pull complete

4c46c17073c8: Pull complete

1aeaad2f488e: Pull complete

80634091df26: Pull complete

85d94c496d09: Pull complete

Digest: sha256:8c3e27cac1acc8a8421e97ef2f04f62c93535c6d602b068b2b16a40d908a2d30

Status: Downloaded newer image for rabbitmq:3-management

9698a30abf007e09fcfa41f4295488b7880bf6cc45a6042b86416220266e2a7a

C:\Users\rabbitmq20>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

9698a30abf00 rabbitmq:3-management "docker-entrypoint.s…" 23 minutes ago Up 23 minutes 4369/tcp, 5671-5672/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:8080->15672/tcp some-rabbit

C:\Users\rabbitmq20>docker exec -it 9698a30abf00 sh

# rabbitmqctl status

Status of node rabbit@my-rabbit ...

Runtime

OS PID: 272

OS: Linux

Uptime (seconds): 1465

Is under maintenance?: false

RabbitMQ version: 3.8.16

Node name: rabbit@my-rabbit

Erlang configuration: Erlang/OTP 23 [erts-11.2.2.1] [source] [64-bit] [smp:2:2] [ds:2:2:10] [async-threads:1]

Erlang processes: 398 used, 1048576 limit

Scheduler run queue: 1

Cluster heartbeat timeout (net\_ticktime): 60

Plugins

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

Enabled plugins:

\* rabbitmq\_prometheus

\* prometheus

\* rabbitmq\_management

\* amqp\_client

\* rabbitmq\_web\_dispatch

\* cowboy

\* cowlib

\* rabbitmq\_management\_agent

Data directory

Node data directory: /var/lib/rabbitmq/mnesia/rabbit@my-rabbit

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

Config files

\* /etc/rabbitmq/rabbitmq.conf

Log file(s)

\* <stdout>

Alarms

(none)

Memory

Total memory used: 0.1048 gb

Calculation strategy: rss

Memory high watermark setting: 0.4 of available memory, computed to: 0.8252 gb

code: 0.0332 gb (29.25 %)

other\_system: 0.0252 gb (22.2 %)

other\_proc: 0.0233 gb (20.54 %)

allocated\_unused: 0.0167 gb (14.75 %)

plugins: 0.0093 gb (8.18 %)

other\_ets: 0.0033 gb (2.9 %)

atom: 0.0015 gb (1.29 %)

mgmt\_db: 0.0005 gb (0.4 %)

metrics: 0.0002 gb (0.2 %)

binary: 0.0001 gb (0.13 %)

mnesia: 0.0001 gb (0.08 %)

quorum\_ets: 0.0 gb (0.04 %)

msg\_index: 0.0 gb (0.03 %)

connection\_other: 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\_procs: 0.0 gb (0.0 %)

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

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

reserved\_unallocated: 0.0 gb (0.0 %)

File Descriptors

Total: 2, limit: 1048479

Sockets: 0, limit: 943629

Free Disk Space

Low free disk space watermark: 0.05 gb

Free disk space: 62.1642 gb

Totals

Connection count: 0

Queue count: 0

Virtual host count: 1

Listeners

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

Interface: [::], port: 15692, protocol: http/prometheus, purpose: Prometheus exporter API over HTTP

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

# ^C

# ^C

# ^C

# ^C

# exit

C:\Users\rabbitmq20>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

rabbitmq 3-management 246db2517862 4 days ago 186MB

C:\Users\rabbitmq20>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

9698a30abf00 rabbitmq:3-management "docker-entrypoint.s…" 2 hours ago Up 2 hours 4369/tcp, 5671-5672/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:8080->15672/tcp some-rabbit

C:\Users\rabbitmq20>dokcer rm -f 9698a30abf00

'dokcer' is not recognized as an internal or external command,

operable program or batch file.

C:\Users\rabbitmq20>docker rm -f 9698a30abf00

9698a30abf00

C:\Users\rabbitmq20>docker rmi -f 246db2517862

Untagged: rabbitmq:3-management

Untagged: rabbitmq@sha256:8c3e27cac1acc8a8421e97ef2f04f62c93535c6d602b068b2b16a40d908a2d30

Deleted: sha256:246db2517862b705e7a7187bf0e9d04595033385b7656fe3f8ca909b3db5cad3

Deleted: sha256:1086a912caafcc33719eb3d9d10473ac0f819efbce7b4b7ee77fdd7e87f6755f

Deleted: sha256:d986470198c75ae39cb69d6ebe8e85360e3ea9244d77696a1392d33cdccd321d

Deleted: sha256:f79b5675d03e15b245f5eeb3be91f71141c9c1e18172f3e2d01c47eb881d9e67

Deleted: sha256:2dcbb0dbeb243663fa6972807e626f7a59012e57437de98544e59e3a121ba643

Deleted: sha256:55aa0349e2f65ad7666d9dcb27046da2bf408a8d17e508f747ba5bc33f1df6b4

Deleted: sha256:070d0ec9385d68dd7739f0b7ff5007836543de21c6050bd453982c3b19cc40a2

Deleted: sha256:95ec7d7bb8e5345715f78321ff0c4eb3bf1597c59258e37c2292774032ef757d

Deleted: sha256:048cd3895a8aeb879408b8e502e43a1f9da7bacc24999aed663aa3185a494295

Deleted: sha256:64339751a08ca8d5504cecb25e8ecd3d6e6bd04ef20a9a00c58767fde03d7893

Deleted: sha256:3679cc383654e11ed83840f948f542663d2af3c00e698417e7788b608f8a3aa0

Deleted: sha256:c6ba5a1f46bb1847ee5523c97c9811f116d5ec5ab9c4671e73f228fe48b35fd7

Deleted: sha256:4f56df2e02e984a9a18f667011ae7abe7262a57118633a1a5a476e5be8e62ac8

Deleted: sha256:50858308da3d192ec20027838c7aaf983333731dc2dcc0cd03c4522495a4cee8

C:\Users\rabbitmq20>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Users\rabbitmq20>docker run -d --hostname rabbit1 --name rabbit1 -e RABBITMQ\_ERLANG\_COOKIE='rabbitcluster' -p 30000:5672 -p 30001:15672 rabbitmq:management

Unable to find image 'rabbitmq:management' locally

management: Pulling from library/rabbitmq

4bbfd2c87b75: Pull complete

d2e110be24e1: Pull complete

889a7173dcfe: Pull complete

913ba9f31013: Pull complete

019520f99983: Pull complete

2256ef99ff86: Pull complete

b356943af928: Pull complete

b614c3545a3f: Pull complete

ea2a4aecbe6b: Pull complete

4c46c17073c8: Pull complete

1aeaad2f488e: Pull complete

80634091df26: Pull complete

85d94c496d09: Pull complete

Digest: sha256:8c3e27cac1acc8a8421e97ef2f04f62c93535c6d602b068b2b16a40d908a2d30

Status: Downloaded newer image for rabbitmq:management

b70105fc87d25adb0e8b7fed0112ef86f898d6e76b9bef3a82184830a66ef066

C:\Users\rabbitmq20>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

rabbitmq management 246db2517862 4 days ago 186MB

C:\Users\rabbitmq20>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b70105fc87d2 rabbitmq:management "docker-entrypoint.s…" 32 seconds ago Up 29 seconds 4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:30000->5672/tcp, 0.0.0.0:30001->15672/tcp rabbit1

C:\Users\rabbitmq20>docker run -d --hostname rabbit2 --name rabbit2 --link rabbit1:rabbit1 -e RABBITMQ\_ERLANG\_COOKIE='rabbitcluster' -p 30002:5672 -p 30003:15672 rabbitmq:management

0eeacb64aee50a7dcb82717f9936b1cb7971b29f380b36726a2d8c38bb87e037

C:\Users\rabbitmq20>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0eeacb64aee5 rabbitmq:management "docker-entrypoint.s…" 10 seconds ago Up 9 seconds 4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:30002->5672/tcp, 0.0.0.0:30003->15672/tcp rabbit2

b70105fc87d2 rabbitmq:management "docker-entrypoint.s…" 2 minutes ago Up 2 minutes 4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:30000->5672/tcp, 0.0.0.0:30001->15672/tcp rabbit1

C:\Users\rabbitmq20>docker run -d --hostname rabbit3 --name rabbit3 --link rabbit1:rabbit1 --link rabbit2:rabbit2 -e RABBITMQ\_ERLANG\_COOKIE='rabbitcluster' -p 30004:5672 -p 30005:15672 rabbitmq:management

185dd8abd4f2c7523c9e39eef845634a1382102aa3868953478006e67dab2ca5

C:\Users\rabbitmq20>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

185dd8abd4f2 rabbitmq:management "docker-entrypoint.s…" 4 seconds ago Up 3 seconds 4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:30004->5672/tcp, 0.0.0.0:30005->15672/tcp rabbit3

0eeacb64aee5 rabbitmq:management "docker-entrypoint.s…" About a minute ago Up About a minute 4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:30002->5672/tcp, 0.0.0.0:30003->15672/tcp rabbit2

b70105fc87d2 rabbitmq:management "docker-entrypoint.s…" 3 minutes ago Up 3 minutes 4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:30000->5672/tcp, 0.0.0.0:30001->15672/tcp rabbit1

C:\Users\rabbitmq20>docker exec -it rabbit2 \bash

OCI runtime exec failed: exec failed: container\_linux.go:349: starting container process caused "exec: \"\\\\bash\": executable file not found in $PATH": unknown

C:\Users\rabbitmq20>docker exec -it rabbit2 sh

# rabbitmqctl stop\_app

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Stopping rabbit application on node rabbit@rabbit2 ...

# rabbitmqctl join\_cluster rabbit@cluster1

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Clustering node rabbit@rabbit2 with rabbit@cluster1

Error: unable to perform an operation on node 'rabbit@cluster1'. Please see diagnostics information and suggestions below.

Most common reasons for this are:

\* Target node is unreachable (e.g. due to hostname resolution, TCP connection or firewall issues)

\* CLI tool fails to authenticate with the server (e.g. due to CLI tool's Erlang cookie not matching that of the server)

\* Target node is not running

In addition to the diagnostics info below:

\* See the CLI, clustering and networking guides on https://rabbitmq.com/documentation.html to learn more

\* Consult server logs on node rabbit@cluster1

\* If target node is configured to use long node names, don't forget to use --longnames with CLI tools

DIAGNOSTICS

===========

attempted to contact: [rabbit@cluster1]

rabbit@cluster1:

\* unable to connect to epmd (port 4369) on cluster1: nxdomain (non-existing domain)

Current node details:

\* node name: 'rabbitmqcli-272-rabbit@rabbit2'

\* effective user's home directory: /var/lib/rabbitmq

\* Erlang cookie hash: MmSybvRiduJ80yQcGOPbuw==

# rabbitmqctl join\_cluster rabbit@rabbit1

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Clustering node rabbit@rabbit2 with rabbit@rabbit1

# rabbitmqctl start\_app

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Starting node rabbit@rabbit2 ...

# exit

C:\Users\rabbitmq20>docker exec -it rabbit3 sh

# rabbitmqctl stop\_app

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Stopping rabbit application on node rabbit@rabbit3 ...

# rabbitmqctl join\_cluster rabbit@rabbit1

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Clustering node rabbit@rabbit3 with rabbit@rabbit1

# rabbitmqctl start\_app

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Starting node rabbit@rabbit3 ...

# exit

C:\Users\rabbitmq20>docker run -d --hostname rabbit4 --name rabbit4 --link rabbit1:rabbit1 --link rabbit2:rabbit2 --link rabbit3:rabbit3 -e RABBITMQ\_ERLANG\_COOKIE='rabbitcluster' -p 30006:5672 -p 30007:15672 rabbitmq:management

a5c4eecd4364eef4134e7267169f9535420e4793c972b033c03648dff386c2a6

C:\Users\rabbitmq20>docker exec -it rabbit4 sh

# rabbitmqctl stop\_app

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Stopping rabbit application on node rabbit@rabbit4 ...

# rabbitmqctl join\_cluster rabbit@rabbit1

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Clustering node rabbit@rabbit4 with rabbit@rabbit1

# rabbitmqctl start\_app

RABBITMQ\_ERLANG\_COOKIE env variable support is deprecated and will be REMOVED in a future version. Use the $HOME/.erlang.cookie file or the --erlang-cookie switch instead.

Starting node rabbit@rabbit4 ...

# exit