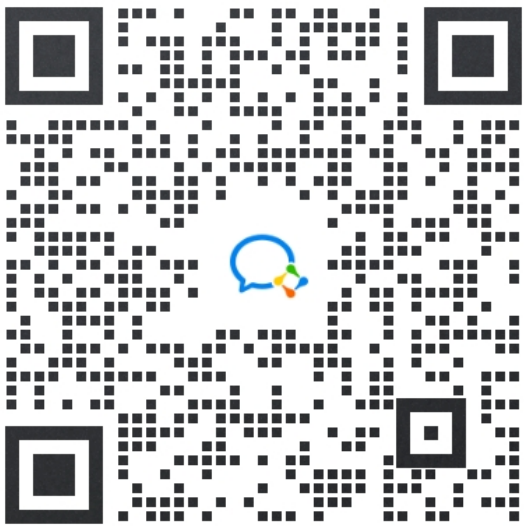
6.2 单机多节点搭建集群

版权说明



1. Ubuntu

1.1 搭建RabbitMQ单节点

1. 安装RabbitMQ

参考前面课件: [RabbitMQ单机安装]

如果已经安装过了, 此步可忽略

2. 确认RabbitMQ运行没问题

```
root@hcss-ecs-2618:~# rabbitmqctl status    #查看RabbitMQ状态
Status of node rabbit@hcss-ecs-2618 ...    #节点名称，下面会用到
Runtime

OS PID: 83170
OS: Linux
Uptime (seconds): 2511652
Is under maintenance?: false
RabbitMQ version: 3.13.1
RabbitMQ release series support status: supported
Node name: rabbit@hcss-ecs-2618
Erlang configuration: Erlang/OTP 26 [erts-14.2.3] [source] [64-bit] [smp:2:2] [ds:2:2:10] [async-threads:1] [jit:ns]
Crypto library: OpenSSL 3.0.2 15 Mar 2022
Erlang processes: 405 used, 1048576 limit
Scheduler run queue: 1
Cluster heartbeat timeout (net_ticktime): 60

Plugins

//...  中间省略

Totals

Connection count: 0
Queue count: 6
Virtual host count: 2

Listeners

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

```

> 端口号介绍:
>
> - **25672**这是Erlang分布式节点通信的默认端口, Erlang是RabbitMQ的底层通信协议.
> - **15672**这是 Web管理界面的默认端口, 通过这个端口可以访问RabbitMQ的Web管理控制台, 用于查看和管理消息队列.

> - **5672**这是 AMQP (Advanced Message Queuing Protocol) 协议的默认端口, 用于客户端与RabbitMQ服务器之间的通信.

3. 再启动两个节点

现在已经安装的RabbitMQ端口号是5672, 15672

再启动两个RabbitMQ服务, 节点名称和端口号分别设置为

Node name	AMQP协议端口号	Web管理界面端口
rabbit2	5673	15673
rabbit3	5674	15674

启动命令如下

```
RABBITMQ_NODE_PORT=5673 RABBITMQ_SERVER_START_ARGS="-rabbitmq_management listener [{port,15673}]" RABBITMQ_NODENAME=rabbit2 rabbitmq-server -detached

RABBITMQ_NODE_PORT=5674 RABBITMQ_SERVER_START_ARGS="-rabbitmq_management listener [{port,15674}]" RABBITMQ_NODENAME=rabbit3 rabbitmq-server -detached
```

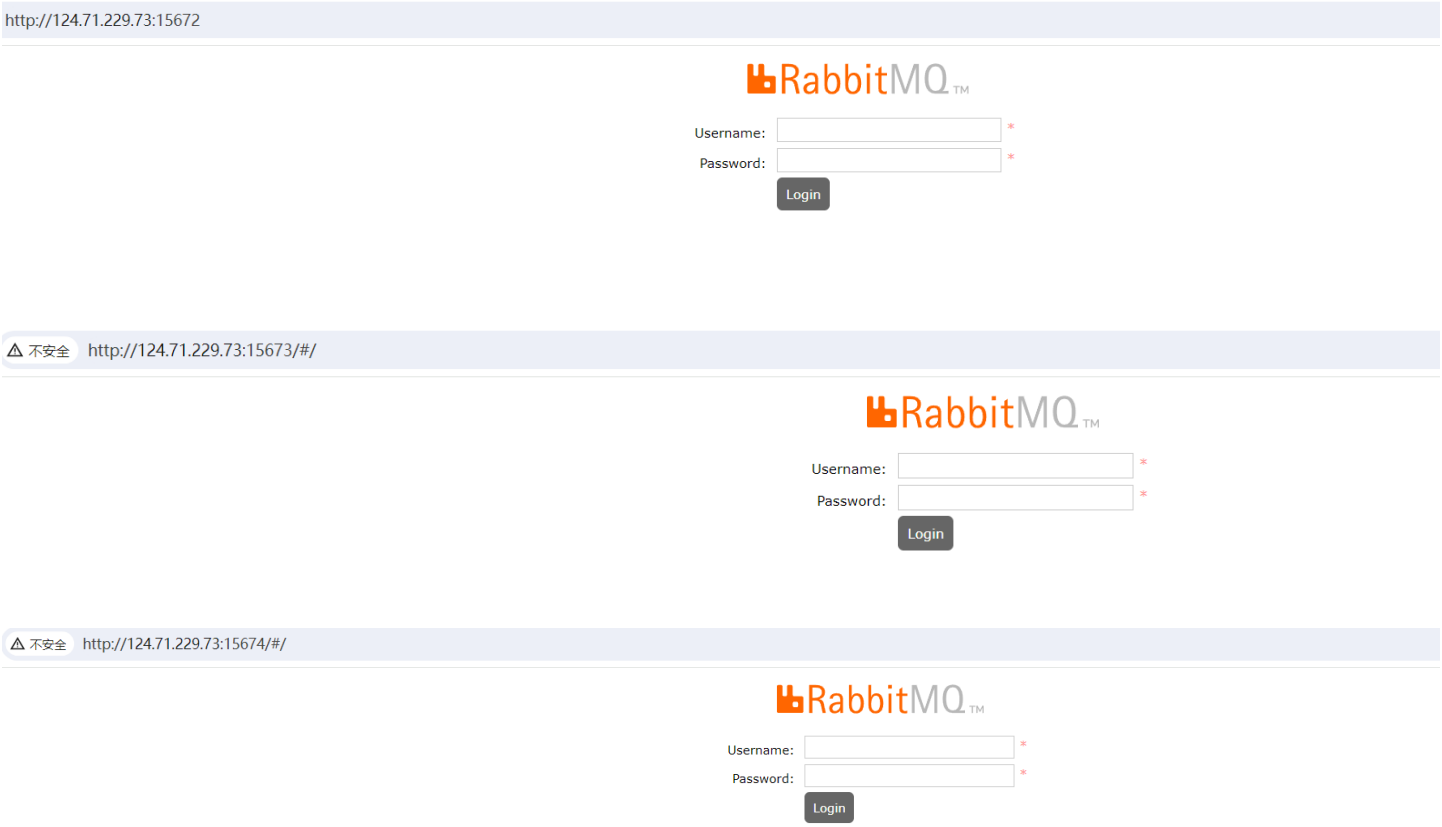
执行结果:

```
root@hcss-ecs-2618:~# RABBITMQ_NODE_PORT=5673 RABBITMQ_SERVER_START_ARGS="-rabbitmq_management listener [{port,15673}]" RABBITMQ_NODENAME=rabbit2 rabbitmq-server -detached
root@hcss-ecs-2618:~# RABBITMQ_NODE_PORT=5674 RABBITMQ_SERVER_START_ARGS="-rabbitmq_management listener [{port,15674}]" RABBITMQ_NODENAME=rabbit3 rabbitmq-server -detached
root@hcss-ecs-2618:~#
```

4. 验证RabbitMQ启动成功

在云服务器开通 15673, 15674端口号

分别测试:http://124.71.229.73:15672/ http://124.71.229.73:15673/ http://124.71.229.73:15674/



1.2 搭建集群

1. 停止服务并重置

停止目的是为了重置

```
rabbitmqctl -n rabbit2 stop_app
rabbitmqctl -n rabbit2 reset

rabbitmqctl -n rabbit3 stop_app
rabbitmqctl -n rabbit3 reset
```

运行结果

```
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit2 stop_app
Stopping rabbit application on node rabbit2@hcss-ecs-2618 ...
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit2 reset
Resetting node rabbit2@hcss-ecs-2618 ...
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit3 stop_app
Stopping rabbit application on node rabbit3@hcss-ecs-2618 ...
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit3 reset
Resetting node rabbit3@hcss-ecs-2618 ...
root@hcss-ecs-2618:~#

```

2. **把rabbit2, rabbit3添加到集群**

rabbit@hcss-ecs-2618 是主节点的node Name, 可以通过 `rabbitmqctl status` 查询

```
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit2 join_cluster rabbit@hcss-ecs-2618
Clustering node rabbit2@hcss-ecs-2618 with rabbit@hcss-ecs-2618
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit3 join_cluster rabbit@hcss-ecs-2618
Clustering node rabbit3@hcss-ecs-2618 with rabbit@hcss-ecs-2618
root@hcss-ecs-2618:~#
```

3. **重启rabbit2,rabbit3**

```
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit2 start_app
Starting node rabbit2@hcss-ecs-2618 ...
root@hcss-ecs-2618:~# rabbitmqctl -n rabbit3 start_app
Starting node rabbit3@hcss-ecs-2618 ...
root@hcss-ecs-2618:~#

```

4. **查看集群状态**

```
rabbitmqctl cluster_status -n rabbit
```

运行结果

```
root@hcss-ecs-2618:~# rabbitmqctl cluster_status -n rabbit
Cluster status of node rabbit@hcss-ecs-2618 ...
Basics

Cluster name: rabbit@hcss-ecs-2618
Total CPU cores available cluster-wide: 6

Disk Nodes

rabbit2@hcss-ecs-2618
rabbit3@hcss-ecs-2618
rabbit@hcss-ecs-2618

Running Nodes
```

```
15
16 rabbit2@hcss-ecs-2618
17 rabbit3@hcss-ecs-2618
18 rabbit@hcss-ecs-2618
19
20 Versions
21
22 rabbit@hcss-ecs-2618: RabbitMQ 3.13.2 on Erlang 26.2.5
23 rabbit2@hcss-ecs-2618: RabbitMQ 3.13.2 on Erlang 26.2.5
24 rabbit3@hcss-ecs-2618: RabbitMQ 3.13.2 on Erlang 26.2.5
25
26 CPU Cores
27
28 Node: rabbit@hcss-ecs-2618, available CPU cores: 2
29 Node: rabbit2@hcss-ecs-2618, available CPU cores: 2
30 Node: rabbit3@hcss-ecs-2618, available CPU cores: 2
31
32 Maintenance status
33
34 Node: rabbit@hcss-ecs-2618, status: not under maintenance
35 Node: rabbit2@hcss-ecs-2618, status: not under maintenance
36 Node: rabbit3@hcss-ecs-2618, status: not under maintenance
37
38 Alarms
39
40 (none)
41
42 Network Partitions
43
44 (none)
45
46 Listeners
47
48 Node: rabbit@hcss-ecs-2618, interface: [::], port: 15672, protocol: http,
   purpose: HTTP API
49 Node: rabbit@hcss-ecs-2618, interface: [::], port: 25672, protocol:
   clustering, purpose: inter-node and CLI tool communication
50 Node: rabbit@hcss-ecs-2618, interface: [::], port: 5672, protocol: amqp,
   purpose: AMQP 0-9-1 and AMQP 1.0
51 Node: rabbit2@hcss-ecs-2618, interface: [::], port: 15673, protocol: http,
   purpose: HTTP API
52 Node: rabbit2@hcss-ecs-2618, interface: [::], port: 25673, protocol:
   clustering, purpose: inter-node and CLI tool communication
53 Node: rabbit2@hcss-ecs-2618, interface: [::], port: 5673, protocol: amqp,
   purpose: AMQP 0-9-1 and AMQP 1.0
54 Node: rabbit3@hcss-ecs-2618, interface: [::], port: 15674, protocol: http,
   purpose: HTTP API
```

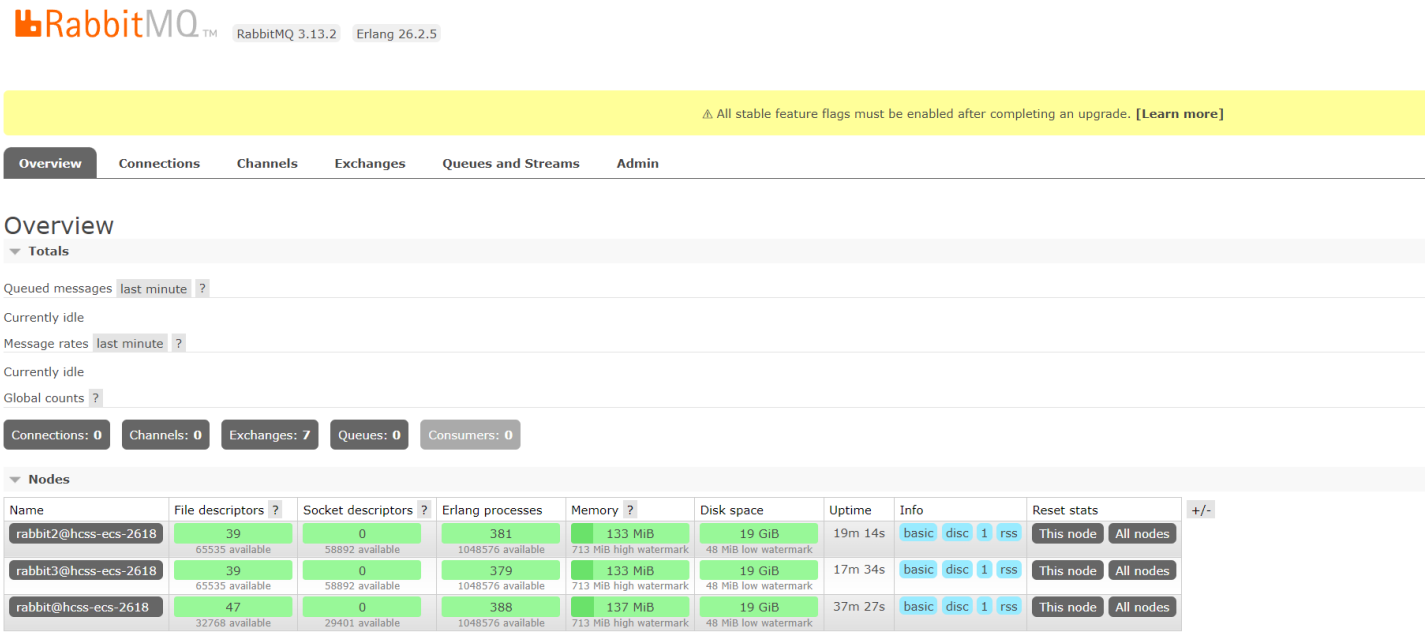
```
Node: rabbit3@hcss-ecs-2618, interface: [::], port: 25674, protocol: clustering, purpose: inter-node and CLI tool communication
Node: rabbit3@hcss-ecs-2618, interface: [::], port: 5674, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

Feature flags

Flag: classic_mirrored_queue_version, state: enabled
Flag: classic_queue_type_delivery_support, state: enabled
Flag: detailed_queues_endpoint, state: disabled
Flag: direct_exchange_routing_v2, state: enabled
Flag: drop_unroutable_metric, state: enabled
Flag: empty_basic_get_metric, state: enabled
Flag: feature_flags_v2, state: enabled
Flag: implicit_default_bindings, state: enabled
Flag: khepri_db, state: disabled
Flag: listener_records_in_ets, state: enabled
Flag: maintenance_mode_status, state: enabled
Flag: message_containers, state: enabled
Flag: quorum_queue, state: enabled
Flag: quorum_queue_non_voters, state: enabled
Flag: restart_streams, state: enabled
Flag: stream_filtering, state: enabled
Flag: stream_queue, state: enabled
Flag: stream_sac_coordinator_unblock_group, state: enabled
Flag: stream_single_active_consumer, state: enabled
Flag: stream_update_config_command, state: enabled
Flag: tracking_records_in_ets, state: enabled
Flag: user_limits, state: enabled
Flag: virtual_host_metadata, state: enabled
root@hcss-ecs-2618:~#


```

通过主节点(rabbit)管理界面, 可以看到集群其他节点



2. CentOS

2.1 搭建RabbitMQ单节点

1. 安装RabbitMQ

参考前面课件: [RabbitMQ单机安装]

如果已经安装过了, 此步可忽略

2. 确认RabbitMQ运行没问题

```
[root@VM-24-3-centos ~]# rabbitmqctl status   #查看RabbitMQ状态
Status of node rabbit@VM-24-3-centos ...  #节点名称, 后面会使用
Runtime

OS PID: 999
OS: Linux
Uptime (seconds): 587245
Is under maintenance?: false
RabbitMQ version: 3.8.30
Node name: rabbit@VM-24-3-centos
Erlang configuration: Erlang/OTP 23 [erts-11.2.2.10] [source] [64-bit]
[smp:2:2] [ds:2:2:10] [async-threads:1] [hipe]
Erlang processes: 369 used, 1048576 limit
Scheduler run queue: 1
Cluster heartbeat timeout (net_ticktime): 60

//...  中间省略

Totals
```

```
19
20 Connection count: 0
21 Queue count: 0
22 Virtual host count: 1
23
24 Listeners
25
26 Interface: [::], port: 15672, protocol: http, purpose: HTTP API
27 Interface: [::], port: 25672, protocol: clustering, purpose: inter-node and
   CLI tool communication
28 Interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0
29 [root@VM-24-3-centos ~]#
30
```

3. 再启动两个节点

现在已经安装的RabbitMQ端口号是5672, 15672

再启动两个RabbitMQ服务, 节点名称和端口号分别设置为

Node name	AMQP协议端口号	Web管理界面端口
rabbit2	5673	15673
rabbit3	5674	15674

启动命令如下

```
1 RABBITMQ_NODE_PORT=5673 RABBITMQ_SERVER_START_ARGS="-rabbitmq_management
  listener [{port,15673}]" RABBITMQ_NODENAME=rabbit2 rabbitmq-server -detached
2
3 RABBITMQ_NODE_PORT=5674 RABBITMQ_SERVER_START_ARGS="-rabbitmq_management
  listener [{port,15674}]" RABBITMQ_NODENAME=rabbit3 rabbitmq-server -detached
```

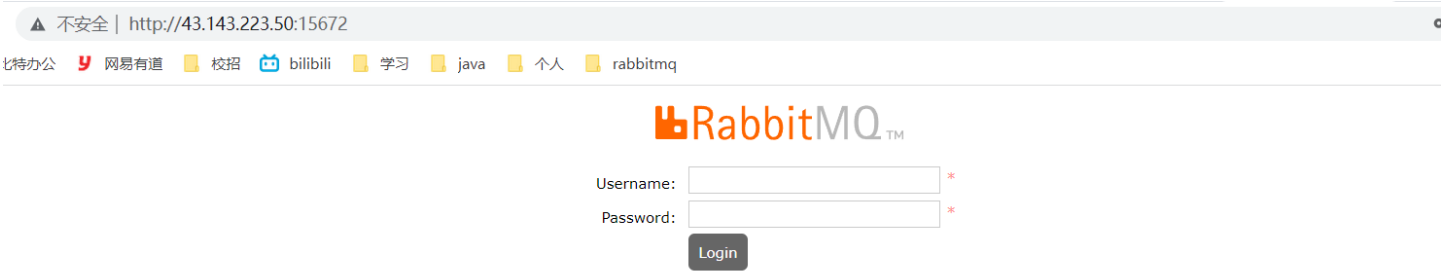
执行结果:

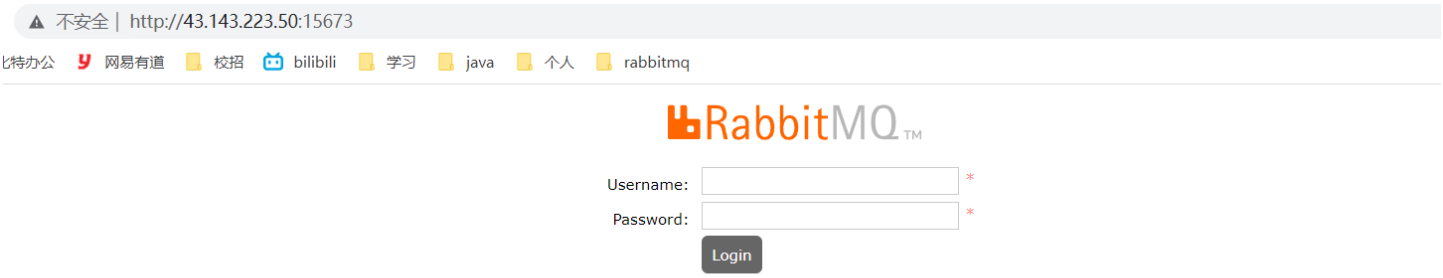
```
1 [root@VM-24-3-centos ~]# RABBITMQ_NODE_PORT=5673 RABBITMQ_SERVER_START_ARGS="-
  rabbitmq_management listener [{port,15673}]" RABBITMQ_NODENAME=rabbit2
  rabbitmq-server -detached
2 [root@VM-24-3-centos ~]# RABBITMQ_NODE_PORT=5674 RABBITMQ_SERVER_START_ARGS="-
  rabbitmq_management listener [{port,15674}]" RABBITMQ_NODENAME=rabbit3
  rabbitmq-server -detached
3 [root@VM-24-3-centos ~]#
```

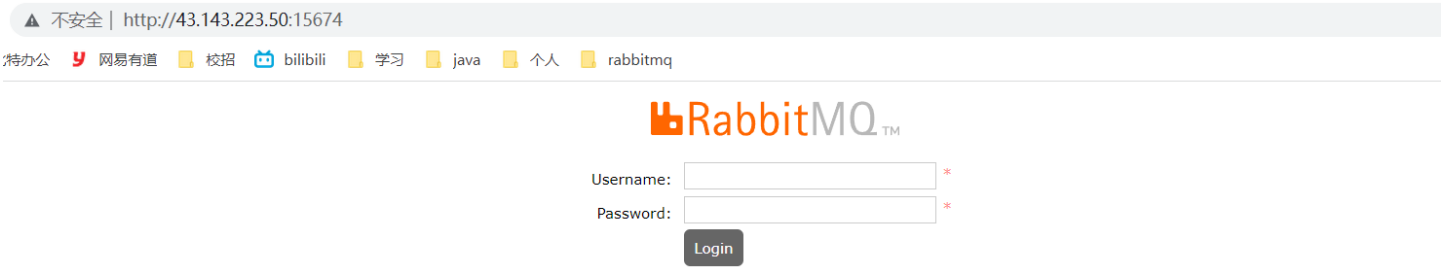
4. 验证RabbitMQ启动成功

在云服务器开通 15673, 15674端口号

分别测试:http://43.143.223.50:15672/ http://43.143.223.50:15673/ http://43.143.223.50:15674/







2.2 搭建集群

1. 停止服务并重置

停止目的是为了重置

```
rabbitmqctl -n rabbit2 stop_app
rabbitmqctl -n rabbit2 reset

rabbitmqctl -n rabbit3 stop_app
rabbitmqctl -n rabbit3 reset
```

运行结果

```
[root@VM-24-3-centos rabbitmq]#  rabbitmqctl -n rabbit2 stop_app
Stopping rabbit application on node rabbit3@VM-24-3-centos ...
[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit2 reset
Resetting node rabbit3@VM-24-3-centos ...
[root@VM-24-3-centos rabbitmq]#
[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit3 stop_app
Stopping rabbit application on node rabbit4@VM-24-3-centos ...
[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit3 reset
Resetting node rabbit4@VM-24-3-centos ...
[root@VM-24-3-centos rabbitmq]#
```

2. 把rabbit2, rabbit3添加到集群

rabbit@VM-24-3-centos 是主节点的node Name, 可以通过 `rabbitmqctl status` 查询

```
[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit2 join_cluster rabbit@VM-24-3-centos
Clustering node rabbit3@VM-24-3-centos with rabbit@VM-24-3-centos
[root@VM-24-3-centos rabbitmq]#

[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit3 join_cluster rabbit@VM-24-3-centos
Clustering node rabbit4@VM-24-3-centos with rabbit@VM-24-3-centos
[root@VM-24-3-centos rabbitmq]#
```

3. 重启rabbit2,rabbit3

```
[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit2 start_app
Starting node rabbit3@VM-24-3-centos ...
[root@VM-24-3-centos rabbitmq]# rabbitmqctl -n rabbit3 start_app
Starting node rabbit4@VM-24-3-centos ...
[root@VM-24-3-centos rabbitmq]#

```

4. 查看集群状态

```
rabbitmqctl cluster_status -n rabbit
```

运行结果

```
[root@VM-24-3-centos rabbitmq]# rabbitmqctl cluster_status -n rabbit
Cluster status of node rabbit@VM-24-3-centos ...
Basics

Cluster name: rabbit@VM-24-3-centos

Disk Nodes

rabbit3@VM-24-3-centos
rabbit2@VM-24-3-centos
rabbit@VM-24-3-centos

Running Nodes

rabbit3@VM-24-3-centos
rabbit2@VM-24-3-centos
rabbit@VM-24-3-centos

Versions

rabbit3@VM-24-3-centos: RabbitMQ 3.8.30 on Erlang 23.3.4.11
rabbit4@VM-24-3-centos: RabbitMQ 3.8.30 on Erlang 23.3.4.11
rabbit@VM-24-3-centos: RabbitMQ 3.8.30 on Erlang 23.3.4.11

Maintenance status

Node: rabbit3@VM-24-3-centos, status: not under maintenance
Node: rabbit4@VM-24-3-centos, status: not under maintenance
Node: rabbit@VM-24-3-centos, status: not under maintenance

Alarms

(none)

Network Partitions

(none)

Listeners

Node: rabbit3@VM-24-3-centos, interface: [::], port: 15673, protocol: http,
purpose: HTTP API
Node: rabbit3@VM-24-3-centos, interface: [::], port: 25673, protocol:
clustering, purpose: inter-node and CLI tool communication
Node: rabbit3@VM-24-3-centos, interface: [::], port: 5673, protocol: amqp,
purpose: AMQP 0-9-1 and AMQP 1.0
```

```
Node: rabbit4@VM-24-3-centos, interface: [::], port: 15674, protocol: http, purpose: HTTP API
Node: rabbit4@VM-24-3-centos, interface: [::], port: 25674, protocol: clustering, purpose: inter-node and CLI tool communication
Node: rabbit4@VM-24-3-centos, interface: [::], port: 5674, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0
Node: rabbit@VM-24-3-centos, interface: [::], port: 15672, protocol: http, purpose: HTTP API
Node: rabbit@VM-24-3-centos, interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool communication
Node: rabbit@VM-24-3-centos, interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0

Feature flags

Flag: drop_unroutable_metric, state: enabled
Flag: empty_basic_get_metric, state: enabled
Flag: implicit_default_bindings, state: enabled
Flag: maintenance_mode_status, state: enabled
Flag: quorum_queue, state: enabled
Flag: user_limits, state: enabled
Flag: virtual_host_metadata, state: enabled
[root@VM-24-3-centos rabbitmq]#

```

通过主节点(rabbit)管理界面, 可以看到集群其他节点