

# Manuelles Hinzufügen des Rabbitmq-Clusters

## Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Problem](#)

[Überprüfung, ob RabbitMQ außerhalb des Clusters ist](#)

[Lösung](#)

## Einführung

In diesem Dokument wird beschrieben, wie RabbitMQ manuell zu einem Cluster hinzugefügt wird, wenn der Cluster beschädigt ist.

## Voraussetzungen

### Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

### Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Mindestens 2 RabbitMQ-Server
- Ein Load Balancer

## Problem

Dieser Artikel führt Sie durch die Überprüfung des RabbitMQ-Clusters und das manuelle Hinzufügen dieser Instanz zum Cluster. CloudCenter bietet einen Assistenten zur Konfiguration von High Availability (HA) für RabbitMQ, aber in einigen Fällen sagt es, dass die HA erfolgreich konfiguriert ist, nachdem sie den Assistenten verlassen hat, aber der RabbitMQ-Cluster ist nicht richtig geformt.

### Überprüfung, ob RabbitMQ außerhalb des Clusters ist

Schritt 1: Melden Sie sich mit der CLI-Konsole bei allen RabbitMQ-Servern an.

Schritt 2: Überprüfen Sie, ob der RabbitMQ-Server auf allen Instanzen ausgeführt wird.

```
#ps -ef | grep rabbit
```

Output:

```
rabbitmq 1677      1  0 14:47 ?           00:00:00 /usr/lib/erlang/erts-6.4/bin/epmd -daemon
root     1973      1  0 14:47 ?           00:00:00 /bin/sh /etc/rc.d/init.d/rabbitmq-server start
root     2000     1973  0 14:47 ?           00:00:00 /bin/bash -c ulimit -S -c 0 >/dev/null 2>&1 ;
/usr/sbin/rabbitmq-server
root     2003     2000  0 14:47 ?           00:00:00 su rabbitmq -s /bin/sh -c
/usr/lib/rabbitmq/bin/rabbitmq-server
rabbitmq 2026     2003  0 14:47 ?           00:00:05 /usr/lib/erlang/erts-6.4/bin/beam -W w -K true -
A30 -P 1048576 -- -root /usr/lib/erlang -programe erl -- -home /var/lib/rabbitmq -- -pa
/usr/lib/rabbitmq/lib/rabbitmq_server-3.5.1/sbin/../ebin -noshell -noinput -s rabbit boot -sname
rabbit@ip-172-31-32-101 -boot start_sasl -config /etc/rabbitmq/rabbitmq -kernel
inet_default_connect_options [{nodelay,true}] -sasl errlog_type error -sasl sasl_error_logger
false -rabbit error_logger {file,"/var/log/rabbitmq/rabbit@ip-172-31-32-101.log"} -rabbit
sasl_error_logger {file,"/var/log/rabbitmq/rabbit@ip-172-31-32-101-sasl.log"} -rabbit
enabled_plugins_file "/etc/rabbitmq/enabled_plugins" -rabbit plugins_dir
"/usr/lib/rabbitmq/lib/rabbitmq_server-3.5.1/sbin/../plugins" -rabbit plugins_expand_dir
"/var/lib/rabbitmq/mnesia/rabbit@ip-172-31-32-101-plugins-expand" -os_mon start_cpu_sup false -
os_mon start_disk_sup false -os_mon start_memsup false -mnesia dir
"/var/lib/rabbitmq/mnesia/rabbit@ip-172-31-32-101" -kernel inet_dist_listen_min 25672 -kernel
inet_dist_listen_max 25672
rabbitmq 2242     2026  0 14:47 ?           00:00:00 inet_gethost 4
rabbitmq 2243     2242  0 14:47 ?           00:00:00 inet_gethost 4
root     2602     2588  0 15:04 pts/0      00:00:00 grep --color=auto rabbit
OR
#/sbin/service rabbitmq-server status
```

Status of node 'rabbit@ip-172-31-32-101' ...

```
[{pid,2026},
 {running_applications,
  [{rabbitmq_management,"RabbitMQ Management Console","3.5.1"},
   {rabbitmq_web_dispatch,"RabbitMQ Web Dispatcher","3.5.1"},
   {webmachine,"webmachine","1.10.3-rmq3.5.1-gite9359c7"},
   {mochiweb,"MochiMedia Web Server","2.7.0-rmq3.5.1-git680dba8"},
   {rabbitmq_management_agent,"RabbitMQ Management Agent","3.5.1"},
   {rabbit,"RabbitMQ","3.5.1"}],
```

----- Text omitted for brevity

Schritt 3: Überprüfen Sie den Cluster-Status aller Instanzen mit den folgenden Befehlen:

```
[root@ip-172-31-32-101 ~]# rabbitmqctl cluster_status
Cluster status of node 'rabbit@ip-172-31-32-101' ...
[{nodes,[{disc,['rabbit@ip-172-31-32-101']}]}],
 {running_nodes,['rabbit@ip-172-31-32-101']},
 {cluster_name,<<"rabbit@ip-172-31-32-101.us-east-2.compute.internal">>},
 {partitions,[]}]
```

In dieser Ausgabe können Sie festlegen, dass nur ein Knoten im Cluster ausgeführt wird.

## Lösung

In diesem Szenario fügen Sie Ihrem Cluster **rabbit@ip-172-31-32-101** hinzu **rabbit@ip-172-31-45-110.us-east-2.compute.internal**

Schritt 1: Um den zweiten RabbitMQ-Server zu verbinden, müssen Sie sicherstellen, dass die Anwendung beendet wird und dem Cluster beitrifft.

Schritt 2: Wechseln Sie zum RabbitMQ2-Server, und beenden Sie die Anwendung.

```
[root@ip-172-31-32-101 ~]# rabbitmqctl stop_app
Stopping node 'rabbit@ip-172-31-32-101' ...
[root@ip-172-31-32-101 ~]#
```

Schritt 3: Treten Sie dem RabbitMQ2-Server zum RabbitMQ1-Cluster bei.

```
[root@ip-172-31-32-101 ~]# rabbitmqctl join_cluster rabbit@ip-172-31-45-110
Clustering node 'rabbit@ip-172-31-32-101' with 'rabbit@ip-172-31-45-110' ...
```

Schritt 4: Starten Sie die Anwendung rabbitmq2.

```
[root@ip-172-31-32-101 ~]# rabbitmqctl start_app
Starting node 'rabbit@ip-172-31-32-101' ...
```

Sie können sehen, dass die beiden Knoten in einem Cluster verbunden sind, wenn Sie den Befehl **cluster\_status** auf einem der Knoten ausführen.

Server1 : Output

```
[root@ip-172-31-45-110 ~]# rabbitmqctl cluster_status
Cluster status of node 'rabbit@ip-172-31-45-110' ...
[{nodes,[{disc,['rabbit@ip-172-31-32-101','rabbit@ip-172-31-45-110']}]},
 {running_nodes,['rabbit@ip-172-31-32-101','rabbit@ip-172-31-45-110']},
 {cluster_name,<<"rabbit@ip-172-31-45-110.us-east-2.compute.internal">>},
 {partitions,[]}]
```

Server2 : Output

```
[root@ip-172-31-32-101 ~]# rabbitmqctl cluster_status
Cluster status of node 'rabbit@ip-172-31-32-101' ...
[{nodes,[{disc,['rabbit@ip-172-31-32-101','rabbit@ip-172-31-45-110']}]},
 {running_nodes,['rabbit@ip-172-31-45-110','rabbit@ip-172-31-32-101']},
 {cluster_name,<<"rabbit@ip-172-31-45-110.us-east-2.compute.internal">>},
 {partitions,[]}]
```