

Como adicionar manualmente o conjunto de Rabbitmq

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Problema](#)

[Verifique se RabbitMQ é fora do conjunto](#)

[Solução](#)

Introdução

Este documento descreve como adicionar manualmente RabbitMQ a um conjunto se o conjunto é quebrado.

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- Server de RabbitMQ do mínimo 2
- Um equilibrador da carga

Problema

Este artigo guia-o em como verificar o conjunto de RabbitMQ e adicionar manualmente aqueles exemplo ao conjunto. CloudCenter fornece um assistente para configurar a Alta disponibilidade (HA) para RabbitMQ contudo, em bastante alguns exemplo diz que o HA está configurado com sucesso depois que retira o assistente mas o conjunto de RabbitMQ não está formado corretamente.

Verifique se RabbitMQ é fora do conjunto

Etapa 1. Entre a todo o server de RabbitMQ com o uso do console CLI.

Etapa 2. Verifique se o server de RabbitMQ é executado em todos os exemplos.

```
#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 -prognome 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_disksup 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

Etapa 3. Verifique o estado do conjunto de todo o exemplo com estes comandos:

```
[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,[]}]
```

Nesta saída, você pode identificar que há somente um nó que é executado no conjunto.

Solução

Nesta encenação, você adiciona **rabbit@ip-172-31-32-101** a seu conjunto **rabbit@ip-172-31-45-110.us-east-2.compute.internal**

Etapa 1. A fim ligar o segundo server de RabbitMQ, você precisa de assegurar-se de que o aplicativo esteja parado e se junte ao conjunto.

Etapa 2. Comute ao server RabbitMQ2 e pare o aplicativo.

```
[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,[]}]
```

Etapa 3. Junte-se ao server RabbitMQ2 ao conjunto RabbitMQ1.

```
[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,[]}]
```

Etapa 4. Comece o aplicativo rabbitmq2.

```
[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,[]}]
```

Você pode ver que os dois Nós estão juntados em um conjunto quando você executa o comando **do cluster_status em qualquer um dos Nós.**

```
[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,[]}]
```