

Come aggiungere manualmente un cluster Rabbitmq

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Problema](#)

[Verifica se RabbitMQ è fuori dal cluster](#)

[Soluzione](#)

Introduzione

Questo documento descrive come aggiungere manualmente RabbitMQ a un cluster se il cluster è interrotto.

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Almeno 2 server RabbitMQ
- Un servizio di bilanciamento del carico

Problema

In questo articolo viene illustrato come verificare il cluster RabbitMQ e aggiungere manualmente tali istanze al cluster. CloudCenter fornisce una procedura guidata per configurare High Availability (HA) per RabbitMQ. Tuttavia, in un certo numero di casi, dice che la HA è stata configurata correttamente dopo essere uscita dalla procedura guidata, ma il cluster RabbitMQ non è formato correttamente.

Verifica se RabbitMQ è fuori dal cluster

Passaggio 1. Accedere a tutti i server RabbitMQ utilizzando la console CLI.

Passaggio 2. Verificare se il server RabbitMQ è in esecuzione su tutte le istanze.

```
#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 -progname 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
```

Passaggio 3. Verificare lo stato del cluster di tutte le istanze con questi comandi:

```
[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 questo output è possibile identificare che nel cluster è in esecuzione un solo nodo.

Soluzione

In questo scenario, aggiungere **rabbit@ip-172-31-32-101** al cluster **rabbit@ip-172-31-45-110.us-**

east-2.compute.internal

Passaggio 1. Per collegare il secondo server RabbitMQ, è necessario assicurarsi che l'applicazione sia arrestata e si unisca al cluster.

Passaggio 2. Passare al server RabbitMQ2 e arrestare l'applicazione.

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

Passaggio 3. Aggiungere il server RabbitMQ2 al cluster RabbitMQ1.

```
[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' ...
```

Passaggio 4. Avviare l'applicazione rabbitmq2.

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

È possibile verificare che i due nodi sono uniti in un cluster quando si esegue il comando **cluster_status** in uno dei nodi.

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, []}]
```