

Comment ajouter manuellement la batterie de Rabbitmq

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Problème](#)

[Vérifiez si RabbitMQ est hors de batterie](#)

[Solution](#)

Introduction

Ce document décrit comment ajouter manuellement RabbitMQ à une batterie si la batterie est cassée.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

[Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Serveur de RabbitMQ du minimum 2
- Un équilibreur de charge

Problème

Cet article vous guide sur la façon dont vérifier la batterie de RabbitMQ et ajouter manuellement ceux citent à la batterie. CloudCenter fournit un assistant pour configurer la Haute disponibilité (ha) pour RabbitMQ cependant, dans tout à fait quelques uns exemple il indique que l'ha est avec succès configuré après qu'il quitte l'assistant mais la batterie de RabbitMQ n'est pas formée correctement.

Vérifiez si RabbitMQ est hors de batterie

Étape 1. Ouvrez une session à tout le serveur de RabbitMQ avec l'utilisation de la console CLI.

Étape 2. Vérifiez si le serveur de RabbitMQ fonctionne sur tous les exemples.

```
#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

Étape 3. Vérifiez le statut de batterie de tout l'exemple avec ces commandes :

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

Dans cette sortie, vous pouvez identifier qu'il y a seulement un noeud qui fonctionne dans la batterie.

Solution

Dans ce scénario, vous ajoutez **rabbit@ip-172-31-32-101** à votre batterie **rabbit@ip-172-31-45-110.us-east-2.compute.internal**

Étape 1. Afin de joindre le deuxième serveur de RabbitMQ, vous devez s'assurer que l'application est arrêtée et joint la batterie.

Étape 2. Commutez au serveur RabbitMQ2 et arrêtez l'application.

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

Étape 3. Joignez le serveur RabbitMQ2 à la batterie 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, []}]
```

Étape 4. Commencez l'application 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, []}]
```

Vous pouvez voir que les deux Noeuds sont joints dans une batterie quand vous exécutez la commande de **cluster_status** sur l'un ou l'autre des Noeuds.

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