

Как вручную добавить кластер Rabbitmq

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Проблема](#)

[Проверьте, ли RabbitMQ Вне Кластера](#)

[Решение](#)

Введение

Этот документ описывает, как вручную добавить RabbitMQ к кластеру, если сломан кластер.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Минимальные 2 сервера RabbitMQ
- Один балансировщик загрузки

Проблема

Эта статья ведет вас о том, как проверить, что RabbitMQ кластеризуют и вручную добавляют тех экземпляр к кластеру. CloudCenter предоставляет мастера для настройки Высокой доступности (HA) для RabbitMQ, однако, в довольно многих инстанцируют, это говорит, что HA успешно настроен после того, как это выходит из мастера, но кластер RabbitMQ не сформирован должным образом.

Проверьте, ли RabbitMQ Вне Кластера

Шаг 1. Войдите ко всему серверу RabbitMQ с использованием консоли CLI.

Шаг 2. Проверьте, работает ли сервер RabbitMQ на всех экземплярах.

```
#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

Шаг 3. Проверьте кластерный статус всего экземпляра с этими командами:

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

В этих выходных данных можно определить это существует только один узел, который выполняется в кластере.

Решение

В этом сценарии вы добавляете **rabbit@ip-172-31-32-101** к вашему кластеру **rabbit@ip-172-31-45-110.us-east-2.compute.internal**

Шаг 1. Для соединения второго сервера RabbitMQ необходимо гарантировать, что приложение остановлено и присоединяется к кластеру.

Шаг 2. Коммутатор к серверу RabbitMQ2 и останавливает приложение.

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

Шаг 3. Соедините сервер RabbitMQ2 с кластером 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, []}]
```

Шаг 4. . Запустите rabbitmq2 приложение.

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

Вы видите, что к этим двум узлам присоединяются в кластере, когда вы выполняете **cluster_status** команду на любом из узлов.

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