Red Hat CloudForms 4.2

Configuring High Availability

Installing and configuring database high availability in a Red Hat CloudForms environment

Last Updated: 2017-12-18

Red Hat CloudForms 4.2 Configuring High Availability

Installing and configuring database high availability in a Red Hat CloudForms environment

Red Hat CloudForms Documentation Team
cloudforms-docs@redhat.com

Legal Notice

Abstract

This guide provides instructions on installing, configuring, and testing database high availability in Red Hat CloudForms. Information and procedures in this book are relevant to CloudForms Management Engine administrators. If you have a suggestion for improving this guide or have found an error, please submit a Bugzilla report at http://bugzilla.redhat.com against Red Hat CloudForms Management Engine for the Documentation component. Please provide specific details, such as the section number, guide name, and CloudForms version so we can easily locate the content.

Table of Contents

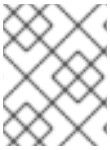
CHAPTER 1. ENVIRONMENT OVERVIEW

This guide describes how to configure database high availability in a Red Hat CloudForms environment. This configuration allows for disaster mitigation: a failure in the primary database does not result in downtime, as the standby database takes over the failed database's processes. This is made possible by database replication between two or more database servers. In CloudForms, these servers are *database-only CloudForms appliances* which do not have **evmserverd** processes enabled.

In this configuration, only one database is writable at any given time. This procedure also does not provide scalability or a multi-master database setup. While a Red Hat CloudForms environment is comprised of an engine tier and a database tier, this configuration affects only the database tier and does not provide load balancing for the appliances.

This guide describes two types of appliances used in high availability:

- *Database-only CloudForms appliances*, which do not have **evmserverd** processes enabled or a user interface.
- *CloudForms appliances*, which are standard appliances containing a user interface and which have other **evmserverd** processes enabled.



NOTE

Manual steps are required to reintroduce the failed database node back as the standby server. See Section 3.3, "Reintroducing the Failed Node".

1.1. REQUIREMENTS

For a high availability Red Hat CloudForms environment, you need a virtualization host containing at minimum four virtual machines with CloudForms installed, consisting of:

- One virtual machine for the primary external database containing a minimum of 4GB dedicated disk space
- One virtual machine for the standby external database containing a minimum of 4GB dedicated disk space
- Two virtual machines for the CloudForms appliances

See Planning in the *Deployment Planning Guide* for information on setting up the correct disk space for the database-only appliances.

The database-only appliances should reside on a highly reliable local network in the same location.

IMPORTANT

It is essential to use the same Red Hat CloudForms appliance template version to install each virtual machine in this environment.

See the Red Hat Customer Portal to obtain the appliance download for the platform you are running CloudForms on.

Correct time synchronization is required before installing the cluster. After installing the appliances, configure time synchronization on all appliances using **chronyd**.

NOTE

Red Hat recommends using a DNS server for a high availability configuration, as DNS names can be updated more quickly than IP addresses when restoring an operation in a different location, network, or datacenter.

CHAPTER 2. INSTALLING THE APPLIANCES

This chapter outlines the steps for installing and configuring the Red Hat CloudForms components needed for high availability: a database cluster comprised of primary and standby database-only appliances, and two (at minimum) CloudForms appliances.

2.1. INSTALLING THE PRIMARY DATABASE-ONLY APPLIANCE

The primary database-only appliance functions as an external database to the CloudForms appliances.

1. Deploy a CloudForms appliance with an extra partition for the database at a size appropriate for your deployment. For recommendations on disk space, see Database Requirements in the *Deployment Planning Guide*.
2. Configure time synchronization on the appliance:
   a. Edit `/etc/chronyd.conf` with valid NTP server information.
   b. Re-synchronize time information across the appliances:

```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

3. SSH into the CloudForms appliance to enter the appliance console.
4. Configure the hostname by selecting **Set Hostname**.
5. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
6. Select **Configure Database**.
7. Select **Create key** to create the encryption key. You can create a new key, or use an existing key on your system by selecting **Fetch key from remote machine** and following the prompts.
8. Select **Create Internal Database**.
9. Select the database disk. CloudForms then activates the configuration.
10. For **Should this appliance run as a standalone database server?** , select **Y**. Selecting this option configures this appliance as a database-only appliance, and therefore the CFME application and `evmserverd` processes will not run. This is required in highly available database deployments.

WARNING

This configuration is not reversible.

11. Create the database password.

NOTE

Do not create a region at this stage in the procedure.

You can check the configuration on the appliance console details screen. If configured successfully, **Local Database Server** shows as **`running (primary)`**.

2.2. INSTALLING A CLOUDFORMS APPLIANCE

Install and configure a CloudForms appliance to point to the primary database server. This appliance does not serve as a database server.

From this appliance, you can then create a database region and configure the primary database.

NOTE

After installing and configuring an empty database-only appliance in Section 2.1, "Installing the Primary Database-Only Appliance", the steps in this section create the database schema used by CloudForms on the primary database-only appliance, and prepare the initial data. Region metadata is required to configure the primary database-only appliance as a primary node in the replication cluster.

This must be configured from the CloudForms appliance before the primary and secondary database-only appliances can be configured.

1. Deploy a CloudForms appliance with an extra partition for the database at a size appropriate for your deployment. For recommendations on disk space, see Database Requirements in the *Deployment Planning Guide*.
2. Configure time synchronization on the appliance:
   a. Edit **`/etc/chronyd.conf`** with valid NTP server information.
   b. Re-synchronize time information across the appliances:

```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

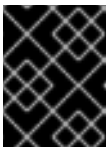
3. SSH into the CloudForms appliance to enter the appliance console.
4. Configure the hostname by selecting **Set Hostname**.
5. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
6. Select **Configure Database**.
7. Configure this appliance to use the encryption key from the primary database-only appliance:
   a. For **Encryption Key**, select **Fetch key from remote machine** .
   b. Enter the hostname for the primary database-only appliance you previously configured containing the encryption key.
   c. Enter the primary database-only appliance's username.

d. Enter the primary database-only appliance's password.

e. Enter the path of the remote encryption key. (For example, `/var/www/miq/vmdb/certs/v2_key`.)

8. Configure the database:

a. Select **Create Region in External Database** , since the database is external to the appliances.



IMPORTANT

Creating a database region will destroy any existing data and cannot be undone.

b. Provide the primary database-only appliance as the remote key location and its credentials.

c. Assign a unique database region number. Note that creating a database region will destroy any existing data and cannot be undone.

d. For **Are you sure you want to continue?** Select `y`.

9. Enter the primary database-only appliance's name and access details:

a. Enter the hostname for the primary database-only appliance.

b. Enter a name to identify the database.

c. Enter the primary database-only appliance's username.

d. Enter a password for the database and confirm the password.

You can check the configuration on the appliance console details screen. When configured successfully, **CFME Server** will show as `running`, and **CFME Database** will show the name of the primary database-only appliance.

2.3. CONFIGURING THE PRIMARY DATABASE-ONLY APPLIANCE

On the primary database-only appliance, initialize the nodes in the database cluster to configure the database replication:

1. In the appliance console menu, select **Configure Database Replication** .

2. Select **Configure Server as Primary** .

3. Set an unique identifier number for the server and enter the database name and credentials:

a. Select a number to uniquely identify the node in the replication cluster.

b. Enter the cluster database name.

c. Enter the cluster database username.

d. Enter the cluster database password and confirm the password.

e. Enter the primary database-only appliance hostname or IP address.

NOTE

The hostname must be visible to all appliances that communicate with this database, including the CloudForms appliances and any global region databases.

f. Confirm that the replication server configuration details are correct, and select **y** to apply the configuration.

2.4. INSTALLING THE STANDBY DATABASE-ONLY APPLIANCE

The standby database-only appliance is a copy of the primary database-only appliance and takes over the role of primary database in case of failure.

1. Deploy a CloudForms appliance with an extra partition for the database that is the same size as the primary database-only appliance, as it will contain the same data. For recommendations on disk space, see Database Requirements in the *Deployment Planning Guide*.

2. Configure time synchronization on the appliance:

   a. Edit `/etc/chronyd.conf` with valid NTP server information.

   b. Re-synchronize time information across the appliances:

   ```
   # systemctl enable chronyd.service
   # systemctl start chronyd.service
   ```

3. SSH into the CloudForms appliance to enter the appliance console.

4. Configure the hostname by selecting **Set Hostname**.

5. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.

2.5. CONFIGURING THE STANDBY DATABASE-ONLY APPLIANCE

The steps to configure the standby database-only appliance are similar to that of the primary database-only appliance, in that they prepare the appliance to be database-only, but as the standby.

On the standby database-only appliance, configure the following:

1. In the appliance console menu, select **Configure Database Replication**.

2. Select **Configure Server as Standby**.

3. Set an unique identifier number for the standby server and enter the database name and credentials:

   a. Select a number to uniquely identify the node in the replication cluster.

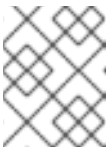
   b. Enter the cluster database name.

   c. Enter the cluster database username.

d. Enter the cluster database password.

e. Enter the primary database-only appliance hostname or IP address.

f. Enter the standby database-only appliance hostname or IP address.



NOTE

The hostname must be visible to all appliances that communicate with this database, including the engine appliances and any global region databases.

g. Select **y** to configure the replication manager for automatic failover.

h. Confirm that the replication standby server configuration details are correct, and select **y** to apply the configuration.

The standby server will then run an initial synchronization with the primary database, and start locally in standby mode.

Verify the configuration on the appliance console details screen for the standby server. When configured successfully, **Local Database Server** shows as **`running (standby)`**.

2.6. INSTALLING ADDITIONAL CLOUDFORMS APPLIANCES

Install a second virtual machine with a CloudForms appliance and any additional appliances in the region using the following steps:

1. Deploy a CloudForms appliance with an extra partition for the database at a size appropriate for your deployment. For recommendations on disk space, see Database Requirements in the *Deployment Planning Guide*.

2. Configure time synchronization on the appliance:

   a. Edit **`/etc/chronyd.conf`** with valid NTP server information.

   b. Re-synchronize time information across the appliances:

   ```
   # systemctl enable chronyd.service
   # systemctl start chronyd.service
   ```

3. SSH into the CloudForms appliance to enter the appliance console.

4. Configure the hostname by selecting **Set Hostname**.

5. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.

6. Select **Configure Database**.

7. Configure this appliance to use the encryption key from the primary database-only appliance:

   a. For **Encryption Key**, select **Fetch key from remote machine** .

   b. Enter the hostname for the primary database-only appliance you previously configured containing the encryption key.

c. Enter the primary database-only appliance's username.

d. Enter the primary database-only appliance's password.

e. Enter the path of the remote encryption key. (For example, `/var/www/miq/vmdb/certs/v2_key`.)

f. Select **Join Region in External Database** from the appliance console menu.

8. Enter the primary database-only appliance's name and access details:

a. Enter the hostname for the primary database-only appliance.

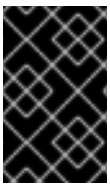
b. Enter a name to identify the database.

c. Enter the primary database-only appliance's username.

d. Enter a password for the database and confirm the password.

CHAPTER 3. CONFIGURING DATABASE FAILOVER

The failover monitor daemon must run on all of the non-database CloudForms appliances to check for failures. In case of a database failure, it modifies the database configuration accordingly.



IMPORTANT

This configuration is crucial for high availability to work in your environment. If the database failover monitor is not configured, the standby database-only appliance will not react and take over operations in case of a primary database failure.

3.1. CONFIGURING THE FAILOVER MONITOR

Configure the failover monitor only on the non-database CloudForms appliances with the following steps:

1. In the appliance console menu, select **Configure Application Database Failover Monitor**.
2. Select **Start Database Failover Monitor**.

3.2. TESTING DATABASE FAILOVER

Test that failover is working correctly between your databases with the following steps:

1. Simulate a failure by stopping the database on the primary server:

```
# systemctl stop rh-postgresql95-postgresql
```

2. To check the status of the database, run:

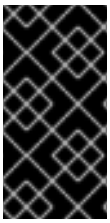
```
# systemctl status rh-postgresql95-postgresql
```

NOTE

You can check the status of the simulated failure by viewing the most recent **`ha_admin.log`** log on the engine appliances:

```
# tail -f /var/www/miq/vmdb/log/ha_admin.log
```

3. Check the appliance console summary screen for the primary database. If configured correctly, the **CFME Database** value in the appliance console summary should have switched from the hostname of the old primary database to the hostname of the new primary on all CloudForms appliances.



IMPORTANT

Upon database server failover, the standby server becomes the primary. However, the failed node cannot switch to standby automatically and must be manually configured. Data replication from the new primary to the failed and recovered node does not happen by default, so the failed node must be reintroduced into the configuration.

3.3. REINTRODUCING THE FAILED NODE

Manual steps are required to reintroduce the failed primary database node back into the cluster as a standby. This allows for greater control over the configuration, and to diagnose the failure.

To reintroduce the failed node:

1. Stop the local pgsql service:

```
# systemctl stop $APPLIANCE_PG_SERVICE
```

2. Run **pg_rewind** on the failed primary server as the **postgres** user:

```
# su - postgres
$ pg_rewind -D $APPLIANCE_PG_DATA --source-server="host=
<new_master_ip> user=root password=<db_password>
dbname=vmdb_production"
```

3. Add the following lines for standby configuration to the **/etc/repmgr.conf** file:

```
failover=automatic
promote_command='repmgr standby promote'
follow_command='repmgr standby follow'
logfile=/var/log/repmgr/repmgrd.log
```

4. Copy the **/var/lib/pgsql/.pgpass** file from the new primary server to the failed primary server. Change the file's ownership to the **postgres** user and group, and the permissions to 600:

```
# chown postgres:postgres /var/lib/pgsql/.pgpass
# chmod 600 /var/lib/pgsql/.pgpass
```

5. Run **repmgr standby follow** as the **postgres** user on the failed primary server to add it as a standby server:

```
# su - postgres
$ repmgr -f /etc/repmgr.conf -D $APPLIANCE_PG_DATA -h
<new_master_ip> -U root -d vmdb_production standby follow
```

If the **repmgr standby follow** command times out and **postgresql.log** reports a message similar to **requested WAL segment 000000020000000000000004 has already been removed**, you can correct this by removing the contents of the data directory and reinitializing the standby to re-add the node. This occurs when the write ahead log (WAL) required to catch up the standby server is no longer available on the primary.

Correct this by running the following steps:

a. Delete all database data and the replication manager configuration file from failed node:

```
# rm -rf /var/opt/rh/rh-postgresql95/lib/pgsql/data/*
# rm /etc/repmgr.conf
```

b. Delete the failed database node entry from new primary database-only appliance:

i. Check the failed node ID from the output and delete the entry; the ID is the same as the cluster ID provided during installation.

ii. Delete the node. For example, if the cluster ID is **1**, run **vmdb_production=#delete from repl_nodes where id = 1** to delete the failed node:

```
# psql vmdb_production
vmdb_production=#select * from repl_nodes;
vmdb_production=#delete from repl_nodes table where id =
$cluster_node_id_of_failed node;
```

c. Delete the replication slot information for the failed database node.

i. Check the **replication_slot** information for the failed node in the **pg_replication_slots** table:

- The failed node is marked with **f** in the **active** column in the **pg_replication_slots** table. If it is **t**, there is a standby database-only appliance connected with ID 1; this appliance must be shut down before deleting the replication slot.
- When the standby node is disconnected, the active column will be changed to **f**.

ii. Delete the slot:

- The slot number is the same as the **cluster_id**; then delete the replication slot. For example, if the cluster ID is **1**, run **vmdb_production=#select pg_drop_replication_slot('repmgr_slot_1')** to delete the slot:

```
# psql vmdb_production
vmdb_production=#select * from pg_replication_slots;
vmdb_production=#select
pg_drop_replication_slot('repmgr_slot_$failed_node_cluster_
id');
```

d. Reinitialize the standby database as described in Section 2.5, “Configuring the Standby Database-Only Appliance” to re-add the node.
For more information on the WAL log, see the PostgreSQL documentation.

6. Start and enable **repmgrd** for automatic failover:

```
# systemctl start rh-postgresql95-repmgr
# systemctl enable rh-postgresql95-repmgr
```

7. To apply the changes, stop the cluster, then restart the service as the **postgres** user:

```
# su - postgres
$ pg_ctl -D $APPLIANCE_PG_DATA stop
$ pg_ctl -D $APPLIANCE_PG_DATA status
$ exit

# systemctl start $APPLIANCE_PG_SERVICE
```

Your CloudForms environment is now re-configured for high availability.

CHAPTER 4. APPLYING UPDATES IN A HIGH AVAILABILITY ENVIRONMENT

Applying software package minor updates (referred to as *errata*) to appliances in a high availability environment must be performed in a specific order to avoid migrating your databases to the next major CloudForms version.

Prerequisites

Ensure each appliance is registered to Red Hat Subscription Manager and subscribed to the update channels required by CloudForms in order to access updates.

To verify if your appliance is registered and subscribed to the correct update channels, run:

```
# yum repolist
```

Appliances must be subscribed to the following channels:

- **cf-me-5.7-for-rhel-7-rpms**
- **rhel-7-server-rpms**
- **rhel-server-rhscl-7-rpms**

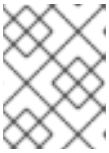
If any appliance shows it is not registered or is missing a subscription to any of these channels, see *Registering and Updating Red Hat CloudForms*in General Configuration to register and subscribe the appliance.

Updating the Appliances

Follow this procedure to update appliances in your environment without migrating the database to the next major version of CloudForms. Note the appliance to perform each step on: some steps are to be performed only on the database-only appliances, and other steps only on the CloudForms appliances, while some steps apply to all appliances.

1. Power off the CloudForms appliances.
2. Power off the database-only appliances.
3. Back up each appliance:
   a. Back up the database of your appliance. Take a snapshot if possible.
   b. Back up the following files for disaster recovery, noting which appliance each comes from:
      - **/var/www/miq/vmdb/GUID**
      - **/var/www/miq/vmdb/REGION**
   c. Note the hostnames and IP addresses of each appliance. This information is available on the summary screen of the appliance console.
4. Start each database-only appliance.
5. Start each CloudForms appliance again, and stop **evmserverd** on each just after boot:

   ```
   # systemctl stop evmserverd
   ```



NOTE

This step is not required on the database-only appliances, as **evmserverd** does not run on them.

6. Apply updates by running the following on each appliance:

```
# yum update
```

7. On one of the CloudForms (non-database) appliances, apply any database schema updates included in the errata, and reset the Red Hat and ManageIQ automation domains:

```
# vmdb
# rake db:migrate
# rake evm:automate:reset
```

8. Power off the CloudForms appliances.
9. Reboot the database-only appliances.
10. Wait five minutes, then start the CloudForms appliances again.

The appliances in your high availability environment are now up to date.