Red Hat CloudForms 5.0

High Availability Guide

Configuring and managing high availability in a Red Hat CloudForms environment

Last Updated: 2020-08-05

Red Hat CloudForms 5.0 High Availability Guide

Configuring and managing 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 configuring and managing 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

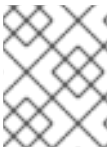
This guide describes how to configure and manage database high availability in a Red Hat CloudForms environment. This configuration allows for disaster mitigation: a failure in the primary database does not result in downtime, as the standby database takes over the failed database's processes. This is 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. This is configured from the **appliance_console** menu at the time of deployment.

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.
- *Non-database CloudForms appliances*, which are standard appliances containing a user interface and which have **evmserverd** processes enabled.

Unlike the high availability method in older versions of CloudForms which uses **pacemaker**, the built-in database high availability in CloudForms 4.2 and newer is achieved by **repmgr** database replication with PostgreSQL.

In this configuration, a failover monitor daemon is configured and running on each non-database CloudForms appliance. The failover monitor watches the **repmgr** metadata about the database-only appliances present in the cluster. When the primary database-only appliance goes down, the non-database CloudForms appliances start polling each of the configured standby database-only appliances to monitor which one comes up as the new primary. The promotion is orchestrated either by **repmgrd** on the database-only appliances or is done manually. When the non-database CloudForms appliances find that a standby has been promoted, CloudForms reconfigures the setup by writing the new IP address in the **database.yml** file to point to the new primary.



NOTE

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

Note, this procedure also does not provide scalability or a multi-master database setup. While a 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.

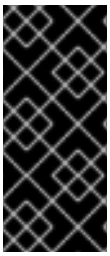
1.1. REQUIREMENTS

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 non-database 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) non-database CloudForms appliances.

2.1. INSTALLING THE PRIMARY DATABASE-ONLY APPLIANCE

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

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

NOTE

See the installation guide for your host platform (such as *Installing Red Hat CloudForms on Red Hat Virtualization*) for detailed steps on deploying an appliance with an extra disk.

2. Configure time synchronization on the appliance by editing **/etc/chrony.conf** with valid NTP server information.

3. From your host environment, open the appliance and configure the network:

    a. Log in to the appliance and run the **appliance_console** command to access the appliance console menu.

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

4. Resynchronize time information across the appliances:

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

5. In the appliance console, configure the following:

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

    b. Select **Configure Database**.

    c. 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.

    d. Select **Create Internal Database**.

    e. Select the database disk. CloudForms then activates the configuration.

    f. 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.

g. Create the database password.

NOTE

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

You have now created the empty database.

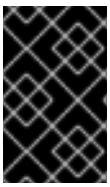
You can check the configuration on the appliance console details screen. If configured successfully, **Local Database Server** shows as **running (primary)**.

Running the **psql vmdb_production** command also provides information about the database.

2.2. INSTALLING THE FIRST CLOUDFORMS APPLIANCE

Install and configure a CloudForms appliance to point to the primary database server. You can then create a database region and configure the primary database. This appliance does not serve as a database server.

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 populate the database with the initial data.



IMPORTANT

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. There is no requirement for an extra disk on this appliance.
2. Configure time synchronization on the appliance by editing **/etc/chrony.conf** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
   a. Log in to the appliance and run the **appliance_console** command to access the appliance console menu.
   b. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
4. Re-synchronize time information across the appliances:

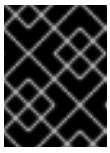
```
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

5. In the appliance console, configure the following:
    a. Configure the hostname by selecting **Set Hostname**.
    b. Select **Configure Database**.
    c. Configure this appliance to use the encryption key from the primary database-only appliance:
        i. For **Encryption Key**, select **Fetch key from remote machine**
        ii. Enter the hostname for the primary database-only appliance you previously configured containing the encryption key.
        iii. Enter the primary database-only appliance's username.
        iv. Enter the primary database-only appliance's password.
        v. Enter the path of the remote encryption key. (For example, **/var/www/miq/vmdb/certs/v2_key**.)

IMPORTANT

All appliances in the same region must use the same v2 key.

    d. Configure the database:
        i. 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.

        ii. Assign a unique database region number.
        iii. Enter the port number.
        iv. For **Are you sure you want to continue?**Select **y**.
    e. Enter the primary database-only appliance's name and access details:
        i. Enter the hostname for the primary database-only appliance.
        ii. Enter a name to identify the database.
        iii. Enter the primary database-only appliance's username.
        iv. Enter a password for the database and confirm the password.

This initializes the database, which takes a few minutes.

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 hostname of the primary database-only appliance.

2.3. CONFIGURING THE PRIMARY DATABASE-ONLY APPLIANCE

On the primary database-only appliance you created in Section 2.1, "Installing the Primary Database-Only Appliance", initialize the nodes in the database cluster to configure the database replication. Run these steps from the appliance console:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Primary**.
3. Set a 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 name of the database you configured previously.
   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 non-database CloudForms appliances and any global region databases.

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

This configures database replication in the cluster.

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.

Follow these steps to create a new standby appliance, or to add another standby appliance to the cluster.

1. Deploy a CloudForms appliance with an extra (and unpartitioned) disk 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 by editing **/etc/chrony.conf** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
   a. Log in to the appliance and run the **appliance_console** command to access the appliance

console menu.

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

4. Re-synchronize time information across the appliances:

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

5. In the appliance console, configure the hostname by selecting **Set Hostname**.

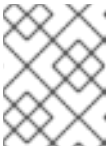
You can now configure this appliance as a standby database-only appliance in the cluster.

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 from the appliance console:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Standby**.
3. Select the database disk. CloudForms then activates the configuration.
4. Set a 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 and confirm 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. This takes a few minutes.

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. There is no requirement for an extra disk on this appliance.
2. Configure time synchronization on the appliance by editing **/etc/chrony.conf** with valid NTP server information.
3. From your host environment, open the appliance and configure the network:
    a. Log in to the appliance and run the **appliance_console** command to access the appliance console menu.
    b. Configure networking as desired by selecting the **Set DHCP Network Configuration** or **Set Static Network Configuration** option.
4. Re-synchronize time information across the appliances:

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

5. In the appliance console, configure the following:
    a. Configure the hostname by selecting **Set Hostname**.
    b. Select **Configure Database**.
    c. Configure this appliance to use the encryption key from the primary database-only appliance:
        i. For **Encryption Key**, select **Fetch key from remote machine**
        ii. Enter the hostname for the primary database-only appliance you previously configured containing the encryption key.
        iii. Enter the port number.
        iv. Enter the primary database-only appliance's username.
        v. Enter the primary database-only appliance's password.
        vi. Enter the path of the remote encryption key. (For example, **/var/www/miq/vmdb/certs/v2_key**.)
        vii. Select **Join Region in External Database** from the appliance console menu.
    d. Enter the primary database-only appliance's name and access details:
        i. Enter the hostname for the primary database-only appliance.
        ii. Enter a name to identify the database.

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

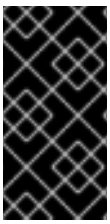
iv. Enter a password for the database and confirm the password.

This configuration takes a few minutes to process.

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 hostname of the primary database-only appliance.

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, non-database CloudForms appliances will not react to the failover event and will not be reconfigured against the new primary database host.

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 postgresql.service
```

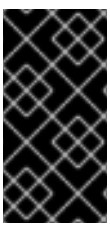
2. To check the status of the database, run:

```
# systemctl status postgresql.service
```

NOTE

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

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

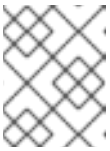
Upon database server failover, the standby server becomes the primary. However, the failed node cannot switch to standby automatically and must be manually configured. Data replication from the new primary to the failed and recovered node does not happen by default, so the failed node must be reintroduced 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, reinitialize the standby database. On the standby database-only appliance, configure the following:

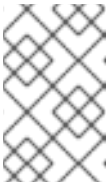
1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Standby**.
3. Select **y** to remove all previous data from the server and configure it as a new standby database.
4. Set a unique identifier number for the standby server and enter the database name and credentials:
    a. Select a number to uniquely identify the node in the replication cluster. This number can be the same as the node's original identification number.
    b. Enter the cluster database name.
    c. Enter the cluster database username.
    d. Enter the cluster database password.
    e. Enter the new primary database-only appliance hostname or IP address.
    f. Enter the new 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.



NOTE

If re-using the node's identification number, select **y** to overwrite the existing node ID (this cannot be undone). Additionally, select **y** to overwrite and reconfigure the replication settings in **/etc/repmgr.conf** when prompted.

    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)**.

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

CHAPTER 4. CONFIGURING THE HAPROXY LOAD BALANCER

After configuring the appliances as described in Chapter 2, *Installing the Appliances*, configure a load balancer to direct traffic to the non-database CloudForms appliances.

Prerequisites

An important part of configuring the HAProxy load balancer requires ensuring that links to the interface use the load balancer. To do so, you configure the URL override for each appliance in the region as follows.

1. On each appliance, open the advanced regional settings at **zone > appliance> advanced**.
2. Locate the URL entry under the **ui** component, as shown in the following example:

```
:ui:
  :custom_menu:
  :mark_translated_strings: false
  :display_ops_database: false
  :url: http://loadbalancer.example.com
```

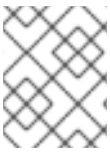
3. Replace **loadbalancer.example.com** by the fully qualified domain name for your load balancer. This domain name is then used for all links.

Procedure

The following steps highlight the configuration requirements for the load balancer, which in this case is HAProxy. The load balancer is assigned a virtual IP address for the CloudForms user interface and is pointed to one of the many non-database CloudForms appliances behind the load balancer in a round robin fashion.

Additionally, to avoid the HAProxy server being a single point of failure, two redundant HAProxy servers are configured in active-passive mode. The failover is orchestrated by using the **keepalived** daemon. The **keepalived** daemon monitors the health of the active load balancer and in case of a failure, the virtual IP is failed over to the passive load balancer, which then becomes active. The virtual IP is configured by **keepalived**.

The virtual IP is the single IP address that is used as the point of access to the CloudForms appliance user interfaces and is configured in the HAProxy configuration along with a load balancer profile. When an end user accesses the virtual IP, it directs traffic to the appropriate CloudForms appliance based on the configured HAProxy policy.



NOTE

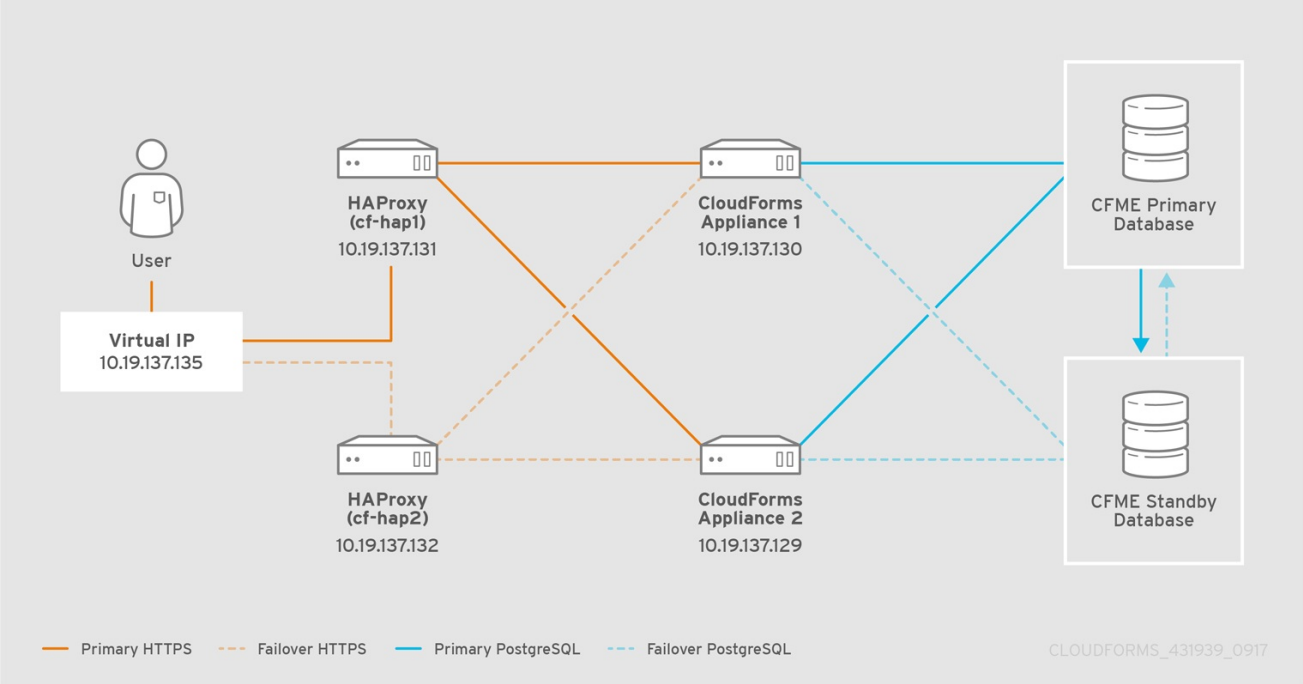
Additional configuration is required to run HAProxy on Red Hat OpenStack Platform. See the OpenStack Networking Guide for more information.

This configuration uses two HAProxy servers and a virtual IP (configured by **keepalived**). The following example procedure uses the following IP addresses and names; substitute values for your environment as needed:

- HAProxy1: 10.19.137.131 (cf-hap1.example.com)
- HAProxy2: 10.19.137.132 (cf-hap2.example.com)
- Virtual IP (to be configured by **keepalived**): 10.19.137.135 (cf-ha.example.com)

- CFME Appliance 1: 10.19.137.130 (cfme1.example.com)
- CFME Appliance 2: 10.19.137.129 (cfme2.example.com)

The following diagram shows the HAProxy configuration in this procedure:



To configure HAProxy load balancing:

1. Install two servers (virtual or physical) running Red Hat Enterprise Linux 7.2 or above, to be used as the HAProxy servers.

2. Configure subscriptions on both HAProxy servers (**cf-hap1** and **cf-hap2**) so that the **rhel-7-server-rpms** repository is enabled:

```
[root@cf-hap1 ~]# subscription-manager repos --list-enabled
+----------------------------------------+
Available Repositories in /etc/yum.repos.d/redhat.repo
+----------------------------------------+
Repo Name:   Red Hat Enterprise Linux Server 7 Server (RPMs)

Repo ID:     rhel-7-server-rpms

Repo URL: https://cdn.redhat.com/content/dist/rhel/server/7/$release/$basearch/os
Enabled: 1
```

```
[root@cf-hap2 ~]# subscription-manager repos --list-enabled
+----------------------------------------+
Available Repositories in /etc/yum.repos.d/redhat.repo
+----------------------------------------+
Repo Name:   Red Hat Enterprise Linux Server 7 Server (RPMs)

Repo ID:     rhel-7-server-rpms

Repo URL: https://cdn.redhat.com/content/dist/rhel/server/7/$release/$basearch/os
Enabled: 1
```

3. Configure the firewall on both HAProxy servers.

   a. On the **cf-hap1** server, run the following:

NOTE

keepalived uses VRRP (Virtual Redundancy Router Protocol) to monitor the servers and determine which node is the master and which node is the backup. VRRP communication between routers uses multicast IPv4 address 224.0.0.18 and IP protocol number 112.

```
[root@cf-hap1 ~]# firewall-cmd --permanent  --add-port=80/tcp --add-port=443/tcp --add-port=8443/tcp && firewall-cmd --reload

[root@cf-hap1 ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 \ --in-interface eth0 --destination 224.0.0.18 --protocol vrrp -j ACCEPT

[root@cf-hap1 ~]#firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 \ --out-interface eth0 --destination 224.0.0.18 --protocol vrrp -j ACCEPT

[root@cf-hap1 ~]# firewall-cmd --reload
```

   b. On the **cf-hap2** server, repeat the same commands by running the following:

```
[root@cf-hap2 ~]# firewall-cmd --permanent  --add-port=80/tcp --add-port=443/tcp --add-port=8443/tcp && firewall-cmd --reload

[root@cf-hap2 ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 \ --in-interface eth0 --destination 224.0.0.18 --protocol vrrp -j ACCEPT

[root@cf-hap2 ~]# firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 \ --out-interface eth0 --destination 224.0.0.18 --protocol vrrp -j ACCEPT

[root@cf-hap2 ~]# firewall-cmd --reload
```

4. Install and configure **keepalived** on both servers.

   a. On the **cf-hap1** server, run the following:

```
[root@cf-hap1 ~]# yum install keepalived -y

[root@cf-hap1 ~]# cat /etc/keepalived/keepalived.conf
vrrp_script chk_haproxy {
script "killall -0 haproxy" # check the haproxy process
interval 2 # every 2 seconds
weight 2 # add 2 points if OK
}
vrrp_instance VI_1 {
interface eth0              # interface to monitor
state MASTER              # MASTER on haproxy1, BACKUP on haproxy2
virtual_router_id 51
priority 101              # 101 on haproxy1, 100 on haproxy2
virtual_ipaddress {
10.19.137.135/21 # virtual ip address
}
```

```
track_script {
chk_haproxy
}
}
```

b. On the **cf-hap2** server, run the following:

```
[root@cf-hap2 ~]# yum install keepalived -y

[root@cf-hap2 ~]# cat /etc/keepalived/keepalived.conf
vrrp_script chk_haproxy {
script "killall -0 haproxy" # check the haproxy process
interval 2 # every 2 seconds
weight 2 # add 2 points if OK
}
vrrp_instance VI_1 {
interface eth0              # interface to monitor
state BACKUP              # MASTER on haproxy1, BACKUP on haproxy2
virtual_router_id 51
priority 100            # 101 on haproxy1, 100 on haproxy2
virtual_ipaddress {
10.19.137.135/21 # virtual ip address
}
track_script {
chk_haproxy
}
}
```

c. On both servers, configure IP forwarding and non-local binding by appending the following to the **sysctl.conf** file. In order for the **keepalived** service to forward network packets properly to the real servers, each router node must have IP forwarding turned on in the kernel. On the **cf-hap1** server, run the following:

```
[root@cf-hap1 ~]# cat /etc/sysctl.conf
# System default settings live in /usr/lib/sysctl.d/00-system.conf.
# To override those settings, enter new settings here, or in an /etc/sysctl.d/<name>.conf
file
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
```

d. On the **cf-hap2** server, run the following:

```
[root@cf-hap2 ~]# cat /etc/sysctl.conf
# System default settings live in /usr/lib/sysctl.d/00-system.conf.
# To override those settings, enter new settings here, or in an /etc/sysctl.d/<name>.conf
file
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
```

e. Verify that the **sysctl.conf** settings were saved on each server:

```
[root@cf-hap1 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
```

```
[root@cf-hap2 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
```

5. Install HAProxy on both servers:

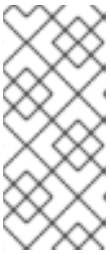
```
[root@cf-hap1 ~]# yum install haproxy -y

[root@cf-hap2 ~]# yum install haproxy -y
```

6. Configure the appropriate IPs for load balancing on the **cf-hap1** server as follows:

```
[root@cf-hap1 ~]# cat /etc/haproxy/haproxy.cfg
global
    log                 127.0.0.1 local0
    chroot              /var/lib/haproxy
    pidfile             /var/run/haproxy.pid
    maxconn             4000
    user                haproxy
    group               haproxy
    daemon
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option                  http-server-close
    option     forwardfor       except 127.0.0.0/8
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive     10s
    timeout check           10s
# CloudForms Management UI URL
listen apache
  bind 10.19.137.135:80
  mode tcp
  balance source
  server cfme1 10.19.137.130:80 check inter 1s
  server cfme2 10.19.137.129:80  check inter 1s
#
listen apache-443
  bind 10.19.137.135:443
  mode tcp
  balance source
  server cfme1 10.19.137.130:443 check inter 1s
  server cfme2 10.19.137.129:443  check inter 1s
```

```
#
listen apache-8443
  bind 10.19.137.135:8443
  mode tcp
  balance source
  server cfme1 10.19.137.130:8443 check inter 1s
  server cfme2 10.19.137.129:8443  check inter 1s
```



NOTE

- The virtual IP in this configuration is 10.19.137.135 (cf-haproxy.example.com).
- The IP of CFME Appliance 1 is 10.19.137.130 (cfme1.example.com).
- The IP of CFME Appliance 2 is 10.19.137.129 (cfme2.example.com).

7. Configure the appropriate IPs for load balancing on the **cf-hap2** server as well:

```
[root@cf-hap2 ~]# cat /etc/haproxy/haproxy.cfg
global
    log                 127.0.0.1 local0
    chroot              /var/lib/haproxy
    pidfile             /var/run/haproxy.pid
    maxconn             4000
    user                haproxy
    group               haproxy
    daemon
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option                  http-server-close
    option     forwardfor       except 127.0.0.0/8
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive     10s
    timeout check           10s
# CloudForms Management UI URL
listen apache
  bind 10.19.137.135:80
  mode tcp
  balance source
  server cfme1 10.19.137.130:80 check inter 1s
  server cfme2 10.19.137.129:80  check inter 1s
#
listen apache-443
  bind 10.19.137.135:443
  mode tcp
  balance source
```

```
    server cfme1 10.19.137.130:443 check inter 1s
    server cfme2 10.19.137.129:443  check inter 1s
#
listen apache-8443
    bind 10.19.137.135:8443
    mode tcp
    balance source
    server cfme1 10.19.137.130:8443 check inter 1s
    server cfme2 10.19.137.129:8443  check inter 1s
```

8. On each server, start the **keepalived** and **haproxy** services:

```
[root@cf-hap1~]# systemctl enable keepalived
[root@cf-hap1~]# systemctl start keepalived
[root@cf-hap1~]# systemctl enable haproxy
[root@cf-hap1~]# systemctl start haproxy
```

```
[root@cf-hap2~]# systemctl enable keepalived
[root@cf-hap2~]# systemctl start keepalived
[root@cf-hap2~]# systemctl enable haproxy
[root@cf-hap2~]# systemctl start haproxy
```

4.1. VERIFYING THE HAPROXY CONFIGURATION

Verify the HAProxy configuration by inspecting the following:

On the master node (**cf-hap1**):

```
[root@cf-hap1 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:32:4e brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.131/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 10.19.137.135/21 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:324e/64 scope global mngtmpaddr dynamic
       valid_lft 2591800sec preferred_lft 604600sec
    inet6 fe80::201:a4ff:feac:324e/64 scope link
       valid_lft forever preferred_lft forever
```

On the backup node (**cf-hap2**):

```
[root@cf-hap2 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:33:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.132/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:33a6/64 scope global noprefixroute dynamic
       valid_lft 2591982sec preferred_lft 604782sec
    inet6 fe80::201:a4ff:feac:33a6/64 scope link
       valid_lft forever preferred_lft forever
```

Notice the virtual IP 10.19.137.135 has been started by **keepalived** (VRRP).

Simulate a failure on the master node:

```
[root@cf-hap1 ~]# systemctl stop keepalived
```

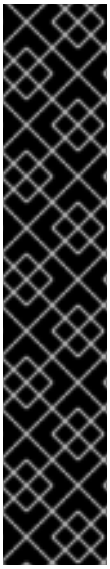
Notice the virtual IP failover on the master node (**cf-hap1**):

```
[root@cf-hap1 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:32:4e brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.131/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:324e/64 scope global mngtmpaddr dynamic
       valid_lft 2591800sec preferred_lft 604600sec
    inet6 fe80::201:a4ff:feac:324e/64 scope link
       valid_lft forever preferred_lft forever
```

The backup node (**cf-hap2**) shows the following:

```
[root@cf-hap2 ~]# ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:01:a4:ac:33:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.19.137.132/21 brd 10.19.143.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet 10.19.137.135/21 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:1388:201:a4ff:feac:33a6/64 scope global noprefixroute dynamic
       valid_lft 2591982sec preferred_lft 604782sec
    inet6 fe80::201:a4ff:feac:33a6/64 scope link
       valid_lft forever preferred_lft forever
```

Your environment is now configured for high availability.



IMPORTANT

The following additional configuration in the CloudForms user interface worker appliances and the load balancer are recommended for improved performance in worker appliances:

- For each CloudForms appliance behind the load balancer, change the **session_store** setting to **sql** in the appliance's advanced settings.
- Configure sticky sessions in the load balancer.
- Configure the load balancer to test for appliance connectivity using the **https://appliance_name/ping** URL.

See Using a Load Balancer in the *Deployment Planning Guide* for more details on these configuration steps.

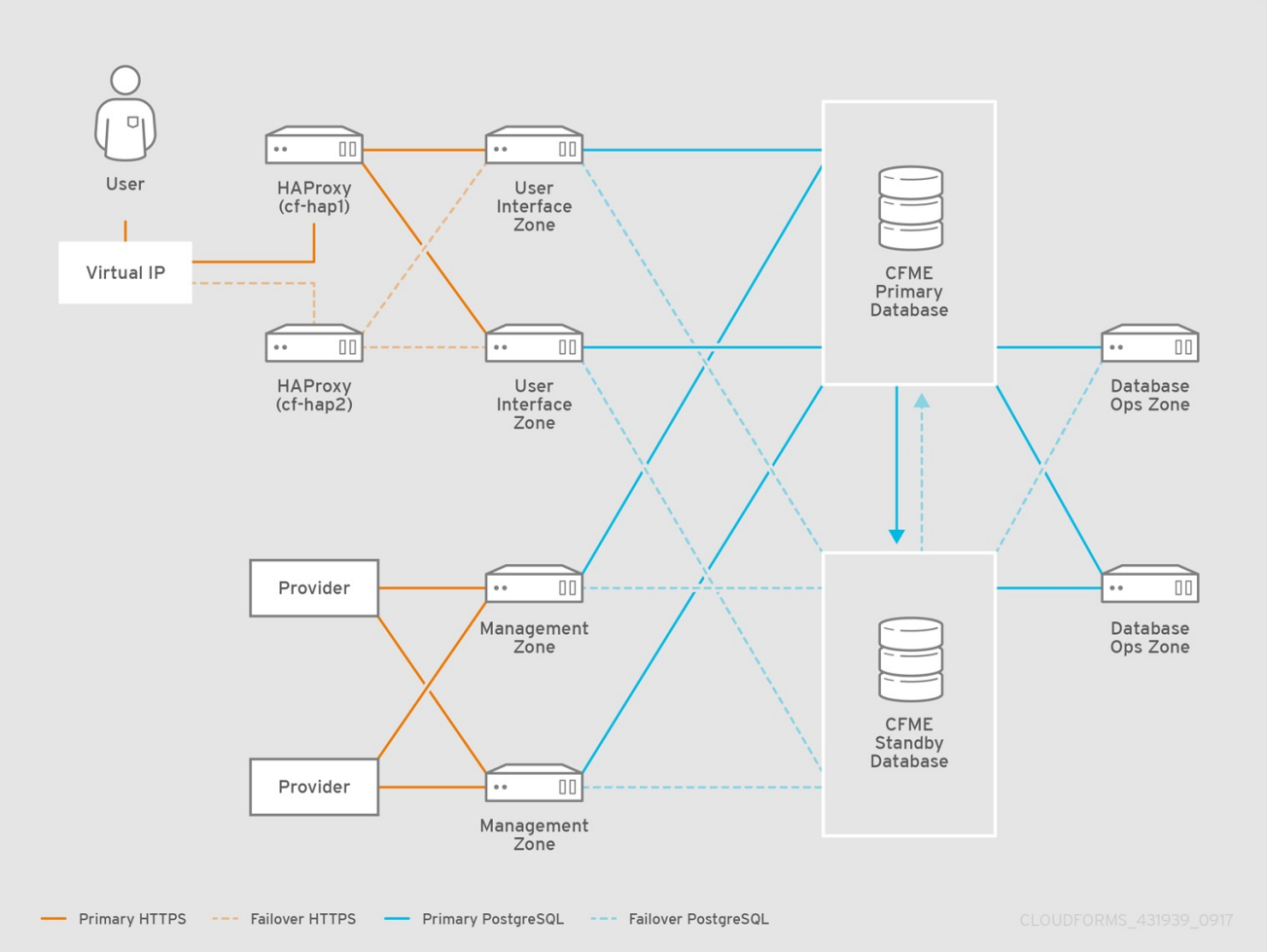
CHAPTER 5. SCALING A HIGHLY AVAILABLE CLOUDFORMS ENVIRONMENT

After creating high availability for the database tier and the user interface tier, the rest of the infrastructure should be sized appropriately for the roles and the environments that they manage. These roles and tiers use built-in high availability mechanisms like primary, secondary, and tertiary failover.

You can configure additional worker appliances as needed using the steps in Section 2.6, "Installing Additional CloudForms Appliances", and then assign zones and server roles. The non-database CloudForms appliances and roles can be configured in any order.

The following diagram shows an example of a highly available database configuration that contains worker appliances, providers, and the HAProxy load balancer configured in Chapter 4, *Configuring the HAProxy Load Balancer*.

The worker appliances in the diagram are labeled by server role (**User Interface**, **Management**, and **Database Ops**) and corresponding zone to show how a highly available environment might be scaled with server roles and zones.



See Regions and Servers in *General Configuration* for more information on configuring servers and roles.

See Deploying CloudForms at Scale for further recommendations on scaling your CloudForms environment.

CHAPTER 6. UPDATING A HIGHLY AVAILABLE CLOUDFORMS 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:

- **ansible-2.8-for-rhel-8-x86_64-rpms**
- **cfme-5.11-for-rhel-8-x86_64-rpms**
- **rhel-8-for-x86_64-appstream-rpms**
- **rhel-8-for-x86_64-baseos-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 non-database CloudForms appliances, while some steps apply to all appliances.

1. Power off the non-database 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 non-database CloudForms appliance again, and stop **evmserverd** on each just after boot:

```
# systemctl stop evmserverd
```

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

```
# yum update
```

7. On one of the non-database CloudForms 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 non-database CloudForms appliances.

9. Reboot the database-only appliances.

10. Wait five minutes, then start the non-database CloudForms appliances again.

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

CHAPTER 7. UPDATING HOSTNAMES ON DATABASE-ONLY APPLIANCES

When updating the hostnames of database-only appliances in a cluster, you must also re-configure high availability on the primary and standby database-only appliances.

7.1. PREPARING TO UPDATE APPLIANCE HOSTNAMES

Before updating the hostnames of your appliances, complete the following:

1. Note the hostnames of the active database (primary) appliance and the standby database appliances in the appliance console.
   You can verify this from the appliance summary screen of a database-only appliance, where the status for **Local Database Server** shows *primary* or *standby*, for example:

   ```
   ...
   Local Database Server:   running (primary)
   ...
   ```

2. On the non-database CloudForms appliances, stop the failover monitor:

   a. In the appliance console menu, select **Configure Application Database Failover Monitor**.

   b. Select **Stop Database Failover Monitor**.

3. On the *standby* database-only appliances, stop the **postgresql** database service:

   ```
   # systemctl stop postgresql.service
   ```

4. Stop **evmserverd** on each non-database CloudForms appliance:

   ```
   # systemctl stop evmserverd
   ```

7.2. UPDATING THE PRIMARY DATABASE-ONLY APPLIANCE HOSTNAME

Run the following steps on your primary database-only appliance:

1. In the appliance console menu, configure the new hostname or IP address from the **Configure Network** option.

2. Restart the appliance.

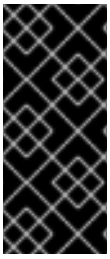
3. Update the **host** key to the primary database appliance's new hostname in **/var/www/miq/vmdb/config/database.yml**.

7.3. UPDATING THE STANDBY DATABASE-ONLY APPLIANCE HOSTNAME

Run the following steps on your standby database-only appliances:

1. In the appliance console menu, configure the new hostname or IP address from the **Configure Network** option.
2. Restart the appliance.
3. Update the **host** key to the standby database appliance's new hostname in **/var/www/miq/vmdb/config/database.yml**.

Repeat these steps on any additional standby database-only appliances.



IMPORTANT

Your primary and standby appliances must be reachable to each other by their hostnames. If all appliances are on the same network and are resolvable, no additional updates are needed. If your appliances exist on different networks, edit the **/etc/hosts** file on each database appliance to include entries for the IP address and hostname of each other database appliance in the cluster.

Proceed to re-configure high availability on the primary and standby database-only appliances.

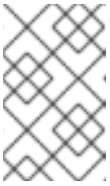
7.4. RE-CONFIGURING HIGH AVAILABILITY ON DATABASE-ONLY APPLIANCES

Re-configure replication on the database-only appliances, and restart services on your cluster.

7.4.1. Configuring the Primary Database-Only Appliance

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

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Primary**.
3. Set a 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 new primary database-only appliance hostname or IP address.



NOTE

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

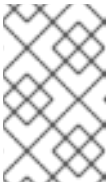
   f. Confirm that the replication server configuration details are correct, and select **y** to apply the configuration.

7.4.2. Configuring the Standby Database-Only Appliance

The steps to re-configure the standby database-only appliances 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 appliances, configure the following:

1. In the appliance console menu, select **Configure Database Replication**.
2. Select **Configure Server as Standby**.
3. Select the database disk. CloudForms then activates the configuration.
4. Set a 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 new primary database-only appliance hostname or IP address.
    f. Enter the new standby database-only appliance hostname or IP address.

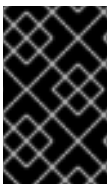


NOTE

The hostname or IP address 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.
5. Verify the configuration on the appliance console details screen for the standby server. When configured successfully, **Local Database Server** shows as **running (standby)**.

Repeat these steps on any additional standby database-only appliances.



IMPORTANT

If you are using non-dedicated database appliances, also stop **evmserverd** on those appliances before changing their hostnames, and reconfigure **database.yml** before restarting.

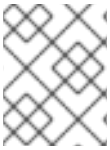
7.4.3. Restarting Services

1. Start **evmserverd** on each non-database CloudForms appliance:

```
# systemctl start evmserverd
```

After **evmserverd** has started successfully, all appliances will be able connect to the database.

2. Restart the failover monitor on the non-database CloudForms appliances:
   a. In the appliance console menu, select **Configure Application Database Failover Monitor**.
   b. Select **Start Database Failover Monitor**.



NOTE

You can view a summary of the updated appliances by running **repmgr cluster show** on one of the database appliances.

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