Red Hat CloudForms 4.7

Appliance Hardening Guide

Instructions on enhancing the security of your Red Hat CloudForms appliances

Last Updated: 2020-02-19

Red Hat CloudForms 4.7 Appliance Hardening Guide

Instructions on enhancing the security of your Red Hat CloudForms appliances

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

Legal Notice

Abstract

This guide provides a few procedures to help enhance security to your CloudForms appliances. This includes customizing passwords, restricting unnecessary communication, configuring firewall ports, and encryption key and certificate generation. 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

INTRODUCTION TO HARDENING RED HAT CLOUDFORMS

The Red Hat CloudForms appliance is distributed as a virtual machine image, which provides users with a simple installation process. The appliance also contains a set of factory default settings that can expose appliances to vulnerabilities if left unset. This guide provides a set of procedures to enhance security on your Red Hat CloudForms appliance. This ensures your appliance has enhanced protection against any unwarranted intrusion.

It is recommended to perform these steps immediately after installing all appliances in your infrastructure.

CHAPTER 1. SSH SECURITY

1.1. SETTING THE ROOT PASSWORD ON THE APPLIANCE

The Red Hat CloudForms appliance is a virtual machine image that runs on a Red Hat Enterprise Linux-based operating system. This means users can access the base operating system through SSH. This is why it is advisable to change the default password. Continuing to use the default password leaves the appliance vulnerable to any user attempting to gain root access.

Changing the **root** password on the appliance 'uses' **the** same process as changing any user password on a Linux-based system.

1. Access your appliance through SSH as the **root** user:

```
[user@localhost ~]$ ssh root@10.1.1.205
```

   Substitute **10.1.1.205** with the address of your appliance.

2. Enter the **passwd** command, which changes the password for the current user:

```
[root@ ~]# passwd
```

3. Enter and Confirm and new password for the **root** user.

```
Changing password for user root.
New password: ************
Confirm password: ************
```

4. Log out of the appliance.

The Red Hat CloudForms appliance now has a non-default **root** password. This prevents unauthorized access to your appliance through SSH.

1.2. SETTING SSH KEYS ON THE APPLIANCE

Another recommended practice is to use SSH keys to access the appliance from a single machine. An SSH key provides access from one machine to another through the SSH protocol. The following procedure shows how to create an SSH key on your local machine and add it to the appliance.

1. Check the **.ssh**/ directory in your home directory for any existing key pairs:

```
[user@localhost ~]$ ls ~/.ssh/
```

   A key pair usually consists of two files. One file is the private key, which stays on your local machine, and the other is the public key, which you copy to another machine. But files are named the same except the public key ends with a **.pub** extension.

   If a key pair already exists, you can use this key pair. Otherwise, use the next few steps to create your own.

2. On your local machine, start the key pair generation process using the **ssh-keygen** command:

```
[user@localhost ~]$ ssh-keygen -t rsa
```

3. A prompt asks for the file and location to store these keys:

```
Enter file in which to save the key (/home/user/.ssh/id_rsa):
```

   Accept the default path if you do not have a **id_rsa** key pair.

4. Another prompt asks for a passphrase:

```
Enter passphrase (empty for no passphrase):
```

   This encrypts the key pair with a password. This protects the key pair if it ever falls into the wrong hands. Alternatively, you can leave the passphrase empty, which provides an automatic login between your local machine and the remote machine.

5. The **ssh-keygen** command generates two files:

   - The private key - the default is */home/user/.ssh/id_rsa*
   - The public key - the default is */home/user/.ssh/id_rsa.pub*
     Copy the public key to the appliance using the **ssh-copy-id** command:

```
[user@localhost ~]$ ssh-copy-id ~/.ssh/id_rsa.pub root@10.1.1.205
```

     The command copies the public key to the appliance. You might receive a prompt for the password of the root user on the appliance.

6. Test the SSH key authentication:

```
[user@localhost ~]$ ssh root@10.1.1.205
```

   This authenticates using the SSH key pair. If you entered a passphrase for the key, the command prompts you for the passphrase.

7. As an additional security measure, edit the **/etc/ssh/sshd_config** on the appliance and modify the following parameter:

```
PermitRootLogin without-password
```

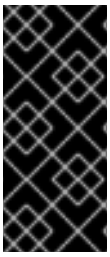
   This forces the **root** user account to use certificates instead of passwords for SSH login. This means only your local system can access the appliance.

The appliance now restricts access to only a single machine using the SSH key.

CHAPTER 2. APPLIANCE SECURITY

2.1. SETTING THE PASSWORD FOR THE ADMINISTRATIVE USER

Red Hat CloudForms uses a unique **admin** user to control all functions in the web-based user interface. After installing the appliance, change the default password of the **admin** to restrict administrative access to the appliance's UI.



IMPORTANT

Red Hat CloudForms appliances are designed for **admin** users with **root** access. Red Hat does not recommend or support CloudForms appliance configurations with users lacking **root** access.

Use default credentials (Username: admin | Password: smartvm) for the initial login.

Changing the **admin** password uses the same process as changing any standard user in the appliance.

1. Access the appliance through your web browser and log in.
2. From the settings menu, select **Configuration**.
3. In the accordion tree on the left, click on **Access Control**, then select the **Administrator** under the **Users** section. This displays the details for the **admin** user.
4. On the details page, select **Configuration → Edit this user** from the toolbar.
5. Enter a new password in the **Change Password / Confirm Password** fields.
6. Click **Save** at the bottom of the page.
7. Log out of the user interface.
8. Test your new password by logging into the user interface. Additionally, test your new password in the appliance console.

The Red Hat CloudForms appliance now has a non-default **admin** password. This restricts access to your appliance's administrative functions.

2.2. CONFIGURING HOST-BASED ACCESS CONTROL RULES ON YOUR IPA SERVER

Red Hat CloudForms provides support for external authentication using an IPA server. However, there are certain recommendations to enhance security to your appliance, such as creating a specific user group and host group that can access the appliance authentication service.

Run the following steps on your IPA server:

1. Create a user group and restrict access to only the Red Hat CloudForms users:

```
[root@ipa ~]# ipa group-add cloudforms_users --desc="cloudforms Users"
[root@ipa ~]# ipa group-add-member cloudforms_users --users=testuser1,testuser2
```

2. Create a host group and restrict access to your appliance hosts:

```
[root@ipa ~]# ipa hostgroup-add cloudforms_hosts --desc "Red Hat CloudForms hosts"
[root@ipa ~]# ipa hostgroup-add-member cloudforms_hosts --
hosts=appliance1.example.com,appliance2.example.com
```

3. Add rules to allow the host group and user group access to the Red Hat CloudForms HTTP service:

```
[root@ipa ~]# ipa hbacrule-add cloudforms_access --srchostcat=all
[root@ipa ~]# ipa hbacrule-add-service cloudforms_access --hbacsvcs httpd-auth
[root@ipa ~]# ipa hbacrule-add-user cloudforms_access --groups cloudforms_users
[root@ipa ~]# ipa hbacrule-add-host cloudforms_access --hostgroups cloudforms_hosts
```

4. Remove the default rule on your IPA server to allow access to all:

```
[root@ipa ~]# ipa hbacrule-disable allow_all
```

This ensures only users in the **cloudforms_users** group can access the authentication service (**http-auth**) on the appliances in the **cloudforms_hosts** host group.

CHAPTER 3. SERVER SECURITY

3.1. CONFIGURING FIREWALL PORTS

A new appliance starts with a few standard ports open:

- 22 for SSH communication
- 80 for HTTP access to the appliance
- 443 for HTTPS access to the appliance
- 5432 for the appliance database

You might need to restrict or open access to certain services on your appliance in the future. In such situations, use the following method:

- Use **firewalld** to enable a service or port, specifying the zone in use. For example, to open the LDAP port:

```
[root@ ~]# firewall-cmd --zone=manageiq --permanent --add-port=389/tcp
```

The following table lists the appliance's main services and their respective ports.

Table 3.1. Ports Used by Red Hat CloudForms

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
Administrator (Internet Browser)	CFME appliance (User Interface)	HTTPS	443		Access to CFME appliance User Interface
Administrator (Internet Browser)	CFME appliance (User Interface)	HTTP	80		Redirect Web Browser to HTTPS service (443)
Service Catalog or other integration through Web Service	CFME appliance (Web Service)	HTTPS	443		Access to CFME appliance Web Service
CFME appliance	NFS Server	NFS	2049	2049	Embedded NFS VM scanning

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance (User Interface)	Any Virtual Machine	TCP	903		VM Remote Console (if using MKS plug-in)
CFME appliance (User Interface)	Any Virtual Machine	TCP	5900-5999		VM Remote Console (if using VNC)
CFME appliance (any role)	CFME appliance running the VMDB	PostgreSQL Named Pipes	5432		CFME appliance connectivity to the CFME Database (PostgreSQL)
CFME Subordinate Region VMDB appliance (Database Operations)	CFME Master Region VMDB appliance	PostgreSQL Named Pipes	5432		Regional VMDB node replication up to Master VMDB node (PostgreSQL only)
CFME Subordinate Region VMDB appliance	CFME Master Region VMDB appliance (Web Services and/or User Interface)	PostgreSQL Named Pipes	5432		Subscription validation (PostgreSQL only)
CFME appliance(Authentication through LDAP)	LDAP Server (AD or other)	LDAP	389		LDAP integration
CFME appliance (Authentication through LDAPs)	LDAP Server (AD or other)	LDAPs	636		LDAPS integration
SNMP Agent	CFME appliance (Notifier)	SNMP (UDP)		161	SNMP Polling

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance (Notifier)	SNMP Server	SNMP (TCP)	162		SNMP Trap Send
CFME appliance (Notifier)	Mail server	SMTP	25		SNMP Trap Send
CFME appliance (any role)	NTP Server	NTP		123	Time Source
CFME appliance	CFME SmartProxy installed on VMware ESX Server	HTTPS	1139		Communication with SmartProxy
CFME appliance	DNS Server	UDP		53	DNS Lookups

The following tables detail the ports used by Red Hat CloudForms to communicate with providers.

Table 3.2. Red Hat Enterprise Virtualization Ports Used by Red Hat CloudForms

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance (SmartProxy)	RHEV-M Server	HTTPS	8443		API communications to RHEV-M environment (Inventory, Operations, SmartProxy)

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance (C&U)	RHEV-M Server	PostgreSQL	5432		RHEV-M History Database (Database connectivity not enabled by default). See How to access the RHEV-M Postgres DB from a remote machine.
CFME appliance	RHEV-H Hosts or RHEL Hypervisors	SSH	22		SSH connections.
CFME appliance	RHEV-H Hosts or RHEL Hypervisors	DirectLUN			Direct LUN hook must be installed and enabled for embedded VM scanning on FC or iSCSI storage devices. Not a tcp/udp connection.

Table 3.3. Red Hat OpenStack Platform Ports Used by Red Hat CloudForms

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	RHOS (Keystone)	HTTP REST API	5000		Authentication and Service Entry Point
CFME appliance	RHOS (Nova)	HTTP REST API	8774		Compute Resources
CFME appliance (C&U)	RHOS (Ceilometer)	HTTP REST API	8777		Metrics for Capacity and Utilization

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	RHOS (Glance)	HTTP REST API	9292		Authentication and Service Entry Point
CFME appliance	RHOS (AMQP)	AMQP	5672		Events Integration
CFME appliance	RHOS (Neutron)	HTTP REST API	9696		Networking
CFME appliance	RHOS (Cinder)	HTTP REST API	8776		Block Storage

Table 3.4. OpenShift Container Platform Ports Used by CloudForms Management Engine

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME Appliance	OpenShift Master Node(s) (or Load Balancer)	HTTPS	8443 or 443		Required for communication to the OpenShift API. Dependent on OpenShift configuration.
CFME Appliance	OpenShift Infrastructure Node(s) (or Load Balancer)	HTTPS	443		Metrics and logging

Table 3.5. VMware vSphere Ports Used by Red Hat CloudForms

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance(Management System Inventory, Management System Operations, C & U Data Collection, SmartProxy)	vCenter	HTTPS	443		CFME appliance running any of these roles will initiate communication with vCenter on this port
CFME appliance (SmartProxy)	ESX, ESXi Host	HTTPS	443		CFME appliance
CFME appliance (SmartProxy)	ESX Hosts (if analyzing VMs through host)	SOAP over HTTPS	902		Communication from CFME appliance to hosts
CFME appliance (SmartProxy)	vCenter (if analyzing VMs through VC)	SOAP over HTTPS	902		Communication from CFME appliance to vCenters
CFME appliance(SmartProxy)	ESX Hosts (not needed for ESXi)	SSH	22		CFME appliance console access (ssh) to ESX hosts

Table 3.6. SCVMM Ports Used by Red Hat CloudForms

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	Hyper-V Host (VMM agent)	WinRM/RPC/ NetBIOS/SMB (over TCP)	80/135/139/445		Communication from CFME appliance to Host

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	Hyper-V Host (file transfer)	HTTPS (using BITS)	443		Communication from CFME appliance to Host
CFME appliance	VM Guest Agent (file transfer)	HTTPS (using BITS)	443		Communication from CFME appliance to VM Guest Agent
CFME appliance	VMware ESX 3.0/3.5 Host (file transfer)	SFTP	22		Communication from CFME appliance to ESX Host
CFME appliance	VMware ESXi Host (file transfer)	SSH/HTTPS (using BITS)	443		Communication from CFME appliance to ESX Host
CFME appliance	WSUS Server (data channel)	HTTP	80/443		Communication from CFME appliance to Server
CFME appliance	SQL Server database (remote)	TDS	1433		CFME appliance connectivity to the Database
CFME appliance	Load Balancer	Load balancer config provider	80/443		
CFME appliance	Hyper-V host in untrusted domain or perimeter network (File Transfer)	TCP	443		CFME appliance connectivity to the host
CFME appliance	Hyper-V Host (file transfer)	BITS	443		Communication from CFME appliance to Host

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	VMware Web Services	WCF	443		

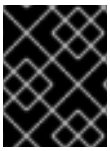
Table 3.7. Azure Ports Used by Red Hat CloudForms

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME appliance	SQL Management (*.database.windows.net)	TDS	1433		CFME appliance connectivity to the Database
CFME appliance	Upload into Storage (*.blob.core.windows.net)	HTTP/HTTPS	80/443		
CFME appliance	Service Bus Relay HTTP Mode (*.servicebus.windows.net)	SB over HTTP	80		
CFME appliance	Service Bus Pubsub over REST (*.servicebus.windows.net)	HTTPS	443		
CFME appliance	Access Control (*.accesscontrol.windows.net)	HTTPS	443		

Table 3.8. Google Compute Engine Ports Used by CloudForms Management Engine

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose

Initiator (CFME Role if applicable)	Receiver (CFME Role if applicable)	Application	TCP Port	UDP Port	Purpose
CFME Appliance	Google Cloud SDK	HTTPS	443		Communication from CFME Appliance to Google Cloud Platform resources



IMPORTANT

To provide your Red Hat CloudForms infrastructure with an extra layer of security, use a network layer firewall to restrict port access.

3.2. GENERATING SSL CERTIFICATES FOR YOUR APPLIANCE AND DATABASE

It is important to enhance the security of SSL communication of your appliances, which, depending on your setup, may include your database appliance. The appliance image ships with a default SSL certificate. It is recommended to replace this certificate with your own certificate, either signed by a trusted Certificate Authority (CA) or self-signed.

3.2.1. Creating a Certificate Signing Request

The first step is to determine the host name of your appliance or database appliance by running the following command:

```
$ hostname
```

The next step is to create a Certificate Signing Request (CSR) using the **openssl** command:

```
[root@ ~]# openssl req -new -newkey rsa:2048 -out appliance.csr -keyout appliance.key
```

This command generates a 2048-bit RSA private key and asks for a passphrase for the key.

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'appliance.key'
Enter PEM pass phrase: **********
Verifying - Enter PEM pass phrase: **********
```

The command then provides a questionnaire requesting certain details for the key. Fill out this questionnaire. Use the output of the **hostname** command to specify the **Common Name**.

For example:

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:North Carolina
Locality Name (eg, city) [Default City]:Raleigh
Organization Name (eg, company) [Default Company Ltd]:Red Hat CloudForms
Organizational Unit Name (eg, section) []:Customer Content Services
Common Name (eg, your name or your server's hostname) []:$(hostname)
Email Address []:example@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Running the command produces two files:

- **appliance.key** - The private key
- **appliance.csr** - The Certificate Signing Request (CSR)

At this stage, you would send the CSR to a trusted Certificate Authority (CA) and in return they would send you a signed certificate.

3.2.2. Creating a Self-signed Certificate

As an alternative to obtaining a signed certificate, you can use the **appliance.key** and **appliance.csr** files to create a self-signed certificate by running the following **openssl** commands:

```
[root@ ~]# openssl rsa -in appliance.key -out server.cer.key
[root@ ~]# openssl x509 -in appliance.csr -out server.cer -req -signkey server.cer.key -days 3650
```

This produces two files:

- **server.cer.key** - The private key for your signed certificate
- **server.cer** - The self-signed certificate

3.2.3. Enabling Your Certificate

Despite whether you used a trusted CA or self-signed the certificate, you should now have your own certificate for your appliance.

Copy the certificate and key files to the certificate directory on the appliance:

```
[root@ ~]# cp ~/server.cer.key /var/www/miq/vmdb/certs/server.cer.key
[root@ ~]# cp ~/server.cer /var/www/miq/vmdb/certs/server.cer
```

After the certificate and key files have been copied, restart the appliance:

```
[root@ ~]# systemctl restart evmserverd
```

The appliance now uses your own certificate.

If your environment consists of multiple appliances connecting to a single database appliance, you can use your certificate and key files to set up SSL for the database connection. For more information, see Section 4.2, “Configuring the Database to use SSL” .

IMPORTANT

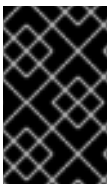
Updates from the Red Hat Content Delivery Network might overwrite these certificate and key files. Make sure to copy your own certificate and key files to the certificate directory after performing an update to your appliance.

NOTE

See also the following article for information on replacing SSL certificates in Red Hat CloudForms : https://access.redhat.com/articles/449033.

3.3. CREATING CUSTOM ENCRYPTION KEYS

To avoid storing passwords in plain text, Red Hat CloudForms appliances use an encryption key to encode and decode passwords. Each appliance stores the key in the **/var/www/miq/vmdb/certs/v2_key**. Changing the encryption key is recommended during setting up new Red Hat CloudForms appliances only.



IMPORTANT

Red Hat does not recommend changing the encryption key for an existing appliance as the ability to decrypt the password will be lost, affecting all stored passwords in Red Hat CloudForms.

To generate a new encryption key:

1. Log in to the console of your master appliance as the **root** user.
2. Run the **appliance_console** command. The Red Hat CloudForms appliance information screen appears.
3. Press any key to view the appliance menu.
4. Select **Generate Custom Encryption Key**.
5. A prompt asks if for confirmation to overwrite the existing key. Enter **Y**.
6. Enter **1** for **1) Create key**.
7. The appliance generates the new key. Press any key to complete this procedure.

This completes the procedure for generating the new key. If you have external Red Hat CloudForms appliances, you must share this key to ensure your whole Red Hat CloudForms infrastructure is using consistent encryption. Failure to use the same key results in encryption and decryption problems.

To copy an encryption key:

1. Log in to the console of an external appliance as the **root** user.
2. Run the **appliance_console** command. The Red Hat CloudForms appliance information screen appears.

3. Press any key to view the appliance menu.
4. Select **Generate Custom Encryption Key**.
5. A prompt asks if for confirmation to overwrite the existing key. Enter **Y**.
6. Select **Fetch key from remote machine**.
7. Enter the hostname or IP address of the master appliance.
8. Enter the username for SSH access to the master appliance. Use the default **root** user.
9. Enter the password for SSH access to the master appliance.
10. Enter the location of the remote key. Accept the default as **/var/www/miq/vmdb/certs/v2_key**.
11. The appliance copies the new key from the remote server. Press any key to complete this procedure.

After distributing the new key, all appliances require an update to the database configuration. For all appliances, log in as the **root** user and run the following commands replacing **dbpassword** with your database password:

```
[root@{productname_short_l} ~]# fix_auth --databaseyml --hostname localhost --password
dbpassword
[root@{productname_short_l} ~]# systemctl restart evmserverd
```

This completes the new encryption key generation for your Red Hat CloudForms infrastructure.

3.4. APPLYING SCAP STANDARDS

The Security Content Automation Protocol (SCAP) is a set of standards to assist with vulnerability management and policy compliance. Red Hat CloudForms provides a set of SCAP standards to apply to your appliance. View these SCAP rules in the **/var/www/miq/vmdb/productization/appliance_console/config/scap_rules.yml** file.

To apply the SCAP standards to your appliance's server:

1. Log in to the appliance as the **root** user.
2. Enter the **appliance_console** command. The Red Hat CloudForms Appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Harden Appliance Using SCAP Configuration**.
5. The appliance console displays the following:

   ```
   Harden Appliance Using SCAP Configuration

   Locking down the appliance for SCAP...
   ```

   The appliance applies the SCAP settings from the **scap_rules.yml** file.
6. When complete, press any key to return to the summary screen.

The appliance now meets the SCAP standards set in the **scap_rules.yml** file.

CHAPTER 4. DATABASE SECURITY

4.1. RESTRICTING HOSTS ACCESS TO THE DATABASE

Strengthening the host-based authentication (HBA) settings on a database appliance helps with preventing unauthorized access from external hosts. The HBA settings restrict access to an IP address range so that only hosts within that range have access.

Restricting access to the database requires modifications to the **/var/opt/rh/rh-postgresql95/lib/pgsql/data/pg_hba.conf** file. This file contains a text-based table with some initial settings:

```
# TYPE    DATABASE USER  ADDRESS       METHOD
local     all      all                 peer map=usermap
host      all      all   all           md5
#hostssl  all      all   all           md5
```

This format for this table uses the following header columns:

TYPE

This defines the access type, either local access from the database host (**local**), remote access from an external host regardless of encryption (**host**), external access with encryption (**hostssl**), or external access without encryption (**nohostssl**).

DATABASE

The name of the database the host can access. Use **all** for all databases.

USER

The name of the user the host can use to access the database. Use **all** for all users.

ADDRESS

The IP address of the host or address range of hosts with access to the database. This can either be:

- A single address:

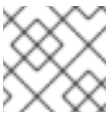
```
host    all     all   192.168.1.10          md5
```

- An address range using a CIDR mask:

```
host    all     all   192.168.1.0/24        md5
```

- An address range using a separate subnet mask value

```
host    all     all   192.168.1.0  255.255.255.0          md5
```



NOTE

ADDRESS is not required for **local** connections.

METHOD

The authentication method, which includes:

- **trust** - Allow the connection unconditionally. This method allows anyone that can connect to the PostgreSQL database server to login as any PostgreSQL user they wish, without the need for a password or any other authentication.
- **reject** - Reject the connection unconditionally. This is useful for "filtering out" certain hosts from a group, for example a **reject** line could block a specific host from connecting, while a later line allows the remaining hosts in a specific network to connect.
- **md5** - Require the client to supply an MD5-encrypted password for authentication.
- **password** - Require the client to supply an unencrypted password for authentication. Since the password is sent in clear text over the network, this should not be used on untrusted networks.
- **ident** - Obtain the operating system user name of the client by contacting the ident server on the client and check if it matches the requested database user name. Ident authentication can only be used on TCP/IP connections. When specified for local connections, peer authentication will be used instead.
- **peer** - Obtain the client's operating system user name from the operating system and check if it matches the requested database user name. This is only available for local connections.

Using a combination of these options, you create a series of rules that govern which hosts can access your database and which hosts are denied. For example, you might change the default HBA rules to only allow remote access to the Red Hat CloudForms database (**vmdb_production**) from hosts in a certain subnet. The modified HBA table would looks like this:

```
# TYPE  DATABASE        USER  ADDRESS         METHOD
local   all             all                   peer map=usermap
host    vmdb_production   all   192.168.1.0/24  md5
#hostssl all             all   all             md5
```

These restrictions help when structuring your Red Hat CloudForms appliances in relationships. For example, use these database restrictions to grant access only between a master database appliance in one region and appliances connecting from a separate region.

4.2. CONFIGURING THE DATABASE TO USE SSL

Red Hat CloudForms initially connects to the database through an unencrypted communication. If using multiple appliances connecting to a single database appliance, you can set up the database connection to use SSL. An SSL connection encrypts the communication between the CloudForms and the database.

The procedures in this section use the SSL certificate and key files listed below. These files can be found on your main CloudForms database appliance.

NOTE

The appliance image ships with a default SSL certificate and it is recommended to change this certificate. You can use a certificate signed by a trusted CA or, alternatively, generate a self-signed certificate.

See Section 3.2, "Generating SSL Certificates for Your Appliance and Database" for more information on generating an SSL certificate.

- **/var/www/miq/vmdb/certs/server.cer** - Signed or self-signed certificate for the database appliance.
- **/var/www/miq/vmdb/certs/server.cer.key** - Private key for server certificate.
- **/var/www/miq/vmdb/certs/root.crt** - The root CA certificate used to sign the CA certificate for the CloudForms database. You can either use a self-signed certificate or a certificate signed by a trusted CA to generate your root certificate.

It is also recommended to stop all CloudForms services before configuring the database to use SSL.

To configure SSL on the database appliance:

1. Log in as **root** to the appliance where the database resides.
2. Stop the **evmserverd** and **rh-postgresql95-postgresql** services:

```
[root@appliance2 ~]# systemctl stop evmserverd
[root@appliance2 ~]# systemctl stop rh-postgresql95-postgresql
```

3. Install the server key file in the correct location and set the ownership and permissions for it:

```
[root@appliance2 ~]# install -m 600 -o postgres -g postgres \
/var/www/miq/vmdb/certs/server.cer.key /var/www/miq/vmdb/certs/postgres.key
```

4. Install the server certificate file in the correct location and set the ownership and permissions for it:

```
[root@appliance2 ~]# install -m 644 -o postgres -g postgres \
/var/www/miq/vmdb/certs/server.cer /var/www/miq/vmdb/certs/postgres.crt
```

5. Install the database appliance certificate file as the root certificate in the correct location and set the ownership and permissions for it.
   If you are using a self-signed certificate, run:

```
[root@appliance2 ~]# install -m 644 -o postgres -g postgres
/var/www/miq/vmdb/certs/server.cer /var/www/miq/vmdb/certs/root.crt
```

   If you are using a third-party certificate, edit this command to install your root certificate.

6. Make sure that the security context is set correctly for the files in **/var/www/miq/certs**:

```
[root@appliance2 ~]# restorecon -R -v /var/www/miq/vmdb/certs
```

7. Open the **/var/opt/rh/rh-postgresql95/lib/pgsql/data/postgresql.conf** file and uncomment and edit the **ssl** option:

```
ssl=on
```

   In the same file, locate the options **ssl_cert_file**, **ssl_key_file**, and **ssl_ca_file** that specify the location of SSL certificates and edit them so that they are uncommented and point to the correct certificate files:

```
ssl_cert_file = '/var/www/miq/vmdb/certs/postgres.crt'  # (change requires restart)
ssl_key_file  = '/var/www/miq/vmdb/certs/postgres.key'  # (change requires restart)
ssl_ca_file   = '/var/www/miq/vmdb/certs/root.crt'      # (change requires restart)
```

8. Open the **/var/opt/rh/rh-postgresql95/lib/pgsql/data/pg_hba.conf** file and locate the two lines that contain the following:

```
host      all      all   all          md5
#hostssl  all      all   all          md5
```

Modify the two lines to comment the **host** entry and uncomment the **hostssl** entry:

```
#host     all      all   all          md5
hostssl   all      all   all          md5
```

This changes the incoming communication protocol to use SSL and refuse any unencrypted PostgreSQL connections.

9. Start the **rh-postgresql95-postgresql** and **evmserverd** services so that the changes take effect:

```
[root@cloudforms1 ~]# systemctl start rh-postgresql95-postgresql
[root@cloudforms1 ~]# systemctl start evmserverd
```

The database appliance now only accepts connections from connecting appliances using SSL. The following procedure sets up connecting appliances to communicate to the database using SSL. Use this procedure for each connecting appliance:

1. Log in as **root** to the connecting appliance.

2. Create the **.postgresql** directory in your **root** user home directory.

```
[root@cloudforms2 ~]# mkdir /root/.postgresql
```

The PostgreSQL client library, which Red Hat CloudForms also uses, looks to this directory for custom configuration files.

3. Copy the root certificate file from the database appliance to the **/root/.postgresql** directory on the connecting appliance:

```
[root@cloudforms2 ~]# scp
root@[database_appliance_fqdn]:/var/www/miq/vmdb/certs/root.crt /root/.postgresql/root.crt
```

Where **[database_appliance_fqdn]** is the fully qualified domain name of the database appliance.

4. Test the connection between the connecting appliance and the database appliance using the **psql**:

```
[root@localhost ~]# psql -h [database_appliance_fqdn] -d vmdb_production
Password: ********
psql (9.2.8)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
```

```
Type "help" for help.

vmdb_production=#
```

The **psql** displays information about the SSL connection, which indicates that the configuration succeeded. Enter **\q** to leave **psql**.

Complete this procedure for each external appliance. This enhances the security of all database transactions in your Red Hat CloudForms infrastructure.

4.2.1. Hardening TLS Protocol Version

After configuring the database to use SSL, protocol TLS version 1.2 is used as default. The older versions of this protocol (TLS 1.0 and 1.1) are still available for clients to choose. You can disable older versions by inserting the following lines into **/var/opt/rh/rh-postgresql95/lib/pgsql/data/postgresql.conf**:

```
ssl_ciphers = 'TLSv1.2:!aNULL'
ssl_prefer_server_ciphers=true
```