Red Hat CloudForms 3.1
Installing CloudForms on VMware vSphere

How to Install and Configure the CloudForms Management Engine Appliance on a VMware vSphere environment

CloudForms Documentation Team
Red Hat CloudForms 3.1 Installing CloudForms on VMware vSphere

How to Install and Configure the CloudForms Management Engine Appliance on a VMware vSphere environment

CloudForms Documentation Team
Abstract

This guide provides installation and configuration instructions for the CloudForms Management Engine Appliance. Information and procedures in this book are relevant to CloudForms Management Engine administrators.
Table of Contents

1. Introduction to Red Hat CloudForms ......................................................... 2
   1.1. Architecture .................................................................................. 2
   1.2. Requirements .............................................................................. 3
   1.3. Terminology ............................................................................... 3
   1.4. Getting Help and Giving Feedback ............................................. 7

2. Installing CloudForms .............................................................................. 7
   2.1. Obtaining the CloudForms Management Engine Appliance ........ 8
   2.2. Uploading the Appliance on VMware vSphere ............................... 8

3. Configuring CloudForms ........................................................................... 9
   3.1. Changing Configuration Settings .................................................... 9
   3.2. Configuring a Database for CloudForms Management Engine .... 10
   3.3. Configuring an Internal Database ............................................... 10
   3.4. Configuring an External Database .............................................. 11
   3.5. Configuring a Worker Appliance for CloudForms Management Engine 12

4. Additional Requirements ......................................................................... 13
   4.1. Installing VMware VDDK on CloudForms Management Engine ... 13

5. Planning .................................................................................................. 14
   5.1. Regions ...................................................................................... 14
   5.2. CPU Sizing Assistant for a Dedicated VMDB Host ..................... 15
   5.3. Database Sizing Assistant ......................................................... 17
   5.4. Example PostgreSQL Configuration File .................................... 17

6. Security .................................................................................................. 29

A. Revision History .................................................................................... 32
1. Introduction to Red Hat CloudForms

CloudForms Management Engine delivers the insight, control, and automation enterprises need to address the challenges of managing virtual environments. This technology enables enterprises with existing virtual infrastructures to improve visibility and control, and those starting virtualization deployments to build and operate a well-managed virtual infrastructure.

Red Hat CloudForms 3.1 is comprised of a single component, the CloudForms Management Engine. It has the following feature sets:

- **Insight**: Discovery, Monitoring, Utilization, Performance, Reporting, Analytics, Chargeback, and Trending.
- **Control**: Security, Compliance, Alerting, and Policy-Based Resource, and Configuration Enforcement.
- **Automate**: IT Process, Task and Event, Provisioning, and Workload Management and Orchestration.
- **Integrate**: Systems Management, Tools and Processes, Event Consoles, Configuration Management Database (CMDB), Role-based Administration (RBA), and Web Services.

1.1. Architecture

The diagram below describes the capabilities of CloudForms Management Engine. Its features are designed to work together to provide robust management and maintenance of your virtual infrastructure.

![Figure 1. Features](image)

The architecture comprises the following components:

- The CloudForms Management Engine Appliance (Appliance) which is supplied as a secure, high-performance, preconfigured virtual machine. It provides support for HTTPS communications.
The CloudForms Management Engine Server (Server) resides on the Appliance. It is the software layer that communicates between the SmartProxy and the Virtual Management Database. It includes support for HTTPS communications.

The Virtual Management Database (VMDB) resides either on the Appliance or another computer accessible to the Appliance. It is the definitive source of intelligence collected about your Virtual Infrastructure. It also holds status information regarding Appliance tasks.

The CloudForms Management Engine Console (Console) is the Web interface used to view and control the Server and Appliance. It is consumed through Web 2.0 mash-ups and web services (WS Management) interfaces.

The SmartProxy can reside on the Appliance or on an ESX Server. If not embedded in the Server, the SmartProxy can be deployed from the Appliance. Each storage location must have a SmartProxy with visibility to it. The SmartProxy acts on behalf of the Appliance communicating with it over HTTPS on standard port 443.

### 1.2. Requirements

To use CloudForms Management Engine, the following requirements must be met:

- One of the following web browsers:
  - Mozilla Firefox for versions supported under Mozilla’s Extended Support Release (ESR) [1]
  - Internet Explorer 8 or higher
  - Google Chrome for Business
- A monitor with minimum resolution of 1280x1024.
- Adobe Flash Player 9 or above. At the time of publication, you can access it at [http://www.adobe.com/products/flashplayer/](http://www.adobe.com/products/flashplayer/).
- The CloudForms Management Engine Appliance must already be installed and activated in your enterprise environment.
- The SmartProxy must have visibility to the virtual machines and cloud instances that you want to control.
- The resources that you want to control must have a SmartProxy associated with them.

**Important**

Due to browser limitations, Red Hat supports logging in to only one tab for each multi-tabbed browser. Console settings are saved for the active tab only. For the same reason, CloudForms Management Engine does not guarantee that the browser’s Back button will produce the desired results. Red Hat recommends using the breadcrumbs provided in the Console.

### 1.3. Terminology

The following terms are used throughout this document. Review them before proceeding.

**Account Role**
A designation assigned to a user allowing or restricting a user to parts and functions of the CloudForms Management Engine console.

Action

An execution that is performed after a condition is evaluated.

Alert

CloudForms Management Engine alerts notify administrators and monitoring systems of critical configuration changes and threshold limits in the virtual environment. The notification can take the form of either an email or an SNMP trap.

Analysis Profile

A customized scan of hosts, virtual machines, or instances. You can collect information from categories, files, event logs, and registry entries.

Cloud

A pool of on-demand and highly available computing resources. The usage of these resources are scaled depending on the user requirements and metered for cost.

CloudForms Management Engine Appliance

A virtual machine on which the virtual management database (VMDB) and CloudForms Management Engine server reside.

CloudForms Management Engine Console

A web-based interface into the CloudForms Management Engine Appliance.

CloudForms Management Engine Role

A designation assigned to a CloudForms Management Engine server that defines what a CloudForms Management Engine server can do.

CloudForms Management Engine Server

The application that runs on the CloudForms Management Engine Appliance and communicates with the SmartProxy and the VMDB.

Cluster

Hosts that are grouped together to provide high availability and load balancing.

Condition

A test of criteria triggered by an event.

Discovery

Process run by the CloudForms Management Engine server which finds virtual machine and cloud providers.

Drift

The comparison of a virtual machine, instance, host, cluster to itself at different points in time.

Event
A trigger to check a condition.

**Event Monitor**

Software on the CloudForms Management Engine Appliance which monitors external providers for events and sends them to the CloudForms Management Engine server.

**Host**

A computer on which virtual machine monitor software is loaded.

**Instance/Cloud Instance**

A on-demand virtual machine based upon a predefined image and uses a scalable set of hardware resources such as CPU, memory, networking interfaces.

**Managed/Registered VM**

A virtual machine that is connected to a host and exists in the VMDB. Also, a template that is connected to a provider and exists in the VMDB. Note that templates cannot be connected to a host.

**Managed/Unregistered VM**

A virtual machine or template that resides on a repository or is no longer connected to a provider or host and exists in the VMDB. A virtual machine that was previously considered registered may become unregistered if the virtual machine was removed from provider inventory.

**Provider**

A computer on which software is loaded which manages multiple virtual machines that reside on multiple hosts.

**Policy**

A combination of an event, a condition, and an action used to manage a virtual machine.

**Policy Profile**

A set of policies.

**Refresh**

A process run by the CloudForms Management Engine server which checks for relationships of the provider or host to other resources, such as storage locations, repositories, virtual machines, or instances. It also checks the power states of those resources.

**Regions**

Regions are used to create a central database for reporting and charting. Regions are used primarily to consolidate multiple VMDBs into one master VMDB for reporting.

**Resource**

A host, provider, instance, virtual machine, repository, or datastore.

**Resource Pool**

A group of virtual machines across which CPU and memory resources are allocated.
Repository

A place on a datastore resource which contains virtual machines.

SmartProxy

The SmartProxy is a software agent that acts on behalf of the CloudForms Management Engine Appliance to perform actions on hosts, providers, storage and virtual machines.

The SmartProxy can be configured to reside on the CloudForms Management Engine Appliance or on an ESX server version. The SmartProxy can be deployed from the CloudForms Management Engine Appliance, and provides visibility to the VMFS storage. Each storage location must have a SmartProxy with visibility to it. The SmartProxy acts on behalf of the CloudForms Management Engine Appliance. If the SmartProxy is not embedded in the CloudForms Management Engine server, it communicates with the CloudForms Management Engine Appliance over HTTPS on standard port 443.

SmartState Analysis

Process run by the SmartProxy which collects the details of a virtual machine or instance. Such details include accounts, drivers, network information, hardware, and security patches. This process is also run by the CloudForms Management Engine server on hosts and clusters. The data is stored in the VMDB.

SmartTags

Descriptors that allow you to create a customized, searchable index for the resources in your clouds and infrastructure.

Storage Location

A device, such as a VMware datastore, where digital information resides that is connected to a resource.

Tags

Descriptive terms defined by a CloudForms Management Engine user or the system used to categorize a resource.

Template

A template is a copy of a preconfigured virtual machine, designed to capture installed software and software configurations, as well as the hardware configuration, of the original virtual machine.

Unmanaged Virtual Machine

Files discovered on a datastore that do not have a virtual machine associated with them in the VMDB. These files may be registered to a provider that the CloudForms Management Engine server does not have configuration information on. Possible causes may be that the provider has not been discovered or that the provider has been discovered, but no security credentials have been provided.

Virtual Machine

A software implementation of a system that functions similar to a physical machine. Virtual machines utilize the hardware infrastructure of a physical host, or a set of physical hosts, to provide a scalable and on-demand method of system provisioning.

Virtual Management Database (VMDB)
Database used by the CloudForms Management Engine Appliance to store information about your resources, users, and anything else required to manage your virtual enterprise.

**Virtual Thumbnail**

An icon divided into smaller areas that summarize the properties of a resource.

**Zones**

CloudForms Management Engine Infrastructure can be organized into zones to configure failover and to isolate traffic. Zones can be created based on your environment. Zones can be based on geographic location, network location, or function. When first started, new servers are put into the default zone.

### 1.4. Getting Help and Giving Feedback

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at [http://access.redhat.com](http://access.redhat.com). Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products
- submit a support case to Red Hat Global Support Services (GSS)
- access other product documentation

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at [https://www.redhat.com/mailman/listinfo](https://www.redhat.com/mailman/listinfo). Click on the name of any mailing list to subscribe to that list or to access the list archives.

**Documentation Feedback**

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, please submit a report to GSS through the customer portal.

When submitting a report, be sure to mention the manual's identifier: *Installing CloudForms on VMware vSphere*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

### 2. Installing CloudForms

CloudForms Management Engine is able to be installed and ready to configure in a few quick steps. After downloading CloudForms Management Engine as a virtual machine image template from the Red Hat Customer Portal, the installation process takes you through the steps of uploading the appliance to a supported virtualization or cloud provider.

**Important**

After installing the CloudForms Management Engine Appliance, you must configure the database for Red Hat CloudForms. See [Section 3.2, “Configuring a Database for CloudForms Management Engine”](#).
2.1. Obtaining the CloudForms Management Engine Appliance


2. Click Downloads in the menu bar.

3. Click A-Z to sort the product downloads alphabetically.

4. Click Red Hat CloudForms to access the product download page.

5. Click the CFME VMware Virtual Appliance download link.

2.2. Uploading the Appliance on VMware vSphere

Uploading the CloudForms Management Engine Appliance file onto VMware vSphere systems has the following requirements:

- 44 GB of space on the chosen vSphere datastore.
- Administrator access to the vSphere Client.
- Depending on your infrastructure, allow time for the upload.

**Note**

These are the procedural steps as of the time of writing. For more information, consult the VMware documentation.

Use the following procedure to upload the CloudForms Management Engine Appliance OVF template from your local file system using the vSphere Client.

1. In the vSphere Client, select File → Deploy OVF Template. The Deploy OVF Template wizard appears.

2. Specify the source location and click Next.
   - Select Deploy from File to browse your file system for the OVF template, for example cfme-vsphere-5.3-47.x86_64.vsphere.ova
   - Select Deploy from URL to specify a URL to an OVF template located on the internet.

3. View the OVF Template Details page and click Next.

4. Select the deployment configuration from the drop-down menu and click Next. The option selected typically controls the memory settings, number of CPUs and reservations, and application-level configuration parameters.

5. Select the host or cluster on which you want to deploy the OVF template and click Next.

6. Select the host on which you want to run the CloudForms Management Engine appliance, and click Next.

7. Navigate to, and select the resource pool where you want to run the CloudForms Management Engine appliance and click Next.
8. Select a datastore to store the deployed CloudForms Management Engine Appliance, and click Next. Ensure to select a datastore large enough to accommodate the virtual machine and all of its virtual disk files.

9. Select the disk format to store the virtual machine virtual disks, and click Next.
   - Select **Thin Provisioned** if the storage is allocated on demand as data is written to the virtual disks.
   - Select **Thick Provisioned** if all storage is immediately allocated.

10. For each network specified in the OVF template, select a network by right-clicking the Destination Network column in your infrastructure to set up the network mapping and click Next.

11. The IP Allocation page does not require any configuration changes. Leave the default settings in the IP Allocation page and click Next.

12. Set the user-configurable properties and click Next. The properties to enter depend on the selected IP allocation scheme. For example, you are prompted for IP related information for the deployed virtual machines only in the case of a fixed IP allocation scheme.

13. Review your settings and click Finish.

**Result:**

The progress of the import task appears in the vSphere Client Status panel.

### 3. Configuring CloudForms

Although the CloudForms Management Engine Appliance comes configured to be integrated immediately into your environment, you can make some changes to its configuration.

---

**Note**

The CloudForms Management Engine Appliance is intended to have minimal configuration options.

---

### 3.1. Changing Configuration Settings

The procedure describes how to make changes to the configuration settings on the CloudForms Management Engine appliance.

**Procedure 1. To Change Settings on the CloudForms Management Engine Appliance**

1. After starting the appliance, log in with a user name of **admin** and the default password of **smartvm**. The CloudForms Management Engine Appliance summary screen displays.

2. Press Enter to manually configure settings.

3. Press the number for the item you want to change, and press Enter. The options for your selection are displayed.

4. Follow the prompts to make the changes.
3.2. Configuring a Database for CloudForms Management Engine

Before using CloudForms Management Engine, configure the database options for it. CloudForms Management Engine provides two options for database configuration:

- Install an *Internal* PostgreSQL database to the appliance
- Configure the appliance to use an *External* PostgreSQL database

**Note**

See Section 5.2, “CPU Sizing Assistant for a Dedicated VMDB Host” for CPU requirement guidelines.

3.3. Configuring an Internal Database

**Important**

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions on how to add a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance.

**Procedure 2. To Configure a CloudForms Management Engine Internal Database**

1. Start the appliance and open a terminal from your virtualization or cloud provider.

2. The login screen appears:

   To administer this appliance, browse to https://192.168.0.40/
   Username:

   Enter the administration username and password.

3. The status screen appears and displays details about the CloudForms Management Engine Appliance. Press **Enter** to change to the configuration menu.

4. Select **8) Configure Database** from the menu.

5. You are prompted to create or fetch an encryption key.
If this is the first CFME appliance, choose 1) **Create key**.

If this is not the first CFME appliance, choose 2) **Fetch key from remote machine** to fetch the key from the first CFME appliance. All CFME appliances in a multi-region deployment must use the same key.

6. Choose 1) **Internal** for the database location.

7. Choose a disk for the database. For example:

   1) /dev/vdb: 20480

   Choose disk:

   Enter 1 to choose /dev/vdb for the database location.

8. When prompted, enter a unique three digit region ID to create a new region.

   **Important**

   Creating a new region destroys any existing data on the chosen database.

9. Confirm the configuration when prompted.

   **Result:**

   CloudForms Management Engine configures the internal database.

### 3.4. Configuring an External Database

The *postgresql.conf* file used with CloudForms Management Engine databases requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. Due to these requirements, Red Hat recommends that external CloudForms Management Engine databases use a *postgresql.conf* file based on the standard file used by the CloudForms Management Engine appliance. For an example *postgresql.conf* file, see [Section 5.4, “Example PostgreSQL Configuration File”](#).

Ensure you configure the settings in the *postgresql.conf* to suit your system. For example, customize the `shared_buffers` setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances expected to connect to the PostgreSQL instance, it may be necessary to alter the `max_connections` setting.

Because the *postgresql.conf* file controls the operation of all databases managed by a single instance of PostgreSQL, do not mix CloudForms Management Engine databases with other types of databases in a single PostgreSQL instance.

**Note**

CloudForms Management Engine 3.x requires PostgreSQL version 9.2.4.
Procedure 3. To Configure a CloudForms Management Engine External Database

1. Start up the appliance and open a terminal console from your virtualization or cloud provider.

2. The login screen appears:

   ![Login Screen](image)

   To administer this appliance, browse to https://192.168.0.40/

   Username:

   Enter the administration username and password.

3. The status screen appears and displays details about the CloudForms Management Engine Appliance. Press Enter to change to the configuration menu.

4. Select 8) Configure Database from the menu.

5. You are prompted to create or fetch a security key.

   - If this is the first CFME appliance, select the option to create a key.
   - If this is not the first CFME appliance, select the option to fetch the key from the first CFME appliance. All CFME appliances in a multi-region deployment must use the same key.

6. Choose 2) External for the database location.

7. Enter the database hostname or IP address when prompted.

8. Enter the database name or leave blank for the default (vmdb_production).

9. Enter the database username or leave blank for the default (root).

10. Enter the chosen database user's password.

11. Confirm the configuration if prompted.

**Result:**

CloudForms Management Engine configures the external database.

3.5. Configuring a Worker Appliance for CloudForms Management Engine

You can configure a worker appliance through the terminal. These steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database.

Procedure 4. To Configure a Worker Appliance for CloudForms Management Engine

1. Start up the appliance and open a terminal console from your virtualization or cloud provider.

2. The login screen appears:

   ![Login Screen](image)

   To administer this appliance, browse to https://192.168.0.40/

   Username:

   Enter the administration username and password.
3. The status screen appears and displays details about the CloudForms Management Engine Appliance. Press **Enter** to change to the configuration menu.

4. Select **8) Configure Database** from the menu.

5. You are prompted to create or fetch a security key. Select the option to fetch the key from the first CFME appliance. All CFME appliances in a multi-region deployment must use the same key.

6. Choose **2) External** for the database location.

7. Enter the database hostname or IP address when prompted.

8. Enter the database name or leave blank for the default (**vmdb_production**).

9. Enter the database username or leave blank for the default (**root**).

10. Enter the chosen database user's password.

11. Confirm the configuration if prompted.

**Result:**

CloudForms Management Engine configures the external database.

### 4. Additional Requirements

#### 4.1. Installing VMware VDDK on CloudForms Management Engine

Execution of SmartState Analysis on virtual machines within a VMware environment requires the Virtual Disk Development Kit (VDDK). CloudForms Management Engine supports VDDK 5.5.

**Procedure 5. To Install the VDDK on a CloudForms Management Engine Appliance**

1. Download VDDK 5.5 (**VMware-vix-disklib-5.5.0-1284542.x86_64.tar.gz** at the time of this writing) from the VMware website.

   **Note**

   If you do not already have a login ID to VMware, then you will need to create one. At the time of this writing, the file can be found by navigating to **Downloads → All Downloads, Drivers & Tools → VMware vSphere → Drivers & Tools**. Expand **Automation Tools and SDKs**, and select **vSphere Virtual Disk Development Kit 5.5**. Alternatively, find the file by searching for it using the **Search** on the VMware site.

2. Download and copy the **VMware-vix-disklib-5.5.0-1284542.x86_64.tar.gz** file to the **/root** directory of the appliance.

3. Start an SSH session into the appliance.

4. Extract and install VDDK 5.5. using the following commands:
# cd /root
# tar -xvf VMware-vix-disklib-5.5.0-1284542.x86_64.tar.gz
# cd vmware-vix-disklib-distrib
# ./vmware-install.pl

5. Accept the defaults during the installation

Installing VMware VIX DiskLib API.
You must read and accept the VMware VIX DiskLib API End User License Agreement to continue.
Press enter to display it.
Do you accept? (yes/no) yes

Thank you.
What prefix do you want to use to install VMware VIX DiskLib API?
The prefix is the root directory where the other folders such as man, bin, doc, lib, etc. will be placed.
[/usr]

(Press Enter)

The installation of VMware VIX DiskLib API 5.5.0 build-1284542 for Linux completed successfully. You can decide to remove this software from your system at any time by invoking the following command: "/usr/bin/vmware-uninstall-vix-disklib.pl".
Enjoy,
--the VMware team

6. Run `ldconfig` in order for CloudForms Management Engine to find the newly installed VDDK library.

Note
Use the following command to verify the VDDK files are listed and accessible to the appliance:

```
# ldconfig -p | grep vix
```

7. Restart the CloudForms Management Engine Appliance.

Result:
The VDDK is now installed on the CloudForms Management Engine Appliance. This enables use of the SmartState Analysis Server Role on the appliance.

5. Planning

5.1. Regions
Use regions for centralizing data which is collected from public and private virtualization
environments. A region is ultimately represented as a single database for the VMDB. Regions are particularly useful when multiple geographical locations need to be managed as they enable all the data collection to happen at each particular location and avoids data collection traffic across slow links between networks.

When multiple regions are being used, each with their own unique ID, a master region can be created to centralize the data of all the children regions into a single master database. To do this, configure each child region to replicate its data to the master region database (Red Hat recommends use of region 99). This parent and child region is a one-to-many relationship.

Regions can contain multiple zones, which in turn contain appliances. Zones are used for further segregating network traffic along with enabling failover configurations. Each appliance has the capability to be configured for a number of specialized server roles. These roles are limited to the zone containing the appliance they run on.

Only one failover type of each server role can run in a zone. If multiple appliances have the same failover role, the extras are used as backups that activate only if the primary appliance fails. Non-failover server roles can run on multiple appliances simultaneously in a zone, so resources can be adjusted according to the workload those roles are responsible for.

**Note**

- Replicating a parent region to a higher-level parent is not supported.
- Parent regions can be configured after the child regions are online.

### 5.2. CPU Sizing Assistant for a Dedicated VMDB Host

This chart is based on empirical data gathered from a real world scenario with a dedicated external PostgreSQL database machine for VMDB. This chart can be used to predict the number of appliances that can be supported by a dedicated VMDB host. The table is intended to be used as a starting point in planning for how many CPUs (cores) will be needed based on the number of (default configured) appliances that will be accessing the VMDB on this machine.
Worker Process Per Appliance: When all of the worker types (i.e., roles) are set to their default counts, the rough maximum number of workers that will be activated at CloudForms Management Engine startup (based on initial CloudForms Management Engine 3.0 worker roles) is about 20 workers (or more accurately 20 separate Linux processes) each of which generates workload (more or less independently) on the VMDB when the system is idling.
**Database Appliance CPU Count**: The CPU count refers to the number of cores. The processors are of reasonable speed (2.5+ GHz, faster is better) and are assumed to be dedicated to processing the database requests being generated by the CloudForms Management Engine workload (appliances and workers).

**CPU consumption**: There is a cost to the host on which the PostgreSQL VMDB is hosted to adding CloudForms Management Engine appliances into a region.

- The 0.00435 CPU consumed per Postgres session was the average utilization measured when the appliance was idling. This figure is a percent, thus .435% x 20 / 2 gets us to 4.35% in the first entry.
- If CPU utilization goes over 30% when idling, it is a sign that the CPU is being over utilized and you might hit performance bottlenecks under full load.

### 5.3. Database Sizing Assistant

Red Hat recommends allocating the virtual machine disk fully at the time of creation. Three main factors affect the size of your database over time:

- **Virtual Machine Count**: the most important factor in the calculation of virtual machine database (VMDB) size over time.
- **Host Count**: the number of hosts associated with the provider.
- **Storage Count**: the number of individual storage elements as seen from the perspective of the provider or host. It is not the total number of virtual disks for all virtual machines.

<table>
<thead>
<tr>
<th>Virtual Machine Count</th>
<th>Host Count</th>
<th>Storage Count</th>
<th>Estimated VMDB Size in GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>5</td>
<td>50</td>
<td>3.5</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>100</td>
<td>17</td>
</tr>
<tr>
<td>5000</td>
<td>50</td>
<td>500</td>
<td>173</td>
</tr>
</tbody>
</table>

*Figure 3. Database Size Estimates in Gigabytes (GB)*

### 5.4. Example PostgreSQL Configuration File

The following file is an example of a `postgresql.conf` file modified for an external database. It is similar to the default file installed with the CloudForms Management Engine appliance for an internal database.

```
# -----------------------------
# PostgreSQL configuration file - MIQ Dedicated Appliance Configuration
# -----------------------------

# This file consists of lines of the form:
# 
#   name = value
#
# (The "=" is optional.) Whitespace may be used. Comments are introduced with
# "#" anywhere on a line. The complete list of parameter names and
```
allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default
values.
# Re-commenting a setting is NOT sufficient to revert it to the default
value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a
# SIGHUP
# signal. If you edit the file on a running system, you have to SIGHUP
the
# server for the changes to take effect, or use "pg_ctl reload". Some
# parameters, which are marked below, require a server shutdown and
# restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the
server, e.g.,
# "postgres -c log_connections=on". Some parameters can be changed at
run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes        Time units:  ms  = milliseconds
#        MB = megabytes                   s   = seconds
#        GB = gigabytes                   min = minutes
#        h   = hours
#        d   = days
#
#----------------------------------------------------------------------
# FILE LOCATIONS
#----------------------------------------------------------------------
#
# The default values of these variables are driven from the -D command-
line
# option or PGDATA environment variable, represented here as ConfigDir.

data_directory = 'ConfigDir'  # use data in another directory
    # (change requires restart)
hba_file = 'ConfigDir/pg_hba.conf' # host-based authentication file
    # (change requires restart)
ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
    # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is
written.
#external_pid_file = '(none)'  # write an extra PID file
    # (change requires restart)

#----------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION

# - Connection Settings -

listen_addresses = '10.132.50.128'  # MIQ Value;
#listen_addresses = 'localhost'  # what IP address(es) to listen on;
    # comma-separated list of addresses;
    # defaults to 'localhost', '*' = all
    # (change requires restart)
#port = 5432    # (change requires restart)
max_connections = 1600   # MIQ Value increased for customer
#max_connections = 100   # (change requires restart) Note: Increasing
max_connections costs ~400 bytes of shared memory per connection slot,
plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = ''  # (change requires restart)
#unix_socket_group = ''   # (change requires restart)
#unix_socket_permissions = 0777  # begin with 0 to use octal notation
    # (change requires restart)
#bonjour = off    # advertise server via Bonjour
    # (change requires restart)
#bonjour_name = ''  # defaults to the computer name
    # (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min   # 1s-600s
#ssl = off    # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
    # (change requires restart)
#ssl_renegotiation_limit = 512MB # amount of data between renegotiations
#password_encryption = on
#db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres'  # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -
# see "man 7 tcp" for details

tcp_keepalives_idle = 3   # MIQ Value;
#tcp_keepalives_idle = 0   # TCP_KEEPIDLE, in seconds;
    # 0 selects the system default
tcp_keepalives_interval = 75   # MIQ Value;
#tcp_keepalives_interval = 0   # TCPKEEPINTVL, in seconds;
    # 0 selects the system default
tcp_keepalives_count = 9      # MIQ Value;
#tcp_keepalives_count = 0     # TCP_KEEPCNT;
    # 0 selects the system default
--------
# RESOURCE USAGE (except WAL)
#-----------------------------------------------
--------

# - Memory -

#shared_buffers = 128MB  # MIQ Value SHARED CONFIGURATION
shared_buffers = 4GB   # MIQ Value DEDICATED CONFIGURATION increased for customer
#shared_buffers = 32MB  # min 128kB
  # (change requires restart)
#temp_buffers = 8MB    # min 800kB
#max_prepared_transactions = 0  # zero disables the feature
  # (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you actively intend to use prepared transactions.
#work_mem = 1MB    # min 64kB
#maintenance_work_mem = 16MB  # min 1MB
#max_stack_depth = 2MB   # min 100kB

# - Kernel Resource Usage -

#max_files_per_process = 1000  # min 25
  # (change requires restart)
#shared_preload_libraries = ''  # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0ms  # 0-100 milliseconds
#vacuum_cost_page_hit = 1  # 0-10000 credits
#vacuum_cost_page_miss = 10  # 0-10000 credits
#vacuum_cost_page_dirty = 20  # 0-10000 credits
#vacuum_cost_limit = 200  # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms  # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100  # 0-1000 max buffers written/round
#bgwriter_lru_multiplier = 2.0  # 0-10.0 multiplier on buffers scanned/round

# - Asynchronous Behavior -

#effective_io_concurrency = 1  # 1-1000. 0 disables prefetching

#-----------------------------------------------
--------
# WRITE AHEAD LOG
#-----------------------------------------------
--------
# - Settings -

#wal_level = minimal   # minimal, archive, or hot_standby
  # (change requires restart)
#fsync = on    # turns forced synchronization on or off
#synchronous_commit = on   # synchronization level; on, off, or local
#wal_sync_method = fsync   # the default is the first option
  # supported by the operating system:
  #   open_datasync
  #   fdatasync (default on Linux)
  #   fsync
  #   fsync_writethrough
  #   open_sync
#full_page_writes = on   # recover from partial page writes
wal_buffers = 16MB   # MIQ Value;
#wal_buffers = -1   # min 32kB, -1 sets based on shared_buffers
  # (change requires restart)
#wal_writer_delay = 200ms  # 1-10000 milliseconds
#commit_delay = 0   # range 0-100000, in microseconds
#commit_siblings = 5   # range 1-1000

# - Checkpoints -

checkpoint_segments = 15  # MIQ Value;
#checkpoint_segments = 3  # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 5min  # range 30s-1h
checkpoint_completion_target = 0.9 # MIQ Value;
#checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
#checkpoint_warning = 30s  # 0 disables

# - Archiving -

#archive_mode = off  # allows archiving to be done
  # (change requires restart)
#archive_command = ''  # command to use to archive a logfile segment
#archive_timeout = 0  # force a logfile segment switch after this
  # number of seconds; 0 disables

# - Master Server -

# These settings are ignored on a standby server

#max_wal_senders = 0   # max number of walsender processes
  # (change requires restart)
#wal_sender_delay = 1s  # walsender cycle time, 1-10000 milliseconds
#wal_keep_segments = 0   # in logfile segments, 16MB each; 0 disables
#vacuum_defer_cleanup_age = 0   # number of xacts by which cleanup is
#replication_timeout = 60s  # in milliseconds; 0 disables
#synchronous_standby_names = ''  # standby servers that provide sync rep
  # comma-separated list of application_name
  # from standby(s); '*' = all

# - Standby Servers -

# These settings are ignored on a master server

#hot_standby = off  # "on" allows queries during recovery
  # (change requires restart)
#max_standby_archive_delay = 30s  # max delay before canceling queries
  # when reading WAL from archive;
  # -1 allows indefinite delay
#max_standby_streaming_delay = 30s  # max delay before canceling queries
  # when reading streaming WAL;
  # -1 allows indefinite delay
#wal_receiver_status_interval = 10s  # send replies at least this often
  # 0 disables
#hot_standby_feedback = off  # send info from standby to prevent
  # query conflicts

# QUERY TUNING

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_material = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0  # measured on an arbitrary scale
#random_page_cost = 4.0  # same scale as above
#cpu_tuple_cost = 0.01  # same scale as above
#cpu_index_tuple_cost = 0.005  # same scale as above
#cpu_operator_cost = 0.0025  # same scale as above
#effective_cache_size = 128MB

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
# geqo_effort = 5   # range 1-10
# geqo_pool_size = 0   # selects default based on effort
# geqo_generations = 0   # selects default based on effort
# geqo_selection_bias = 2.0   # range 1.5-2.0
# geqo_seed = 0.0   # range 0.0-1.0

# - Other Planner Options -

default_statistics_target = 100   # range 1-10000
constraint_exclusion = partition   # on, off, or partition
cursor_tuple_fraction = 0.1   # range 0.0-1.0
from_collapse_limit = 8
join_collapse_limit = 8   # 1 disables collapsing of explicit
# JOIN clauses

-----------------------------------------------------------------------------------

# ERROR REPORTING AND LOGGING
-----------------------------------------------------------------------------------

# - Where to Log -

log_destination = 'stderr'   # Valid values are combinations of
# stderr, csvlog, syslog, and eventlog,
# depending on platform.  csvlog
# requires logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on   # Enable capturing of stderr and csvlog
# into log files. Required to be on for
# csvlogs.
# (change requires restart)

# These are only used if logging_collector is on:
log_directory = '/www/postgres/log'   # Customer specific setting
#log_directory = 'pg_log'   # directory where log files are written,
# can be absolute or relative to PGDATA
log_filename = 'postgresql-%Y-%m-%d.log'   # log file name pattern,
# can include strftime() escapes
log_file_mode = 0644   # creation mode for log files,
# begin with 0 to use octal notation
log_truncate_on_rotation = on   # If on, an existing log file
with the
# same name as the new log file will be
# truncated rather than appended to.
# But such truncation only occurs on
# time-driven rotation, not on restarts
# or size-driven rotation. Default is
# off, meaning append to existing files
# in all cases.
log_rotation_age = 1d   # Automatic rotation of logfiles
# will
# happen after that time.  0 disables.

log_rotation_size = 0                   # Automatic rotation of logfiles
      will
      # happen after that much log output.
      # 0 disables.

# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'

#silent_mode = off   # Run server silently.
      # DO NOT USE without syslog or
      # logging_collector
      # (change requires restart)

# - When to Log -

#client_min_messages = notice   # values in order of decreasing detail:
      # debug5
      # debug4
      # debug3
      # debug2
      # debug1
      # log
      # notice
      # warning
      # error

#log_min_messages = warning     # values in order of decreasing detail:
      # debug5
      # debug4
      # debug3
      # debug2
      # debug1
      # info
      # notice
      # warning
      # error
      # log
      # fatal
      # panic

#log_min_error_statement = error # values in order of decreasing detail:
      # debug5
      # debug4
      # debug3
      # debug2
      # debug1
      # info
      # notice
      # warning
      # error
      # log
      # fatal
      # panic (effectively off)
log_min_duration_statement = 5000 # MIQ Value - ANY statement > 5 seconds
#log_min_duration_statement = -1 # -1 is disabled, 0 logs all statements
  # and their durations, > 0 logs only
  # statements running at least this number
  # of milliseconds

# - What to Log -
#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_error_verbosity = default # terse, default, or verbose messages
#log_hostname = off
log_line_prefix = '%t:%r:%c:%u@%d:[%p]:' # MIQ Value;
#log_line_prefix = '' # special values:
  # %a = application name
  # %u = user name
  # %d = database name
  # %r = remote host and port
  # %h = remote host
  # %p = process ID
  # %t = timestamp without milliseconds
  # %m = timestamp with milliseconds
  # %i = command tag
  # %e = SQL state
  # %c = session ID
  # %l = session line number
  # %s = session start timestamp
  # %v = virtual transaction ID
  # %x = transaction ID (0 if none)
  # %q = stop here in non-session
    # processes
  # %% = '%'
  # e.g. '<%u%%d>'
log_lock_waits = on   # MIQ Value - used to track possible deadlock issues
#log_lock_waits = off   # log lock waits >= deadlock_timeout
#log_statement = 'none'   # none, ddl, mod, all
#log_temp_files = -1   # log temporary files equal or larger
  # than the specified size in kilobytes;
    # -1 disables, 0 logs all temp files
#log_timezone = '(defaults to server environment setting)'

#-------------------------------------------------------------
#-------------------------------------------------------------
# RUNTIME STATISTICS
#-------------------------------------------------------------
#-------------------------------------------------------------
# - Query/Index Statistics Collector -

track_activities = on
track_counts = on  # MIQ Value;
track_functions = none  # none, pl, all
track_activity_query_size = 1024  # (change requires restart)
update_process_title = on
stats_temp_directory = 'pg_stat_tmp'

# - Statistics Monitoring -

log_parser_stats = off
log_planner_stats = off
log_executor_stats = off
log_statement_stats = off

# AUTOVACUUM PARAMETERS

autovacuum = on  # MIQ Value;
autovacuum = on  # Enable autovacuum subprocess? 'on'
    # requires track_counts to also be on.
log_autovacuum_min_duration = 0  # MIQ Value;
    #log_autovacuum_min_duration = -1 # -1 disables, 0 logs all actions and
    # their durations, > 0 logs only
    # actions running at least this number
    # of milliseconds.
autovacuum_max_workers = 1  # max number of autovacuum subprocesses
    # (change requires restart)
autovacuum_naptime = 30min  # MIQ Value;
    #autovacuum_naptime = 1min  # time between autovacuum runs
autovacuum_vacuum_threshold = 500  # MIQ Value;
    #autovacuum_vacuum_threshold = 50 # min number of row updates before
    # vacuum
autovacuum_analyze_threshold = 500  # MIQ Value;
    #autovacuum_analyze_threshold = 50 # min number of row updates before
    # analyze
autovacuum_vacuum_scale_factor = 0.05  # MIQ Value;
    #autovacuum_vacuum_scale_factor = 0.2 # fraction of table size before
    # vacuum
autovacuum_analyze_scale_factor = 0.1  # fraction of table size before
    # analyze
autovacuum_freeze_max_age = 200000000  # maximum XID age before forced
    # (change requires restart)
autovacuum_vacuum_cost_delay = 20ms  # default vacuum cost delay for
    # autovacuum, in milliseconds;
    # -1 means use vacuum_cost_delay
autovacuum_vacuum_cost_limit = -1  # default vacuum cost limit for
    # autovacuum, -1 means use
# vacuum_cost_limit

# CLIENT CONNECTION DEFAULTS

# - Statement Behavior -

#search_path = '"$user",public'  # schema names
#default_tablespace = ''  # a tablespace name, '' uses the default
#temp_tablespaces = ''   # a list of tablespace names, '' uses
#    only default tablespace
#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#default_transaction_deferrable = off
#session_replication_role = 'origin'
#statement_timeout = 0   # in milliseconds, 0 is disabled
#statement_timeout = 43200000   # MIQ statment timeout of 12 hours as a
default
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#bytea_output = 'hex'   # hex, escape
#xmlbinary = 'base64'
#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, mdy'
#intervalstyle = 'postgres'
timezone = '(defaults to server environment setting)'
timezone_abbreviations = 'Default'     # Select the set of available
time zone
    # abbreviations. Currently, there are
    #   Default
    #   Australia
    #   India
    # You can create your own file in
    # share/timezonesets/.
extra_float_digits = 0   # min -15, max 3
client_encoding = sql_ascii  # actually, defaults to database
    # encoding

default_text_search_config = 'pg_catalog.english'
# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''

# LOCK MANAGEMENT

# deadlock_timeout = 5s  # MIQ Value - one second is too low, 5 seconds is more "interesting"
#deadlock_timeout = 1s
#max_locks_per_transaction = 64  # min 10
#  # (change requires restart)
# Note: Each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections +
# max_prepared_transactions)
# lock table slots.
#max_pred_locks_per_transaction = 64  # min 10
#  # (change requires restart)

# VERSION/PLATFORM COMPATIBILITY

# - Previous PostgreSQL Versions -

#array_nulls = on
#backslash_quote = safe_encoding # on, off, or safe_encoding
#default_with_oids = off
#escape_string_warning = off  # MIQ Value no sure why this is enabled
#escape_string_warning = on
#lo_compat_privileges = off
#quote_all_identifiers = off
#sql_inheritance = on
#standard_conforming_strings = off  # MIQ Value not sure why this is enabled
#standard_conforming_strings = on
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equal = off

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''
#exit_on_error = off  # terminate session on any error?
#restart_after_crash = on   # reinitialize after backend crash?

# CUSTOMIZED OPTIONS

#custom_variable_classes = ''  # list of custom variable class names

6. Security

Table 1. Ports Used by CloudForms Management Engine

<table>
<thead>
<tr>
<th>Initiator (CFME Role if applicable)</th>
<th>Receiver (CFME Role if applicable)</th>
<th>Application</th>
<th>TCP Port</th>
<th>UDP Port</th>
<th>Purpose/Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator (Internet Browser)</td>
<td>CFME Appliance (User Interface)</td>
<td>HTTPS</td>
<td>443</td>
<td></td>
<td>Access to CFME Appliance User Interface</td>
</tr>
<tr>
<td>Administrator (Internet Browser)</td>
<td>CFME Appliance (User Interface)</td>
<td>HTTP</td>
<td>80</td>
<td></td>
<td>Redirect Web Browser to HTTPS service (443)</td>
</tr>
<tr>
<td>Service Catalog or other integration through Web Service</td>
<td>CFME Appliance (Web Service)</td>
<td>HTTPS</td>
<td>443</td>
<td></td>
<td>Access to CFME Appliance Web Service</td>
</tr>
<tr>
<td>CFME Appliance</td>
<td>NFS Server</td>
<td>NFS</td>
<td>2049</td>
<td>2049</td>
<td>Embedded NFS VM scanning</td>
</tr>
<tr>
<td>CFME Appliance (User Interface)</td>
<td>Any Virtual Machine</td>
<td>TCP</td>
<td>903</td>
<td></td>
<td>VM Remote Console (if using MKS plugin)</td>
</tr>
<tr>
<td>CFME Appliance (User Interface)</td>
<td>Any Virtual Machine</td>
<td>TCP</td>
<td>5900</td>
<td></td>
<td>VM Remote Console (if using VNC)</td>
</tr>
<tr>
<td>CFME Appliance (any role)</td>
<td>CFME Appliance running the VMDB (or MS SQL)</td>
<td>PostgreSQ L Named Pipes</td>
<td>5432 (1433 MS SQL)</td>
<td></td>
<td>CFME Appliance connectivity to the CFME Database (PostgreSQL or MS SQL)</td>
</tr>
<tr>
<td>CFME Subordinate Region VMDB Appliance(Database Synchronization)</td>
<td>CFME Master Region VMDB Appliance</td>
<td>PostgreSQ L Named Pipes</td>
<td>5432 (1433 MS SQL)</td>
<td></td>
<td>Regional VMDB node replication up to Master VMDB node (PostgreSQL only)</td>
</tr>
<tr>
<td>Initiator (CFME Role if applicable)</td>
<td>Receiver (CFME Role if applicable)</td>
<td>Application</td>
<td>TCP Port</td>
<td>UDP Port</td>
<td>Purpose/Reason</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------</td>
<td>-------------</td>
<td>----------</td>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>CFME Appliance (Authentication through LDAP)</td>
<td>LDAP Server (AD or other)</td>
<td>LDAP</td>
<td>389</td>
<td></td>
<td>LDAP integration</td>
</tr>
<tr>
<td>CFME Appliance (Authentication through LDAPs)</td>
<td>LDAP Server (AD or other)</td>
<td>LDAPS</td>
<td>636</td>
<td></td>
<td>LDAPS integration</td>
</tr>
<tr>
<td>SNMP Agent</td>
<td>CFME Appliance (Notifier)</td>
<td>SNMP (UDP)</td>
<td>161</td>
<td></td>
<td>SNMP Polling</td>
</tr>
<tr>
<td>CFME Appliance (Notifier)</td>
<td>SNMP Server</td>
<td>SNMP (TCP)</td>
<td>162</td>
<td></td>
<td>SNMP Trap Send</td>
</tr>
<tr>
<td>CFME Appliance (Notifier)</td>
<td>Mail server</td>
<td>SMTP</td>
<td>25</td>
<td></td>
<td>SNMP Trap Send</td>
</tr>
<tr>
<td>CFME Appliance (any role)</td>
<td>NTP Server</td>
<td>NTP</td>
<td>123</td>
<td></td>
<td>Time Source</td>
</tr>
<tr>
<td>CFME Appliance</td>
<td>CFME SmartProxy installed on Windows or Linux</td>
<td>HTTPS</td>
<td>1139</td>
<td></td>
<td>Communication with SmartProxy</td>
</tr>
<tr>
<td>CFME SmartProxy installed on Windows or Linux</td>
<td>CFME Appliance</td>
<td>HTTPS</td>
<td>443</td>
<td></td>
<td>SmartProxy Heartbeat</td>
</tr>
<tr>
<td>CFME Appliance</td>
<td>DNS Server</td>
<td>UDP</td>
<td>53</td>
<td></td>
<td>DNS Lookups</td>
</tr>
</tbody>
</table>

**Table 2. VMware vSphere Ports Used by CloudForms Management Engine**
<table>
<thead>
<tr>
<th>Initiator (CFME Role if applicable)</th>
<th>Receiver (CFME Role if applicable)</th>
<th>Application</th>
<th>TCP Port</th>
<th>UDP Port</th>
<th>Purpose/Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFME Appliance (SmartProxy)</td>
<td>vCenter (if analyzing VMs through VC)</td>
<td>SOAP over HTTPS</td>
<td>902</td>
<td></td>
<td>Communication from CFME Appliance to vCenters.</td>
</tr>
<tr>
<td>CFME Appliance (SmartProxy)</td>
<td>ESX Hosts (not needed for ESXi)</td>
<td>SSH</td>
<td>22</td>
<td></td>
<td>CFME Appliance console access (ssh) to ESX hosts</td>
</tr>
</tbody>
</table>
### A. Revision History

<table>
<thead>
<tr>
<th>Revision 1.3.0</th>
<th>Tue Feb 3 2015</th>
<th>CloudForms Docs Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update product name to Red Hat CloudForms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision 1.2.0</th>
<th>Mon Nov 17 2014</th>
<th>CloudForms Docs Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final packages for Red Hat CloudForms 3.1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BZ#1095573 - Moved information on regions and zones to the Planning chapter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BZ#1161994 - Updated instructions for obtaining the CloudForms Management Engine appliance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BZ#1095573 - Updated information on regions and zones in the Requirements section</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision 1.1.0</th>
<th>Wed Oct 22 2014</th>
<th>CloudForms Docs Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat CloudForms 3.1 (CFME 5.3) asynchronous documentation release.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BZ#1148585 - Updated steps for adding Internal and External Databases to include creating or fetching an encryption key</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision 1.0.0</th>
<th>Thu Oct 9 2014</th>
<th>CloudForms Docs Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial book creation.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---