



## **Red Hat CloudForms 4.6**

# **Installing Red Hat CloudForms on Microsoft Azure**

How to install and configure Red Hat CloudForms on a Microsoft Azure cloud environment



# Red Hat CloudForms 4.6 Installing Red Hat CloudForms on Microsoft Azure

---

How to install and configure Red Hat CloudForms on a Microsoft Azure cloud environment

Red Hat CloudForms Documentation Team

[cloudforms-docs@redhat.com](mailto:cloudforms-docs@redhat.com)

## Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This guide provides instructions on how to install and configure Red Hat CloudForms on a Microsoft Azure cloud environment. 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

<b>CHAPTER 1. INSTALLING RED HAT CLOUDFORMS</b> .....	<b>3</b>
1.1. OBTAINING THE RED HAT CLOUDFORMS VIRTUAL APPLIANCE	3
1.2. UPLOADING AND PROVISIONING THE RED HAT CLOUDFORMS VIRTUAL APPLIANCE IN MICROSOFT AZURE	3
1.2.1. Using the Azure PowerShell Script	4
1.2.2. Using the Azure Command-Line Interface	6
1.2.2.1. Installing the Azure Command-Line Interface	6
1.2.2.2. Creating Resources for the Appliance in Microsoft Azure Using the Azure Command-Line Interface	7
1.2.2.3. Uploading and Provisioning the CloudForms Virtual Appliance Using the Azure Command-Line Interface	10
<b>CHAPTER 2. ENABLING CLOUDFORMS USER INTERFACE ACCESS</b> .....	<b>14</b>
<b>CHAPTER 3. CONFIGURING RED HAT CLOUDFORMS</b> .....	<b>15</b>
3.1. ACCESSING THE APPLIANCE CONSOLE	15
3.2. CONFIGURING A DATABASE	15
3.2.1. Configuring an Internal Database	15
3.2.2. Configuring an External Database	17
3.3. CONFIGURING A WORKER APPLIANCE	18
<b>CHAPTER 4. LOGGING IN AFTER INSTALLING RED HAT CLOUDFORMS</b> .....	<b>20</b>
4.1. CHANGING THE DEFAULT LOGIN PASSWORD	20
<b>APPENDIX A. APPENDIX</b> .....	<b>21</b>
A.1. APPLIANCE CONSOLE COMMAND-LINE INTERFACE (CLI)	21
A.2. STORAGE SKU TYPES	24



# CHAPTER 1. INSTALLING RED HAT CLOUDFORMS

Installing Red Hat CloudForms consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the CloudForms appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

## 1.1. OBTAINING THE RED HAT CLOUDFORMS VIRTUAL APPLIANCE

1. Go to [access.redhat.com](https://access.redhat.com) and log in to the Red Hat Customer Portal using your customer account details.
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. From the list of installers and images, click the **Download Now** link for **CFME Azure Virtual Appliance**.

## 1.2. UPLOADING AND PROVISIONING THE RED HAT CLOUDFORMS VIRTUAL APPLIANCE IN MICROSOFT AZURE

You can upload and provision the appliance in an Azure environment using the following two methods:

- Using the Azure PowerShell script
- Using the Azure Command-Line Interface (Azure CLI)

To upload the Red Hat CloudForms appliance file in Microsoft Azure, ensure the following requirements are met:

- Approximately 2 GB of space for each VHD image; 44+ GB of space, 12 GB RAM, and 4 VCPUs for the Red Hat CloudForms appliance.
- [Microsoft Azure Account](#).
- Administrator access to the Azure portal.
- Depending on your infrastructure, allow time for the upload.



## IMPORTANT

Azure requires that the uploaded Virtual Hard Disk (VHD) files are in a fixed format. The Azure PowerShell script and Azure CLI do not automatically convert a dynamic .vhd file to fixed during upload. For CloudForms 4.5.2 and newer, the virtual appliance image files (cfme-azure-5.8.2.3-1.x86\_64.vhd and newer) are available for download in a fixed format and therefore can be uploaded and provisioned in an Azure environment without requiring any additional manipulations with the image.

### 1.2.1. Using the Azure PowerShell Script



## NOTE

Make sure Azure Resource Manager cmdlets are available; see [Azure Resource Manager Cmdlets](#) for the latest installation information.

1. Log in to **Azure Resource Manager** using the cmdlet:

```
## Customize for Your Environment
$SubscriptionName = "my subscription"

Login-AzureRmAccount
Select-AzureRmSubscription -SubscriptionName $SubscriptionName
```

When prompted, enter your user name and password for the Azure Portal.

2. Upload the .vhd file to a storage account. As shown in the example script below, you will first create a **Resource Group** through the Portal UI or PowerShell. Additionally, create the storage container defined in "BlobDestinationContainer" in advance.

Example Script:

```
## Customize for Your Environment
$SubscriptionName = "my subscription"

$ResourceGroupName = "test"
$StorageAccountName = "test"

$BlobNameSource = "cfme-test.vhd"
$BlobSourceContainer = "templates"
$LocalImagePath = "C:\tmp\$BlobNameSource"

##

# Upload VHD to a "templates" directory. You can pass a few
arguments, such as `NumberOfUploaderThreads 8`. The default number
of uploader threads is `8`. See https://msdn.microsoft.com/en-
us/library/mt603554.aspx

Add-AzureRmVhd -ResourceGroupName $ResourceGroupName -Destination
https://$StorageAccountName.blob.core.windows.net/$BlobSourceContain
er/$BlobNameSource -LocalFilePath $LocalImagePath -
NumberOfUploaderThreads 8
```



3. Create a virtual machine. Then, define your VM and VHD name, your system/deployment name and size. Next, you will set the appropriate Storage, Network and Configuration options for your environment.

Example Script:

```
## Customize for Your Environment

$BlobNameDest = "cfme-test.vhd"
$BlobDestinationContainer = "vhds"
$VMName = "cfme-test"
$DeploySize= "Standard_A3"
$vmUserName = "user1"

$InterfaceName = "test-nic"
$VNetName = "test-vnet"
$PublicIPName = "test-public-ip"

$SSHKey = <your ssh public key>

##

$StorageAccount = Get-AzureRmStorageAccount -ResourceGroup
$ResourceGroupName -Name $StorageAccountName

$SourceImageUri =
"https://$StorageAccountName.blob.core.windows.net/templates/$BlobNameSource"
$Location = $StorageAccount.Location
$OSDiskName = $VMName

# Network
$Subnet1Name = "default"
$VNetAddressPrefix = "10.1.0.0/16"
$VNetSubnetAddressPrefix = "10.1.0.0/24"
$PIp = New-AzureRmPublicIpAddress -Name $PublicIPName -
ResourceGroupName $ResourceGroupName -Location $Location -
AllocationMethod Dynamic -Force
$SubnetConfig = New-AzureRmVirtualNetworkSubnetConfig -Name
$Subnet1Name -AddressPrefix $VNetSubnetAddressPrefix
$VNet = New-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName
$ResourceGroupName -Location $Location -AddressPrefix
$VNetAddressPrefix -Subnet $SubnetConfig -Force
$Interface = New-AzureRmNetworkInterface -Name $InterfaceName -
ResourceGroupName $ResourceGroupName -Location $Location -SubnetId
$VNet.Subnets[0].Id -PublicIpAddressId $PIp.Id -Force

# Specify the VM Name and Size
$VirtualMachine = New-AzureRmVMConfig -VMName $VMName -VMSize
$DeploySize

# Add User
$cred = Get-Credential -UserName $vmUserName -Message "Setting user
credential - use blank password"
$VirtualMachine = Set-AzureRmVMOperatingSystem -VM $VirtualMachine -
Linux -ComputerName $VMName -Credential $cred
```

```

# Add NIC
$VirtualMachine = Add-AzureRmVMNetworkInterface -VM $VirtualMachine
-Id $Interface.Id

# Add Disk
$OSDiskUri = $StorageAccount.PrimaryEndpoints.Blob.ToString() +
$BlobDestinationContainer + "/" + $BlobNameDest

$VirtualMachine = Set-AzureRmVMOSDisk -VM $VirtualMachine -Name
$OSDiskName -VhdUri $OSDiskUri -CreateOption fromImage -
SourceImageUri $SourceImageUri -Linux

# Set SSH key
Add-AzureRmVMSshPublicKey -VM $VirtualMachine -Path
"/home/$VMUserName/.ssh/authorized_keys" -KeyData $SSHKey

# Create the VM
New-AzureRmVM -ResourceGroupName $ResourceGroupName -Location
$Location -VM $VirtualMachine

```



## NOTE

These are the procedural steps as of the time of writing. For more information, see the following Azure documentation.

- <https://azure.microsoft.com/en-us/documentation/articles/powershell-azure-resource-manager>

The steps covered in the following article are for a Windows machine, however, most of the items are common between Windows and Linux.

- <https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-windows-create-powershell>

## 1.2.2. Using the Azure Command-Line Interface

Complete the following steps to upload and provision the CloudForms virtual appliance using the Azure CLI.

### 1.2.2.1. Installing the Azure Command-Line Interface



## NOTE

For a complete Azure CLI 2.0 command reference, see [Azure CLI 2.0: Command reference - az](#).

1. Import the Microsoft repository key.

```

$ sudo rpm --import
https://packages.microsoft.com/keys/microsoft.asc

```

2. Create a local Azure CLI repository entry.



```
$ sudo sh -c 'echo -e "[azure-cli]\nname=Azure
CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
cli\nenabled=1\nngpgcheck=1\nngpgkey=https://packages.microsoft.com/ke
ys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
```

### 3. Update the yum package index.

```
$ yum check-update
```

### 4. Install the Azure CLI.

```
$ sudo yum install azure-cli
```

### 5. Log in to Azure.

```
$ az login
```

Example:

To sign in, use a web browser to open the page <https://aka.ms/devicelogin> and enter the code GJP8Y33XY to authenticate.

```
[
  {
    "cloudName": "AzureCloud",
    "id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "isDefault": true,
    "name": "Demo Azure account",
    "state": "Enabled",
    "tenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "user": {
      "name": "clouduser",
      "type": "user"
    }
  }
]
```

#### 1.2.2.2. Creating Resources for the Appliance in Microsoft Azure Using the Azure Command-Line Interface

Complete the following steps to create resources in Microsoft Azure using the Azure CLI.



#### NOTE

- If you already have resources you can use, you can skip this section and go directly to [Section 1.2.2.3, “Uploading and Provisioning the CloudForms Virtual Appliance Using the Azure Command-Line Interface”](#).
- For a complete Azure CLI 2.0 command reference, see [Azure CLI 2.0: Command reference - az](#).

#### 1. Create a resource group in an Azure region.

■

```
$ az group create --name <resource-group> --location <azure-region>
```

Example:

```
[clouduser@localhost]$ az group create --name azrhelclirgrp --
location southcentralus
{
  "id": "/subscriptions//resourceGroups/azrhelclirgrp",
  "location": "southcentralus",
  "managedBy": null,
  "name": "azrhelclirgrp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

## 2. Create a storage account; see [SKU type descriptions](#).

```
$ az storage account create -l <azure-region> -n <storage-account-
name> -g <resource-group --sku <sku_type>
```

Example:

```
[clouduser@localhost]$ az storage account create -l southcentralus -
n azrhelclistact -g azrhelclirgrp --sku Standard_LRS
{
  "accessTier": null,
  "creationTime": "2017-04-05T19:10:29.855470+00:00",
  "customDomain": null,
  "encryption": null,
  "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.S
torage/storageAccounts/azrhelclistact",
  "kind": "Storage",
  "lastGeoFailoverTime": null,
  "location": "southcentralus",
  "name": "azrhelclistact",
  "primaryEndpoints": {
    "blob": "https://azrhelclistact.blob.core.windows.net/",
    "file": "https://azrhelclistact.file.core.windows.net/",
    "queue": "https://azrhelclistact.queue.core.windows.net/",
    "table": "https://azrhelclistact.table.core.windows.net/"
  },
  "primaryLocation": "southcentralus",
  "provisioningState": "Succeeded",
  "resourceGroup": "azrhelclirgrp",
  "secondaryEndpoints": null,
  "secondaryLocation": null,
  "sku": {
    "name": "Standard_LRS",
    "tier": "Standard"
  },
  "statusOfPrimary": "available",
  "statusOfSecondary": null,
```

```
"tags": {},
  "type": "Microsoft.Storage/storageAccounts"
}
```

### 3. Get the storage account connection string.

```
$ az storage account show-connection-string -n <storage-account-name> -g <resource-group>
```

Example:

```
[clouduser@localhost]$ az storage account show-connection-string -n
azrhelclistact -g azrhelclirsgrp
{
  "connectionString":
  "DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
}
```

### 4. Export the connection string. Copy the connection string and paste it into the following command. This connects your system to the storage account.

```
$ export AZURE_STORAGE_CONNECTION_STRING="<storage-connection-string>"
```

Example:

```
[clouduser@localhost]$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
```

### 5. Create the storage container.

```
$ az storage container create -n <container-name>
```

Example:

```
[clouduser@localhost]$ az storage container create -n
azrhelclistcont
{
  "created": true
}
```

### 6. Create a virtual network.

```
$ az network vnet create -g <resource group> --name <vnet-name> --
subnet-name <subnet-name>
```

Example:

```
[clouduser@localhost]$ az network vnet create --resource-group
azrhelclirsgrp --name azrhelclivnet1 --subnet-name azrhelclisubnet1
{
  "newVNet": {
```

```

    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "dhcpOptions": {
      "dnsServers": []
    },
    "etag": "W/\\"\\",
    "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.N
etwork/virtualNetworks/azrhelclivnet1",
    "location": "southcentralus",
    "name": "azrhelclivnet1",
    "provisioningState": "Succeeded",
    "resourceGroup": "azrhelclirgrp",
    "resourceGuid": "0f25efee-e2a6-4abe-a4e9-817061ee1e79",
    "subnets": [
      {
        "addressPrefix": "10.0.0.0/24",
        "etag": "W/\\"\\",
        "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.N
etwork/virtualNetworks/azrhelclivnet1/subnets/azrhelclisubnet1",
        "ipConfigurations": null,
        "name": "azrhelclisubnet1",
        "networkSecurityGroup": null,
        "provisioningState": "Succeeded",
        "resourceGroup": "azrhelclirgrp",
        "resourceNavigationLinks": null,
        "routeTable": null
      }
    ],
    "tags": {},
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": null
  }
}

```

### 1.2.2.3. Uploading and Provisioning the CloudForms Virtual Appliance Using the Azure Command-Line Interface

You can now upload and provision the appliance in an Azure environment using the Azure Command-Line Interface (Azure CLI).

1. Upload the image to the storage container. It may take several minutes. Note: Enter **az storage container list** to get the list of storage containers.

```

$ az storage blob upload --account-name <storage-account-name> --
container-name <container-name> --type page --file <path-to-vhd> --
name <image-name>.vhd

```

Example:

```

$ az storage blob upload --account-name azrhelclistact --container-

```

```
name azrhelclistcont --type page --file cfme-azure-5.8.2.3-
1.x86_64.vhd --name cfme-azure-5.8.2.3-1.x86_64.vhd
```

```
Finished[#####]
##] 100.0000%
```

2. Get the URL for the uploaded .vhd file using the following command. You will need to use this URL in the next step.

```
$ az storage blob url -c <container-name> -n <image-name>.vhd
```

Example:

```
$ az storage blob url -c azrhelclistcont -n cfme-azure-5.8.2.3-
1.x86_64.vhd
```

```
"https://azrhelclistact.blob.core.windows.net/azrhelclistcont/cfme-
azure-5.8.2.3-1.x86_64.vhd"
```

3. Create a reusable image from a blob and then use a managed disk.

Example:

```
$ az image create -n <image-name> -g <cfme-appliance-group> --os-
type <linux> --source
<https://cfmestorageaccount.blob.core.windows.net/cfmestoragecontain
er/cfme-azure-5.8.2.3-1.x86_64.vhd>
```

4. Create the virtual machine. Note that the following command uses `--generate-ssh-keys`. In this example, the private/public key pair `/home/clouduser/.ssh/id_rsa` and `/home/clouduser/.ssh/id_rsa.pub` are created.

```
$ az vm create --resource-group <resource-group> --location <azure-
region> --use-unmanaged-disk --name <vm-name> --storage-account
<storage-account-name> --os-type linux --admin-username
<administrator-name> --generate-ssh-keys --image <URL>
```

Example:

```
az vm create --resource-group azrhelclirgrp --location
southcentralus --use-unmanaged-disk --name cfme-appliance-1 --
storage-account azrhelclistact --os-type linux --admin-username
clouduser --generate-ssh-keys --image
https://azrhelclistact.blob.core.windows.net/azrhelclistcont/cfme-
azure-5.8.2.3-1.x86_64.vhd
```

```
{
  "fqdns": "",
  "id":
"/subscriptions//resourceGroups/azrhelclirgrp/providers/Microsoft.C
ompute/virtualMachines/cfme-appliance-1",
  "location": "southcentralus",
  "macAddress": "00-0X-XX-XX-XX-XX",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
```

```
"publicIpAddress": "12.84.121.147",  
"resourceGroup": "azrhelclirgrp"  
}
```

Make a note of the public IP address. You will need this to log in to the virtual machine in the next step.

5. Start an SSH session and log in to the appliance.

```
$ ssh -i <path-to-ssh-key> <admin-username@public-IP-address>
```

Example:

```
$ ssh -i /home/clouduser/.ssh/id_rsa clouduser@12.84.121.147  
The authenticity of host '12.84.121.147' can't be established.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '12.84.121.147' (ECDSA) to the list of  
known hosts.
```

```
Welcome to the Appliance Console
```

```
For a menu, please type: appliance_console
```

6. Enter **sudo appliance\_console** at the prompt. The summary screen appears.

You have successfully provisioned a CloudForms virtual appliance in Microsoft Azure.



**NOTE**

The exported storage connection string does not persist after a system reboot. If any of the commands in the above steps fail, export the storage connection string again using the following commands:

1. Get the storage account connection string.

```
$ az storage account show-connection-string -n <storage-account-name> -g <resource-group>
```

Example:

```
$ az storage account show-connection-string -n
azrhelclistact -g azrhelclirgrp
{
  "connectionString":
  "DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
}
```

2. Export the connection string. Copy the connection string and paste it into the following command. This connects your system to the storage account.

```
$ export AZURE_STORAGE_CONNECTION_STRING="<storage-connection-string>"
```

Example:

```
$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=azrhelclistact;AccountKey=NreGk...=="
```

**Possible Next Steps:**

- You will need to create a data disk for the database; see <https://docs.microsoft.com/en-us/azure/virtual-machines/linux/add-disk> for information about how to add a persistent disk to store your data.
- See [Database Requirements](#) for some general guidelines for your database requirements.
- For information about Azure ports used by CloudForms, see [https://access.redhat.com/documentation/en-us/red\\_hat\\_cloudforms/4.5/html-single/appliance\\_hardening\\_guide/#chap\\_red\\_hat\\_cloudforms\\_security\\_guide\\_firewall](https://access.redhat.com/documentation/en-us/red_hat_cloudforms/4.5/html-single/appliance_hardening_guide/#chap_red_hat_cloudforms_security_guide_firewall).

**IMPORTANT**

After uploading the Red Hat CloudForms appliance, you must configure the database for CloudForms; see [Configuring a Database for Red Hat CloudForms](#).

## CHAPTER 2. ENABLING CLOUDFORMS USER INTERFACE ACCESS

To access the Red Hat CloudForms virtual appliance user interface, you need to enable access over ports **80** and **443** to the virtual machine. You can do this using the CLI or from within the Azure portal.

- To enable a port using the CLI, enter `az vm open-port --port <port-number> --resource-group <resource-group> --name <vm-name>`.
- To enable a port using the Microsoft Azure portal, open the properties for the resource group where the appliance is located, click on **Network Security Group**, and add HTTP and HTTPS access.

## CHAPTER 3. CONFIGURING RED HAT CLOUDFORMS

After installing CloudForms and running it for the first time, you must perform some basic configuration. To configure CloudForms, you must at a minimum:

1. Add a disk to the infrastructure hosting your appliance.
2. Configure the database.

Configure the CloudForms appliance using the internal appliance console.

### 3.1. ACCESSING THE APPLIANCE CONSOLE

1. Start the appliance and open a terminal console.
2. Log in to the appliance using the SSH key.
3. Enter the `appliance_console` command. The Red Hat CloudForms appliance summary screen displays.
4. Press **Enter** to manually configure settings.
5. Press the number for the item you want to change, and press **Enter**. The options for your selection are displayed.
6. Follow the prompts to make the changes.
7. Press **Enter** to accept a setting where applicable.



#### NOTE

The CloudForms appliance console automatically logs out after five minutes of inactivity.

### 3.2. CONFIGURING A DATABASE

CloudForms uses a database to store information about the environment. Before using CloudForms, configure the database options for it; CloudForms provides the following two options for database configuration:

- Install an internal PostgreSQL database to the appliance
- Configure the appliance to use an external PostgreSQL database

#### 3.2.1. Configuring an Internal Database



#### IMPORTANT

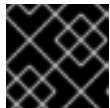
Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Red Hat CloudForms only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Log in to the appliance using the SSH key.
3. Enter the `appliance_console` command. The Red Hat CloudForms appliance summary screen displays.
4. Press **Enter** to manually configure settings.
5. Select **5) Configure Database** from the menu.
6. You are prompted to create or fetch an encryption key.
  - If this is the first Red Hat CloudForms appliance, choose **1) Create key**.
  - If this is not the first Red Hat CloudForms appliance, choose **2) Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

**NOTE**

All CloudForms appliances in a multi-region deployment must use the same key.

7. Choose **1) Create Internal Database** for the database location.
8. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

**IMPORTANT**

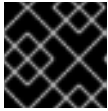
Red Hat recommends using a separate disk for the database.

If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

- ```
1) /dev/vdb: 20480
2) Don't partition the disk
```

- Enter **1** to choose `/dev/vdb` for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is `/var/opt/rh/rh-postgresql195/lib/pgsql`, which can be found in the environment variable `$APPLIANCE_PG_MOUNT_POINT`.
  - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).
9. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
    - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
    - Select **N** to configure the appliance with the full administrative user interface.

- When prompted, enter a unique number to create a new region.



### IMPORTANT

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

- Create and confirm a password for the database.

Red Hat CloudForms then configures the internal database.

### 3.2.2. Configuring an External Database

Based on your setup, you will choose to configure the appliance to use an external PostgreSQL database. For example, we can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The `postgresql.conf` file used with Red Hat CloudForms 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 Red Hat CloudForms databases use a `postgresql.conf` file based on the standard file used by the Red Hat CloudForms appliance.

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.



### NOTE

- Red Hat CloudForms 4.x requires PostgreSQL version 9.4.
- Because the `postgresql.conf` file controls the operation of all databases managed by a single instance of PostgreSQL, do not mix Red Hat CloudForms databases with other types of databases in a single PostgreSQL instance.

- Start the appliance and open a terminal console.
- Log in to the appliance using the SSH key.
- Enter the `appliance_console` command. The Red Hat CloudForms appliance summary screen displays.
- Press **Enter** to manually configure settings.
- Select **5) Configure Database** from the menu.
- You are prompted to create or fetch a security key.
  - If this is the first Red Hat CloudForms appliance, choose **1) Create key**.
  - If this is not the first Red Hat CloudForms appliance, choose **2) Fetch key from remote machine** to fetch the key from the first appliance.

**NOTE**

All CloudForms appliances in a multi-region deployment must use the same key.

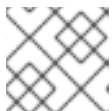
7. Choose **2) Create Region in External Database** for the database location.
8. Enter the database hostname or IP address when prompted.
9. Enter the database name or leave blank for the default (**vmdb\_production**).
10. Enter the database username or leave blank for the default (**root**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.

Red Hat CloudForms will then configure the external database.

### 3.3. CONFIGURING A WORKER APPLIANCE

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database.

1. Start the appliance and open a terminal console.
2. Log in to the appliance using the SSH key.
3. Enter the **appliance\_console** command. The Red Hat CloudForms appliance summary screen displays.
4. Press **Enter** to manually configure settings.
5. Select **5) Configure Database** from the menu.
6. You are prompted to create or fetch a security key. Since this is not the first Red Hat CloudForms appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy key from another appliance.

**NOTE**

All CloudForms appliances in a multi-region deployment must use the same key.

7. Choose **3) Join Region in External Database** for the database location.
8. Enter the database hostname or IP address when prompted.
9. Enter the port number or leave blank for the default (**5432**).
10. Enter the database name or leave blank for the default (**vmdb\_production**).
11. Enter the database username or leave blank for the default (**root**).

12. Enter the chosen database user's password.
13. Confirm the configuration if prompted.

## CHAPTER 4. LOGGING IN AFTER INSTALLING RED HAT CLOUDFORMS

Once Red Hat CloudForms is installed, you can log in and perform administration tasks.

Log in to Red Hat CloudForms for the first time after installing by:

1. Navigate to the URL for the login screen. (<https://xx.xx.xx.xx> on the virtual machine instance)
2. Enter the default credentials (Username: **admin** | Password: **smartvm**) for the initial login.
3. Click **Login**.

### 4.1. CHANGING THE DEFAULT LOGIN PASSWORD

Change your password to ensure more private and secure access to Red Hat CloudForms.

1. Navigate to the URL for the login screen. (<https://xx.xx.xx.xx> on the virtual machine instance)
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.



## APPENDIX A. APPENDIX

### A.1. APPLIANCE CONSOLE COMMAND-LINE INTERFACE (CLI)

Currently, the `appliance_console_cli` feature is a subset of the full functionality of the `appliance_console` itself, and covers functions most likely to be scripted using the command-line interface (CLI).

1. After starting the Red Hat CloudForms appliance, log in with a user name of `root` and the default password of `smartvm`. This displays the Bash prompt for the root user.
2. Enter the `appliance_console_cli` or `appliance_console_cli --help` command to see a list of options available with the command, or simply enter `appliance_console_cli --option <argument>` directly to use a specific option.

**Table A.1. Database Configuration Options**

| Option                       | Description                                                                                |
|------------------------------|--------------------------------------------------------------------------------------------|
| <code>--region (-r)</code>   | region number (create a new region in the database - requires database credentials passed) |
| <code>--internal (-i)</code> | internal database (create a database on the current appliance)                             |
| <code>--dbdisk</code>        | database disk device path (for configuring an internal database)                           |
| <code>--hostname (-h)</code> | database hostname                                                                          |
| <code>--port</code>          | database port (defaults to <b>5432</b> )                                                   |
| <code>--username (-U)</code> | database username (defaults to <b>root</b> )                                               |
| <code>--password (-p)</code> | database password                                                                          |
| <code>--dbname (-d)</code>   | database name (defaults to <b>vmdb_production</b> )                                        |

**Table A.2. v2\_key Options**

| Option                        | Description                                |
|-------------------------------|--------------------------------------------|
| <code>--key (-k)</code>       | create a new v2_key                        |
| <code>--fetch-key (-K)</code> | fetch the v2_key from the given host       |
| <code>--force-key (-f)</code> | create or fetch the key even if one exists |

| Option                     | Description                                                                        |
|----------------------------|------------------------------------------------------------------------------------|
| <code>--sshlogin</code>    | ssh username for fetching the <code>v2_key</code> (defaults to <code>root</code> ) |
| <code>--sshpassword</code> | ssh password for fetching the <code>v2_key</code>                                  |

Table A.3. IPA Server Options

| Option                            | Description                                                                                      |
|-----------------------------------|--------------------------------------------------------------------------------------------------|
| <code>--host (-H)</code>          | set the appliance hostname to the given name                                                     |
| <code>--ipaserver (-e)</code>     | IPA server FQDN                                                                                  |
| <code>--ipaprincipal (-n)</code>  | IPA server principal (default: <code>admin</code> )                                              |
| <code>--ipapassword (-w)</code>   | IPA server password                                                                              |
| <code>--ipadomain (-o)</code>     | IPA server domain (optional). Will be based on the appliance domain name if not specified.       |
| <code>--iparealm (-l)</code>      | IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified. |
| <code>--uninstall-ipa (-u)</code> | uninstall IPA client                                                                             |

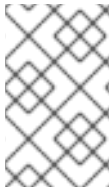
**NOTE**

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the `appliance_console`, external authentication can be optionally configured via the `appliance_console_cli` (command-line interface).
- Specifying `--host` will update the hostname of the appliance. If this step was already performed via the `appliance_console` and the necessary updates made to `/etc/hosts` if DNS is not properly configured, the `--host` option can be omitted.

Table A.4. Certificate Options

| Option                                   | Description                                              |
|------------------------------------------|----------------------------------------------------------|
| <code>--ca (-c)</code>                   | CA name used for certmonger (default: <code>ipa</code> ) |
| <code>--postgres-client-cert (-g)</code> | install certs for postgres client                        |
| <code>--postgres-server-cert</code>      | install certs for postgres server                        |

| Option                           | Description                                                                    |
|----------------------------------|--------------------------------------------------------------------------------|
| <code>--http-cert</code>         | install certs for http server (to create certs/httpd* values for a unique key) |
| <code>--extauth-opts (-x)</code> | external authentication options                                                |



## NOTE

The certificate options augment the functionality of the `certmonger` tool and enable creating a certificate signing request (CSR), and specifying `certmonger` the directories to store the keys.

**Table A.5. Other Options**

| Option                      | Description                                                                                     |
|-----------------------------|-------------------------------------------------------------------------------------------------|
| <code>--logdisk (-l)</code> | log disk path                                                                                   |
| <code>--tmpdisk</code>      | initialize the given device for temp storage (volume mounted at <code>/var/www/miq_tmp</code> ) |
| <code>--verbose (-v)</code> | print more debugging info                                                                       |

## Example Usage

```
$ ssh root@appliance.test.company.com
```

To create a new database locally on the server using `/dev/sdb`:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the `v2_key` from a host `some.example.com` to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where `db.example.com` is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --hostname db.example.com --password mydatabasepassword
```

To configure external authentication:

-

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli --uninstall-ipa
```

## A.2. STORAGE SKU TYPES

Table A.6. Storage SKU types

| SKU Type       | Description                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Standard_LRS   | Locally Redundant Storage: Synchronous copies of data are created within a single data center.                                                                    |
| Standard_GRS   | Geographically Redundant Storage: Same as Standard_LRS with additional asynchronous copies stored in a secondary data center in a separate geographical location. |
| Standard_RAGRS | Read-Access Geographically Redundant Storage: Same as Standard_GRS with additional read access to the secondary data center.                                      |
| Standard_ZRS   | Zone Redundant Storage: For block blobs only. Stores three copies of data across multiple data centers within the region or across regions.                       |
| Premium_LRS    | Premium Locally Redundant Storage: Same as Standard_LRS, but uses Premium Storage disks.                                                                          |