
*Comprehensive, hands-on training that solves real world problems*

# Red Hat Enterprise Virtualization

## Student Workbook



RH318-RHEV3.5-en-5-20150515

# RED HAT ENTERPRISE VIRTUALIZATION

# Red Hat Enterprise Virtualization 3.5 RH318
# Red Hat Enterprise Virtualization
# Edition 5 20150515

Authors: Robert Locke, Razique Mahroua, Forrest Taylor, George Hacker, Rudolf Kastl, Scott McBrien, Wander Boessenkool
Editor: Brandon Nolta, Barbara O'Brien



Contributors: Susan Lauber

# Document Conventions

## Notes and Warnings

### Note

"Notes" are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

### Important

"Important" boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled "Important" will not cause data loss, but may cause irritation and frustration.

### Warning

"Warnings" should not be ignored. Ignoring warnings will most likely cause data loss.

### References

"References" describe where to find external documentation relevant to a subject.

# Introduction

## Red Hat Enterprise Virtualization

*Red Hat Enterprise Virtualization* (RH318) is targeted at system administrators responsible for managing virtual machines for their organization. It features the use of the Red Hat Enterprise Virtualization suite.

This course will discuss how to manage large numbers of virtual machines and physical virtualization hosts using Red Hat Enterprise Virtualization. Students will learn how to install and use a RHEV-M management console; set up physical hosts running either the RHEV-H hypervisor or Red Hat Enterprise Linux; create and manage virtual machines; and create and manage images on NFS and iSCSI storage servers.

### Objectives

- Create, deploy, manage, and migrate Linux virtual machines in a large-scale environment using Red Hat Enterprise Virtualization.

### Audience

- System administrators who want to gain deeper knowledge about large-scale virtualization of systems, whether those systems are Linux-based or Microsoft Windows-based.

### Prerequisites

- Students should enter the class with RHCSA or RHCT certification *or* have equivalent Linux skills.

# Orientation to the Classroom Environment

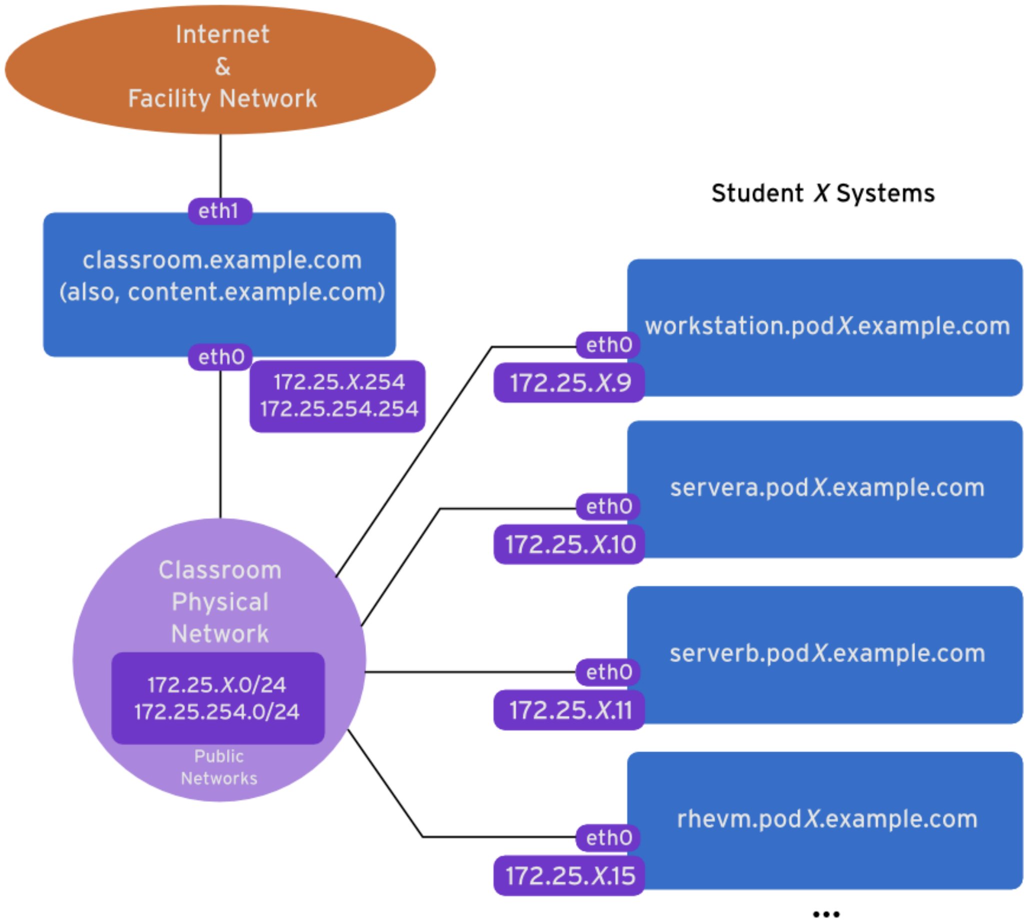


*Figure 0.1: Classroom diagram*

In this course, students will do most hands-on practice exercises and lab work with a set of computer systems, which will be referred to as **`workstation`**, **`rhevm`**, **`servera`**, and **`serverb`**. These machines have the host names workstation.podX.example.com, rhevm.pod*X*.example.com, servera.pod*X*.example.com, and serverb.podX.example.com, where the *X* in the computers' host names will be a number that will vary from student to student. All machines have a standard user account, *student*, with the password *student*. The *root* password on all systems is *redhat*.

We recommend that you use the console of **`workstation`** and then **`ssh`** into the other systems to manage them when possible. This should simplify your desktop experience when doing the labs. A lab that *requires* console access will state such in the lab instructions.

> **Important**
>
> Instructions on how to control your stations vary depending on whether you are taking this course in a physical classroom or in a virtual classroom.
>
> Read the *Instructor-Led Training (ILT)* section if you are taking this course in a classroom that is using local computer hardware and an in-person instructor.
>
> Read the *Virtual Training (VT)* section if you are taking this course using a remote classroom accessed through your web browser and have a remote instructor.

# Instructor-led training (ILT)

In a live instructor-led classroom, students will be assigned two physical computers with the first preinstalled as foundation*X*.ilt.example.com. This system will be used to access the virtual machines (**`workstation`**, **`rhevm`**, and **`serverb`**) running on this host. Students should log into this machine as user *kiosk* with the password *redhat*. The second will be installed by the student as part of the coursework (**`servera`**).

Each student is on the IPv4 network 172.25.*X*.0/24, where the *X* matches the number of their pod in workstation.podX.example.com, rhevm.pod*X*.example.com, servera.pod*X*.example.com, and serverb.podX.example.com systems. The instructor runs a central utility server, classroom.example.com, which acts as a router for the classroom networks and provides DNS, DHCP, HTTP, and other content services.

**Classroom Machines**

| Machine name | IP addresses | Role |
|---|---|---|
| workstation.podX.example.com | 172.25.*X*.9 | Student "client" |
| rhevm.pod*X*.example.com | 172.25.*X*.15 | Student "RHEV Manager" |
| servera.pod*X*.example.com | 172.25.*X*.10 | Student "first RHEV Host" |
| serverb.podX.example.com | 172.25.*X*.11 | Student "second RHEV Host" |
| classroom.example.com | 172.25.254.254 | Classroom utility server |

### Controlling your stations

On foundation*X*, a special command called **`rht-vmctl`** is used to work with the virtual machines. The commands in the following table should be run as the *kiosk* user on foundation*X*, and can be used with **`workstation`** (as in the examples) or the other virtual machine names **`rhevm`** and **`serverb`**.

**`rht-vmctl` Commands**

| Action | Command |
|---|---|
| Start **`workstation`** machine. | **`rht-vmctl start workstation`** |
| View "physical console" to log in and work with **`workstation`** machine. | **`rht-vmctl view workstation`** |
| Reset **`workstation`** machine to its previous state and restart virtual machine. **Caution: Any work generated on the disk will be lost**. | **`rht-vmctl reset workstation`** |

At the start of a lab exercise, if an instruction to reset your workstation appears, that means the command **`rht-vmctl reset workstation`** should be run in a prompt on the foundation*X* system as user *kiosk*. Likewise, if an instruction to reset your rhevm appears, that means the command **`rht-vmctl reset rhevm`** should be run on foundation*X* as user *kiosk*.

# Virtual training (VT)

In a VT classroom, students will be assigned remote computers that will be accessed through a web application hosted at *live.redhat.com* [https://live.redhat.com]. Students should log into this machine using the user credentials they provided when registering for the class.

Each student is on the IPv4 network 172.25.*X*.0/24, where the *X* matches the number of their pod in workstation.podX.example.com, rhevm.pod*X*.example.com, servera.pod*X*.example.com, and serverb.podX.example.com systems. The instructor runs a central utility server, classroom.example.com, which acts as a router for the classroom networks and provides DNS, DHCP, HTTP, and other content services.

**Classroom Machines**

| Machine name | IP addresses | Role |
|---|---|---|
| workstation.podX.example.com | 172.25.*X*.9 | Student "client" |
| rhevm.pod*X*.example.com | 172.25.*X*.15 | Student "RHEV Manager" |
| servera.pod*X*.example.com | 172.25.*X*.10 | Student "first RHEV Host" |
| serverb.podX.example.com | 172.25.*X*.11 | Student "second RHEV Host" |
| classroom.example.com | 172.25.254.254 | Classroom utility server |

### Controlling your stations

The top of the console describes the state of your machine.

**Machine States**

| State | Description |
|---|---|
| none | Your machine has not yet been started. When started, your machine will boot into a newly initialized state (the disk will have been reset). |
| starting | Your machine is in the process of booting. |
| running | Your machine is running and available (or, when booting, soon will be.) |
| stopping | Your machine is in the process of shutting down. |
| stopped | Your machine is completely shut down. Upon starting, your machine will boot into the same state as when it was shut down (the disk will have been preserved). |
| impaired | A network connection to your machine cannot be made. Typically this state is reached when a student has corrupted networking or firewall rules. If the condition persists after a machine reset, or is intermittent, please open a support case. |

Depending on the state of your machine, a selection of the following actions will be available to you.

**Machine Actions**

| Action | Description |
|---|---|
| power on | Start ("power on") the machine. |

| Action | Description |
|---|---|
| power off | Stop ("power off") the machine, preserving the contents of its disk. |
| reset | Stop ("power off") the machine and select **Reset** to reset your image, resetting the disk to its initial state. **Caution: Any work generated on the disk will be lost.** |
| Refresh Page | Refresh the page will reprobe the machine state. |

At the start of a lab exercise, if an instruction to reset workstation appears, that means you should press the **reset** button in the workstation console. Likewise, if an instruction to reset your rhevm appears, that means you should press the **reset** button in the rhevm console.

# Internationalization

## Language support

Red Hat Enterprise Linux 7 officially supports 22 languages: English, Assamese, Bengali, Chinese (Simplified), Chinese (Traditional), French, German, Gujarati, Hindi, Italian, Japanese, Kannada, Korean, Malayalam, Marathi, Odia, Portuguese (Brazilian), Punjabi, Russian, Spanish, Tamil, and Telugu.

## Per-user language selection

Users may prefer to use a different language for their desktop environment than the system-wide default. They may also want to set their account to use a different keyboard layout or input method.

### Language settings

In the GNOME desktop environment, the user may be prompted to set their preferred language and input method on first login. If not, then the easiest way for an individual user to adjust their preferred language and input method settings is to use the **Region & Language** application. Run the command **`gnome-control-center region`**, or from the top bar, select **(User)** > **Settings**. In the window that opens, select **Region & Language**. The user can click the **Language** box and select their preferred language from the list that appears. This will also update the **Formats** setting to the default for that language. The next time the user logs in, these changes will take full effect.

These settings affect the GNOME desktop environment and any applications, including **`gnome-terminal`**, started inside it. However, they do not apply to that account if accessed through an **`ssh`** login from a remote system or a local text console (such as **`tty2`**).

> **Note**
>
> A user can make their shell environment use the same **`LANG`** setting as their graphical environment, even when they log in through a text console or over **`ssh`**. One way to do this is to place code similar to the following in the user's **`~/.bashrc`** file. This example code will set the language used on a text login to match the one currently set for the user's GNOME desktop environment:
>
> ```
> i=$(grep 'Language=' /var/lib/AccountService/users/${USER} \
>   | sed 's/Language=//')
> if [ "$i" != "" ]; then
>     export LANG=$i
> fi
> ```
>
> Japanese, Korean, Chinese, or other languages with a non-Latin character set may not display properly on local text consoles.

Individual commands can be made to use another language by setting the **`LANG`** variable on the command line:

```
[user@host ~]$ LANG=fr_FR.utf8 date
```

```
jeu. avril 24 17:55:01 CDT 2014
```

Subsequent commands will revert to using the system's default language for output. The **locale** command can be used to check the current value of **LANG** and other related environment variables.

**Input method settings**

GNOME 3 in Red Hat Enterprise Linux 7 automatically uses the **IBus** input method selection system, which makes it easy to change keyboard layouts and input methods quickly.

The **Region & Language** application can also be used to enable alternative input methods. In the **Region & Language** application's window, the **Input Sources** box shows what input methods are currently available. By default, **English (US)** may be the only available method. Highlight **English (US)** and click the **keyboard** icon to see the current keyboard layout.

To add another input method, click the **+** button at the bottom left of the **Input Sources** window. An **Add an Input Source** window will open. Select your language, and then your preferred input method or keyboard layout.

Once more than one input method is configured, the user can switch between them quickly by typing **Super**+**Space** (sometimes called **Windows**+**Space**). A *status indicator* will also appear in the GNOME top bar, which has two functions: It indicates which input method is active, and acts as a menu that can be used to switch between input methods or select advanced features of more complex input methods.

Some of the methods are marked with gears, which indicate that those methods have advanced configuration options and capabilities. For example, the Japanese **Japanese (Kana Kanji)** input method allows the user to pre-edit text in Latin and use **Down Arrow** and **Up Arrow** keys to select the correct characters to use.

US English speakers may find also this useful. For example, under **English (United States)** is the keyboard layout **English (international AltGr dead keys)**, which treats **AltGr** (or the right **Alt**) on a PC 104/105-key keyboard as a "secondary-shift" modifier key and dead key activation key for typing additional characters. There are also Dvorak and other alternative layouts available.

### Note

Any Unicode character can be entered in the GNOME desktop environment if the user knows the character's Unicode code point, by typing **Ctrl**+**Shift**+**U**, followed by the code point. After **Ctrl**+**Shift**+**U** has been typed, an underlined **u** will be displayed to indicate that the system is waiting for Unicode code point entry.

For example, the lowercase Greek letter lambda has the code point U+03BB, and can be entered by typing **Ctrl**+**Shift**+**U**, then **03bb**, then **Enter**.

# System-wide default language settings

The system's default language is set to US English, using the UTF-8 encoding of Unicode as its character set (**en_US.utf8**), but this can be changed during or after installation.

From the command line, *root* can change the system-wide locale settings with the **localectl** command. If **localectl** is run with no arguments, it will display the current system-wide locale settings.

To set the system-wide language, run the command **`localectl set-locale LANG=`*`locale`***, where *`locale`* is the appropriate **`$LANG`** from the "Language Codes Reference" table in this chapter. The change will take effect for users on their next login, and is stored in **`/etc/locale.conf`**.

```
[root@host ~]# localectl set-locale LANG=fr_FR.utf8
```

In GNOME, an administrative user can change this setting from **Region & Language** and clicking the **Login Screen** button at the upper-right corner of the window. Changing the **Language** of the login screen will also adjust the system-wide default language setting stored in the **`/etc/locale.conf`** configuration file.

### Important

Local text consoles such as **`tty2`** are more limited in the fonts that they can display than **`gnome-terminal`** and **`ssh`** sessions. For example, Japanese, Korean, and Chinese characters may not display as expected on a local text console. For this reason, it may make sense to use English or another language with a Latin character set for the system's text console.

Likewise, local text consoles are more limited in the input methods they support, and this is managed separately from the graphical desktop environment. The available global input settings can be configured through **`localectl`** for both local text virtual consoles and the X11 graphical environment. See the **`localectl`**(1), **`kbd`**(4), and **`vconsole.conf`**(5) man pages for more information.

## Language packs

When using non-English languages, you may want to install additional "language packs" to provide additional translations, dictionaries, and so forth. To view the list of available langpacks, run **`yum langavailable`**. To view the list of langpacks currently installed on the system, run **`yum langlist`**. To add an additional langpack to the system, run **`yum langinstall`** ***`code`***, where *`code`* is the code in square brackets after the language name in the output of **`yum langavailable`**.

### References

**`locale`**(7), **`localectl`**(1), **`kbd`**(4), **`locale.conf`**(5), **`vconsole.conf`**(5), **`unicode`**(7), **`utf-8`**(7), and **`yum-langpacks`**(8) man pages

Conversions between the names of the graphical desktop environment's X11 layouts and their names in **`localectl`** can be found in the file **`/usr/share/X11/xkb/rules/base.lst`**.

# Language Codes Reference

**Language Codes**

| Language | $LANG value |
|---|---|
| English (US) | en_US.utf8 |
| Assamese | as_IN.utf8 |
| Bengali | bn_IN.utf8 |
| Chinese (Simplified) | zh_CN.utf8 |
| Chinese (Traditional) | zh_TW.utf8 |
| French | fr_FR.utf8 |
| German | de_DE.utf8 |
| Gujarati | gu_IN.utf8 |
| Hindi | hi_IN.utf8 |
| Italian | it_IT.utf8 |
| Japanese | ja_JP.utf8 |
| Kannada | kn_IN.utf8 |
| Korean | ko_KR.utf8 |
| Malayalam | ml_IN.utf8 |
| Marathi | mr_IN.utf8 |
| Odia | or_IN.utf8 |
| Portuguese (Brazilian) | pt_BR.utf8 |
| Punjabi | pa_IN.utf8 |
| Russian | ru_RU.utf8 |
| Spanish | es_ES.utf8 |
| Tamil | ta_IN.utf8 |
| Telugu | te_IN.utf8 |

# CHAPTER 1

# RED HAT ENTERPRISE VIRTUALIZATION OVERVIEW

| Overview | |
|---|---|
| **Goal** | Describe virtualization in Linux and the architecture of the Red Hat Enterprise Virtualization platform. |
| **Objectives** | • Define and discuss general virtualization, Kernel-based Virtual Machine (KVM), and Red Hat virtualization product concepts.<br>• Describe the Red Hat Enterprise Virtualization platform. |
| **Sections** | • Virtualization in Linux (and Quiz)<br>• Red Hat Enterprise Virtualization Architecture (and Quiz) |
| **Lab** | • Preparing the Classroom Environment |

# Virtualization in Linux

## Objectives

After completing this section, students should be able to:

- Define and discuss general virtualization, Kernel-based Virtual Machine (KVM), and Red Hat virtualization concepts.

## Virtualization

*System virtualization* allows a single computer to be partitioned or divided into multiple virtual computers, which may each run its own operating system simultaneously. These virtual machines are isolated from each other. From the perspective of each operating system, it is running on its own private hardware. They may have their own network interfaces and IP addresses, file systems, and other peripherals. Different virtual machines need not run the same operating system or version of the operating system.

A *hypervisor* is the software that manages and supports the virtualization environment. It runs the virtual machines for each virtualized operating system, providing access to virtual CPUs, memory, disks, networking, and other peripherals while restricting the virtual machines from having direct access to the real hardware and each other. The physical machine or operating system providing the hypervisor is called the *host*.

A *guest* is a *virtual machine* running on the hypervisor.

### Virtualization types

*Full virtualization* is a form of system virtualization that allows unmodified operating systems and software to be run on a virtual machine exactly as if it were running directly on real hardware. The virtual machine environment looks just like bare metal, complete with virtual peripherals. Access to physical hardware is controlled by the hypervisor so that virtual machines cannot interfere with each other.

*Native virtualization* or *hardware-assisted full virtualization* allows almost all code to be run directly by the CPU without any changes, for efficiency. The hypervisor only needs to intervene when the code uses *sensitive instructions* that would interfere with the state of the hypervisor or its supporting environment. These sensitive instructions need to be intercepted by the hypervisor and replaced with safe equivalents before they execute on the CPU. In order to easily support this, all sensitive instructions in the CPU's instruction set architecture (ISA) must be *privileged*; that is, when run in user mode, the sensitive instructions must first be trapped so control can be passed to the hypervisor. The hypervisor replaces these sensitive instructions with safe instructions that emulate the operation.

The development of Intel Virtualization Technology (VT-x) and AMD Virtualization (AMD-V) on most modern 32-bit and 64-bit x86 processors makes hardware-assisted full virtualization possible. In Linux, the KVM hypervisor requires hardware-assisted full virtualization on x86 with Intel VT-x or AMD-V. Red Hat Enterprise Virtualization uses native virtualization.

*Paravirtualization* is a technique where the hypervisor provides the guest operating system with special interfaces so that it can communicate more efficiently with the hypervisor. Typically, this requires modifications to the guest operating system or the installation of special hypervisor-aware drivers, the latter being available with KVM.

Normally, the hypervisor presents a number of emulated hardware devices as part of the virtual environment; for example, it may present a device that looks like an e1000 network card. This allows the guest operating system to use the standard driver for a physical e1000 network card to make the virtual NIC work.

One way to improve performance further is to have the hypervisor stop presenting devices that emulate real hardware and instead present virtual devices that require special paravirtualized drivers to work. The cooperation between the kernel and the hypervisor allows paravirtualized drivers to have much lower overhead than native drivers.

In addition, paravirtualization allows the hypervisor to help guest operating systems resolve timing issues that may arise as guests alternate running on a physical CPU. Therefore, even on systems which do not require paravirtualized approaches to allow virtualization, this sort of cooperation between the kernel and the hypervisor can be useful. In this context, paravirtualization is sometimes referred to as *cooperative virtualization*.

Lastly, a new enablement technology, *virt-in-virt* or *nested virt*, is where the hypervisor mirrors the CPU feature set to a guest. This can be used to test hypervisors inside a virtual machine. New CPU and kernel module features aid the enablement of this technology. While partially available, it is not currently supported in Red Hat Enterprise Linux.

### Hardware-assisted virtualization technologies

In x86 and x86-64 processor architectures, code is run on the CPU in one of four privilege levels, or rings. Normally, privileged code (the Linux kernel) runs in ring 0, where it has full access to hardware interfaces and all registers. User-space processes run in ring 3, which has limited access to hardware and protected pages in memory. With the hypervisor, guest kernel, and applications vying for access, users are faced with *ring compression*.

In the first generation, new processor instructions are available on Intel VT/AMD-V-capable CPUs, which can place the CPU in a new execution mode. When executing instructions for a hardware-assisted virtual machine, the CPU switches to a *nonroot* or *guest* mode in which the virtual machine's kernel can run in ring 0 and the user space can run in ring 3. When an instruction that must be trapped by the hypervisor is issued, the CPU leaves this mode and returns to the normal *root* or *host* mode, in which the privileged hypervisor is running in ring 0. Each virtual CPU for each virtual machine has a virtual machine control block or structure associated with it, about a page (4 KiB) in size, which stores information about the state of the processor in that virtual machine's *guest* mode.

Second-generation x86 hardware virtualization support provides Memory Management Unit (MMU) virtualization. Normally, the CPU needs to spend many cycles handling the mappings of memory pages to virtual machines. MMU virtualization allows this work to be offloaded to special hardware, improving performance. On Intel, this is called *Extended Page Tables* (EPT) and was introduced on Nehalem microarchitecture processors. On AMD, this is *Rapid Virtualization Indexing* (RVI), which was introduced on quad-core Opteron processors in 2007. This is also sometimes referred to as *nested page tables support*.

Third-generation x86 hardware virtualization support is focused on I/O virtualization. The enabling technology on the motherboard chipset is secure PCI passthrough; this allows physical PCI devices to be directly attached to virtual machines without exposing the hypervisor or other virtual machines to attacks through direct memory transfers or PCI bus snooping. This can deliver near-native I/O performance for virtual machines. On Intel, this is called *Virtualization Technology for Directed I/O* (VT-d); on AMD, this is called *I/O Virtualization Technology* (AMD-Vi), originally called IOMMU.

The other third-generation technology is at the PCI-device level, *Single Root I/O Virtualization* (SR-IOV). SR-IOV allows an SR-IOV-capable PCIe device to be securely divided into multiple virtual devices, which can then be individually attached directly to different virtual machines through VT-d/IOMMU. Essentially, multiple VMs can share one hardware PCIe device without interfering with each other. For example, a 10GbE network card could be divided into 15 virtual network cards in hardware and tied to individual virtual machines. One example is the Intel 82576 Gigabit Ethernet card; two drivers exist, the **igb** driver used for the physical hardware, and the **igbvf** driver used for the virtual functions passed through to VMs.

# Kernel-based Virtual Machine (KVM)

KVM, the *Kernel-based Virtual Machine*, is a modern hypervisor technology integrated directly into the Linux kernel. It allows a Linux kernel running on bare metal to act directly as the hypervisor itself.

The KVM project was started by Qumranet (now part of Red Hat) in October 2006; it was ready to be submitted to the upstream kernel developers by *December 2006* and was accepted in the 2.6.20 kernel *one month later*, adding about 40,000 lines of code to Linux. KVM is fully supported in Red Hat Enterprise Linux on the x86-64 architecture starting with RHEL 5.4 and is the basis of the thin hypervisor product provided in Red Hat Enterprise Virtualization.

The KVM design works very well because many features needed by a hypervisor are already implemented by the Linux kernel: processor scheduling, memory management, physical device drivers, and so on. One example of the benefits of this integration can be found in power management. Power management is normally difficult to implement in operating systems. In KVM, a small amount of glue code allowed the hypervisor to be tied into automatic frequency scaling and suspend/resume support for virtual machines. As another example, KVM also inherits real-time and SELinux features from the kernel.

### Thick vs. thin virtualization host

The intended use of a virtualized node can have a significant effect on the best Red Hat Enterprise Linux installation plan for a KVM host. There are two basic approaches, a *thin host* that is minimally installed with just the core requirements for managing the virtual hosts and a *thick host* that consists of a much more complete user environment.

For a machine that will support guests used to provide network services, a minimal KVM host is generally better. No network services should actually be served by the host operating system, with the possible exception of services such as **ssh** for management access. The host operating system should only be used for virtualization management and *guests* should provide production network services. If the host operating system is unstable, it can affect its guests; if it is compromised, the attacker can control and compromise the guest virtual machines.

A *thick host* can be useful for development, testing, demonstrations, or simply having the ability to work in two operating systems simultaneously. For example, a software developer may be testing new code that has the ability to crash the operating system in the guest. From the KVM host, the test virtual machine can be suspended and core dumps of main memory collected, then debugging tools can be used to track down the problem. If the guest virtual machine locks up, it can be reset or analyzed from the host without interrupting the host's operating system.

### Hardware considerations

Many modern x86 and x86-64 processors from Intel and AMD support processor extensions that allow native virtualization. Before attempting to use hardware-assisted full virtualization on x86 to set up guest operating systems, it is critical to verify that the hardware CPU actually provides appropriate support for Intel Virtualization Technology (VT-x) or AMD Virtualization (AMD-V).

The **x86info** utility can provide this information, as can the file **/proc/cpuinfo**. If Intel VT-x is supported, the **vmx** flag will be listed; if AMD-V is supported, the **svm** flag will appear. Only one of these two flags must be present.

```
[root@serverb ~]# grep --color -E '(vmx|svm)' /proc/cpuinfo
```

Additionally, the BIOS must have virtualization support enabled. Some vendors disable this at the factory for certain models of hardware, some do not. The procedure for enabling virtualization support in the BIOS varies from vendor to vendor.

*Executable space protection* is used to prevent *buffer overflow* attacks and is used to further protect multiple virtual machines from each other. The *NX bit* (No eXecute) is called the XD bit (eXecute Disable) by Intel and Enhanced Virus Protection by AMD. This feature, while not required for use of KVM in Red Hat Enterprise Linux, is a requirement for the Red Hat Enterprise Virtualization Hypervisor.

```
[root@serverb ~]# grep --color -E 'nx' /proc/cpuinfo
```

### Memory management and Kernel Same-Page Merging (KSM)

One useful aspect about KVM memory management is that the physical memory seen by the guest operating system in its virtual machine is just a chunk of virtual memory allocated to a **qemu-kvm** process on the host. The host can handle this chunk of memory like it would for regular processes; it can page it in and out, share it with other processes, have it backed by a file on secondary storage, and so on. It is automatically reclaimed when the virtual machine is shut down. The remainder of the host's virtual memory is usable by the host/hypervisor for normal purposes or supporting functions.

> **Note**
>
> Paging out host virtual memory to swap is a *last resort* in a situation where the host is under heavy memory pressure. Since this feature is available, though, KVM is able to overcommit memory on the host if necessary.

*Kernel Same-page Merging* (KSM) is a feature in Red Hat Enterprise Linux that allows identical memory pages to be shared between processes. KVM uses this mechanism to allow memory overcommit. In two similar virtual machines, there is a good chance that there are many identical pages in memory. KSM allows the host to scan virtual machine memory to find identical pages and consolidate them as one shared memory page, reducing physical memory consumption.

Participating processes register memory pages for use with KSM. A user-space utility, **ksmctl**, is used to control the **ksm.ko** kernel module that implements KSM. The command **ksmctl start *npages sleep*** starts KSM, scans *npages* memory pages, and sleeps for *sleep* ms before scanning that many pages again. If identical pages are found, they are consolidated into one copy-on-write shared memory page. When a guest attempts to write to the shared page, it gets its own new copy with its changes, and other guests can continue to share the original shared page. The command **ksmctl stop** stops scanning, and **ksmctl info** prints out information about KSM activity.

KSM does cause overhead, and potentially extra page faults, so it is not normally enabled unless the system is under memory pressure. Red Hat Enterprise Virtualization is able to do this automatically when it detects host resources are becoming constrained.

### Device model and VirtIO

For unmodified operating systems to run in a virtual machine using virtualization extensions, they need to have something that acts like real hardware that their unmodified drivers can use. In many cases, it is infeasible to allow guests to have direct access to physical hardware on the system. One issue is that any device that allows DMA that can be directly used by a guest can be used to bypass the hypervisor and access regions of physical memory allocated to other guests. This compromises the security and isolation of all guests on the host, which is a problem with mechanisms such as PCI passthrough on most current x86 hardware. AMD IOMMU and Intel VT-d are meant to fix this issue, but are not yet widely supported features. Another issue is that most hardware devices are not designed to be accessed from multiple operating systems at the same time, so there may not be enough hardware to go around for each VM. To some extent, PCIe SR-IOV will help address these issues.

In the meantime, the KVM hypervisor does not permit guests to have low-level access to real hardware. Instead, it presents emulated hardware to guests through a *device model*. Only peripheral hardware such as network interfaces, video cards, disk controllers, and the BIOS are emulated; the CPU is not emulated, and the code itself still runs on the processor directly.

A modified version of the QEMU emulator, **`qemu-kvm`**, is used as the KVM device model. Each guest runs its own modified QEMU device model to provide hardware devices and isolate them from other guests. The hardware selected is relatively basic and old so that it is more likely that a wide variety of guest operating systems have drivers that support them; while the interfaces may be old, performance provided is often faster than the original hardware being emulated.

Virtual hardware emulated by the device model is not as fast as it could be. The fact that real hardware has to be emulated in software means that the host needs to use more CPU emulating the hardware and performance is lower in guests than would be ideal.

As previously mentioned, the way around this is to have the hypervisor and the guest operating system cheat. The hypervisor can present paravirtualized devices to the guest that are optimized for I/O; the guest operating system installs additional drivers so it can talk to those paravirtualized devices as if they were normal hardware-based network cards, storage adapters, and so on.

One issue is that many different hypervisor vendors provide differing paravirtualized device interfaces to their hypervisors, requiring different drivers for guests. IBM and Red Hat have led the VirtIO project, which is working on an open-standard framework for hypervisor-independent and bus-independent paravirtualized drivers. The VirtIO project provides a way to allow standardized drivers to be easily adapted to different hypervisor designs using a shim layer. KVM currently implements this standard.

The upstream Linux kernel includes VirtIO drivers for Linux guests, including a network driver, block device driver, and clock driver. These drivers have been backported into the kernels shipped with Red Hat Enterprise Linux 5.3/4.8 and later. During installation of a guest, the kernel should automatically detect the paravirtualized devices provided and load the VirtIO drivers. VirtIO drivers for Microsoft Windows are also provided by Red Hat through the Red Hat Network service.

# Red Hat virtualization technologies

### Red Hat Cloud Infrastructure (RHCI)

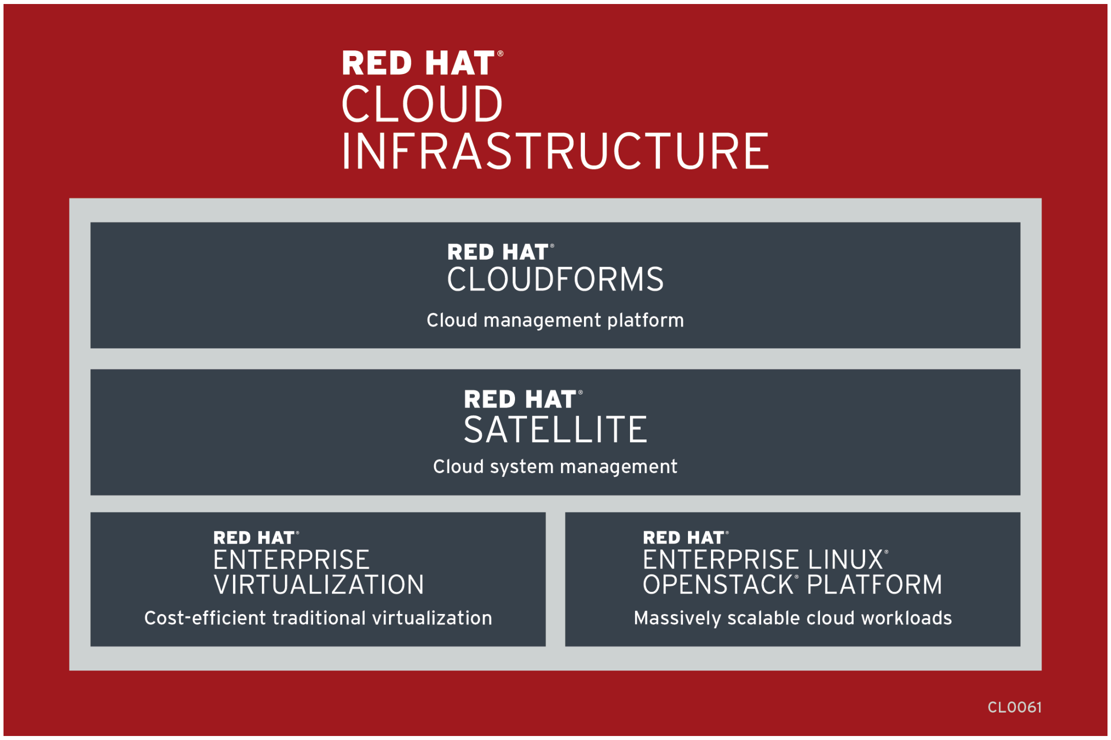


*Figure 1.1: Red Hat Cloud Infrastructure*

Red Hat Cloud Infrastructure tightly integrates four components (Red Hat CloudForms, Red Hat Satellite, Red Hat Enterprise Virtualization, and Red Hat Enterprise Linux OpenStack Platform) along with Red Hat Enterprise Linux, building and managing traditional virtualization workloads, private Infrastructure-as-a-Service (IaaS) clouds, and ramping up to public cloud infrastructures.

While this course focuses on the installation, configuration, and administration of Red Hat Enterprise Virtualization, there are courses available for each of the other components. At the time of this writing, the following courses were available for the other three components:

- CL220: Red Hat CloudForms Hybrid Cloud Management
- RH403: Red Hat Satellite 6 Administration
- CL210: Red Hat OpenStack Administration

There are also other courses that delve deeper into individual components of the Red Hat Enterprise Linux OpenStack Platform. Check the website for more information.

### Upstream projects

Note the upstream project names for each of the RHCI components:

1. Red Hat CloudForms
    - ManageIQ—**`http://manageiq.org`**

2. Red Hat Satellite
   - Foreman—**http://www.theforeman.org**
   - Puppet—**http://www.puppetlabs.com**
   - Katello—**http://www.katello.org**
   - Pulp—**http://www.pulpproject.org**
   - Candlepin—**http://www.candlepinproject.org**
3. Red Hat Enterprise Virtualization
   - oVirt—**http://www.ovirt.org**
4. Red Hat Enterprise Linux OpenStack Platform
   - RDO—**http://openstack.redhat.com**
   - OpenStack—**http://www.openstack.org**

### Comparison of RHEV and RHEL-OSP

The most basic difference between RHEV and RHEL-OSP is the expected workload of the virtual machines: *traditional* vs. *cloud*. Traditional workloads are satisfied with RHEV while cloud workloads are satisfied with RHEL-OSP.

**VM Workload Comparison**

| Traditional | Cloud |
|---|---|
| Is one big, stateful VM. | Contains small, stateless VMs. |
| Has a one-to-one application to VM. | Has one application spanning many VMs. |
| Has a life cycle in years. | Has a life cycle of hours to months. |
| Scales up by making the VM bigger. | Scales out by adding VMs. |
| Is not designed for VM failure - needs fault tolerance. | Tolerates VM failures by creating new ones. |

Recent releases of Red Hat Enterprise Virtualization are beginning to integrate directly with Red Hat Enterprise Linux OpenStack Platform components to allow for a more singular implementation of Red Hat Cloud Infrastructure. OpenStack Glance integration includes the storage of VM templates and ISO images facilitating the use, export, and share of templates and images between RHEV and RHEL-OSP. OpenStack Neutron integration includes an ability to centralize network configurations with RHEL-OSP.

### Migration

*Migration* is the ability to move a running guest from one hardware node (host) to another. Normal migration suspends the guest, copies its memory state to the other node, then resumes the guest on the new node. *Live migration* allows the guest to continue running throughout the migration process, typically only suspending the machine at the end of the process to complete the memory transfer and to resume the guest on the new node. Remote clients should not notice that the guest has changed computers when live migration is used.

Red Hat Enterprise Virtualization currently requires that several conditions must be met in order to use migration to relocate a guest:

- Migration only transfers memory state, not file system data. Both hardware nodes must have access to the virtual block devices (VBDs) through a shared-storage device.
- The MAC address and IP addresses of the virtual network interfaces travel with the relocated guest. Since the guest operating system does not know the virtual machine has migrated, the migration must be performed between two hardware nodes on the same network link, with access to the same broadcast domain, so that network traffic can get properly redirected. Likewise, the migrated virtual machine must be able to find all networks or bridges it is expecting to be connected to after migration.
- The processor architecture of the two hardware nodes must be the same. Migration from a 64-bit x86 node to a 32-bit x86 node will not work. In some cases, there may be issues if the processor supports certain features on one machine that the other does not, even though the basic architecture is the same. For example, if the initial node supports SSSE3 CPU extensions and the new node does not, programs that detected and are using the extended math instructions may crash or malfunction. Potentially, this could include the kernel. If the guest can see physical hardware using advanced methods, such as PCI passthrough, migration will not work because some state may be contained in the hardware.

**High availability**

Since traditional workloads tend to be critical, *high availability* can automatically restart virtual machines that have been unnaturally terminated. This can be triggered by problems on the underlying host, e.g., a host becomes nonoperational due to a hardware failure or is put in maintenance mode for scheduled downtime or becomes unavailable due to lost communication with an external storage resource.

With high availability, the virtual machine is restarted with no user intervention. Resources may be rebalanced as the restarted guests would launch on a host with low current resource utilization.

### References

Additional general information may be available in the *Virtualization Getting Started Guide* for Red Hat Enterprise Linux 7, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Quiz: Virtualization in Linux

Match the following items to their counterparts in the table.

Extended Page Tables (EPT) | Hypervisor | Kernel Same-page Merging (KSM) | Kernel-based Virtual Machine (KVM) | Migration | No eXecute (NX) | Paravirtualization | Thick host | VirtIO | oVirt

| Definition | Term |
| --- | --- |
| Runs the virtual machines for each virtualized operating system | |
| Technique where the hypervisor provides the guest operating system with special, more efficient interfaces | |
| MMU virtualization support introduced by Intel on Nehalem microarchitecture processors | |
| Technology introduced in RHEL 5.4 | |
| Useful for development, testing, demonstrations, or simply to work in two operating systems simultaneously | |

| Definition | Term |
| --- | --- |
| CPU flag required by RHEV but not RHEL KVM for execute space protection | |
| Feature that allows identical memory pages to be shared between processes | |
| Project led by IBM and Red Hat for an open-standard framework for hypervisor-independent and bus-independent paravirtualized drivers | |
| Upstream project for Red Hat Enterprise Virtualization | |
| Ability to move a running guest from one hardware node (host) to another | |

## Solution

Match the following items to their counterparts in the table.

| Definition | Term |
|---|---|
| Runs the virtual machines for each virtualized operating system | Hypervisor |
| Technique where the hypervisor provides the guest operating system with special, more efficient interfaces | Paravirtualization |
| MMU virtualization support introduced by Intel on Nehalem microarchitecture processors | Extended Page Tables (EPT) |
| Technology introduced in RHEL 5.4 | Kernel-based Virtual Machine (KVM) |
| Useful for development, testing, demonstrations, or simply to work in two operating systems simultaneously | Thick host |
| CPU flag required by RHEV but not RHEL KVM for execute space protection | No eXecute (NX) |
| Feature that allows identical memory pages to be shared between processes | Kernel Same-page Merging (KSM) |
| Project led by IBM and Red Hat for an open-standard framework for hypervisor-independent and bus-independent paravirtualized drivers | VirtIO |
| Upstream project for Red Hat Enterprise Virtualization | oVirt |

| Definition | Term |
| --- | --- |
| Ability to move a running guest from one hardware node (host) to another | Migration |

# RHEV Architecture

## Objectives

After completing this section, students should be able to:

- Describe the Red Hat Enterprise Virtualization platform.

## Red Hat Enterprise Virtualization architecture

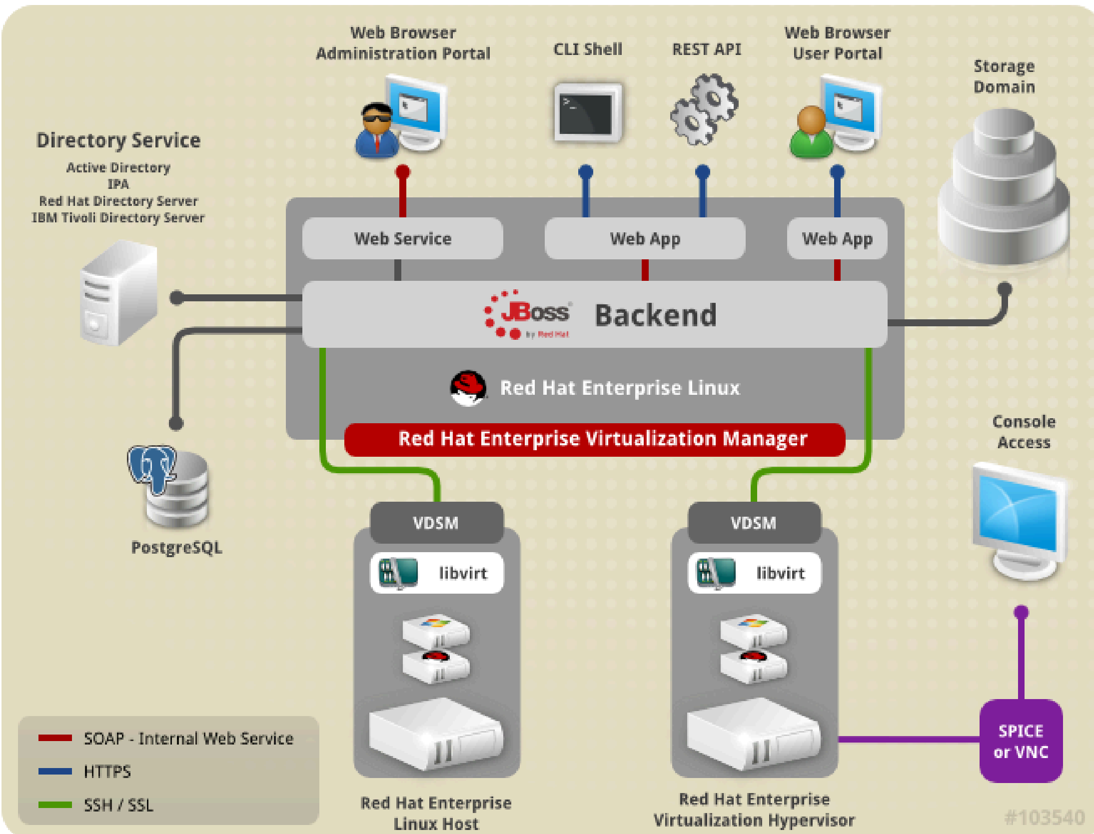


*Figure 1.2: Overview of Red Hat Enterprise Virtualization*

Red Hat Enterprise Virtualization (RHEV) is a virtualization platform that allows centralized management of hosts, virtual servers, and desktops across an enterprise datacenter featuring live migration, high availability, system scheduling, power management, image management, snapshots, thin provisioning, and monitoring.

## Engine

The core component of RHEV is Red Hat Enterprise Virtualization Manager (RHEV-M) or *engine*, which provides a central management platform for both physical and virtual resources. It allows virtual machines to be started, stopped, created from scratch or from templates, migrated from one physical hypervisor node to another, and more. It is built on Red Hat Enterprise Linux and Red Hat JBoss Enterprise Application Platform.

RHEV-M allows the management of virtual machine disk images, installation ISOs, and other storage and network components used in the datacenter. It can access the graphical consoles of these virtual machines through a number of remote access protocols.

## Hosts

Red Hat Enterprise Virtualization Hypervisor (RHEV-H) is a standalone bare-metal hypervisor based on Red Hat Enterprise Linux and KVM, used as a managed physical node.

Alternatively, Red Hat Enterprise Linux systems can be installed as hypervisors, registered to the RHEV-M console, and be used interchangeably with RHEV-H nodes.

The Virtual Desktop Server Manager (VDSM) is a management module that serves as a RHEV Manager (RHEV-M) agent on RHEV hypervisors (RHEV-H) or Red Hat Enterprise Linux hosts. VDSM allows RHEV-M to manage virtual machines, manage storage, and retrieve statistics from both hosts and guests.

On RHEV-managed hosts, the **vdsmd** service runs and listens for XMLRPC communications from the engine; authentication is handled through SSL certificates.

VDSM uses **libvirt** for basic life-cycle commands (start, stop, reboot).

## Interfaces

At the heart of the system is the RHEV-M management console, which provides an *Administration Portal* web interface that can be used to manage large numbers of virtual machines running on physical nodes. The users configured to access RHEV-M are, at present, authenticated by an Active Directory domain controller or LDAP/IPA.

The *User Portal* is used to display and access all assigned desktops. The User Portal can be accessed from thin clients (with only a web browser). A *desktop pool* is a group of identical desktops that can be used on-demand by authorized users.

The User Portal offers both standard- and power-user access. Power users can additionally manage virtual resources that have been assigned to them.

The SPICE protocol is used to provide a user desktop experience that performs as if the user's desktop was running on the local physical client.

The Representational State Transfer (REST) API allows administrators (or developers) to query or control the Red Hat Enterprise Virtualization environment. The REST API supports any programming language that supports HTTP actions.

## Storage

Storage servers provide access to virtual machine disk images to the managed hypervisor nodes in the RHEV system. This allows virtual machines to be easily migrated to other physical nodes within the RHEV environment. These storage servers may be Linux NFS servers, GlusterFS exports, other POSIX-compliant file systems, software iSCSI targets, local storage attached directly to the virtualization hosts, Fibre Channel storage devices, or Parallel NFS (pNFS). Installation of virtual machines can be started through PXE boot from the network, which allows Cobbler and RHN Satellite Kickstart servers to be used instead of (or in addition to) the ISO images stored on the central storage servers for installation.

A central storage repository must be provided that will be used to store and manage:

- Virtual machine disk images.
- Templates (which are used to create new VMs).
- ISO (CD/DVD) images (which can provide installation media).
- A temporary repository to copy, move, or back up VM images.

There are three types of storage domains: *data*, *export*, and *ISO*. The ISO or export storage must be on an NFS share, but data storage may be provided through NFS, iSCSI, FC, POSIX, Gluster, or even local storage. However, all hypervisor nodes need to have access to the storage servers in order to allow VM migration.

Storage servers can be Linux servers, NAS devices, or SAN arrays. It is strongly encouraged that the storage used be highly available and reliable due to it being a single point of failure for the virtual machines in the RHEV system. Multiple storage repositories can also be used to lessen this risk.

## Self-hosted engine

A *Self-hosted engine* is a RHEV environment in which RHEV-M runs on a VM on the hosts managed by that same RHEV-M. The VM is created as part of the host configuration and the engine is installed and configured in parallel to that host configuration process. The primary benefit of the Self-hosted engine is that it requires less hardware to deploy. This setup is very well-suited for doing proof-of-concept deployments and running small RHEV-based environments. Currently, the host must run RHEL and not be RHEV-H.

## RHEV requirements

The Red Hat Enterprise Virtualization Manager requires a system running Red Hat Enterprise Linux 6 with, at a minimum, a dual-core CPU, 4 GB of *available* system RAM, 25 GB of disk space, and a 1 Gbps network interface card. It is recommended that the system be quad-core with 16 GB of *available* system RAM and 50 GB of disk space.

The Red Hat Enterprise Virtualization hypervisor requires 64-bit CPUs with Intel VT-x or AMD-V virtualization support and executable space protection. The system should also have sufficient RAM for the cumulative guests expected to run on the host, up to a maximum 2 TB. The host also requires some local storage to store configuration, logs, kernel dumps, and swap space: a minimum of 10 GB (more if there is a lot of RAM). Lastly, the network interface card should operate at a minimum bandwidth of 1 Gbps.

### Warning

The Red Hat Enterprise Virtualization Hypervisor does not support **`fakeraid`** devices operating in RAID mode. Change the controller mode to be nonRAID in the BIOS.

### Important

In order to use the High Availability feature, all hosts in the cluster must support power management; they must have an out-of-band management agent such as an iLO, DRAC, or RSA port, and it must be configured properly.

The Administration Portal requires Firefox 17 or higher or Internet Explorer 9 or higher on Windows clients. The User Portal supports Internet Explorer 8 or higher on Windows with the SPICE ActiveX control installed or Firefox 17 or higher on Red Hat Enterprise Linux with the SPICE plug-in installed.

### References

Additional requirements information may be available in the *Installation Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Quiz: RHEV Architecture

Match the following items to their counterparts in the table.

| | | |
|---|---|---|
| Engine | NFS | Out-of-band power management agent |
| SPICE | VDSM | |

| Description | Item |
|---|---|
| RHEV Manager is also known as | |
| Management module that serves as a RHEV Manager agent on hosts | |
| Protocol that provides a user desktop experience as if local | |
| Domain type that ISO and export storage domains must use | |
| In order to use the High Availability feature, all hosts in the cluster must support | |

# Solution

Match the following items to their counterparts in the table.

| Description | Item |
|---|---|
| RHEV Manager is also known as | Engine |
| Management module that serves as a RHEV Manager agent on hosts | VDSM |
| Protocol that provides a user desktop experience as if local | SPICE |
| Domain type that ISO and export storage domains must use | NFS |
| In order to use the High Availability feature, all hosts in the cluster must support | Out-of-band power management agent |

# Lab: Preparing the Classroom Environment

In this lab, you will become oriented to the initial classroom environment and prepare your Red Hat Enterprise Linux systems for use in upcoming labs.

| Resources | |
|---|---|
| **Machines** | **`workstation`**, **`rhevm`**, **`serverb`** |
| **Passwords** | student/student and root/redhat |

### Outcome(s)

You should be able to confirm the system configurations (host name, IPv4 address, interface) of **`workstation`**, **`rhevm`**, and **`serverb`**. You should also be able to verify the feasibility of CPU flags for virtualization on **`serverb`**.

**Before you begin**

**`servera`** is to be installed in a later unit. Only **`workstation`**, **`rhevm`**, and **`serverb`** should be running and evaluated in this lab.

Perform the following steps:

1. Identify Desktop (**`workstation`**) System Settings
   1.1. Log into the **`workstation`** system console using the username **`student`** with the password **`student`**. Open a terminal with a shell prompt.

   1.2. At the prompt, run the **`hostname`** command to see what your machine's host name is. Make note of it as follows.

   *Hostname:*

   1.3. At the prompt, run the **`dig`** command on your machine's host name to determine your expected IPv4 address. Make note of it as follows.

   *IPv4 address:*

   1.4. At the prompt, run the **`ip addr show`** command to see what interface your machine's IPv4 address is attached to. Write it down as follows.

   *Interface name:*

2. Identify RHEV Manager (**`rhevm`**) System Settings
   2.1. **`ssh`** into the **`rhevm.podX.example.com`** system from the **`workstation`** terminal using the username **`student`** with the password **`student`**.

   2.2. At the prompt, run the **`hostname`** command to see what your machine's host name is. Make note of it as follows.

   *Hostname:*

   2.3. At the prompt, run the **`dig`** command on your machine's host name to determine your expected IPv4 address. Make note of it as follows.

   *IPv4 address:*

2.4. At the prompt, run the **ip addr show** command to see what interface your machine's IPv4 address is attached to. Make note of it as follows.

*Interface name:*

2.5. At the prompt, **exit** from the **rhevm** session, returning to the **workstation** terminal.

3. Identify Second Host (**serverb**) System Settings

3.1. **ssh** into the **serverb.podX.example.com** system from the **workstation** terminal using the username **student** with the password **student**.

3.2. At the prompt, run the **hostname** command to see what your machine's host name is. Make note of it as follows.

*Hostname:*

3.3. At the prompt, run the **dig** command on your machine's host name to determine your expected IPv4 address. Make note of it as follows.

*IPv4 address:*

3.4. At the prompt, run the **ip addr show** command to see what interface your machine's IPv4 address is attached to. Make note of it as follows.

*Interface name:*

4. Verify Virtualization Support on Second Host (**serverb**)

4.1. Check to make sure your machine's CPU is 64-bit x86-64-capable.

Hint: Look for flag **lm**, “long mode”.

4.2. Check to make sure your machine's CPU supports Intel VT-x or AMD-V virtualization extensions.

Hint: Look for flags **vmx** (Intel) or **svm** (AMD).

4.3. Check to make sure your machine's CPU supports executable space protection.

Hint: Look for flag **nx**, “No eXecute”.

4.4. At the prompt, **exit** from the **serverb** session, returning to the **workstation** terminal.

# Solution

In this lab, you will become oriented to the initial classroom environment and prepare your Red Hat Enterprise Linux systems for use in upcoming labs.

| Resources | |
|---|---|
| **Machines** | **`workstation`**, **`rhevm`**, **`serverb`** |
| **Passwords** | student/student and root/redhat |

Outcome(s)

You should be able to confirm the system configurations (host name, IPv4 address, interface) of **`workstation`**, **`rhevm`**, and **`serverb`**. You should also be able to verify the feasibility of CPU flags for virtualization on **`serverb`**.

**Before you begin**

**`servera`** is to be installed in a later unit. Only **`workstation`**, **`rhevm`**, and **`serverb`** should be running and evaluated in this lab.

Perform the following steps:

1. Identify Desktop (**`workstation`**) System Settings
   1.1. Log into the **`workstation`** system console using the username **`student`** with the password **`student`**. Open a terminal with a shell prompt.

   1.2. At the prompt, run the **`hostname`** command to see what your machine's host name is. Make note of it as follows.

   *Hostname:* workstation.podX.example.com

   ```
   [student@workstation ~]$ hostname
   workstation.podX.example.com
   ```

   where *X* is your pod number.

   1.3. At the prompt, run the **`dig`** command on your machine's host name to determine your expected IPv4 address. Make note of it as follows.

   *IPv4 address:* 172.25.***X***.9

   ```
   [student@workstation ~]$ dig workstation.podX.example.com
   ... output omitted ...
   ;; ANSWER SECTION:
   workstation.podX.example.com.    86400   IN      A    172.25.X.9
   ... output omitted ...
   ```

   In this example, the IPv4 address is **`172.25.X.9`** (where *X* is your pod number).

   1.4. At the prompt, run the **`ip addr show`** command to see what interface your machine's IPv4 address is attached to. Write it down as follows.

   *Interface name:* eth0

   ```
   [student@workstation ~]$ ip addr show
   ```

```
... output omitted ...
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 52:54:00:00:XX:09 brd ff:ff:ff:ff:ff:ff
    inet 172.25.X.9/24 brd 172.25.X.255 scope global eth0
    inet6 fe80::21a:a0ff:fed0:d8a7/64 scope link
       valid_lft forever preferred_lft forever
... output omitted ...
```

In this example, the IPv4 address is **`172.25.X.9`** (where *X* is your pod number).

2. Identify RHEV Manager (**`rhevm`**) System Settings

   2.1. **`ssh`** into the **`rhevm.podX.example.com`** system from the **`workstation`** terminal using the username **`student`** with the password **`student`**.

```
[student@workstation ~]$ ssh student@rhevm.podX.example.com
[student@rhevm ~]$
```

   2.2. At the prompt, run the **`hostname`** command to see what your machine's host name is. Make note of it as follows.

   *Hostname:* rhevm.pod***X***.example.com

```
[student@rhevm ~]$ hostname
rhevm.podX.example.com
```

   where *X* is your pod number.

   2.3. At the prompt, run the **`dig`** command on your machine's host name to determine your expected IPv4 address. Make note of it as follows.

   *IPv4 address:* 172.25.***X***.15

```
[student@rhevm ~]$ dig rhevm.podX.example.com
... output omitted ...
;; ANSWER SECTION:
rhevm.podX.example.com.    86400   IN      A    172.25.X.15
... output omitted ...
```

   In this example, the IPv4 address is **`172.25.X.15`** (where *X* is your pod number).

   2.4. At the prompt, run the **`ip addr show`** command to see what interface your machine's IPv4 address is attached to. Make note of it as follows.

   *Interface name:* eth0

```
[student@rhevm ~]$ ip addr show
... output omitted ...
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 52:54:00:00:XX:0f brd ff:ff:ff:ff:ff:ff
    inet 172.25.X.15/24 brd 172.25.X.255 scope global eth0
    inet6 fe80::21a:a0ff:fed0:d8a7/64 scope link
       valid_lft forever preferred_lft forever
... output omitted ...
```

In this example, the IPv4 address is **172.25.*X*.15** (where *X* is your pod number).

2.5. At the prompt, **exit** from the **rhevm** session, returning to the **workstation** terminal.

3. Identify Second Host (**serverb**) System Settings

3.1. **ssh** into the **serverb.podX.example.com** system from the **workstation** terminal using the username **student** with the password **student**.

```
[student@workstation ~]$ ssh student@serverb.podX.example.com
FIXME - get the output
```

3.2. At the prompt, run the **hostname** command to see what your machine's host name is. Make note of it as follows.

*Hostname:* serverb.podX.example.com

```
[student@serverb ~]$ hostname
serverb.podX.example.com
```

where *X* is your pod number.

3.3. At the prompt, run the **dig** command on your machine's host name to determine your expected IPv4 address. Make note of it as follows.

*IPv4 address:* 172.25.***X***.11

```
[student@serverb ~]$ dig serverb.podX.example.com
... output omitted ...
;; ANSWER SECTION:
serverb.podX.example.com.    86400   IN      A    172.25.X.11
... output omitted ...
```

In this example, the IPv4 address is **172.25.*X*.11** (where *X* is your pod number).

3.4. At the prompt, run the **ip addr show** command to see what interface your machine's IPv4 address is attached to. Make note of it as follows.

*Interface name:* eth0

```
[student@serverb ~]$ ip addr show
... output omitted ...
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 52:54:00:00:XX:0b brd ff:ff:ff:ff:ff:ff
    inet 172.25.X.11/24 brd 172.25.X.255 scope global eth0
    inet6 fe80::21a:a0ff:fed0:d8a7/64 scope link
       valid_lft forever preferred_lft forever
... output omitted ...
```

In this example, the IPv4 address is **172.25.*X*.11** (where *X* is your pod number).

4. Verify Virtualization Support on Second Host (**serverb**)

4.1. Check to make sure your machine's CPU is 64-bit x86-64-capable.

Hint: Look for flag **lm**, "long mode".

```
[student@serverb ~]$ grep --color 'lm' /proc/cpuinfo
flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc
 arch_perfmon pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2
 ssse3 cx16 xtpr pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc
 arch_perfmon pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2
 ssse3 cx16 xtpr pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
... Output Omitted ...
```

As we can see in the example output, the **lm** flag appears on each of the CPUs.

4.2. Check to make sure your machine's CPU supports Intel VT-x or AMD-V virtualization extensions.

Hint: Look for flags **vmx** (Intel) or **svm** (AMD).

```
[student@serverb ~]$ grep -E --color 'vmx|svm' /proc/cpuinfo
flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc
 arch_perfmon pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2
 ssse3 cx16 xtpr pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc
 arch_perfmon pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2
 ssse3 cx16 xtpr pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
... Output Omitted ...
```

As we can see in the example output, the **vmx** flag appears on each of the Intel CPUs.

4.3. Check to make sure your machine's CPU supports executable space protection.

Hint: Look for flag **nx**, "No eXecute".

```
[student@serverb ~]$ grep --color 'nx' /proc/cpuinfo
flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc
 arch_perfmon pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2
 ssse3 cx16 xtpr pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc
 arch_perfmon pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2
 ssse3 cx16 xtpr pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
... Output Omitted ...
```

As we can see in the example output, the **nx** flag appears on each of the CPUs.

4.4. At the prompt, **exit** from the **serverb** session, returning to the **workstation** terminal.

# Summary

### Virtualization in Linux

In this section, you learned:

- Key virtualization terms include hypervisor and guest. These terms help you to differentiate virtualization types (native, full, and paravirtualization); as well as identifying hardware-assisted virtualization technologies by both Intel and AMD.
- Hardware and implementation considerations include the following elements: thick vs. thin hosts, CPU flags, memory management with KSM, and the device model with VirtIO.
- Red Hat Enterprise Virtualization differs in many aspects with Red Hat Enterprise Linux OpenStack Platform, and other Red Hat virtualization technologies.

### RHEV Architecture

In this section, you learned:

- The underlying components of the Red Hat Enterprise Virtualization architecture include: engine, hosts, interfaces, and storage.
- For each of these components, there are specific hardware and software requirements.

## CHAPTER 2

# INSTALLING, CONFIGURING, AND TESTING RHEV MANAGER

| Overview | |
|---|---|
| **Goal** | Install, configure, secure, and troubleshoot RHEV Manager. |
| **Objectives** | • Install and initially configure RHEV Manager, then test and confirm operation.<br>• Secure RHEV Manager with users, roles, and permissions.<br>• Troubleshoot RHEV Manager installation and runtime issues. |
| **Sections** | • Installing and Testing RHEV Manager (and Practice)<br>• Managing Users, Roles, and Permissions (and Practice)<br>• Troubleshooting RHEV Manager (and Practice) |
| **Lab** | • Installing, Configuring, and Testing RHEV Manager |

# Installing and Testing RHEV Manager

## Objectives

After completing this section, students should be able to:

- Install and configure RHEV Manager (engine), data collection, and reports.
- Access and navigate the RHEV Administration Portal.
- Automate installation and configuration of RHEV Manager.
- Apply release upgrades to RHEV Manager.

## Preparations

Confirm the hardware and operating system of the host platform meet specifications before installing Red Hat Enterprise Virtualization Manager (RHEV-M). The minimum and recommended hardware requirements that follow are based on a typical small- to medium-sized installation. The exact requirements vary between deployments based on size and load. Use these recommendations only as a guide.

### Warning

DNS is crucial for the environment to operate correctly and as expected, and RHEV-H, RHEL-Host, and RHEV-M all rely on DNS for name resolution. Before starting installation, an administrator must ensure that the DNS infrastructure is set correctly.

#### Hardware requirements

Consider the following minimum hardware requirements:

| Category | Requirement |
|---|---|
| **Processor** | One dual-core CPU |
| **Memory** | 4 GB of *available* system RAM (not being consumed by existing processes) |
| **Disk space** | 25 GB of locally accessible, writable disk space |
| **Network** | 1 network interface card (NIC) with bandwidth of at least 1 Gbps |

The following table shows the recommended hardware requirements:

| Category | Requirement |
|---|---|
| **Processor** | One quad-core CPU or multiple dual-core CPUs |
| **Memory** | 16 GB of available system RAM |
| **Disk space** | 50 GB of locally accessible, writable disk space |
| **Network** | 1 network interface card (NIC) with bandwidth of at least 1 Gbps |

RHEV-M 3.5 requires Red Hat Enterprise Linux 6.6. It is expected that the operating system has been successfully installed prior to starting installation of the RHEV-M software.

Entitlements to required channels are needed before proceeding with installation. Use **`subscription-manager list`** command to list the entitlement pools.

- **`Red Hat Enterprise Linux Server`** entitlement provides Red Hat Enterprise Linux.
- **`Red Hat Enterprise Virtualization`** entitlement provides Red Hat Enterprise Virtualization Manager, RHEV Hypervisor, RHEL Host, and RHEV Agent channels.
- **`JBoss Enterprise Application Platform`** entitlement provides the supported release of the application platform on which the Red Hat Enterprise Virtualization Manager runs.

Use the **`subscription-manager repos`** command to display the current list of channels a server subscribes to.

- **`Red Hat Enterprise Linux Server`** (v. 6.6 for 64-bit x86_64) channel, also referred to as **`rhel-6-server-rpms`**, provides Red Hat Enterprise Linux 6 Server. The channel entitlement name for this channel is **`Red Hat Enterprise Linux Server (v. 6)`**.
- **`RHEL Server Supplementary`** (v. 6.6 64-bit x86_64) channel, also referred to as **`rhel-6-server-supplementary-rpms`**, provides the supported version of the Java Runtime Environment (JRE) and the **`virtio-win`** package. The channel entitlement name for this channel is **`Red Hat Enterprise Linux Server Supplementary (v. 6)`**.
- **`Red Hat Enterprise Virtualization Manager`** (v.3.5 x86_64) channel, also referred to as **`rhel-6-server-rhevm-3.5-rpms`**, provides Red Hat Enterprise Virtualization Manager. The channel entitlement name for this channel is **`Red Hat Enterprise Virtualization Manager (v3)`**.

### Note

Note that the channel explicitly defines version 3.5. There is another channel for the older RHEV-M 3. Administrators need to make sure to choose the correct one.

- **`JBoss Application Platform (v 6) for Server x86_64`** channel, also referred to as **`jb-eap-6-for-rhel-6-server-rpms`**, provides the supported release of the application platform on which RHEV-M runs. The channel entitlement name for this channel is **`JBoss Enterprise Application Platform (v 4, zip format)`**.
- Enable one or both of the RHEV Hypervisor channels depending on whether you want to run RHEV Hypervisor based on RHEL 6 or RHEL 7, referred to as **`rhel-6-server-rhevh-rpms`** and **`rhel-7-server-rhevh-rpms`**.

Once the system has been subscribed to the proper repositories, install the *rhevm* package. If RHEV reports are desired, also install the *rhevm-dwh* and *rhevm-reports* packages.

To configure RHEV, administrators need to run the **`engine-setup`**, which will guide them through a series of questions regarding the configuration options for both the engine and reports.

# Demonstration: Installing and configuring RHEV manager

In this example, observe the following steps as the instructor demonstrates the RHEV-M installation process on rhevm.pod*X*.example.com.

1. **ssh** into the rhevm.pod*X*.example.com system from the workstation terminal using the username **root** with the password **redhat**.

```
[student@workstation ~]$ ssh root@rhevm.podX.example.com
[root@rhevm ~]#
```

2. Subscribe to the needed RHN channels or **yum** repositories. In the classroom, we have a set of custom **yum** repositories. To activate them, download the file **rhevm.repo** from **http://classroom.example.com/materials/rhevm.repo** and place it in the directory **/etc/yum.repos.d/**.

```
[root@rhevm ~]# wget http://classroom.example.com/materials/rhevm.repo -P /etc/
yum.repos.d/
...
[root@rhevm ~]# yum repolist
...
```

3. RHEV-M should be installed on the most current edition of Red Hat Enterprise Linux 6, so use **yum** to update all of the operating system packages:

```
[root@rhevm ~]# yum -y update
```

> **Note**
>
> If the update includes a new kernel, make sure to reboot before proceeding.

4. If necessary, reconnect to rhevm using **ssh** becoming **root**. Install the required RHEV-M packages. As a minimum, use **yum** to install the *rhevm* package. If administrators intend to use the RHEV-M reporting tool, they need to install the *rhevm-dwh* and *rhevm-reports* packages also:

```
[root@rhevm ~]# yum -y install rhevm rhevm-dwh rhevm-reports
```

5. Once the packages are installed, RHEV Manager as well as the RHEV Reports data warehouse need to be configured. This is accomplished by executing the **engine-setup**, once the packages have finished installing:

```
[root@rhevm ~]# engine-setup --generate-answer=/root/answers.txt
```

Passing the **generate-answer** flag allows the configuration file name and location to be specified.

- Configure Engine on this host (Yes, No) [Yes]: **Enter**
- Configure Data Warehouse on this host (Yes, No) [Yes]: **Enter**
- Configure Reports on this host (Yes, No) [Yes]: **Enter**
- Configure WebSocket Proxy on this host (Yes, No) [Yes]: **Enter**
- Do you want Setup to configure the firewall? (Yes, No) [Yes]: **no**
- Host fully qualified DNS name of this server [rhevm.pod*X*.example.com]: **Enter**
- Where is the Reports database located?: **local**
- Would you like Setup to automatically configure postgresql and create Reports database, or prefer to perform that manually? (Automatic, Manual): **automatic**
- Where is the DWH database located?: **local**
- Would you like Setup to automatically configure postgresql and create DWH database, or prefer to perform that manually? **automatic**
- Where is the Engine database located? (Local, Remote) [Local]: **local**
- Would you like Setup to automatically configure postgresql and create Engine database, or prefer to perform that manually? **automatic**
- Engine admin password: **redhat**
- Confirm engine admin password: **redhat**
- Use weak password? (Yes, No) [No]: **yes**
- Application mode (Virt, Gluster, Both) [Both]: **Enter**
- DWH database secured connection (Yes, No) [No]: **Enter**
- Organization name for certificate [pod*X*]: **Enter**
- Do you wish to set the application as the default page of the web server? (Yes, No) [Yes]: **Enter**
- Do you wish Setup to configure that, or prefer to perform that manually? (Automatic, Manual) [Automatic]: **Enter**
- Configure an NFS share on this server to be used as an ISO Domain? (Yes, No) [Yes]: **Enter**
- Local ISO domain path [/var/lib/exports/iso]: **/exports/rhevisos**
- Local ISO domain ACL - note that the default will restrict access to rhevm.pod*X*.example.com only, for security reasons [rhevm.pod*X*.example.com(rw)]: **Enter**
- Local ISO domain name [ISO_DOMAIN]: **iso*X***

- Reports power users password: **redhat**
- Confirm Reports power users password: **redhat**
- Use weak password? (Yes, No) [No]: **yes**
- Would you like transactions from the Red Hat Access Plugin sent from the RHEV Manager to be brokered through a proxy server? (Yes, No) [No]: **Enter**
- ... Please confirm installation settings (OK, Cancel) [OK]: **Enter**

### Note

While waiting on the **engine-setup** command to complete, use tab-complete to show the other commands that begin with **engine-**. Open a new shell and use the following command:

```
[root@rhevm ~]# engine- <Tab> <Tab>
engine-backup          engine-image-uploader  engine-manage-domains
engine-cleanup         engine-iso-uploader    engine-setup
engine-config          engine-log-collector   engine-upgrade-check
[root@rhevm ~]# rhevm- <Tab> <Tab>
rhevm-cleanup         rhevm-iso-uploader    rhevm-setup
rhevm-config          rhevm-log-collector   rhevm-shell
rhevm-image-uploader  rhevm-manage-domains
```

Each of these commands will be reviewed as the course progresses.

Once the environment is configured, the **engine-setup** script provides within the output some details about accessing the environment, together with some other information.

## Automating RHEV manager installation

Administrators can automate RHEV-M installations by generating an *answer file*. An answer file can be created upon the first installation, then reused. To generate an answer file during the first installation, the **--generate-answer** flag needs to be passed:

```
[root@rhevm ~]# engine-setup --generate-answer=/root/answers.txt
```

The installation will run the same way it runs without the flag, but it also creates a text file that contains instructions for the installer. It is then possible to use this file to automate installation. To load the answer file in order to automate the installation, the **--config-append=file** flag needs to be passed:

```
[root@rhevm ~]# engine-setup --config-append=/root/anwers.txt
```

The installation will proceed, but administrators will not be asked for any input.

> **Note**
>
> If an option is missing in the answer file, the installer will require extra input during the installation.

The answer file contains key-value entries administrators can update to have a template, which can be reused. The format of the answer file is as follows:

```
#action=setup
[environment:default]
OVESETUP_CORE/engineStop=none:None
OVESETUP_DIALOG/confirmSettings=bool:True
OVESETUP_DB/database=str:engine
OVESETUP_DB/fixDbViolations=none:None
OVESETUP_DB/secured=bool:False
OVESETUP_DB/host=str:localhost
OVESETUP_DB/user=str:engine
OVESETUP_DB/securedHostValidation=bool:False
OVESETUP_DB/password=str:0056jKkY
OVESETUP_DB/port=int:5432
OVESETUP_SYSTEM/nfsConfigEnabled=bool:True
...
OVESETUP_APACHE/configureSsl=bool:True
OSETUP_RPMDISTRO/requireRollback=none:None
OSETUP_RPMDISTRO/enableUpgrade=none:None
OVESETUP_AIO/configure=none:None
OVESETUP_AIO/storageDomainDir=none:None
```

> **Note**
>
> Passwords are not displayed in plain text.

# Applying release updates

A normal **`yum update`** will not update RHEV, since the RHEV installation **`locked`** the RHEV packages in **`yum`** using the **`versionlock`** plug-in. A list of locked packages can be found in **`/etc/yum/pluginconf.d/versionlock.list`**.

Administrators can check the availability of RHEV Manager updates by using the **`engine-upgrade-check`** command.

```
[root@rhevm ~]# engine-upgrade-check
VERB: queue package engine-setup for update
VERB: Building transaction
VERB: Empty transaction
VERB: Transaction Summary:
No upgrade
```

Update the **`engine-setup`** package to the most current version, then execute the **`engine-setup`** command as **`root`** without arguments.

This will stop all necessary RHEV Manager services, update the underlying software, then restart the services when the update is complete.

```
[root@rhevm ~]# yum update -y engine-setup
...
[root@rhevm ~]# engine-setup
...
```

# Demonstration: Testing RHEV manager

In this example, observe the following steps as the instructor demonstrates how to test the RHEV-M installation.

1. Verify the **ovirt-engine** service is running:

```
[root@rhevm ~]# service ovirt-engine status
ovirt-engine (pid  7387) is running...
```

2. Verify an ISO NFS share is available:

```
[root@rhevm ~]# showmount -e localhost | grep rhevisos
/exports/rhevisos rhevm.podX.example.com
```

3. Open a web browser on workstation and check the main portal:

```
[root@workstation ~]# firefox https://rhevm.podX.example.com &
```

4. Add a security exception to permit **firefox** to use the self-signed SSL certificate generated with the installation of RHEV-M.

   Click **I Understand the Risks** > **Add Exception...** > **Confirm Security Exception**.

5. Click the **Administration Portal** hyperlink.

6. If all goes well, a RHEV login screen should be displayed. Log in as **admin** with a password of **redhat** and the domain set to **internal**.

   Once successfully logged into RHEV-M, an administrator portal screen will appear similar to the one displayed in the following figure:

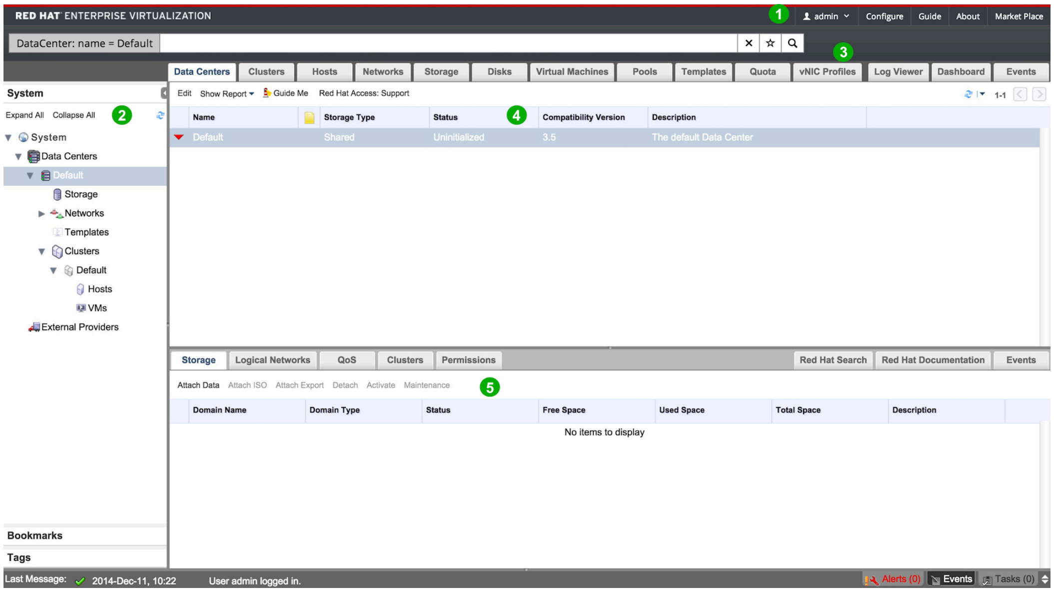


*Figure 2.1: Web Administration Portal features*

## Note

Administrators may need to resize **Firefox** to view all of the content available in the Administration Portal.

6.1. **Header**: This bar contains the name of the logged-in user, the **Sign out** button, and the option to configure user roles.

6.2. **Navigation Pane**: This pane allows administrators to navigate between the **Tree**, **Bookmarks**, and **Tags** tabs. In the **Tree** tab, tree mode allows to see the entire system tree and provides a visual representation of the virtualization environment's architecture.

6.3. **Resources Tabs**: These tabs allows administrators to access the resources of Red Hat Enterprise Virtualization. A **Default Data Center** needs to be existing, a **Default Cluster**, and available **Storage** waiting to be attached to the datacenter. After administrators install a RHEV hypervisor host and connect it to the RHEV Manager, a **Host** waiting to be approved will also appear in this tab.

6.4. **Results List**: When administrators select a tab, this list displays the available resources. It is possible to perform a task on an individual item or multiple items by selecting the item(s) and then clicking the relevant action button. If an action is not possible, the button is disabled.

6.5. **Details Pane**: When administrators select a resource, this pane displays its details in several subtabs. These subtabs also contain action buttons, which can be used to make changes to the selected resource.

## References

Red Hat Enterprise Virtualization 3.5 Installation Guide

- Section 1.2.2. Red Hat Enterprise Virtualization Manager Hardware Requirements
- Chapter 2. Installing Red Hat Enterprise Virtualization:
- Section 4.2. Data Collection Setup and Reports Installation Overview
- Section 5.1. Upgrade Between Minor Releases

# Practice: Installing and Configuring RHEV Manager

In this lab, you will install and configure a RHEV Manager server.

**Outcome(s)**
You should be able to install and configure RHEV Manager software. You should also be able to set up data warehouse reporting. Lastly, you should be able to generate and preserve an **engine-setup** answer file for later use.

**Before you begin**
You should have a server already installed with Red Hat Enterprise Linux 6.6.

Perform the following steps:

1. To install the packages needed to turn this system into a Red Hat Enterprise Virtualization Manager, you need to subscribe it to a new RHN channel or **yum** repository. In the classroom, we have a custom **yum** repository. To activate this repository, download the file **rhevm.repo** from **http://classroom.example.com/materials/rhevm.repo** and place it in the directory **/etc/yum.repos.d/**.

```
[root@rhevm ~]# wget http://classroom.example.com/materials/rhevm.repo -P /etc/
yum.repos.d/
...
[root@rhevm ~]# yum repolist
...
```

2. Update all existing Red Hat Enterprise Linux packages (reboot if necessary).

```
[root@rhevm ~]# yum update -y
```

3. Install the *rhevm*, **rhevm-dwh**, and *rhevm-reports* packages:

```
[root@rhevm ~]# yum -y install rhevm rhevm-dwh rhevm-reports
```

4. Execute **engine-setup --generate-answer=/root/answers.txt** and select the following configuration choices:

| Prompt | Answer |
|---|---|
| **Configure Engine on this host** | yes |
| **Configure Data Warehouse on this host** | yes |
| **Configure Reports on this host** | yes |
| **WebSocket Proxy** | yes |
| **Do you want Setup to configure the firewall?** | no |
| **Host fully qualified DNS name of this server** | rhevm.pod*X*.example.com |
| **Where is the Reports database located?** | local |

| Prompt | Answer |
|---|---|
| **Would you like Setup to automatically configure postgresql and create Reports database, or prefer to perform that manually? (Automatic, Manual)** | automatic |
| **Where is the DWH database located?** | local |
| **Would you like Setup to automatically configure postgresql and create DWH database, or prefer to perform that manually?** | automatic |
| **Where is the Engine database located? (Local, Remote) [Local]:** | local |
| **Would you like Setup to automatically configure postgresql and create Engine database, or prefer to perform that manually?** | automatic |
| **Engine admin password (confirm)** | redhat |
| **Use weak password?** | yes |
| **Application mode (Virt, Gluster, Both)** | both |
| **DWH database secured connection** | no |
| **Organization name for certificate** | pod*X*.example.com |
| **Do you wish to set the application as the default page of the web server?** | yes |
| **Do you wish Setup to configure that, or prefer to perform that manually?** | automatic |
| **Configure an NFS share on this server to be used as an ISO Domain?** | yes |
| **Local ISO domain path** | /exports/rhevisos |
| **Local ISO domain ACL** | rhevm.pod*X*.example.com(rw) |
| **Local ISO domain name (make sure to replace X by your station number)** | iso*X* |
| **Reports power users password (confirm the password)** | redhat |
| **Use weak password?** | yes |
| **Would you like transactions from the Red Hat Access Plugin sent from the RHEV Manager to be brokered through a proxy server?** | no |

```
[root@rhevm ~]# engine-setup --generate-answer=/root/answers.txt
[ INFO  ] Stage: Initializing
[ INFO  ] Stage: Environment setup
          Configuration files: ['/etc/ovirt-engine-setup.conf.d/10-packaging-
dwh.conf', '/etc/ovirt-engine-setup.conf.d/10-packaging-wsp.conf', '/etc/ovirt-
engine-setup.conf.d/10-packaging.conf', '/etc/ovirt-engine-setup.conf.d/20-
packaging-rhevm-reports.conf']
          Log file: /var/log/ovirt-engine/setup/ovirt-engine-setup-20141211130705-
o82mc9.log
          Version: otopi-1.3.0 (otopi-1.3.0-1.el6ev)
[ INFO  ] Stage: Environment packages setup
[ INFO  ] Stage: Programs detection
```

```
[ INFO  ] Stage: Environment setup
[ INFO  ] Stage: Environment customization

          --== PRODUCT OPTIONS ==--

          Configure Engine on this host (Yes, No) [Yes]: <Enter>
          Configure Data Warehouse on this host (Yes, No) [Yes]: <Enter>
          Configure Reports on this host (Yes, No) [Yes]: <Enter>
          Configure WebSocket Proxy on this host (Yes, No) [Yes]: <Enter>

          --== PACKAGES ==--

[ INFO  ] Checking for product updates...
[ INFO  ] No product updates found

          --== ALL IN ONE CONFIGURATION ==--


          --== NETWORK CONFIGURATION ==--

          Setup can automatically configure the firewall on this system.
          Note: automatic configuration of the firewall may overwrite current
 settings.
          Do you want Setup to configure the firewall? (Yes, No) [Yes]: No
[ INFO  ] iptables will be configured as firewall manager.
          Host fully qualified DNS name of this server
 [rhevm.podX.example.com]: <Enter>

          --== DATABASE CONFIGURATION ==--

          Where is the Reports database located? (Local, Remote) [Local]: <Enter>
          Setup can configure the local postgresql server automatically for the
 Reports to run. This may conflict with existing applications.
          Would you like Setup to automatically configure postgresql and create
 Reports database, or prefer to perform that manually? (Automatic, Manual)
 [Automatic]: <Enter>
          Where is the DWH database located? (Local, Remote) [Local]: <Enter>
          Setup can configure the local postgresql server automatically for the DWH
 to run. This may conflict with existing applications.
          Would you like Setup to automatically configure postgresql and
 create DWH database, or prefer to perform that manually? (Automatic, Manual)
 [Automatic]: <Enter>
          Where is the Engine database located? (Local, Remote) [Local]: <Enter>
          Setup can configure the local postgresql server automatically for the
 engine to run. This may conflict with existing applications.
          Would you like Setup to automatically configure postgresql and create
 Engine database, or prefer to perform that manually? (Automatic, Manual)
 [Automatic]: <Enter>

          --== OVIRT ENGINE CONFIGURATION ==--

          Engine admin password: redhat
          Confirm engine admin password: redhat
[WARNING] Password is weak: it is based on a dictionary word
          Use weak password? (Yes, No) [No]: yes
          Application mode (Virt, Gluster, Both) [Both]: <Enter>

          --== PKI CONFIGURATION ==--

          Organization name for certificate [podX.example.com]: <Enter>

          --== APACHE CONFIGURATION ==--
```

```
          Setup can configure the default page of the web server to present the
 application home page. This may conflict with existing applications.
          Do you wish to set the application as the default page of the web server?
 (Yes, No) [Yes]: <Enter>
          Setup can configure apache to use SSL using a certificate issued from the
 internal CA.
          Do you wish Setup to configure that, or prefer to perform that manually?
 (Automatic, Manual) [Automatic]: <Enter>

          --== SYSTEM CONFIGURATION ==--

          Configure an NFS share on this server to be used as an ISO Domain? (Yes,
 No) [Yes]: <Enter>
          Local ISO domain path [/var/lib/exports/iso]: /exports/rhevisos
          Local ISO domain ACL - note that the default will restrict
 access to rhevm.pod0.example.com only, for security reasons
 [rhevm.pod0.example.com(rw)]: <Enter>
          Local ISO domain name [ISO_DOMAIN]: isoX

          --== MISC CONFIGURATION ==--

          Reports power users password: redhat
          Confirm Reports power users password: redhat
[WARNING] Password is weak: it is based on a dictionary word
          Use weak password? (Yes, No) [No]: yes
          Would you like transactions from the Red Hat Access Plugin sent from the
 RHEV Manager to be brokered through a proxy server? (Yes, No) [No]: <Enter>

          --== END OF CONFIGURATION ==--

[ INFO  ] Stage: Setup validation
[WARNING] Less than 16384MB of memory is available

          --== CONFIGURATION PREVIEW ==--

          Application mode                        : both
          Firewall manager                        : iptables
          Update Firewall                         : False
          Host FQDN                               : rhevm.podX.example.com
          Engine database name                    : engine
          Engine database secured connection      : False
          Engine database host                    : localhost
          Engine database user name               : engine
          Engine database host name validation    : False
          Engine database port                    : 5432
          Engine installation                     : True
          NFS setup                               : True
          PKI organization                        : podX
          NFS mount point                         : /exports/rhevisos
          NFS export ACL                          : rhevm.podX.example.com(rw)
          Configure local Engine database         : True
          Set application as default page         : True
          Configure Apache SSL                    : True
          DWH installation                        : True
          DWH database name                       : ovirt_engine_history
          DWH database secured connection         : False
          DWH database host                       : localhost
          DWH database user name                  : ovirt_engine_history
          DWH database host name validation       : False
          DWH database port                       : 5432
          Configure local DWH database            : True
          Reports installation                    : True
          Reports database name                   : ovirt_engine_reports
          Reports database secured connection     : False
```

```
          Reports database host                 : localhost
          Reports database user name            : ovirt_engine_reports
          Reports database host name validation : False
          Reports database port                 : 5432
          Configure local Reports database      : True
          Engine Host FQDN                      : rhevm.podX.example.com
          Configure WebSocket Proxy             : True

          Please confirm installation settings (OK, Cancel) [OK]: <Enter>
[ INFO  ] Stage: Transaction setup
[ INFO  ] Stopping dwh service
[ INFO  ] Stopping reports service
[ INFO  ] Stopping engine service
[ INFO  ] Stopping ovirt-fence-kdump-listener service
[ INFO  ] Stopping websocket-proxy service
[ INFO  ] Stage: Misc configuration
[ INFO  ] Stage: Package installation
[ INFO  ] Stage: Misc configuration
[ INFO  ] Initializing PostgreSQL
[ INFO  ] Creating PostgreSQL 'engine' database
[ INFO  ] Configuring PostgreSQL
[ INFO  ] Creating PostgreSQL 'ovirt_engine_history' database
[ INFO  ] Configuring PostgreSQL
[ INFO  ] Creating PostgreSQL 'ovirt_engine_reports' database
[ INFO  ] Configuring PostgreSQL
[ INFO  ] Creating/refreshing Engine database schema
[ INFO  ] Creating CA
[ INFO  ] Creating/refreshing DWH database schema
[ INFO  ] Deploying Jasper
[ INFO  ] Importing data into Jasper
[ INFO  ] Configuring Jasper Java resources
[ INFO  ] Configuring Jasper Database resources
[ INFO  ] Customizing Jasper
[ INFO  ] Customizing Jasper metadata
[ INFO  ] Customizing Jasper Pro Parts
[ INFO  ] Configuring WebSocket Proxy
[ INFO  ] Generating post install configuration file '/etc/ovirt-engine-
setup.conf.d/20-setup-ovirt-post.conf'
[ INFO  ] Stage: Transaction commit
[ INFO  ] Stage: Closing up

          --== SUMMARY ==--

[WARNING] Less than 16384MB of memory is available
          SSH fingerprint: B8:D9:4E:E6:5F:5C:86:F1:47:17:B2:E5:D1:A1:42:42
          Internal CA 25:B0:D0:02:30:1E:B2:1C:EA:04:BD:3B:AE:9E:05:29:87:08:80:4A
          Web access is enabled at:
              http://rhevm.podX.example.com:80/ovirt-engine
              https://rhevm.podX.example.com:443/ovirt-engine
          Please use the user "admin" and password specified in order to login

          --== END OF SUMMARY ==--

[ INFO  ] Starting engine service
[ INFO  ] Restarting httpd
[ INFO  ] Restarting nfs services
[ INFO  ] Starting dwh service
[ INFO  ] Starting reports service
[ INFO  ] Stage: Clean up
          Log file is located at /var/log/ovirt-engine/setup/ovirt-engine-
setup-20141211130705-o82mc9.log
[ INFO  ] Generating answer file '/var/lib/ovirt-engine/setup/
answers/20141211131733-setup.conf'
[ INFO  ] Stage: Pre-termination
```

```
[ INFO  ] Stage: Termination
[ INFO  ] Execution of setup completed successfully
```

5. Preserve the generated answer file by copying it to workstation.

   ```
   [root@rhevm ~]# scp /root/answers.txt student@workstation.podX.example.com:
   ...
   ```

6. Verify the **ovirt-engine** service is running:

   ```
   [root@rhevm ~]# service ovirt-engine status
   ovirt-engine (pid  7387) is running...
   ```

7. Verify an ISO NFS share is available:

   ```
   [root@rhevm ~]# showmount -e localhost | grep rhevisos
   /exports/rhevisos rhevm.podX.example.com
   ```

8. Open a web browser on workstation:

   ```
   [root@workstation ~]# firefox https://rhevm.podX.example.com &
   ```

9. Add a security exception to permit **firefox** to use the self-signed SSL certificate generated with the installation of RHEV-M.

   Click **I Understand the Risks** > **Add Exception...** > **Confirm Security Exception**.

10. Click the **Administration Portal** hyperlink.

11. If all goes well, a RHEV login screen should be displayed. Log in as **admin** with a password of **redhat** and the domain set to **internal**.

    When you successfully log into RHEV-M, the Administration Portal screen will appear.

### Note

You may need to resize **Firefox** to view all of the content available in the Administration Portal.

# Managing Users, Roles, and Permissions

## Objectives

After completing this section, students should be able to:

- Link RHEV to an external directory service for authentication.
- Define the multilevel administration applied to RHEV and inheritance of authorization.
- Create and manage roles for authorization.
- Identify the purpose of the User Portal.

## The Red Hat Enterprise Virtualization authorization model

Red Hat Enterprise Virtualization has an extensive authorization model to limit and control the actions that users can perform on objects. While *Authorization* is handled by Red Hat Enterprise Virtualization, *Authentication* is handled by an external directory service. The currently available directory services for Red Hat Enterprise Virtualization are Microsoft Active Directory, Red Hat Identity Management (IdM), Red Hat Directory Server 9, and OpenLDAP.

### Warning

It is not currently possible to install the Red Hat Enterprise Virtualization Manager on the same system as Red Hat IdM, as there is a conflict around the **mod_ssl** package.

Red Hat Enterprise Virtualization can be bound to one or more directory services by using the **engine-manage-domains** command from the command line on the host running RHEV-M.

The **engine-setup** command defines a local RHEV-M administrator account (**admin**). For additional user definitions, RHEV-M can be bound to an external directory service. Administrators can use the **engine-manage-domains** command to bind to a directory service:

```
[root@rhevm ~]# engine-manage-domains add --domain=DOMAIN --user=USER --
provider=PROVIDER
```

This utility allows the password to be specified on the command line, but this is not recommended.

### Note

For security reasons, consider using the flag **--password-file=PASS_FILE**.

After making changes with **engine-manage-domains**, the **ovirt-engine** service should be restarted so that it recognizes the change.

The Red Hat Enterprise Virtualization authorization model is based around *Actions*. Actions are tasks that can be performed such as starting or stopping a virtual machine, creating a new template, or migrating a virtual machine to a different hypervisor.

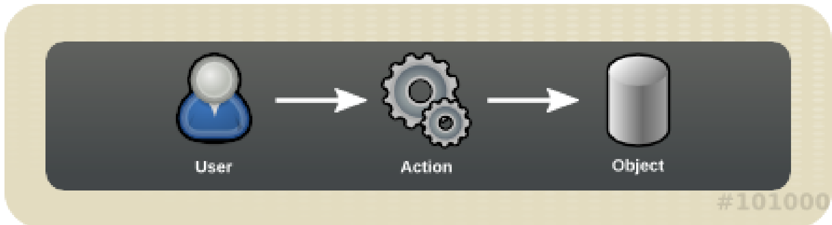


*Figure 2.2: Authorization model*

Each type of action corresponds to a *Permission*.

To simplify maintenance, multiple permissions can be combined into a **`Role`**. The system comes with multiple predefined roles such as **`SuperUser`** and **`PowerUserRole`**.

## Important

Administrators should take care when assigning or combining permissions, as they would want to limit what a user or an operator can control. For example, an administrator may need to manipulate templates but have no need to manage VMs or pools.

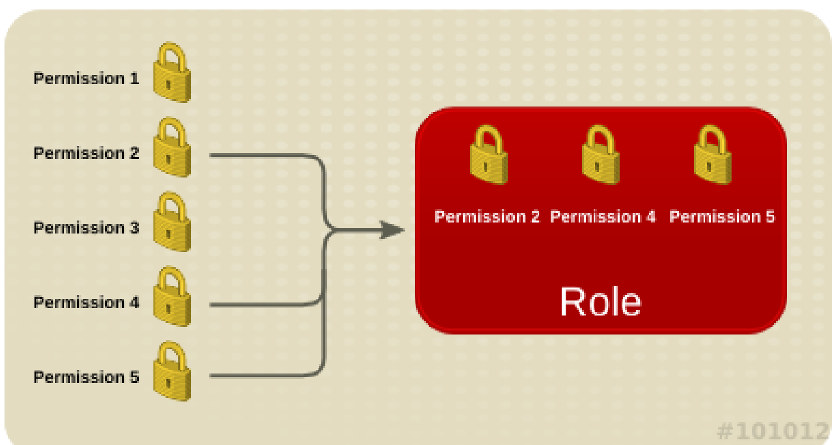


*Figure 2.3: Roles and permissions*

Users can be assigned roles systemwide or a role can be assigned on a specific object (such as a virtual machine or a datacenter). When a user is assigned a role on an object, the user will get the same role on all objects below the assigned one. The following graphic shows how permissions are inherited between objects.

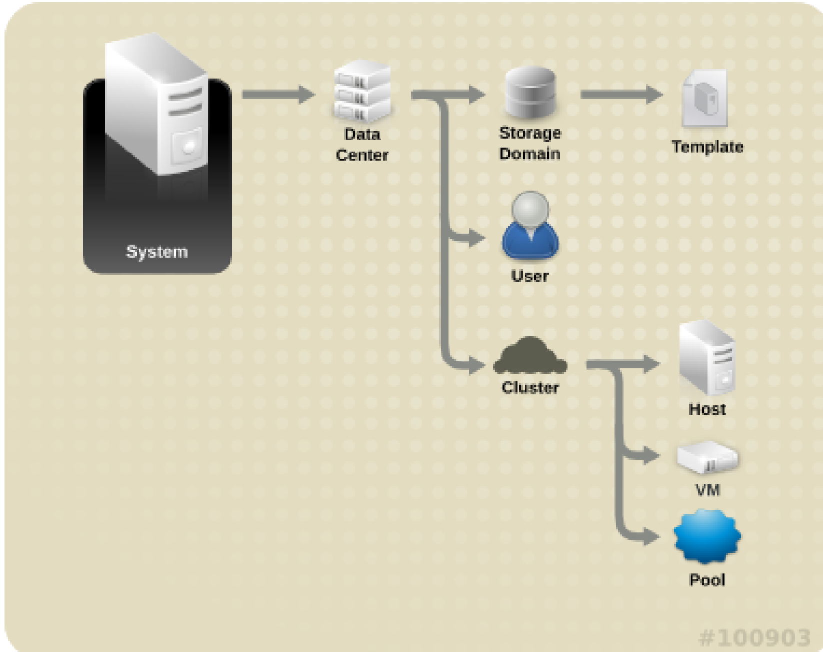


*Figure 2.4: The hierarchical layout of objects in Red Hat Enterprise Virtualization*

### Important

Some actions can affect multiple objects. Copying a template between storage domains, for example, will require the user to have the relevant permissions on both storage domains.

## Managing roles

*Roles* are divided into two major categories: *administrator roles*, which allow access to the Administration Portal, and *user roles*, which allow access to the User Portal.

### Important

For example, if administrators have an *administrator role* on a cluster, they can manage all virtual machines in the cluster using the Administration Portal. However, they cannot access any of these virtual machines in the User Portal; this requires a **`user role`**.

To manage roles, log into the RHEV-M web interface and click the **Configure** link at the top of the screen. This will bring up the following dialog:

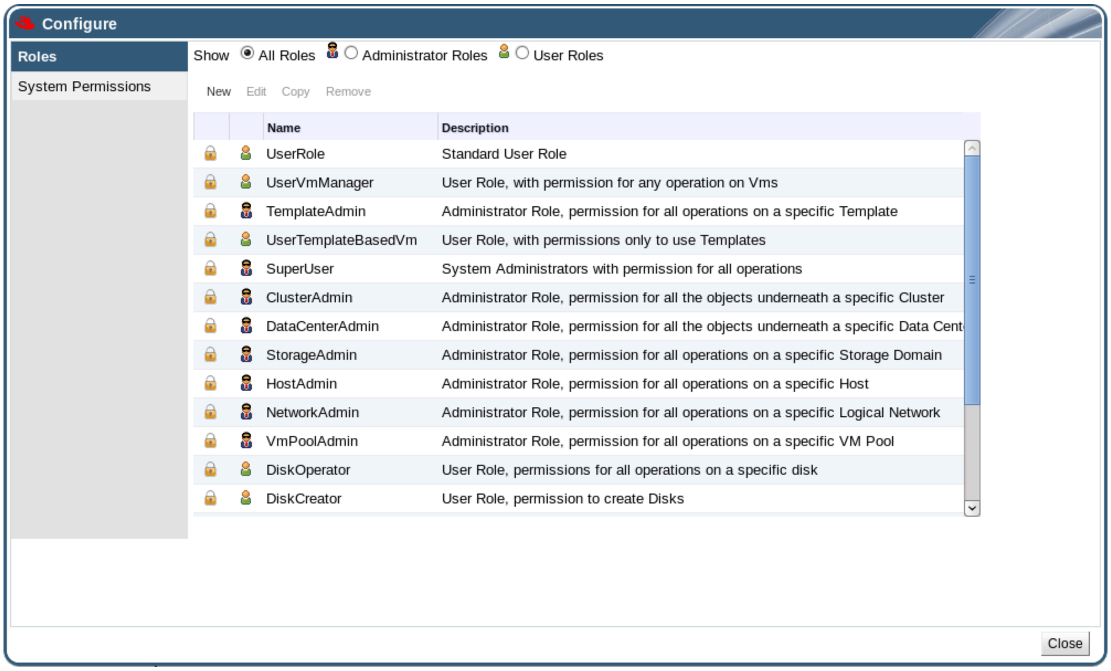


*Figure 2.5: The Configure dialog*

Roles that appear with a padlock in front of their name are locked (system roles) and cannot be edited or removed.

To add a new role, click the **New** button. This will bring up the **New Role** dialog as follows.

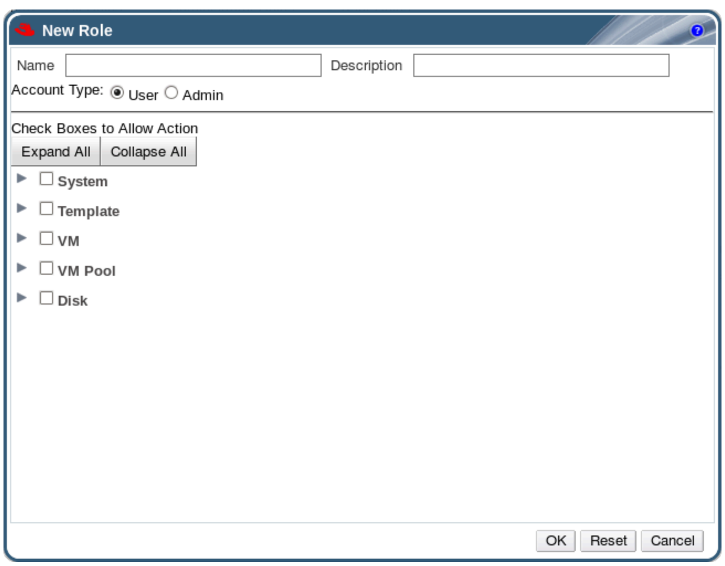


*Figure 2.6: The New Role dialog*

Enter a **Name** and **Description** in this dialog for the new role, and select whether the new role should be a **User** role or an **Admin** role. In the bottom pane, it is possible to select which actions should be allowed inside this role.

To edit an existing role, select the role in the **Configure** dialog and click the **Edit** button. This will bring up a window identical to the **New Role** dialog, except the **Account Type** selection is disabled.

# Adding systemwide users

To add a systemwide user, click the **Configure** link at the top of the RHEV-M web interface. Navigate to the **System Permissions** tab in the **Configure** dialog that appears. Click the **Add** button to add a new user. A new dialog called **Add System Permission to User** will appear.

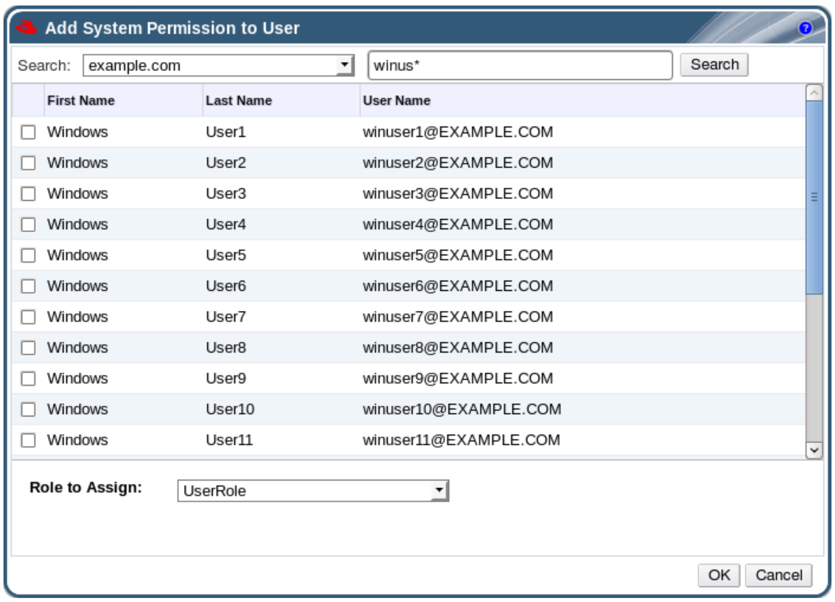


*Figure 2.7: The Add System Permission to User dialog*

In this dialog, select the domain to **Search** and enter the user or group name wanted, using wildcards if needed. Click the **GO** button to perform the search.

### Note

Administrators can search for * to get an overview of all available users and groups in a domain, but this list can be *very* long on some domains.

Check the box in front of all the users and groups to add, and select the intended role from the **Assign role to user** dropdown menu. Click **OK** to add the selected users with the selected role to the System object.

### Important

Users and groups added to the System object (**System Permissions** tab pictured previously) will have access (within their role) to the entire Red Hat Enterprise Virtualization infrastructure. To set more detailed permissions/restrictions, use the **Permissions** tab on individual objects.

# Directory users

RHEV needs to be connected to at least one directory server in order to add new users, and during the initial installation, the user **admin@internal** is created. This account is intended for use when initially configuring the environment, and/or for troubleshooting. Users will be identified by their **User Principal Name** (**UPN**) of the form **user@domain**. Attachment of more than one directory server to the RHEV-M is also possible and supported. In case administrators have more than one directory server attached, they will be able to choose against which one they want to authenticate by selecting the right domain from the pulldown menu.

# Managing multilevel administration

To set fine-grained permissions on objects in Red Hat Enterprise Virtualization, navigate to an object in the RHEV-M web interface and select it. In the lower pane, navigate to the **Permissions** tab. From there, administrators can see which users have what access level to the object, and whether these permissions were set directly or inherited from a parent in the tree.

To add a new user, click the **Add** button in the **Permissions** tab and use the **Add Permission to User** dialog that appears to search for the desired user and assign this user a role. It is also possible to assign permissions to **Everyone**, although this is not typically desired.

> **Note**
>
> The same user or group can have multiple roles on an object. In this case, the permissions will be added to each other.

For a detailed view of who has what kind of access to the system, navigate to the **Users** tab in the RHEV-M web interface. This screen will give an overview of all the users that are currently recognized by the system. For a detailed view of the permissions for a user, select a user in this list and navigate to the **Permissions** tab in the lower pane. This pane can also be used to revoke (**Remove**) permissions from a user.

# Demonstration: Managing users, roles, and permissions

In this example, observe the following steps as the instructor demonstrates how to manage users, roles, and permission on rhevm.pod*X*.example.com.

1. Join the **example.com** domain:

```
[root@rhevm ~]# engine-manage-domains add --domain=example.com --user=rhevadmin --
provider=IPA
Enter password: redhat
The domain example.com has been added to the engine as an authentication source but
 no users from that domain have been granted permissions within the oVirt Manager.
Users from this domain can be granted permissions by editing the domain using action
 edit and specifying --add-permissions or from the Web administration interface
 logging in as admin@internal user.
oVirt Engine restart is required in order for the changes to take place (service
 ovirt-engine restart).
Manage Domains completed successfully
```

   Use **redhat** when prompted for a password. After this command has finished, restart RHEV-M to activate the changes:

```
[root@rhevm ~]# service ovirt-engine restart
```

2. To make sure the RHEV instance is bound to the **example.com** domain, run:

```
[root@rhevm ~]# engine-manage-domains list
Domain: example.com
        User: rhevadmin@EXAMPLE.COM
Manage Domains completed successfully
```

3. Give the user **rhevadmin** the **SuperUser** permission.

   Log into the RHEV-M Administration Portal as user **admin** with a password of **redhat**. In the top bar, select **Configure**, then select the **System Permissions** tab, then click the **Add** button to create a new user permission.

   Select the **example.com (example.com)** domain, then type **rhevadmin** in the search field. Once the user appears in the search field, check it.

   From the dropdown list, select the **SuperUser** role. Click **OK** to set the user permission.

4. Test the new **SuperUser** by logging in as **rhevadmin**.

   From the top-right corner, click the user example.com, then **Sign Out** to log out. Log back in, using **rhevadmin** as the username and **redhat** as the password. Select **example.com** as the domain.

5. Create a new role that allows basic operations and remote logins on virtual machines, but nothing else.

   From the top-right corner, click **Configure**. In the top bar, from the **Roles** tab, click **New** to define a new role.

   Name the new role **VMUserNoCD** and give it a description of **use VMs, no CD**. Keep the **Account Type** set to **User**. Expand the **VM** tree and expand the **Basic Operations** tree. Make sure only **Basic Operations** and **Remote Log In** are checked.

   Click **OK** to confirm the addition of this new role. You will add this permission to an object in a later step.

6. Create an additional user and give it appropriate permissions.

   With the **Configure** dialog still open, navigate to the **System Permissions** tab. Click **Add** to add a new system user.

   Make sure **Search** is set to **example.com** and enter **vmadmin*X*** as a search term, where *X* is your pod number. In the results window, place a check in front of **vmadmin*X***. Select **PowerUserRole** from the **Assign role to user** dropdown menu. Click **OK** to confirm the addition.

7. Create an additional user and give it appropriate permissions.

   Click the **System Permissions** tab, then click **Add** to add a new user. Search the **example.com** domain for **rheluser*X*** (make sure to replace X by your station number), and assign this account the earlier created **VMUserNoCD** role. Do not forget to place a checkmark in front of **rheluser*X*** before clicking **OK**.

8. On **workstation.podX.example.com**, install the *spice-xpi* package.

```
[root@workstation ~]# yum -y install spice-xpi
```

9. If you have **Firefox** open on **workstation.podX.example.com**, close it.

10. Start a new **Firefox** instance on **workstation.podX.example.com** and navigate to **https://rhevm.pod*X*.example.com**. Click **User Portal** to navigate to the RHEV-M User Portal.

    On the error page presented by **Firefox**:

    - Click **I Understand the Risks** to expand the certificate exception dialog in Firefox.
    - Select **Add Exception...**.
    - In the resulting pop-up, click the **Get Certificate** button.
    - Select **Confirm Security Exception** at the bottom of the pop-up window.

11. Make sure you can log in with the newly defined role.

    Log in as the user **vmadmin*X***, using **redhat** for the password. Make sure to choose example.com as the domain.

# The User Portal

In addition to the *Administration Portal* that has been used so far, RHEV-M also provides a *User Portal*. The User Portal is where normal users connect to manage and use their virtual machines. The User Portal is available at **https://rhevm.pod*X*.example.com/ovirt-engine/userportal**. The first time users connect to the User Portal, they will either have to make a security exception in the browser to trust the HTTPS certificate, or they will have to install the RHEV-M CA certificate from **http://rhevm.pod*X*.example.com/ca.crt** as a trusted CA certificate in the browser.

In order to use the *SPICE* protocol to connect to the console of virtual machines, a browser extension must be installed. On Red Hat Enterprise Linux clients with the **Firefox** browser, this can be done by installing the *spice-xpi* package. After installation of the *spice-xpi* package, restart **Firefox** to activate the new extensions. Windows clients running Internet Explorer will be asked to install the SPICE **ActiveX** extensions the first time they try to connect to a console. Follow the on-screen instructions to install the client.

# Using virtual desktops

When users connect to the User Portal, a login screen like the following will appear. Enter the **User Name**, **Password**, and **Domain**. Do not check **Connect Automatically** unless only one virtual machine is tied to the user account and it should always connect to the console of that virtual machine immediately.

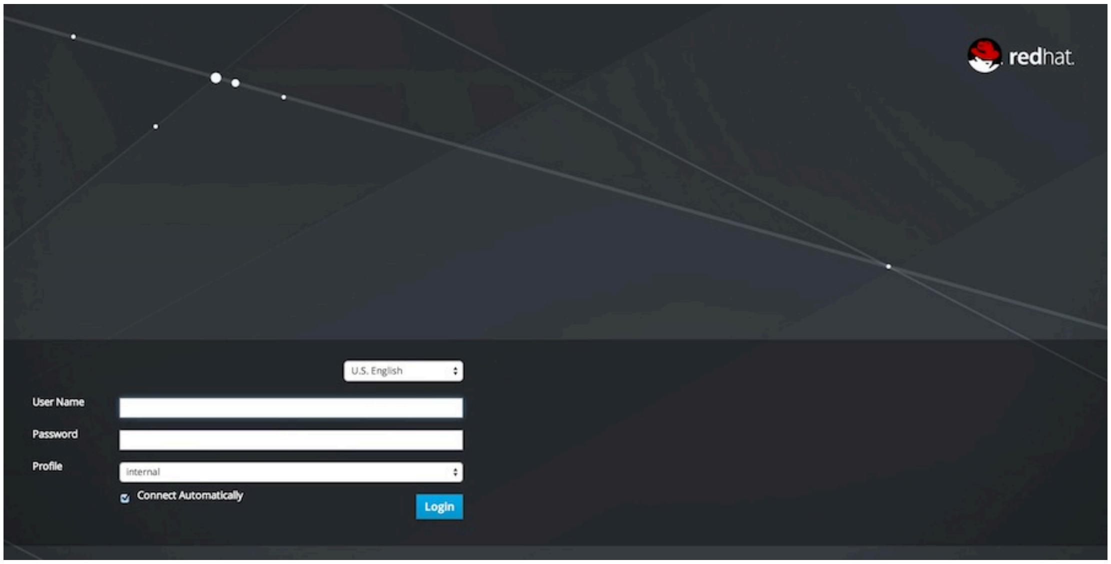


*Figure 2.8: The User Portal Login screen*

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 10. Pools
- Section 12. Users and Roles

Red Hat Enterprise Virtualization 3.5 User Portal Guide

Red Hat Enterprise Virtualization 3.5 Power User Portal Guide

# Practice: Managing Users, Roles, and Permissions

In this lab, you will create three new users and set their permissions accordingly.

**Outcome(s)**
You should be able to create additional users and set appropriate permissions.

Perform the following steps:

1. Run the following commands to join the **example.com** domain:

```
[root@rhevm ~]# engine-manage-domains add --domain=example.com --user=rhevadmin --
provider=IPA
Enter password: redhat
The domain example.com has been added to the engine as an authentication source but
 no users from that domain have been granted permissions within the oVirt Manager.
Users from this domain can be granted permissions by editing the domain using action
 edit and specifying --add-permissions or from the Web administration interface
 logging in as admin@internal user.
oVirt Engine restart is required in order for the changes to take place (service
 ovirt-engine restart).
Manage Domains completed successfully
```

   Use **redhat** when prompted for a password. After this command has finished, restart RHEV-M to activate the changes:

```
[root@rhevm ~]# service ovirt-engine restart
```

2. Make sure your RHEV instance is bound to the **example.com** domain by running:

```
[root@rhevm ~]# engine-manage-domains list
Domain: example.com
        User: rhevadmin@EXAMPLE.COM
Manage Domains completed successfully
```

3. Give the user **rhevadmin** the **SuperUser** permission.

   Log into the RHEV-M Administration Portal as user **admin** with a password of **redhat**. In the top bar, select **Configure**, then select the **System Permissions** tab, then click the **Add** button to create a new user permission.

   Select the **example.com (example.com)** domain, then type **rhevadmin** in the search field. Once the user appears in the search field, check it.

   From the dropdown list, select the **SuperUser** role. Click **OK** to set the user permission.

4. Test the new **SuperUser** by logging in as **rhevadmin**.

   From the top-right corner, click the user example.com, then **Sign Out** to log out. Log back in, using **rhevadmin** as the username and **redhat** as the password. Select **example.com** as the domain.

5. Create a new role that allows basic operations and remote logins on virtual machines, but nothing else.

   From the top-right corner, click **Configure**. In the top bar, from the **Roles** tab, click **New** to define a new role.

   Name the new role **`VMUserNoCD`** and give it a description of **`use VMs, no CD`**. Keep the **Account Type** set to **`User`**. Expand the **VM** tree and expand the **Basic Operations** tree. Make sure only **Basic Operations** and **Remote Log In** are checked.

   Click **OK** to confirm the addition of this new role. You will add this permission to an object in a later step.

6. Create an additional user and give it appropriate permissions.

   With the **Configure** dialog still open, navigate to the **System Permissions** tab. Click **Add** to add a new system user.

   Make sure **Search** is set to **`example.com`** and enter **`vmadminX`** as a search term, where *X* is your pod number. In the results window, place a check in front of **`vmadminX`**. Select **`PowerUserRole`** from the **Assign role to user** dropdown menu. Click **OK** to confirm the addition.

7. Create an additional user and give it appropriate permissions.

   Click the **System Permissions** tab, then click **Add** to add a new user. Search the **`example.com`** domain for **`rheluserX`** (make sure to replace X by your station number), and assign this account the earlier created **`VMUserNoCD`** role. Do not forget to place a checkmark in front of **`rheluserX`** before clicking **OK**.

8. On **`workstation.podX.example.com`**, install the *spice-xpi* package.

```
[root@workstation ~]# yum -y install spice-xpi
```

9. If you have **Firefox** open on **`workstation.podX.example.com`**, close it.

10. Start a new **Firefox** instance on **`workstation.podX.example.com`** and navigate to **`https://rhevm.podX.example.com`**. Click **User Portal** to navigate to the RHEV-M User Portal.

    On the error page presented by **Firefox**:

    - Click **I Understand the Risks** to expand the certificate exception dialog in Firefox.
    - Select **Add Exception...**.
    - In the resulting pop-up, click the **Get Certificate** button.
    - Select **Confirm Security Exception** at the bottom of the pop-up window.

    You should be automatically redirected to the RHEV **User Portal**.

11. Make sure you can log in with the newly defined role.

Log in as the user **vmadmin*X***, using **redhat** for the password. Make sure to choose example.com as the domain.

# Troubleshooting RHEV Manager

## Objectives

After completing this section, students should be able to:

- Locate recommended firewall changes.
- Identify the importance of synchronized time.
- Reset the **admin@internal** password.
- View or set RHEV configuration values with **engine-config**.
- Read or collect RHEV log files.
- Remove RHEV Manager.

## Firewall configuration

During the previous installation, the **engine-setup** command offered to automatically configure the firewall to allow communication. An example of some lines that might be used with **iptables** was deposited in **/etc/ovirt-engine/**.

```
[root@rhevm ~]# cat /etc/ovirt-engine/iptables.example
# Generated by ovirt-engine installer
#filtering rules
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 111 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 111 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 662 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 662 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 875 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 875 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 892 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 892 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2049 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 32769 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 32803 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 5432 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 7410 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 6100 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT

#drop all rule
-A INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Review the documentation for additional pointers on ports that need to be available or services that need to be modified (like NFS).

## Confirming the system time

When RHEV-M is installed, a certification authority certificate is created that signs the host certificate used by the web interface presented by RHEV-M. Problems using SSL encryption can surface if the system clock of the Red Hat Enterprise Linux server used to host RHEV-M is not accurate when RHEV-M is installed. Adjusting the time zone is not sufficient. Ensure the system clock and hardware clock are set to the correct time.

### Important

The **ntpdate** will not work when the NTP server is running on the local host. In this case, use the **service** command to stop the **ntpd** service, synchronize with an accurate time server, then restart the **ntpd** service.

If necessary, use the **ntpdate** command to synchronize with an NTP time server. The **hwclock** command displays the time stored in the BIOS real-time clock. When the **-w** option is specified, **hwclock** persistently writes the current system time into the server's real-time clock.

```
[root@rhevm ~]# service ntpd stop
Shutting down ntpd:                                        [  OK  ]
[root@rhevm ~]# ntpdate classroom.example.com
16 Dec 23:41:17 ntpdate[30767]: adjust time server 172.25.254.254 offset 0.000541 sec
[root@rhevm ~]# hwclock -w
[root@rhevm ~]# hwclock
Tue 16 Jan 2014 11:41:30 PM PST  -0.937889 seconds
[root@rhevm ~]# service ntpd start
```

## Internal administrative account

All RHEV-M administration functions must be performed by an administrative account. Usually RHEV-M authentication is provided by an external Windows Active Directory server or an LDAP/IPA server.

An internal administrative account called **admin@internal** should be used when successful external authentication is in doubt. If the password for this account is lost or has been changed, administrative access can be recovered by resetting the password.

To reset the password for the internal administrative account, take the following steps:

1. Log in as **root** on the RHEV-M Linux server.

2. Use the **engine-config -s AdminPassword=interactive** command to reset the internal administrative password. Enter the new password when prompted:

```
[root@rhevm ~]# engine-config -s AdminPassword=interactive
Please enter a password: New password
Please reenter password: New password
```

3. Issue the **service ovirt-engine restart** command to make the change take effect.

# Demonstration: System time and resetting the administrative account password

In this example, observe the following steps as the instructor demonstrates how to synchronize the server clock, as well as resetting the administrative account password.

1. To set the system clock, use **ntpdate** and **hwclock**. Notice that **ntpd** should be configured as well, but that it cannot be running while the **ntpdate** command is issued.

   On rhevm.pod*X*.example.com, ensure the system clock is properly set. In class, use classroom.example.com as the NTP server.

```
[root@rhevm ~]# service ntpd stop
[root@rhevm ~]# ntpdate classroom.example.com
[root@rhevm ~]# grep server /etc/ntp.conf
...
server 172.25.254.254
...
[root@rhevm ~]# service ntpd start
[root@rhevm ~]# hwclock --systohc
```

2. Demonstrate how to set the **admin@internal** password as **root** on your **rhevm.pod*X*.example.com** machine, then reset the **admin@internal** password to **test123**:

```
[root@rhevm ~]# engine-config -s AdminPassword=interactive
Please enter a password: test123
Please reenter password: test123
[root@rhevm ~]# service ovirt-engine restart
Stopping engine-service:                                   [  OK  ]
Starting engine-service:                                   [  OK  ]
```

3. Log out of the RHEV-M web interface and log in using the new password.

> **Note**
>
> When you restart the **ovirt-engine** service, it will take a few minutes for the web interface to allow logins.

4. Reset the **admin@internal** password back to **redhat**.

```
[root@rhevm ~]# engine-config -s AdminPassword=interactive
Please enter a password: redhat
Please reenter password: redhat
[root@rhevm ~]# service ovirt-engine restart
Stopping engine-service:                                   [  OK  ]
Starting engine-service:                                   [  OK  ]
```

## Retrieving RHEV configuration values with RHEV configuration tool

The **engine-config** tool allows administrators to update values for the RHEV installation. For example, passwords can be reset, or configuration settings updated. In order to update a value, the **-s** flag needs to be used. Consider the following example that sets a user **custom value**:

```
[root@rhevm ~]# engine-config -s "UserDefinedVMProperties=macspoof=(true|false)"
```

The previous command creates a new custom value users will be able to use when working with their virtual machines. Administrators can run **engine-config -a** to retrieve all existing values, and update them if necessary:

```
[root@rhevm ~]# engine-config -a
AbortMigrationOnError: false version: 3.0
AbortMigrationOnError: false version: 3.1
AbortMigrationOnError: false version: 3.2
AbortMigrationOnError: false version: 3.3
AsyncTaskPollingRate: 10 version: general
AsyncTaskZombieTaskLifeInMinutes: 3000 version: general
AuditLogAgingThreshold: 30 version: general
...
ClusterRequiredRngSourcesDefault:  version: 3.2
ClusterRequiredRngSourcesDefault:  version: 3.3
ClusterRequiredRngSourcesDefault:  version: 3.4
ClusterRequiredRngSourcesDefault:  version: 3.5
DefaultMTU: 1500 version: general
```

### Warning

It is not advisable to update the engine values without making sure how they work, as this could prevent RHEV from working.

The long options can be abbreviated as single-letter options: **-l**, **-a**, **-g**, and **-s** respectively. Multiple versions of RHEV-M's configuration can be managed with the configuration tool. The **--cver** option specifies the configuration version to be used when retrieving or setting a value for a configuration key. The default configuration version is **general**.

Configuration values are stored in the RHEV-M database. Changes are not saved unless the database is operational and they are applied when JBoss is restarted. Restart the **ovirt-engine** service after any values are modified for the changes to go into effect.

The **-g** flag can be used to retrieve the value for a particular key:

```
[root@rhevm ~]# engine-config -g "AuditLogCleanupTime"
AuditLogCleanupTime: 03:35:35 version: general
```

When working with the **engine-config** tool, it is possible to append the **--log-file** flag in order to save queries in a file, useful for debugging purposes. Instead of manually typing keys and values to update, it is possible to pass a configuration file as an argument:

```
[root@rhevm ~]# engine-config -s PasswordEntry --admin-pass-file=/tmp/mypass
```

# Demonstration: Updating RHEV configuration values with RHEV configuration tool

In this example, observe the following steps as the instructor demonstrates how to use the RHEV configuration tool to update a RHEV configuration value.

1. Display the list of tunable RHEV-M values. Look for values that might determine user session timeouts:

```
[root@rhevm ~]# engine-config -l | grep -i timeout
... output omitted ...
UserSessionTimeOutInterval: Timeout interval in minutes, after which inactive user
 sessions expire. A negative value indicates that sessions never expire. (Value
 Type: Integer)... output omitted ...
```

2. Get the current value of the **UserSessionTimeOutInterval** setting:

```
[root@rhevm ~]# engine-config -g UserSessionTimeOutInterval
UserSessionTimeOutInterval: 30 version: general
```

   **30** is the number of minutes of inactivity before the RHEV-M session times out. **general** is the RHEV version to which this applies (2.2, 3.0, 3.1, 3.5).

3. Change the value of the **UserSessionTimeOutInterval** setting to timeout after two minuts of inactivity:

```
[root@rhevm ~]# engine-config -s UserSessionTimeOutInterval=2
[root@rhevm ~]# engine-config -g UserSessionTimeOutInterval
UserSessionTimeOutInterval: 1 version: general
```

4. Restart the **ovirt-engine** service so the changes are applied:

```
[root@rhevm ~]# service ovirt-engine restart
```

5. Wait for the web interface to become available and log into the RHEV-M Administrative Portal as **rhevadmin** with a password of **redhat** in the **example.com** domain. Wait at least two minutes without manipulating the keyboard or mouse to watch the session time out.

6. Adjust the value of the **UserSessionTimeOutInterval** setting to timeout after one hour of inactivity:

```
[root@rhevm ~]# engine-config -s UserSessionTimeOutInterval=60
[root@rhevm ~]# service ovirt-engine restart
```

# Reading RHEV log files

For debugging and troubleshooting purposes, administrators can work with RHEV log files, all located under **/var/log/**. There are generally three directories:

- **ovirt-engine** contains logs for the engine activity, such as operations or system-related tasks.

- **`ovirt-engine-dwh`** contains logs for the data warehouse. The directory contains logs related to the RHEV data warehouse module.
- **`ovirt-engine-reports`** contains logs for the reporting module.

Log lines have the following format: **`DATE TIME LOG_LEVEL MODULE (ThreadID/TaskID ) MESSAGE`** While working with log files can be tedious, administrators can use **`grep`** to look for specific errors that occurred for a specific module. For example, if a user is unable to log in, the **`server.log`** file is likely to contain the cause:

```
[root@rhevm ~]# grep ERROR /var/log/ovirt-engine/engine.log
2014-12-16 11:18:29,594 ERROR
 [org.ovirt.engine.core.dal.dbbroker.auditloghandling.AuditLogDirector]
 (ajp-/127.0.0.1:8702-3) Correlation ID: null, Call Stack: null, Custom Event ID: -1,
 Message: User rhevadmin cannot login, please verify the username and password.
```

For troubleshooting purposes, it is also possible to append the **`--log-file`** flag to output messages in a specific file, making it easier to isolate to root cause errors that could have arisen during the setup phase:

```
[root@rhevm ~]# grep ERROR /root/debug.log
...
Error:  exception message: Integrity check on decrypted field failed (31) -
 PREAUTH_FAILED
```

If the current **log level** doesn't catch the error, it is possible to adjust it using the **`--log-level`** flag, then run again the command. Allowed values include:

- DEBUG (default log level)
- INFO
- WARN
- ERROR

While **debug** can help to troubleshoot deeper, **`error`** can be used as a first level of filtering. Administrators would use the **`warn`** level if they want to make sure the installation is fully working. **Tracebacks** are errors the Python module handled. It is a stack trace, going from the point of an exception handler, to where the exception was raised. This entries allow administrators to quickly see why the error has been triggered. The following example shows the socket does not exist, probably indicating the service is not running.

```
MainThread::ERROR::2011-12-05 07:04:56,300::vdsm::74::vds::(run) Traceback (most
recent call last):
   File "/usr/share/vdsm/vdsm", line 72, in run
       serve_clients(log)
   File "/usr/share/vdsm/vdsm", line 40, in serve_clients
       cif = clientIF.clientIF(log)
   File "/usr/share/vdsm/clientIF.py", line 111, in init
       self._libvirt = libvirtconnection.get()
   File "/usr/share/vdsm/libvirtconnection.py", line 111, in get
       conn = libvirt.openAuth('qemu:///system', auth, 0)
   File "/usr/lib64/python2.6/site-packages/libvirt.py", line 102, in openAuth
       if ret is None:raise libvirtError('virConnectOpenAuth() failed')
   libvirtError: Failed to connect socket to '/var/run/libvirt/libvirt-sock': No
```

```
such file or directory
```

The **execCmd** can also trigger errors related to an external command that has been invoked. When errors occur, administrators would look at system logs to understand why the error occurred.

# Using log collector

Red Hat Enterprise Virtualization Manager provides a utility that collects all the logs and stores them in an archive: **engine-log-collector**. This command is part of the *engine-log-collector* package. All the system information, logs, and database backup files are placed into a single compressed archive. To start **Log Collector** on the server running RHEV-M, log in as **root** and execute the **engine-log-collector** command. The command output will be saved in the **/tmp/logcollector/** directory.

The **engine-log-collector** command has two subcommands, **list** and **collect**. The syntax for the two forms of the command is given as follows:

```
engine-log-collector [options] list [all, clusters, datacenters]
engine-log-collector [options] collect
```

The options can be used to specify selection parameters and/or authentication credentials. Options can be persistently specified by making changes to the **/etc/rhevm/logcollector.conf** file. The original file contains comment lines with sample variable settings. A couple useful parameters are **rhevm**, which points to the RHEV Manager host to query, and **user**, which specifies the RHEV user to authenticate as.

```
[root@rhevm ~]# engine-log-collector list
Please provide the REST API password for the admin@internal RHEV-M user (CTRL+D to
 skip): redhat
Host list (datacenter=None, cluster=None, host=None):
Data Center          | Cluster              | Hostname/IP Address
dcenter1             | cluster1             | 172.25.X.15
[root@rhevm ~]# engine-log-collector collect
Please provide the REST API password for the admin@internal RHEV-M user (CTRL+D to
 skip): redhat
About to collect information from 1 hypervisors. Continue? (Y/n): Y
INFO: Gathering information from selected hypervisors...
INFO: collecting information from 172.24.1.201
INFO: finished collecting information from 172.24.1.201
Please provide the password for the PostgreSQL user, postgres, to dump the engine
 PostgreSQL database instance (CTRL+D to skip): redhat
INFO: Gathering PostgreSQL the RHEV-M database and log files from localhost...
INFO: Gathering RHEV-M information...
INFO: Log files have been collected and placed in /tmp/logcollector/sosreport-
LogCollector-rhevm-20121129060015-7050.tar.xz.
      The MD5 for this file is dc38ba4dddd051f1309fd8625a6b7050 and its size is 35.0M
```

Sometimes it is useful to review individual log files on a RHEV Manager server. The following table lists the most useful log files:

**Red Hat Enterprise Virtualization Manager Logs**

| Log file | Description |
|---|---|
| **/var/log/ovirt-engine/engine-setup-*.log** | Provides information about the RHEV Manager installation and configuration process. |

| Log file | Description |
|---|---|
| **/var/log/ovirt-engine/rhevm-dwh-setup-*.log** | Contains information about the installation and configuration of the RHEV Manager data warehouse **rhevm-history** database. |
| **/var/log/ovirt-engine/ovirt-engine-reports-setup-*.log** | Log from the command used to install the RHEV Manager Reports modules. |
| **/var/log/ovirt-engine/engine.log** | Reflects all RHEV-Manager GUI crashes, directory lookups, database and other issues. |

## Removing RHEV Manager

Removing RHEV Manager software from a server requires four steps:

1. Shut down RHEV-M and remove its configuration.
2. Remove the RHEV Manager and JBoss software packages.
3. Remove all **ovirt-engine** directories and remove PostgreSQL databases.
4. Clean up any NFS exports created during the RHEV-M installation.

Examine these steps in more detail as follows.

1. First, log in as **root** on the RHEV-M server and run the **engine-cleanup** command to shut down RHEV-M and remove its configuration. This command will destroy data, so execute it only when removing RHEV-M from a server:

```
[root@rhevm ~]# engine-cleanup
[ INFO  ] Stage: Initializing
[ INFO  ] Stage: Environment setup
          Configuration files: ['/etc/ovirt-engine-setup.conf.d/10-packaging-
dwh.conf', '/etc/ovirt-engine-setup.conf.d/10-packaging-wsp.conf', '/etc/ovirt-
engine-setup.conf.d/10-packaging.conf', '/etc/ovirt-engine-setup.conf.d/20-
packaging-rhevm-reports.conf', '/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-
post.conf']
          Log file: /var/log/ovirt-engine/setup/ovirt-engine-remove-20141216123409-
fxre6z.log
          Version: otopi-1.3.0 (otopi-1.3.0-1.el6ev)
[ INFO  ] Stage: Environment packages setup
[ INFO  ] Stage: Programs detection
[ INFO  ] Stage: Environment customization
          Do you want to remove all components? (Yes, No) [Yes]: yes

          --== PRODUCT OPTIONS ==--

          Do you want to remove Engine database content? All data will be lost (Yes,
 No) [No]: yes
[ INFO  ] Stage: Setup validation
          During execution engine service will be stopped (OK, Cancel) [OK]: ok
          All the installed ovirt components are about to be removed, data will be
 lost (OK, Cancel) [Cancel]: ok
[ INFO  ] Stage: Transaction setup
[ INFO  ] Stopping dwh service
[ INFO  ] Stopping reports service
[ INFO  ] Stopping engine service
[ INFO  ] Stopping ovirt-fence-kdump-listener service
[ INFO  ] Stopping websocket-proxy service
[ INFO  ] Stage: Misc configuration
```

```
[ INFO  ] Stage: Package installation
[ INFO  ] Stage: Misc configuration
[ INFO  ] Backing up PKI configuration and keys
[ INFO  ] Backing up database localhost:engine to '/var/lib/ovirt-engine/backups/
engine-20141216123457.ImPi6v.dump'.
[ INFO  ] Clearing Engine database engine
[ INFO  ] Backing up database localhost:ovirt_engine_history to '/var/lib/ovirt-
engine-dwh/backups/dwh-20141216123517.MG0DBs.dump'.
[ INFO  ] Clearing DWH database ovirt_engine_history
[ INFO  ] Backing up database localhost:ovirt_engine_reports to '/var/lib/ovirt-
engine-reports/backups/reports-20141216123523.XBUKrE.dump'.
[ INFO  ] Clearing Reports database ovirt_engine_reports
[ INFO  ] Removing files
[ INFO  ] Reverting changes to files
[ INFO  ] Stage: Transaction commit
[ INFO  ] Stage: Closing up

          --== SUMMARY ==--

          A backup of the Reports database is available at /var/lib/ovirt-engine-
reports/backups/reports-20141216123523.XBUKrE.dump
          A backup of the DWH database is available at /var/lib/ovirt-engine-dwh/
backups/dwh-20141216123517.MG0DBs.dump
          A backup of the Engine database is available at /var/lib/ovirt-engine/
backups/engine-20141216123457.ImPi6v.dump
          ovirt-engine has been removed
          A backup of PKI configuration and keys is available at /var/lib/ovirt-
engine/backups/engine-pki-20141216123457cVti4Q.tar.gz
          Engine setup successfully cleaned up

          --== END OF SUMMARY ==--

[ INFO  ] Stage: Clean up
          Log file is located at /var/log/ovirt-engine/setup/ovirt-engine-
remove-20141216123409-fxre6z.log
[ INFO  ] Generating answer file '/var/lib/ovirt-engine/setup/
answers/20141216123530-cleanup.conf'
[ INFO  ] Stage: Pre-termination
[ INFO  ] Stage: Termination
[ INFO  ] Execution of cleanup completed successfully
```

2. Once RHEV-M has been shut down, remove all relevant RPMs:

```
[root@rhevm ~]# yum remove rhevm* vdsm-bootstrap *jboss* postgresql-server
```

3. Remove the old **ovirt-engine** directories and remove any remaining PostgreSQL databases. This includes the following configuration and data directories:
   - **/etc/ovirt-engine/**
   - **/usr/share/ovirt-engine***
   - **/var/lib/pgsql/**

4. Finally, clean up any NFS shares created by the configuration tool. Remove the relevant entries in **/etc/exports** and reload the **nfs** service to reread its configuration:

```
[root@rhevm ~]# vim /etc/exports
[root@rhevm ~]# service nfs reload
```

Cleanup includes removing the NFS exported directory, especially if the same directory name will be reused if RHEV-M is reinstalled.

## References

Red Hat Enterprise Virtualization 3.5 Manager Release Notes

- Section 3. What's New?

Red Hat Enterprise Virtualization 3.5 Installation Guide

- Section 3. Manager Installation
- Section 3.7. Removing Red Hat Enterprise Virtualization Manager

# Practice: Updating RHEV Configuration Values with RHEV Configuration Tool

In this lab, you will adjust the Administration Portal timeout.

**Outcome(s)**
You should be able to adjust engine configuration settings.

**Before you begin**
You should have a server already installed with Red Hat Enterprise Linux 6.6, and successfully completed the installation of RHEV-M.

Perform the following steps:

1. Log in as **root** on your **rhevm.pod*X*.example.com** RHEV-M server.

2. Display the list of tunable RHEV-M values. Look for values that might determine user session timeouts:

```
[root@rhevm ~]# engine-config -l | grep -i timeout
... output omitted ...
UserSessionTimeOutInterval: Timeout interval in minutes, after which inactive user
 sessions expire. A negative value indicates that sessions never expire. (Value
 Type: Integer)... output omitted ...
```

3. Get the current value of the **UserSessionTimeOutInterval** setting:

```
[root@rhevm ~]# engine-config -g UserSessionTimeOutInterval
UserSessionTimeOutInterval: 30 version: general
```

   **30** is the number of minutes of inactivity before the RHEV-M session times out. **general** is the RHEV version to which this applies (2.2, 3.0, 3.1, 3.5).

4. Change the value of the **UserSessionTimeOutInterval** setting to timeout after two minuts of inactivity:

```
[root@rhevm ~]# engine-config -s UserSessionTimeOutInterval=2
[root@rhevm ~]# engine-config -g UserSessionTimeOutInterval
UserSessionTimeOutInterval: 1 version: general
```

5. Restart the **ovirt-engine** service so the changes are applied:

```
[root@rhevm ~]# service ovirt-engine restart
```

6. Wait for the web interface to become available and log into the RHEV-M Administrative Portal as **rhevadmin** with a password of **redhat** in the **example.com** domain. Wait at least two minutes without manipulating the keyboard or mouse to watch the session time out.

7. Adjust the value of the **UserSessionTimeOutInterval** setting to timeout after one hour of inactivity:

```
[root@rhevm ~]# engine-config -s UserSessionTimeOutInterval=60
[root@rhevm ~]# service ovirt-engine restart
```

# Lab: Installing, Configuring, and Testing RHEV Manager

In this lab, you will install and configure a RHEV Manager server, using the previously generated answer file, that will produce reports and bind to a directory service.

**Outcome(s)**

Once the installation and configuration is complete, you should be able to log in as the local administrator and as **rhevadmin** with a password of **redhat** using the Administration Portal on **workstation**.

**Before you begin**

1. Confirm you had previously backed up the answer file to the home directory of student@workstation.podX.example.com.

```
[student@workstation ~]$ ls answers.txt
answers.txt
```

2. Reset your **rhevm** system to its initial state.

3. While your **rhevm** is being reset, remove the certificate from Firefox that was imported in the earlier workshop (**Menu** > **Preferences**, **Advanced** tab, **Certificates** subtab, **View Certificates** button, **Servers** Tab, scroll down to **pod*X*.example.com**, highlight the **rhevm.pod*X*.example.com** certificate, click **Delete...** button, click **OK**). Also, clear the cache in the **Advanced** tab by going to the **Network** subtab. Under the **Offline Storage** header, click the **Clear Now** button.

**Lab outline**

Install RHEV Manager software on your **rhevm** server. Configure it so that it conforms to the following specifications:

- Use the answer file that has been generated previously.
- Bind to the **example.com** domain for additional RHEV users. You will need to authenticate with the domain server as **rhevadmin** with a password of **redhat**.
- Once you have bound to the **example.com** domain, add **rhevadmin** as a **SuperUser** in RHEV-M.
- Finally, adjust the timeout of the Administration Portal to 45 minutes.

1. Upgrade all existing Red Hat Enterprise Linux packages and reboot if necessary.

2. To install the packages needed to turn this system into a Red Hat Enterprise Virtualization Manager, you need to subscribe it to a new RHN channel or **yum** repository. In the classroom, we have a custom **yum** repository. To activate this repository, download the file **rhevm.repo** from **http://classroom.example.com/materials/rhevm.repo** and place it in the directory **/etc/yum.repos.d/**.

3. Install the **rhevm**, **rhevm-dwh**, and **rhevm-reports** packages.

4. Run the engine installation, using the answer file generated in the previous practice.

5. Bind to the **example.com** domain. Authenticate as user **rhevadmin** with a password of **redhat**.

6. Add **rhevadmin** (password: **redhat**) as a **SuperUser** in RHEV-M.

7. Adjust the Administration Portal timeout value, to 45 minutes, using the RHEV configuration tool.

# Solution

In this lab, you will install and configure a RHEV Manager server, using the previously generated answer file, that will produce reports and bind to a directory service.

### Outcome(s)

Once the installation and configuration is complete, you should be able to log in as the local administrator and as **rhevadmin** with a password of **redhat** using the Administration Portal on **workstation**.

### Before you begin

1. Confirm you had previously backed up the answer file to the home directory of student@workstation.podX.example.com.

```
[student@workstation ~]$ ls answers.txt
answers.txt
```

2. Reset your **rhevm** system to its initial state.

3. While your **rhevm** is being reset, remove the certificate from Firefox that was imported in the earlier workshop (**Menu** > **Preferences**, **Advanced** tab, **Certificates** subtab, **View Certificates** button, **Servers** Tab, scroll down to **pod*X*.example.com**, highlight the **rhevm.pod*X*.example.com** certificate, click **Delete...** button, click **OK**). Also, clear the cache in the **Advanced** tab by going to the **Network** subtab. Under the **Offline Storage** header, click the **Clear Now** button.

### Lab outline

Install RHEV Manager software on your **rhevm** server. Configure it so that it conforms to the following specifications:

- Use the answer file that has been generated previously.
- Bind to the **example.com** domain for additional RHEV users. You will need to authenticate with the domain server as **rhevadmin** with a password of **redhat**.
- Once you have bound to the **example.com** domain, add **rhevadmin** as a **SuperUser** in RHEV-M.
- Finally, adjust the timeout of the Administration Portal to 45 minutes.

1. Upgrade all existing Red Hat Enterprise Linux packages and reboot if necessary.

```
[root@rhevm ~]# yum -y update
...
```

2. To install the packages needed to turn this system into a Red Hat Enterprise Virtualization Manager, you need to subscribe it to a new RHN channel or **yum** repository. In the classroom, we have a custom **yum** repository. To activate this repository, download the file **rhevm.repo** from **http://classroom.example.com/materials/rhevm.repo** and place it in the directory **/etc/yum.repos.d/**.

```
[root@rhevm ~]# wget http://classroom.example.com/materials/rhevm.repo -P /etc/
yum.repos.d/
...
[root@rhevm ~]# yum repolist
...
```

3. Install the **rhevm**, **rhevm-dwh**, and **rhevm-reports** packages.

```
[root@rhevm ~]# yum -y install rhevm rhevm-dwh rhevm-reports
```

4. Run the engine installation, using the answer file generated in the previous practice.

```
[student@workstation ~]$ scp answers.txt root@rhevm.podX.example.com:
```

```
[root@rhevm ~]# engine-setup --config-append=/root/answers.txt
```

5. Bind to the **example.com** domain. Authenticate as user **rhevadmin** with a password of **redhat**.

```
[root@rhevm ~]# engine-manage-domains add --domain=example.com --user=rhevadmin --
provider=IPA
Enter password: redhat
```

```
[root@rhevm ~]# service ovirt-engine restart
```

6. Add **rhevadmin** (password: **redhat**) as a **SuperUser** in RHEV-M.

   On workstation, open Firefox and browse to **https://rhevm.podX.example.com**. Accept the self-signed certificate to connect to the web site. Click the **Web Admin Portal** link. Enter a username of **admin**, password of **redhat**, and choose the **internal** domain.

   Click the **Configure** link at the top of the page. Click the **System Permissions** link in the left pane. Click the **Add** button in the right pane. Search for **rhevadmin** in the **example.com** domain. Check the box next to the **rhevadmin** user that is found. In the **Role to Assign:** dropdown menu, choose **SuperUser**. Click the **OK** button, then the **Close** button.

   Log out as **admin** and log in as **rhevadmin** with a password of **redhat** in the **example.com** domain.

7. Adjust the Administration Portal timeout value, to 45 minutes, using the RHEV configuration tool.

```
[root@rhevm ~]# engine-config -l | grep -i timeout
... output omitted ...
UserSessionTimeOutInterval: Timeout interval in minutes, after which inactive user
 sessions expire. A negative value indicates that sessions never expire. (Value
 Type: Integer)... output omitted ...
[root@rhevm ~]# engine-config -s UserSessionTimeOutInterval=45
```

```
[root@rhevm ~]# service ovirt-engine restart
```

# Summary

**Installing and Testing RHEV Manager**

In this section, you learned:

- Subscription management and installing the required packages for RHEV is the first step before being able to initiate the RHEV interactive installation.
- The Administration Portal contains various elements to work with and manage RHEV.
- RHEV provides a specialized tool (**`rhevm-upgrade`**) for upgrading the engine.

**Managing Users, Roles, and Permissions**

In this section, you learned:

- Connect RHEV to an external directory service, like IPA, to provide users to RHEV.
- Access roles that group permissions can simplify security assignments to both users and user groups.
- Built-in permissions can be used to limit RHEV web interface access.

**Troubleshooting RHEV Manager**

In this section, you learned:

- Identify ports that must be available through a firewall.
- Use **`NTP`** to synchronize time on the RHEV server.
- The RHEV configuration tool can be used to reset the internal administrative account.
- How to read and interpret RHEV log files and RHEV events and to collect those log files for easier access.

## CHAPTER 3

# INSTALLING AND CONFIGURING RHEV HYPERVISOR HOSTS

| Overview | |
|---|---|
| **Goal** | Install, configure, upgrade, and troubleshoot RHEV Hypervisor Hosts. |
| **Objectives** | • Install, configure, and approve RHEV Hypervisor.<br>• Automate RHEV Hypervisor installation and configuration.<br>• Upgrade RHEV Hypervisor software from a RHEV Manager server.<br>• Troubleshoot RHEV Hypervisor installation and runtime problems. |
| **Sections** | • Installing RHEV Hypervisor (and Practice)<br>• Automating RHEV Hypervisor Installation (and Quiz)<br>• Upgrading RHEV Hypervisor (and Practice)<br>• Troubleshooting RHEV Hypervisor (and Practice) |
| **Lab** | • Installing and Configuring RHEV Hypervisor Hosts |

# Installing RHEV Hypervisor

## Objectives

After completing this section, students should be able to:

- Install, configure, and approve RHEV Hypervisor.

## Preparations

Before installing Red Hat Enterprise Virtualization Hypervisor (RHEV-H), confirm the hardware of the host platform meets the specifications. The recommended hardware requirements are outlined as follows. Use these recommendations as a guide only.

Recommended hardware requirements

- All virtualization host CPUs must be 64-bit and support the AMD-V or Intel VT-x hardware virtualization extensions. They must also support the No eXecute CPU flag. A maximum of 160 physical CPUs per virtualization host is currently supported.
- 2 GB of system RAM + additional RAM for each virtual guest. A maximum of 2 TB of RAM per virtualization host is currently supported.
- A minimum of 2 GB of locally accessible, writable disk space, though additional space is recommended for additional swap space.
- One network interface card (NIC) with bandwidth of at least 1 Gbps.

## Demonstration: Install RHEV hypervisor

In this example, observe the following steps as the instructor demonstrates how to install a RHEV hypervisor.

The installation process for the RHEV hypervisor is very simple, with only seven steps:

1. PXE-boot the servera.pod*X*.example.com machine, hit **`Escape`**, and type **`rhevh`** at the PXE **`boot:`** prompt.

   ### Note

   It takes a minute to begin the installation as the machine downloads the RHEV-H image. As you wait for the installation to commence, note how to get the installation media for RHEV-H:

   Creating RHEV-H install media (either a CD-ROM ISO or PXE boot over a network) is available as follows:

   - Install the *rhev-hypervisor6* or *rhev-hypervisor7* package to create boot media for RHEV-H. This will place a **`.iso`** file in **`/usr/share/rhev-hypervisor`**. The ISO can be burned to a CD with a tool like **`wodim`** (formerly **`cdrecord`**) or any other tool of your choosing.
   - The ISO can also be used for a bootable USB stick or PXE boot by using **`rhevh-iso-to-disk`** or **`rhevh-iso-to-pxeboot`** respectively from the *rhev-hypervisor{6,7}-tools* package.

2. Select **`Install Hypervisor 7.1-`*`version`*** and press **`Enter`**.


*Figure 3.1: RHEV-H installation*

3. Select the keyboard layout and press **`Enter`**.

4. Select the local internal hard drive as the boot device and select **<Continue>**.

## Note

Use the arrow keys and **`Tab`** / **`Shift`**+**`Tab`** to select the options and buttons.

5. Select the local internal hard drive as the installation device and select **<Continue>**.

   RHEV-H 7.1 includes the ability to control the sizing of partitions. Take the default values and select **<Continue>** then **<Confirm>**.

6. Use **`redhat`** in the **Password** and **Confirm Password** fields and select **<Install>**.

7. After the installation is complete, press **Reboot**.


*Figure 3.2: RHEV-H installation complete*

# Demonstration: Configure RHEV hypervisor

In this example, observe the following steps as the instructor demonstrates how to configure a RHEV hypervisor.

Once the RHEV hypervisor system reboots, a login prompt will be displayed. Now it is time for further configuration of the newly installed host.

1. Once the installation is complete and the RHEV-H machine has booted, log in as **`admin`** with the password you created earlier (**`redhat`**).


*Figure 3.3: RHEV-H login*

2. Note the different subscreens and their significance, while filling out the configuration information listed as follows:

   The Main menu will appear at the left, and is comprised of the following settings:

| **Subscreen** | |
|---|---|
| **Status** | Corresponds to the option that is selected initially. It displays the current state of the RHEV hypervisor's configuration. The **`Status`** submenu also provides options to lock the display, log out, reboot, or power off the RHEV-H host. |
| **Network** | The submenu allows the RHEV Hypervisor administrator to select and configure a network interface. Both DHCP and static network configuration is available. This submenu also configures the systemwide network settings such as host name, DNS server addresses, and NTP host addresses. Network configuration should be performed first and the interface activated so the RHEV Manager server can be contacted in the next configuration step.<br><br>`Hostname: servera.pod0.example.com`<br>`DNS Server 1: 172.25.254.254`<br>`NTP Server 1: 172.25.254.254`<br>`Interface eth0`<br>`IPv4 Settings` |

| Subscreen | |
|---|---|
| | `Bootprotocol: Static`<br>`IP Address: 172.25.0.10   Netmask: 255.255.255.0`<br>`Gateway:    172.25.0.254` |
| Security | Allows administrators to change the **admin** password and enable/disable SSH authentication with passwords.<br>`Enable SSH password authentication  X` |
| Keyboard | Allows administrators to switch the keyboard layout. |
| SNMP | Allows administrators to configure SNMP for the RHEV-H host. |
| Logging | Allows administrators to specify the maximum size for a log file, and to configure a remote **syslog** server. |
| Kernel Dump | Allows the configuration of **kdump** to collect memory dumps in the case of a kernel crash.<br>This can be useful if you are suffering from weird hypervisor crashes, but do not make any changes here. |
| Remote Storage | Administrators can set a **iSCSI Qualified Name** (**IQN**) for the iSCSI initiator on the RHEV-H node. The installation procedure automatically sets a unique name for you, but administrators might want to change this for easier management. |
| CIM | Allows administrators to configure the *Common Information Model* (CIM) for monitoring and managing virtual machines on the RHEV-H machine. |
| RHEV-M | This submenu prompts for the RHEV Manager host name and port. When these are specified and confirmed, the RHEV Manager host is contacted and the server certificate fingerprint is displayed. Accepting the certificate allows the RHEV Hypervisor host to register with the RHEV Manager server.<br><br>The fingerprint can be evaluated by looking at the original on the RHEV manager system.<br><br>`[root@rhevm ~]# openssl x509 -fingerprint -in /etc/pki/ovirt-engine/apache-ca-.pem -noout`<br><br>Administrators can enter **rhevm.pod*X*.example.com** in the **Management Server** field, and make sure that the port is set to **443**. The **Connect to RHEV Manager and Validate Certificate** box needs to be checked. In the **Set RHEV-M Admin Password** fields, the password defines the **root** account password and enables remote **ssh** access. Press the **<Apply>** button. Approve the certificate fingerprint when it pops up. |

<table>
<tr><th>Subscreen</th><th></th></tr>
<tr><td></td><td>This step adds the RHEV-H machine to the RHEV-M Hosts tab.<br><br><code>Management Server:      rhevm.pod0.example.com</code><br><code>Management Server Port: 443</code><br><code>Password:               redhat</code><br><code>Confirm Password:       redhat</code></td></tr>
<tr><td>Red Hat Network</td><td>Register with Red Hat Network or a Satellite Server to automatically gain inherited entitlements for Red Hat Enterprise Linux guests running on the cluster this hypervisor is in.<br>All RHEV-H hypervisors normally should be registered with either RHN or a Satellite Server, but we do not register our classroom machines.</td></tr>
</table>

3. Navigate back to the **Status** screen and select **<Log Off>**.

# Demonstration: Approve RHEV hypervisor

In this example, observe the following steps as the instructor demonstrates how to approve a RHEV hypervisor.

When the RHEV-H machine has been configured, including the RHEV-M section, the RHEV-H machine will automatically show up in the RHEV-M web interface in the **Hosts** tab. The RHEV-H machine still needs approval, however. Right-click the RHEV-H host and choose **Approve**. Once the RHEV-H host is approved, it can be used in a cluster and used as a hypervisor.

1. Log into your RHEV-M Administration Portal with a user name of **`rhevadmin`**, password of **`redhat`**, and domain of **`example.com`**.

2. Navigate to the **Hosts** tab. Show that the new RHEV-H node has automatically appeared in a state of **Pending Approval**.

3. Right-click the RHEV-H host and select **Approve** from the menu.

4. Your host should now move through the **Installing** and **Unassigned** states to **Up** status.

## References

Additional information may be available in the *Installation Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

Additional information may be available in the *How to Install the Red Hat Enterprise Virtualization Hypervisor 7.0* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Installing RHEV Hypervisor

In this lab, you will install, configure, and approve a new RHEV Hypervisor host.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to have a working RHEV-H host managed by your RHEV-M engine.

**Before you begin**
Make sure the RHEV-M environment configured in the previous lab is still working.

Perform the following steps:

1. Reboot the servera VM and select the PXE boot option when the BIOS screens appear. When the PXE menu appears, type **rhevh** at the **boot:** prompt.

   ```
   boot: rhevh
   ```

2. After the RHEV Hypervisor installer boots, select **Install Hypervisor 7.1-*version*** then hit **Enter**.

   Select your keyboard layout and press **Enter**.

   The internal hard disk drive should be selected when the installer asks for the boot device. Confirm the internal hard drive is selected, then choose **<Continue>** and press **Enter**.

   The internal hard disk drive should be selected when the installer asks for the installation device. Confirm the internal hard drive is selected, then choose **<Continue>** and press **Enter**. Accept the default partition sizes, then choose **<Continue>**.

3. The RHEV-H installer will prompt for the RHEV-H host administrator password. Type **redhat123** in the **Password:** and **Confirm Password:** fields, choose **<Install>**, then press **Enter**.

   At this point the hard disk drive will be formatted, followed by the installation of the RHEV hypervisor. Once the installation finishes, press **Enter** at the **<Reboot>** prompt. Now that the RHEV-H software is installed, it is time to configure the system.

4. Once the RHEV Hypervisor system boots, log into the console as **admin** with a password of **redhat123**. A main menu of status and configuration items will appear at the left side of the screen. Each menu item can be selected by using the up and down arrow keys to highlight a choice followed by **Enter**.

5. Select **Network** from the main menu. When the network configuration screen appears, set the host name and DNS and NTP settings. Provide the following values:
   - **Hostname: servera.pod*X*.example.com**
   - **DNS Server 1: 172.25.254.254**

- **`NTP Server 1: 172.25.254.254`**

Select **<Save>** to confirm your choices.

Next configure the network interface. Use the **`Tab`** key to move the cursor to the network devices and make sure **`eth0`** (or first interface) is highlighted, then press **`Enter`**.

When the network interface configuration screen appears, choose the **`Static`** radio button for **`IPv4 Settings:`**, then provide the following values:

- **`IP Address: 172.25.`*`X`*`.10`**
- **`Netmask: 255.255.255.0`**
- **`Gateway: 172.25.`*`X`*`.254`**

Select **<Save>** to confirm your choices. When the confirmation box appears, select **Close**.

6. This step is only necessary if you are running this in an online learning environment.

## Important

This step is only necessary if you are running this in an online learning environment, since the nested virtualization of the cloud is missing some information needed to run RHEV-H.

Open the virtual keyboard to send **`F2`** to the system to open the Rescue Shell. Select **OK** to be presented with a prompt.

```
[root@servera admin]# mkdir -p /config/usr/share/libvirt
[root@servera admin]# wget http://classroom.example.com/materials/
cpu_map.xml -P /config/usr/share/libvirt
[root@servera admin]# echo "/usr/share/libvirt/cpu_map.xml" >> /config/
files
[root@servera admin]# uuidgen -r > /etc/vdsm/vdsm.id
[root@servera admin]# reboot
```

After it finishes booting, log back into the RHEV-H configuration interface.

7. Select **Security** from the main menu. Navigate to the **`Enable ssh password authentication`** field and hit **`Space`** to toggle it on. Select **<Save>**, then select **Close** to confirm your settings.

8. Select **RHEV-M** from the main menu. Specify the values that will allow your RHEV-H host to communicate with your RHEV-M server.

   - **`Management Server: rhevm.pod`*`X`*`.example.com`**
   - **`Management Server Port: 443`**

   Fill in the fields with the provided information. Provide the RHEV-M password of **`redhat`** and confirm it. Select **<Save & Register>** to save your changes.

When the RHEV-H host contacts the RHEV-M server, it should display a digital fingerprint. Select **<Accept>** to continue the RHEV-M registration. Select **Close** once the registration succeeds.

Select **Status** from the main menu, then select **<Log Off>** to exit the configuration tool and get back to the RHEV-H login prompt.

9. Log into your RHEV-M Administration Portal as **`rhevadmin`** with a password of **`redhat`** in the **`example.com`** domain. Select the **Hosts** tab. The host should appear with a status of **`Pending Approval`**. Click **`servera.pod`*`X`*`.example.com`** to highlight it and then click the **Approve** button. Review the **Edit & Approve Host** window. Click **OK**.

   On the resulting **Power Management Configuration** window, select **OK** to skip configuring power management.

   The host should change state to **`Installing`**, then quickly progress through **`Unassigned`** to **`Up`**.

   If the RHEV-H host enters a **`Non-Operational state`**, reboot the RHEV-H host and check its BIOS settings. Make sure that "Execute Disable Memory Protection Technology" is enabled.

# Automating RHEV Hypervisor Installation

## Objectives

After completing this section, students should be able to:

- Automate RHEV Hypervisor installation and configuration.

RHEV Hypervisor installation can be automated by passing kernel parameters to the RHEV Hypervisor installer. When all required parameters are passed, the installer will proceed to perform an automated installation. Parameters are discussed in the *Red Hat Enterprise Virtualization Installation Guide*. The required parameters, and some of the optional parameters, are as follows.

## Command

Specify one of the following arguments to identify the action to be taken.

- **`install`**—fresh installation.
- **`reinstall`**—ignore existing installation.
- **`upgrade`**—preserve configuration installing newer hypervisor.
- **`uninstall`**—remove existing installation.

## Storage

The simplest way to specify storage is with the **`storage_init=`*`device`*** argument. This is a required argument for automated installation.

```
storage_init=/dev/sda
```

Specifying the device name is the simplest format for this command. There are other formats that allow for selection of USB, SCSI, or CCISS devices by **udev** serial number. Those formats are discussed and examples are given in the RHEV Hypervisor installation documentation.

For further control of disk allocation, the **`storage_vol=:SWAP::CONFIG:LOGGING:DATA`** kernel argument can be specified in addition to **`storage_init`**. The **`iscsi_name=`*`IQN`*** can be used to set an iSCSI qualified name for your initiator.

## Network

Another required kernel parameter for automated installations is the **`BOOTIF`** argument, which specifies the installation network device. There are three different types of interface specifications that can be made with this argument:

```
BOOTIF=link
BOOTIF=ethX
BOOTIF=<MAC>
```

### Note

The MAC address must be surrounded by < and >. For example:

```
BOOTIF=<52:54:00:56:00:01>
```

This argument can be provided automatically by pxelinux when the **IPAPPEND 2** directive is used in the pxelinux configuration file.

When DHCP is not supported, the following kernel parameters can be specified to configure the network:

```
ip=
netmask=
gateway=
dns=IP1:IP2
hostname=
ntp=IP1:IP2
vlan=
```

## RHEV-M

Another parameter required for automated install is the **management_server** option:

```
management_server=RHEVM_FQDN:PORT
```

This option specifies the fully qualified domain name and port number of the RHEV Manager host to contact and with which to register when the installation finishes. The port number does not have to be specified if the default port (443) is used.

## Security and Red Hat Network

Parameters that configure security and Red Hat Network settings are optional. The **rhevm_admin_password=*HASH*** argument sets the local **root** password and enables SSH access to the RHEV Hypervisor host. **openssl** can generate the **HASH** value. The **ssh_pwauth=1** enables SSH password access (instead of key-based authentication). The default only allows key-based authentication.

Additional parameters can be specified to automate registration with Red Hat Network. These parameters are optional and are presented in the RHEV Installation Guide.

### References

Additional information may be available in the *Installation Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Quiz: Automating RHEV Hypervisor Installation

Choose the correct answer to the following questions:

1. What are the three required directives needed to automate RHEV-H installation?

    a. **`storage_init`**
    b. **`BOOTIF`**
    c. **`IPMI`**.
    d. **`management_server`**

2. What value should the **`storage_init`** kernel parameter have?

    a. The name of the NFS server to use.
    b. The name of the disk device to initialize.
    c. The **`IQN`** of the iSCSI server.

3. What should the value of **`management_server`** include?

    a. The IP address of the IPA server to contact for managing RHEV users.
    b. The **`FQDN`** of the IPA server the RHEV Hypervisor should authenticate against.
    c. The **`FQDN`** and the network port of the RHEV Manager server the RHEV Hypervisor host should contact.

4. What are the three ways the network interface can be selected when performing an automated installation?

    a. The interface name.
    b. The interface **`VLAN`** ID.
    c. The interface **`link`**.
    d. The MAC address enclosed in <>.

# Solution

Choose the correct answer to the following questions:

1. What are the three required directives needed to automate RHEV-H installation?

    a. **`storage_init`**
    b. **`BOOTIF`**
    c. `IPMI`.
    d. **`management_server`**

2. What value should the **`storage_init`** kernel parameter have?

    a. The name of the NFS server to use.
    b. **The name of the disk device to initialize.**
    c. The **`IQN`** of the iSCSI server.

3. What should the value of **`management_server`** include?

    a. The IP address of the IPA server to contact for managing RHEV users.
    b. The **`FQDN`** of the IPA server the RHEV Hypervisor should authenticate against.
    c. **The `FQDN` and the network port of the RHEV Manager server the RHEV Hypervisor host should contact.**

4. What are the three ways the network interface can be selected when performing an automated installation?

    a. **The interface name.**
    b. The interface **`VLAN`** ID.
    c. **The interface `link`.**
    d. **The MAC address enclosed in <>.**

# Upgrading RHEV Hypervisor

## Objectives

After completing this section, students should be able to:

- Upgrade RHEV Hypervisor software from a RHEV Manager server.

## Local upgrades using boot media

One method of upgrading a RHEV Hypervisor host requires installation media with a newer version of the RHEV-H software. Boot the RHEV-H host using the installation media and pass the **upgrade** kernel option to the installer. This option tells the installer to upgrade existing software found on the hard disk rather than prompting the user for an installation.

*rhev-hypervisor7* and *rhev-hypervisor7* include the RHEV Hypervisor ISO in the **/usr/share/rhev-hypervisor** directory. Use a tool, such as **wodim**, to burn the ISO to CD-ROM media.

## Remote upgrades using RHEV-M

Another method of upgrading a RHEV-H host requires installing a new RHEV-H ISO package on the RHEV-M server that manages the host. Use **yum** to upgrade the *rhev-hypervisor7* or *rhev-hypervisor7* package installed on the RHEV-M server. This updated ISO image will be available for upgrade when using the RHEV-M Administration Portal to manage the RHEV-H host.

Log into the RHEV-M Administration Portal as an administrative user for the host you want to upgrade. Under the **Hosts** tab, select the host you want to upgrade. Once the host is switched to **Maintenance** mode, a new **Upgrade** button will appear. An **Install Host** window will open and the new hypervisor ISO should be listed in the pulldown menu next to the **RHEV-H-ISO Name** option. Select the new ISO, then click **OK** to begin the upgrade.

## Demonstration: Upgrading a RHEV-H host via RHEV-M

In this example, observe the following steps as the instructor demonstrates how to initiate the upgrade of a RHEV host, through the RHEV-M web interface.

1. Install an updated hypervisor package on rhevm.pod*X*.example.com:

   ```
   [root@rhevm ~]# yum -y install rhev-hypervisor7
   ```

2. Shut down all your virtual machines (if you have any) by selecting them in the **Virtual Machines** tab of the RHEV-M web interface by right-clicking and selecting **Shut down**.

3. Place your hypervisor in **Maintenance** mode by selecting the host from the **Hosts**, right-clicking the host, and choosing **Maintenance**. Press **OK** to confirm.

4. Once the RHEV-H host shows a status of **Maintenance**, right-click the host and choose **Upgrade**.

   Select the latest RHEV-H ISO from the dropdown menu that appears and click on **OK**.

   Your hypervisor will now start installing the latest version of RHEV-H. Once the transfer is complete, your hypervisor will reboot.

### Note

The default time RHEV will wait before checking on a RHEV-H host after a reboot is 300 seconds (5 minutes). You can find this in the **/var/log/ovirt-engine/engine.log** log. The log message will look something like the following:

```
2012-12-05 16:56:53,710 INFO [org.ovirt.engine.core.bll.InstallVdsCommand]
 (pool-4-thread-49) [25248d7d]  Waiting 300 seconds, for server to finish
 reboot process.
```

If you want to decrease (or increase) this value, set the **ServerRebootTimeout** key using **engine-config**. Once the key is set to a new value, restart the **ovirt-engine** service. For instance, to change the value to 90 seconds, run the following commands on rhevm.pod*X*.example.com:

```
[root@rhevm ~]# engine-config -s ServerRebootTimeout=90
[root@rhevm ~]# service ovirt-engine restart
```

### References

Additional information may be available in the *Installation Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Upgrading RHEV Hypervisor

In this lab, you will upgrade your RHEV Hypervisor host to the latest available version.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

**Outcome(s)**
You should be able to upgrade a RHEV-H host managed by your RHEV-M engine.

**Before you begin**
Make sure the RHEV-M environment and RHEV-H configured in the previous labs are still working.

Perform the following steps:

1. Use **ssh** to connect to **rhevm** virtual machine as root.

2. Install the newest *rhev-hypervisor7* package available.

```
[root@rhevm ~]# yum -y install rhev-hypervisor7
```

3. In the RHEV-M management console, select the **Hosts** tab and select the host **servera.pod*X*.example.com**. Click the **Maintenance** button, then click **OK**.

4. Right-click the host, then select the option **Upgrade**.

   An **Install Host** window will open. The new hypervisor ISO file should be listed in the pulldown menu next to the **RHEV-H-ISO Name** option. Select the **rhevh-7.1-*latest-version*.iso**, then click **OK**.

   The host **Status** moves from **Maintenance** to **Installing**, and after some time changes to **Reboot**. When the status finally changes to **Up**, the upgrade process has finished.

> ## Note
>
> It takes a considerable time after a RHEV-H host reboots for RHEV-M to update its status and show it in the **Up** state.
>
> The default time that RHEV-M waits for the host to reboot is 300 seconds (as seen in **/var/log/ovirt-engine/engine.log**). This can be changed using the **engine-config** command to change the **ServerRebootTimeout** key.

5. When the host has finished, new information about the operating system, kernel, etc. will be updated in the **Details** pane below the **General** tab.

# Troubleshooting RHEV Hypervisor

## Objectives

After completing this section, students should be able to:

- Troubleshoot RHEV Hypervisor installation and runtime problems.

## BIOS misconfigurations

Most RHEV Hypervisor installation problems occur because of BIOS misconfigurations on the hardware used for the RHEV-H server. Having an incorrect boot order specified is a familiar problem that is easily recognized, but there are a couple other BIOS misconfigurations that impact RHEV-H installation.

RHEV Hypervisor requires AMD-V or Intel VT hardware virtualization extensions to function. Frequently these extensions can be enabled or disabled in the BIOS configuration for CPU configuration. Confirm that virtualization CPU extensions are enabled in BIOS before starting a RHEV-H installation.

Another BIOS setting that can impact RHEV Hypervisor installation is the one that controls the No eXecute CPU flag. RHEV Hypervisor will install on a machine that has this setting disabled, but it will not successfully activate with the RHEV Manager server. When attempting to activate the RHEV-H server in RHEV Manager, the following error message displays in RHEV-M:

```
Host hostname moved to Non-Operational state as host does not meet the
cluster's minimum CPU level.  Missing CPU features: CpuFlags
```

This error message indicates an NX CPU flag issue.

## Getting system information

Use the RHEV-H administrative interface to get additional system information for troubleshooting purposes. The **Status** screen provides general information about the network and logging configuration information. Additional information can be gleaned by visiting the other configuration submenus without making changes.

Local log files can be accessed from the **Status** screen using the F8 function key. This key will bring up the support menu which lists the most useful log files on the RHEV Hypervisor host. Select a log file with the arrow keys, then hit Enter to view it with the **less** utility. The q key exits **less**, then select **<Back to Menu>** to return back to the **Status** screen.

### Note

Recall that the **engine-log-collector** command will collect log files from the hypervisor and store them in the **sosreport**:

```
INFO: Log files have been collected and placed in /tmp/logcollector/sosreport-
LogCollector-rhevm-20121129060015-7050.tar.xz.
```

The RHEV-H **root** account is locked by default (though unlocked in the earlier exercise). However, support shell access to RHEV-H *is* available using the F2 function key from the administrative interface. Shell access is intended only for troubleshooting purposes. To avoid damaging the RHEV-H server configuration, limit support shell access to perform only the following tasks:

- Check critical log files and using the **dmesg** command.
- Confirm partition and network configuration.
- Run **sosreport** to collect debugging information.
- Gracefully shut down the node.

Other uses for the RHEV-H support shell are not supported by Red Hat.

# Reinstalling a RHEV Hypervisor

If reinstalling a RHEV Hypervisor is desired, the installation program cannot simply be run through again. As part of the installation process, the RHEV-H installer will check to see if the machine is already installed with RHEV-H by checking the disks on the machine. If evidence of an already installed RHEV-H node exists, the installer will provide an error indicating the machine has been already installed.

In order to have the installer reinstall the RHEV-H node, the **storage_init** kernel argument must be provided on the installer's kernel command line. **storage_init** requires an argument of which storage device should be reinitialized prior to the installation program running, for example:

```
reinstall storage_init=/dev/sda
```

## References

Additional information may be available in the *Installation Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Troubleshooting RHEV Hypervisor

In this lab, you will obtain RHEV Hypervisor host configuration information for possible troubleshooting purposes.

| Resources | |
|---|---|
| **Files** | /var/log/ |
| **Application URL** | https://rhevm.pod*X*.example.com |

**Outcome(s)**
You should be able to gather basic network configuration and RHEV-H log information.

**Before you begin**
Make sure the RHEV-M environment and RHEV-H configured in the previous labs are still working.

Perform the following steps:

1. At the RHEV-H login prompt, log in as the **admin** user with the **redhat** password to access the administration configuration menu.

2. Go to the **Status** submenu.

   If the **Status** submenu is not already displayed, use the arrow keys to highlight **Status** from the menu at the left, then hit Enter.

3. Note the information about the RHEV-H host that is available to you on this display. Record your findings.

   Network information such as IP address configuration, the logical network name, physical device name, and MAC address information about where log file information is kept and how many running virtual machines is also displayed.

4. Use F8 to view the log files on the RHEV-H host.

   Type the F8 function key at the status menu to display a list of available log files to view. Use the arrow keys to highlight the particular log file you would like to view, then hit Enter to begin viewing the file with **less**.

   When you are finished viewing a log file, type q to return to the support menu. Use the arrow keys to select **<Back to Menu>** to return to the status menu.

5. Use the RHEV-H support shell to get system information. Type F2, then select **OK** to get a shell prompt.

6. Use the **ip** and **ethtool** commands to display the network status of your RHEV-H host.

   **ip addr** and **ip -s link** display the IP configuration and interface statistics for network interfaces. The **ethtool** command displays negotiated link information and current interface status.

   ```
   [root@servera admin]# ip addr show dev rhevm
   ```

```
3: rhevm: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether 52:54:00:00:XX:0a brd ff:ff:ff:ff:ff:ff
    inet 172.25.X.10/24 brd 172.25.X.255 scope global rhevm
    inet6 fe80::21b:21ff:fe0b:be11/64 scope link
       valid_lft forever preferred_lft forever
[root@servera admin]# ip -s link show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
 1000
    link/ether 52:54:00:00:XX:0a brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    26590430   196490   0       0       0       2435
    TX: bytes  packets  errors  dropped carrier collsns
    37014550   111632   0       0       0       0
[root@servera admin]# ethtool eth0 | head
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric
```

7. Examine some of the log files in the /var/log directory of the RHEV-H server.

```
[root@servera admin]# ls /var/log
audit     dmesg         libvirt       ntpstats       secure       vdsm-reg
boot.log  dmesg.old     libvirtd.log  ovirt.log      spooler      wtmp
btmp      dracut.log    lost+found    ovirt.log-tmp  tallylog
core      iscsiuio.log  maillog       rhsm           vdsm
cron      lastlog       messages      sa             vdsm-config
[root@servera admin]# less /var/log/messages
```

8. Log out from the RHEV-H support shell.

   Exit the **bash** shell to return to the RHEV-H administrative interface.

```
[root@servera admin]# exit
```

9. Select **<Log Off>** from the **Status** submenu to leave the RHEV-H administrative interface.

   If the **Status** submenu is not already displayed, use the arrow keys to highlight **Status** from the menu at the left, then hit Enter. Use the left and right arrow keys to select the **<Log Off>** option.

# Lab: Installing and Configuring RHEV Hypervisor Hosts

In this lab, you will uninstall, reinstall, then upgrade your RHEV-H host.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to reinstall and fully upgrade a RHEV-H host managed by the RHEV-M server.

**Before you begin**
Make sure the RHEV-M environment configured in the previous lab is still working.

**Lab outline**

1. Remove the existing RHEV-H host installation.

2. Install your RHEV Hypervisor server using PXE with the following configuration:

| RHEV-H specifications | |
|---|---|
| **RHEV-H installation** | |
| Boot device | Internal hard disk |
| Installation device | Internal hard disk |
| RHEV-H administrator password | **`redhat`** |
| **RHEV-H configuration** | |
| RHEV-H host name | servera.pod*X*.example.com |
| DNS server | 172.25.254.254 |
| NTP server | 172.25.254.254 |
| Network interface configuration | Static |
| IP address | 172.25.*X*.10 |
| Netmask | 255.255.255.0 |
| Gateway | 172.25.*X*.254 |
| Enable SSH password authentication | Yes |
| RHEV-M management server | rhevm.pod*X*.example.com |
| RHEV-M management server port | 443 |
| RHEV-M admin password | **`redhat`** |

3. Once the RHEV Hypervisor system boots, log into the console as **`admin`** with a password of **`redhat`**. Configure according to the previous settings.

4. Approve the newly installed RHEV-H host.

5. The newest *rhev-hypervisor7* package should already be installed and available on the **rhevm** server.

   Upgrade the newly installed RHEV-H host to the current release.

# Solution

In this lab, you will uninstall, reinstall, then upgrade your RHEV-H host.

| **Resources** | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to reinstall and fully upgrade a RHEV-H host managed by the RHEV-M server.

**Before you begin**
Make sure the RHEV-M environment configured in the previous lab is still working.

**Lab outline**

1. Remove the existing RHEV-H host installation.

   Log into the RHEV-M Administration Portal. Remove the host from your RHEV-M configuration. Select the **Hosts** tab, then select the **servera.pod*X*.example.com** host. Click the **Maintenance** button, then click **OK** when the confirmation dialog appears. Click the **Remove** button, then click **OK** to confirm your choice.

   Now that the hypervisor has been removed from your RHEV-M configuration, wipe your existing RHEV-H server.

   If running **servera** on physical hardware, reboot it and stop the boot process when GRUB appears. Modify the default stanza by appending the following parameters to the **kernel** line:

   ```
   uninstall storage_init=/dev/sda
   ```

   If running **servera** in a virtual environment, then reset the **servera** virtual machine.

2. Install your RHEV Hypervisor server using PXE with the following configuration:

| **RHEV-H specifications** | |
|---|---|
| **RHEV-H installation** | |
| Boot device | Internal hard disk |
| Installation device | Internal hard disk |
| RHEV-H administrator password | **redhat** |
| **RHEV-H configuration** | |
| RHEV-H host name | servera.pod*X*.example.com |
| DNS server | 172.25.254.254 |
| NTP server | 172.25.254.254 |
| Network interface configuration | Static |
| IP address | 172.25.*X*.10 |
| Netmask | 255.255.255.0 |

| RHEV-H specifications | |
|---|---|
| Gateway | 172.25.*X*.254 |
| Enable SSH password authentication | Yes |
| RHEV-M management server | rhevm.pod*X*.example.com |
| RHEV-M management server port | 443 |
| RHEV-M admin password | **redhat** |

PXE-boot your second server. When the PXE menu appears, type **rhevh** at the **boot:** prompt.

After the RHEV Hypervisor installer boots, select **Install Hypervisor 7.1-*version***, then hit **Enter**.

The internal hard disk drive should be selected when the installer asks for the boot device. Confirm the internal hard drive is selected, then choose **<Continue>** and press **Enter**.

The internal hard disk drive should be selected when the installer asks for the installation device. Confirm the internal hard drive is selected, then choose **<Continue>** and press **Enter**. Accept the default partition sizes, then choose **<Continue>**.

The RHEV-H installer will prompt for the RHEV-H host administrator password. Type **redhat** in the **Password:** and **Confirm Password:** fields, choose **<Install>**, and press **Enter**.

At this point, the hard disk drive will be formatted, followed by the installation of the RHEV Hypervisor. Once the installation finishes, press **Enter** at the **<Reboot>** prompt.

3. Once the RHEV Hypervisor system boots, log into the console as **admin** with a password of **redhat**. Configure according to the previous settings.

   First configure the RHEV-H network settings. Select **Network** from the main menu. When the network configuration screen appears, set the host name and DNS and NTP settings. Provide the following values:

   - **Hostname: servera.pod*X*.example.com**
   - **DNS Server 1: 172.25.254.254**
   - **NTP Server 1: 172.25.254.254**

   Select **<Save>** to confirm your choices.

   Next configure the network interface. Use the **Tab** key to move the cursor to the network devices and make sure **eth0** is highlighted, then press **Enter**.

   When the network interface configuration screen appears, choose the **Static** radio button for **IPv4 Settings:**, then provide the following values:

   - **IP Address: 172.25.*X*.10**
   - **Netmask: 255.255.255.0**
   - **Gateway: 172.25.*X*.254**

   Select **<Save>** to confirm your choices.

### Important

This step is only necessary if you are running this in an online learning environment, since the nested virtualization of the cloud is missing some information needed to run RHEV-H.

Open the virtual keyboard to send **F2** to the system to open the Rescue Shell. Select **OK** to be presented with a prompt.

```
[root@servera admin]# mkdir -p /config/usr/share/libvirt
[root@servera admin]# wget http://classroom.example.com/materials/
cpu_map.xml -P /config/usr/share/libvirt
[root@servera admin]# echo "/usr/share/libvirt/cpu_map.xml" >> /config/
files
[root@servera admin]# uuidgen -r > /etc/vdsm/vdsm.id
[root@servera admin]# reboot
```

After it finishes booting, log back into the RHEV-H configuration interface.

Select **Security** from the main menu. Navigate to the **Enable ssh password authentication** field and hit **Space** to toggle it on. Select **<Save>**, then select **Close** to confirm your settings.

Select **RHEV-M** from the main menu. Specify the values that will allow your RHEV-H host to communicate with your RHEV-M server.

- **Management Server: rhevm.pod*X*.example.com**
- **Management Server Port: 443**

Fill in the fields with the provided information. Select **< Save & Register >** to save your changes.

When the RHEV-H host contacts the RHEV-M server, it should display a digital fingerprint. Select **<Accept>** to continue the RHEV-M registration. Select **OK** once the registration succeeds.

Select **Status** from the main menu, then select **<Log Off>** to exit the configuration tool and get back to the RHEV-H login prompt.

4. Approve the newly installed RHEV-H host.

   Log into your RHEV-M web administration interface. Select the **Hosts** tab. Shortly after your RHEV-H system has rebooted, you should see it appear on the list in state **Pending Approval**. Click **servera.pod*X*.example.com** to highlight it and then click the **Approve** button. Review the **Edit and Approve Host** window and click **Ok**. The host should change state to **Installing**, then quickly progress through **Non Responsive** to **Up**.

   If the RHEV-H host enters a **Non-Operational state**, reboot the RHEV-H host and check its BIOS settings. Make sure the “Execute Disable Memory Protection Technology” is enabled.

5. The newest *rhev-hypervisor7* package should already be installed and available on the **rhevm** server.

   Upgrade the newly installed RHEV-H host to the current release.

   In the RHEV-M management console, select the **Hosts** tab and select the host **servera.pod*X*.example.com**. Click the **Maintenance** button, then click **OK**.

   In the **Hosts** tab, select the **servera.pod*X*.example.com**. Click the **Upgrade** button.

   An **Install Host** window will open. The new hypervisor ISO file should be listed in the pulldown menu next to the **RHEV-H-ISO Name** option. Select the **rhevh-7.1-*latest-version*.iso**, then click **OK**.

   The host **Status** changes to **Installing** and after some time changes to **Reboot**. When the status finally changes to **Up** the upgrade process has finished.

   After some time, new information about the operating system, kernel, etc. will be updated in the **Details** pane below the **General** tab.

# Summary

### Installing RHEV Hypervisor

In this section, you learned:

- To confirm requirements for the RHEV hypervisor and ensure the system is compatible for being a RHEV-H host.
- To manually install, configure, and approve the RHEV hypervisor across the network with PXE.

### Automating RHEV Hypervisor Installation

In this section, you learned:

- The use of **`storage_init`**, **`BOOTIF`**, and **`management_server`** directives when automating the installation of the RHEV hypervisor.

### Upgrading RHEV Hypervisor

In this section, you learned:

- To use the local boot media provided in the *rhev-hypervisor7* package to upgrade the RHEV hypervisor.
- The use of RHEV-M to push an upgrade to a RHEV hypervisor.

### Troubleshooting RHEV Hypervisor

In this section, you learned:

- How to identify BIOS misconfigurations to ensure proper installation of the RHEV hypervisor.
- How to gather additional system information before opening a support ticket with Red Hat.
- The additional **`storage_init`** argument should be specified when reinstalling the RHEV hypervisor.

CHAPTER 4

# CREATING RHEV DATA CENTERS AND CLUSTERS

| Overview | |
|---|---|
| **Goal** | Create RHEV Data Centers and Clusters in the RHEV Manager interface. |
| **Objectives** | • Describe essential resources in RHEV.<br>• Create RHEV Data Center.<br>• Create RHEV Cluster. |
| **Sections** | • RHEV Resources (and Quiz)<br>• Creating Data Centers (and Practice)<br>• Creating Clusters (and Practice) |
| **Lab** | • Creating RHEV Data Centers and Clusters |

# RHEV Resources

## Objectives

After completing this section, students should be able to:

- Describe essential resources in RHEV.

## RHEV resources

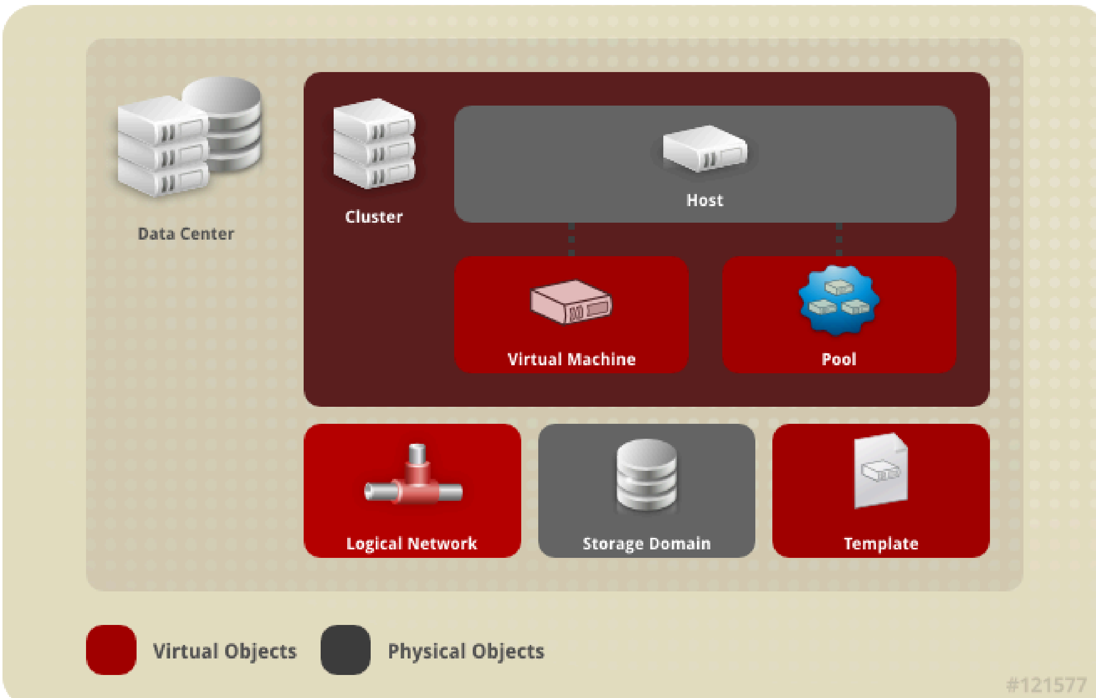


*Figure 4.1: Datacenter objects*

The preceding figure diagrams the various components that make up a single RHEV datacenter. Note the hierarchy of the resources, for example, a physical host belongs to one cluster which belongs to one datacenter, but a storage domain belongs directly to a datacenter. The RHEV-M system may manage multiple datacenters, a datacenter may contain multiple clusters, and a cluster may contain multiple hosts.

## RHEV resource key terms

Working with RHEV requires the administrator to know certain definitions, including those resources previously pictured. The following table defines those key terms.

| **RHEV resource key terms** | |
|---|---|
| *Datacenter* | A datacenter is a collection of a number of clusters of virtual machines, storage, and networks. The datacenter is the highest-level container for all physical and logical resources within a managed virtual environment. |

| RHEV resource key terms | |
|---|---|
| *Storage* | The storage pool is a logical entity that contains a standalone image repository of a certain type, either iSCSI, Fibre Channel, or NFS. Each storage pool may contain several storage domains. Each storage domain may contain virtual machine disks, ISO images, or can be used for the import and export of virtual machine images. |
| *Cluster* | A cluster is a set of physical hosts that are treated as a resource pool for a set of virtual machines. Hosts in a cluster share the same network infrastructure and the same storage. They are a *migration domain* within which virtual machines can be moved from host to host. |
| *Host* | A host is a physical server that runs either RHEV-H or RHEL and hosts one or more virtual machines. RHEV-H nodes register when they are installed but cannot be used until they are approved through the RHEV-M console. |
| *Virtual machine* | Virtual machines can be either virtual servers or virtual desktops running various versions of Linux or Windows. Virtual machines are tied to a particular cluster and therefore to one datacenter. |
| *Pools* | A pool is a group of identical virtual systems that are available on demand by each of the users (not concurrently). |
| *Template* | A template is a model virtual machine with its own set of configurations and settings. A virtual machine that is based on a particular template acquires the configurations and settings of the template. |
| *Logical network* | A datacenter can set up multiple logical networks to segregate different types of traffic between nodes to different networks. Typically, this sort of segregation is done based on the physical topology of the network and various functional requirements. |
| *Snapshot* | A snapshot is a view of a virtual machine's operating system and all its applications at a given point in time. It can be used to save the settings of a virtual machine before an upgrade or before new applications are installed. |
| *Events and monitors* | Alerts, warnings, or other notices about activities within the system can be displayed in both a graphical and textual fashion. |
| *Reports* | RHEV Manager includes a data warehouse that collects monitoring data for hosts, virtual machines, and storage that allows for the creation of reports using any query tool that supports SQL. |

The RHEV Administration Portal allows administrators to create and manage the various RHEV-M resources. Note that many of the previously defined terms correspond to tabs in the portal.

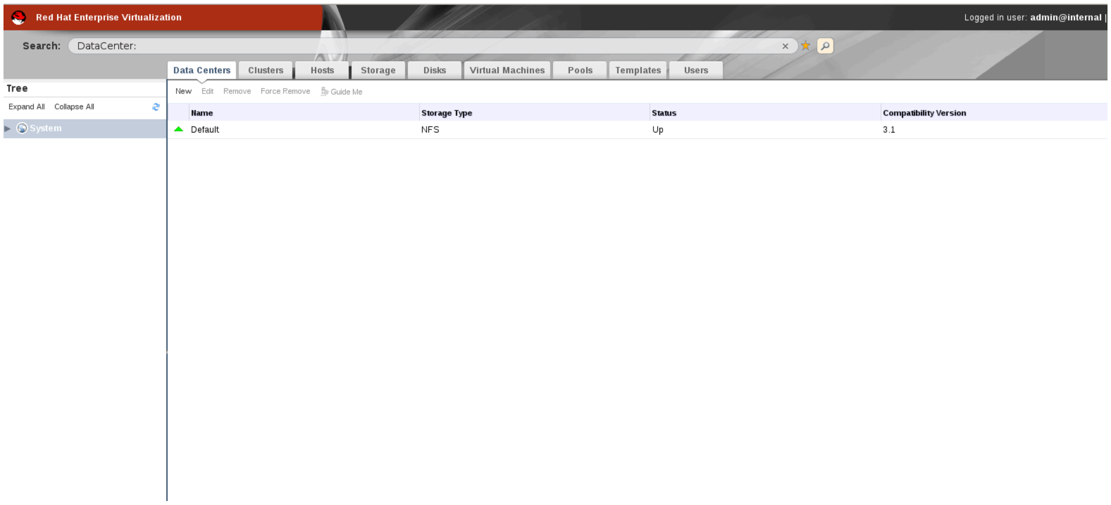


*Figure 4.2: RHEV-M Administration Portal tabs*

## System scheduler

The RHEV System Scheduler manages the allocation of physical resources within the datacenter and is responsible for the placement of virtual machines (VMs) on host systems. The system scheduler continually monitors the utilization of host systems and VMs, dynamically managing the placement of VMs within the datacenter based on policies defined by the system administrator.

When a VM is started, the system scheduler automatically selects the host on which to run the VM based on resource utilization and system policies. The administrator can define a preferred host for a VM, for example, to co-locate two VMs on the same host to reduce their network traffic and optimize their communication.

Main features of the system scheduler include:

- In the event of a hardware failure, any VM configured to be highly available will be restarted on another host in the cluster. The system scheduler manages the selection of the host based on resource utilization within the cluster.
- When the administrator manually initiates a live migration, the system scheduler will automatically select the destination host based on resource utilization. The administrator can override the system scheduler and manually select the destination host.
- The administrator can define policies to manage the resource allocation within a cluster of host systems. Based on these policies, the system scheduler will automatically balance the workload between physical hosts using live migration to move VMs without downtime or service interruption.
- The administrator can define the thresholds at which the system scheduler will automatically use live migration to relocate VMs, defining both the utilization threshold the physical host should reach and the duration for which this threshold must be exceeded. This avoids unnecessary reallocation of resources due to momentary spikes in resource usage. If a cluster is operating at maximum capacity, the administrator can dynamically add another host to the cluster. The system scheduler will automatically rebalance the workload within the cluster.

- Different system scheduler policies can be defined for each cluster in the same virtual datacenter. With the **Even Distribution** policy, administrators can specify the **Maximum Service Level** a host is permitted to have. For example, a host that has reached the maximum service level defined will not have further virtual machines started on it. The time interval (in minutes) can also be specified that a host is permitted to run at the maximum service level before virtual machines are migrated off it.
- **Power Saver** extends the system scheduler by adding policies to reduce power consumption, consolidating more VMs onto a smaller number of hosts. With the power saver policy, you can configure the **Minimum Service Level** at which the power saver policy is triggered. For example, if the utilization of a single host goes down to 10% for 20 minutes or more, the system scheduler can relocate the VMs running on that host to other hosts in the cluster using live migration. The host will now run idle, consuming significantly lower (typically 10-15%) of the power of an active server.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 3.1 Introduction to Data Centers

Red Hat Enterprise Virtualization 3.5 Installation Guide

- Section 10.3. Data Centers
- Section 10.4. Clusters

# Quiz: RHEV Resources

Match the following items to their counterparts in the table.

| Cluster | Datacenter | Pool | Snapshot | System scheduler |
|---|---|---|---|---|
| Template | | | | |

| Definition | Term |
|---|---|
| A collection of a number of clusters of virtual machines, storage, and networks | |
| A set of physical hosts that are treated as a resource pool for a set of virtual machines | |
| A group of identical virtual systems that are available on demand by each of the users | |
| A model virtual machine with its own set of configurations and settings | |
| A view of a virtual machine's operating system and all its applications at a given point in time | |
| Manages the allocation of physical resources within the datacenter and is responsible for the placement of virtual machines on host systems | |

# Solution

Match the following items to their counterparts in the table.

| Definition | Term |
| --- | --- |
| A collection of a number of clusters of virtual machines, storage, and networks | Datacenter |
| A set of physical hosts that are treated as a resource pool for a set of virtual machines | Cluster |
| A group of identical virtual systems that are available on demand by each of the users | Pool |
| A model virtual machine with its own set of configurations and settings | Template |
| A view of a virtual machine's operating system and all its applications at a given point in time | Snapshot |
| Manages the allocation of physical resources within the datacenter and is responsible for the placement of virtual machines on host systems | System scheduler |

# Creating Datacenters

## Objectives

After completing this section, students should be able to:

- Create a RHEV datacenter.

## Datacenter

The top-level organizational object in a RHEV-M configuration is the *Datacenter* (**Data Centers** tab). A datacenter contains all the physical and logical resources in a single managed virtual environment. This may be all the machines in a particular physical datacenter at a particular location, it may correspond with a set of systems and storage belonging to a particular business unit of the organization, or it may indicate some other division which makes sense to the administrator.

A datacenter named "Default" is created automatically. For many sites, there may be no reason to create additional datacenters beyond the initial default.

## The network container

A datacenter can set up multiple logical networks to segregate different types of traffic between nodes to different networks. Typically, this sort of segregation is done based on the physical topology of the network and various functional requirements. For example, administrators may want to separate the data traffic to clients accessing network services on their virtual machines from the NFS or iSCSI storage traffic. To do this, administrators can place that traffic on separate logical networks. Logical networks can be separated by IP address and netmask, as it is possible to use VLAN ID tagging to identify the different network traffic.

## The storage container

Each datacenter has a storage pool associated with it. The storage pool is a collection of individual storage domains which are central repositories of images that are shared by physical hosts. There are three basic kinds of storage domains:

- *Data domains* store guest images (virtual hard disk images) and snapshots of image states. A data domain cannot be shared between datacenters. Data domains may be NFS, iSCSI, or Fibre Channel-based, although currently only one of these technologies may be used at a time for master domains in a particular datacenter.

- *ISO domains* store ISO images (logical CD-ROMs and DVD-ROMs) and VFDs (virtual floppy drive images) that are used for the installation or rescue of virtual machines. An ISO domain can be shared across different datacenters.

- *Export domains* are used to copy or move images between datacenters and RHEV-M installations. The export domain also can be used to back up virtual machines. An export domain can be moved between datacenters, however, it can only be *active* in one datacenter at a time.

One of the hosts in the datacenter is automatically elected the *Storage Pool Manager*, or SPM. This host is responsible for managing changes to the storage pool communicated from RHEV-M. All hosts can change image data in the storage domains, but only the SPM can change storage domain metadata. If the current SPM host goes down, then RHEV-M will relocate the SPM to another host.

The SPM must be running in order to add storage domains. This is the reason that administrators must register a host (hypervisor) before setting up a datacenter from scratch. Once this is done, it is possible to begin the configuration of the storage domains.

In an NFS datacenter, the SPM creates the virtual disk on top of a regular file system, either as a QCOW2 disk for a thin provision (sparse) format or as a normal disk for a preallocated (RAW) format. In an iSCSI or Fibre Channel data center, the SPM creates a volume group (VG) on top of the logical unit numbers (LUNs) provided.

For a virtual disk with a preallocated format, a logical volume (LV) of the specified size (in GB) is created. For a virtual disk with a thin provision format, a 512 MB LV is created initially. The LV is continuously monitored by the host on which the VM is running. As soon as the usage nears a threshold, the host notifies the SPM and the SPM extends the LV by another 512 MB.

From a performance point of view, a virtual disk with a preallocated (RAW) format is significantly faster than a virtual disk with a thin provisioning (QCOW2) format. It is recommended that the thin provision format be used for non-IO intensive virtual desktops and the preallocated (RAW) format be used for virtual servers.

## Reinitializing a datacenter

Administrators can use a recovery procedure that replaces the master data domain of their datacenter with a new master data domain, which is necessary in the event of data corruption of the master data domain. Reinitializing a datacenter allows to restore all other resources associated with the datacenter, including clusters, hosts, and nonproblematic storage domains.

### Note

The **Re-Initialize Data Center** option will appear only when a right click is performed on the target datacenter. It has no menu item.

## Demonstration: Creating a datacenter

At installation time, a datacenter named "Default" is created automatically. For many sites, there may be no reason to create additional datacenters beyond the initial one.

In this example, observe the following steps as the instructor demonstrates the creation of a new datacenter.

1. Navigate to the **Data Centers** tab in the RHEV-M Administration Portal.

2. Click the **New** button to display the **New Data Center** dialog (shown as follows) and fill in the fields using the following table.

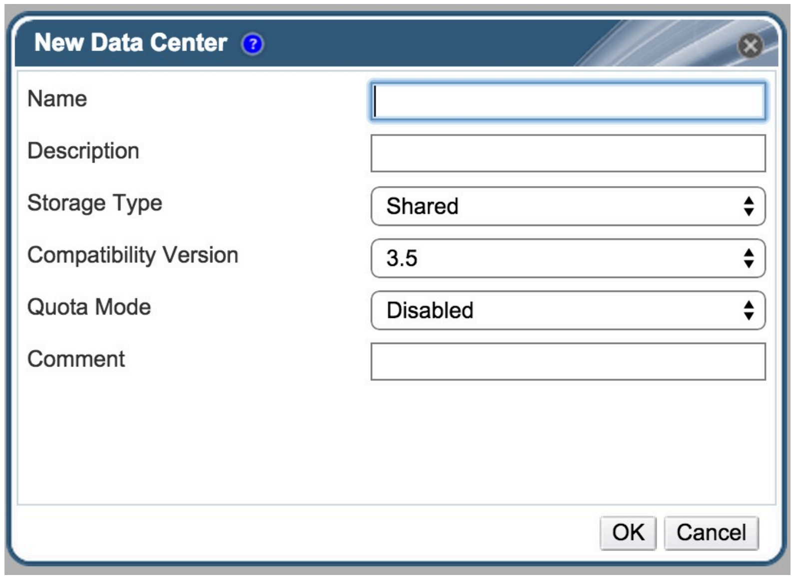


*Figure 4.3: Adding a datacenter*

| Name | Description |
|---|---|
| **Name** | The unique name of the datacenter which can be up to 40 characters.<br>`dcenter0` |
| **Description** | A larger description of the datacenter (optional).<br>`Data Center for Pod 0` |
| **Storage Type** | Identifies the data storage type of the datacenter, shared (NFS, iSCSI, or FCP) or local. All data storage attached to this datacenter will be of this one type.<br>`Shared` |
| **Compatibility Version** | The version of RHEV deployed in this datacenter. Leave 3.5 as the default datacenter version, unless you are upgrading from an earlier version.<br>**In the case of an upgrade, the datacenter will be the last item shifted to 3.5** (first the hosts, then the clusters, finally the datacenter).<br>`3.5` |
| **Quota Mode** | Quota is a resource limitation tool provided with Red Hat Enterprise Virtualization. Choose one of the following:<br>• **`Disabled`**: Select if you do not want to limit resources. |

| Name | Description |
|---|---|
| | • **`Audit`**: Select if you want to edit the Quota settings.<br>• **`Enforced`**: Select to implement Quota.<br>`Disabled` |

3. While a **Guide Me** dialog may be presented to step us through configuring clusters, hosts, and storage, we will choose **Configure Later** as we will perform those steps later.

   The **Guide Me** dialog does not get presented for the first datacenter that you create, only subsequent ones.

4. Note that the new datacenter is in the list of datacenters in a state of "Uninitialized".

R

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 3.1 Introduction to Data Centers

Red Hat Enterprise Virtualization 3.5 Installation Guide

- Section 10.3. Data Centers

# Practice: Creating Datacenters

In this lab, you will create a new datacenter.

**Outcome(s)**
You should be able to create a new datacenter called **newdc*X***.

**Before you begin**
This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.pod*X*.example.com
- Administration Portal connects to rhevm.pod*X*.example.com
- An installed RHEV-H node

Perform the following steps:

1. Log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.

   Open a web browser on workstation and navigate to **`https://rhevm.podX.example.com`**. Fill in the fields as listed previously.

2. Create a New Datacenter

   2.1. Navigate to Datacenters

   Click the **Data Centers** tab.

   2.2. Create a new datacenter with the following values:

   - Name: **newdc*X*** (where *X* is your pod number).
   - Description: **`New Datacenter`**
   - Storage Type: **`Shared`**
   - Compatibility Version: **`3.5`**
   - Quota Mode: **`Disabled`**

   Click the **New** button. The "New Data Center" dialog displays.

   Complete the information using the previous values.

   Click **OK**.

   The "Guide Me" dialog opens, displaying a list of configuration tasks. Click **Configure Later** to close the dialog.

3. Verify the New Datacenter

   3.1. Verify the new datacenter is in the list. What is its status?

   View the new datacenter **newdc*X*** in the list of datacenters with a status of **`Uninitialized`**.

3.2. Access the **Events** entry for the datacenter. Who created the datacenter?

Click the newly created datacenter, **newdc*X***, in the list of datacenters. The settings for that datacenter will appear in the lower pane.

Click the **Events** tab "under Data Center" to see just the events related to this datacenter. There should be an entry identifying "**`rhevadmin`**" as the creator of the new datacenter.

# Creating Clusters

## Objectives

After completing this section, students should be able to:

- Create a RHEV cluster in an existing datacenter.

## Clusters

Hosts, the physical nodes in a datacenter, are organized into clusters (**Clusters** tab). A cluster is a group of hosts that share the same resources such as logical networks and storage domains. A cluster is also a migration domain; that is, virtual machines can migrate between hosts in a cluster freely, but they may not migrate to a host outside the cluster.

Clusters can be assigned guest scheduling policies that determine how virtual machines are balanced among the nodes in the cluster. Three basic policies exist out of the box: a null policy (None), a policy to distribute as evenly as possible (Even Distribution), and a policy to relocate virtual machines off a node if its load is low and the load on other nodes is not too high (Power Saving). This last policy prefers to disable nodes for power savings when utilization is low.

**`Memory overcommit`** settings can be adjusted, either turned off entirely, or to overcommit as much as 200% of the memory on any given node (making the assumption that KSM and other techniques will make it unlikely all this memory would be needed), while allowing virtual machines to continue to function even if a significant number of hosts are down.

## Optimizing clusters

For performance purposes, administrators can work with various optimization features. Such features include memory and CPU optimization, high availability, and KSM tuning.

- Memory optimization: Lets administrators choose on their overcommit policy for the cluster.
- CPU threads: Taking advantage of multithreading built into the CPU. Administrators can treat threads as CPUs. More useful for less CPU-intensive workloads.
- Memory balloon: Memory balloon is a guest device, which may be used to redistribute or reclaim the host memory based on VM needs in a dynamic way. Administrators can enable optimization for the memory balloon usage in the cluster.
- KSM control: Decide whether or not the host shares identical memory pages between different VMs.

## Virtual machines

*Virtual CPUs*: A maximum of 160 physical CPUs are supported by the hypervisor, of which a maximum of 160 virtualized CPUs can be presented to the guest.

*Virtual RAM*: While the hypervisor can support up to 2 TB of physical RAM, a maximum of 2 TB of virtual RAM can be presented to a 64-bit guest and a maximum of 4 GB of virtual RAM can be presented to a 32-bit guest.

*Virtual NICs*: The hypervisor can present to each virtual machine multiple virtual network interface cards (up to eight). Each virtual NIC can map to different virtual networks and physical NICs on the host machine.

*PCI slots:* PCI slots (up to 32 per guest) are used for guest monitors (each video monitor takes a PCI slot), NICs (each NIC takes a PCI slot, "Dual mode" NICs take two slots each) and IDE or VirtIO hard drives. There are also several system devices that use a PCI slot.

## Migrating virtual machines in a cluster

Migrating virtual machines is a network-intensive operation; for hosts running ten or more virtual machines, migrating all of them can be a long and resource-consuming process. The best course of action consists in setting up a policy that best suits both setup and requirements. For example, if a host runs many virtual machines, but only one or two run critical workloads, administrators should set up the option to migrate only highly available virtual machines.

## Demonstration: Creating a new cluster

At installation time, a cluster named "Default" is created automatically. For many sites, there may be no reason to create additional clusters beyond the initial one.

In this example, observe the following steps as the instructor demonstrates the creation of a new cluster.

1. Navigate to the **Clusters** tab.

2. Click the **New** button to display the **New Cluster** dialog.

   The **General** screen allows for setting the required values for creating a new cluster.

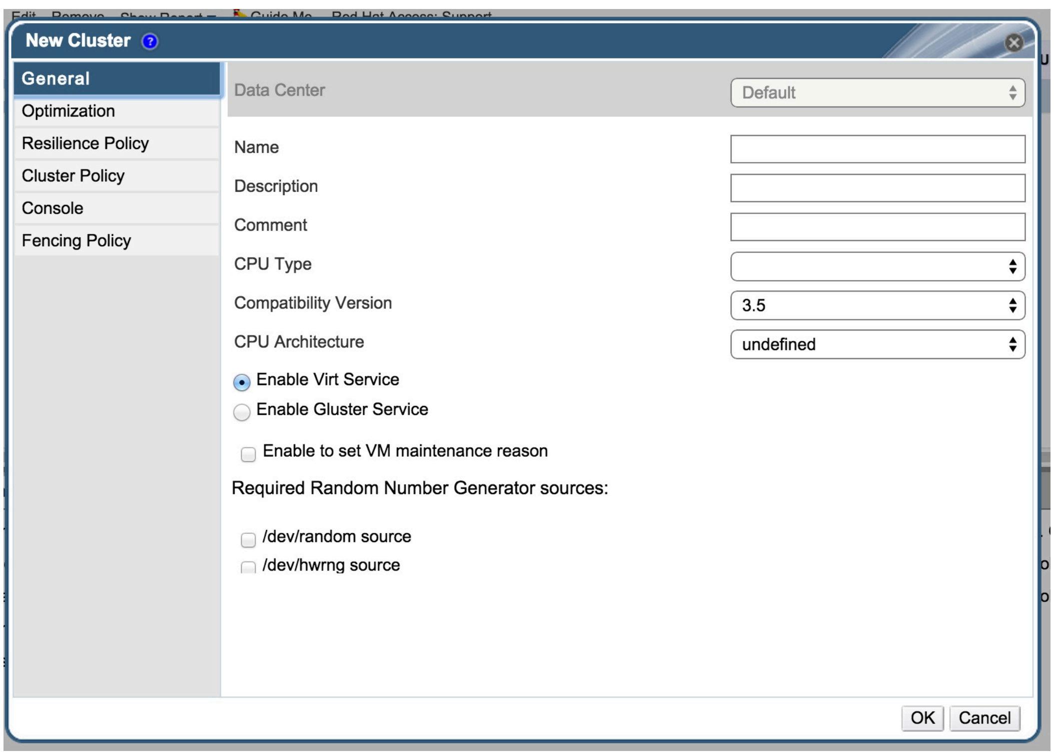


*Figure 4.4: Adding a cluster - general*

| Name | Description |
|---|---|
| Data Center | The name of the datacenter to which this cluster will be registered.<br>`dcenter0` |
| Name | The unique name of the cluster, which can be up to 40 characters.<br>`cluster0` |
| Description | A larger description of the cluster (optional).<br>`Cluster for Pod 0` |
| CPU Type | All hosts within the cluster must run the same type of CPU, identified here. It should be the lowest CPU of any host in the cluster in order to enable migration between hosts.<br>It is possible to check under the **Hosts** tab for the hypervisor host that has been approved to determine the appropriate value here. |
| Compatibility Version | The version of RHEV. Choose 3.5 unless upgrading from an earlier version. In the case of an upgrade, the cluster will be the second item shifted to 3.5 (first the hosts, then the clusters, finally the datacenter).<br>`3.5` |
| Enable Virt Service | Hosts in this cluster will be used to run virtual machines, the default, and should be checked.<br>`checked` |
| Enable Gluster Service | Hosts in this cluster will be used as Red Hat Gluster Storage server nodes and not for running virtual machines. A RHEV-H host cannot be added to this type of cluster.<br>`NOT checked` |
| Enable to set VM maintenance reason | Enables setting a textual reason when switching hosts to maintenance, allowing other admins can see why it was done.<br>`NOT checked` |
| Required Random Number Generator sources | The source for random number generation.<br>`/dev/random` |

The **Memory Optimization** policy will be used to determine whether a virtual machine can be started on a host. Since virtual machines may not be using their full amount of allocated memory, the "overcommit" policy may allow more virtual machines to be allocated to a host than it has physical memory to support. There are three memory optimization policies: **None**, **Server Load** (150%), and **Desktop Load** (200%). The actual percentage of overcommit defined for **Server Load** and **Desktop Load** is adjustable in the RHEV Configuration Tool.

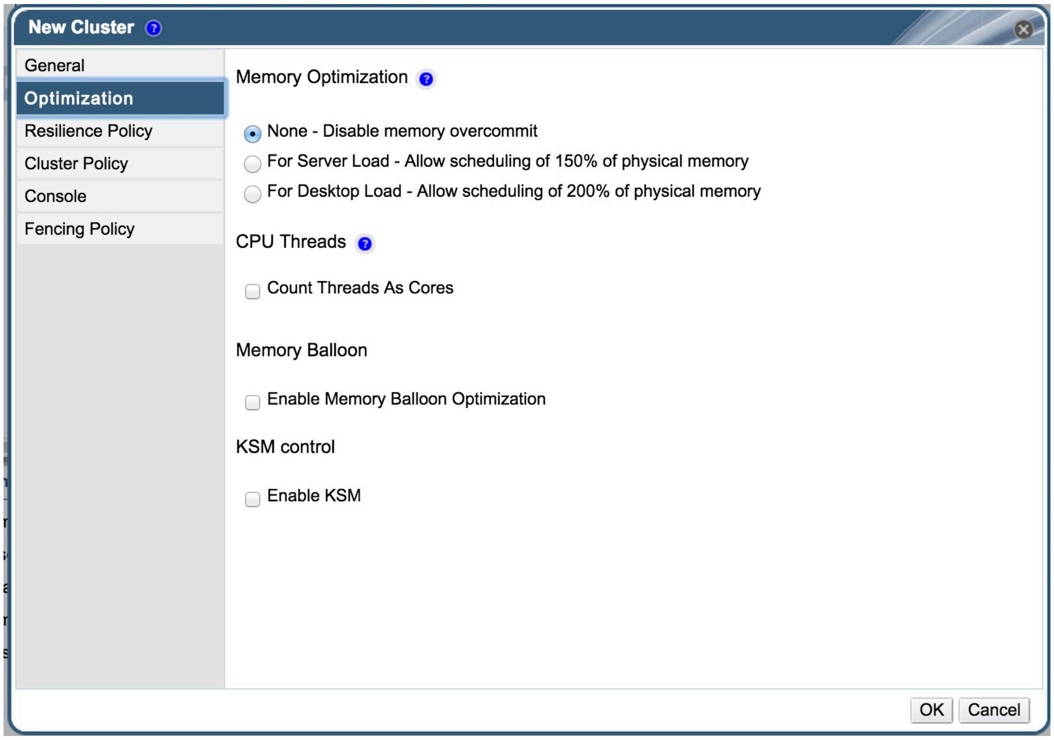


*Figure 4.5: Adding a cluster—memory policy*

| Name | Description |
|---|---|
| **None** | Do not allow any memory overcommit on hosts in this cluster.<br>`None` |
| **Optimized for Server Load** | Allow a certain percentage of memory overcommit on hosts based on "typical" virtual machine server loads. Default: 150%. |
| **Optimized for Desktop Load** | Allow a certain percentage of memory overcommit on hosts based on "typical" virtual machine desktop loads. Default: 200%. |
| **Count Thread As Cores** | Allow guests to use host threads as virtual CPU cores. Enabling this option may be useful for less CPU-intensive workloads. |

| Name | Description |
|---|---|
| | `NOT checked` |
| **Enable Memory Balloon Optimization** | Enables memory overcommitment on virtual machines running on the hosts in this cluster. When this option is set, the Memory Overcommit Manager (**MoM**) will start ballooning where and when possible, with a limitation of the guaranteed memory size of every virtual machine.<br>`Checked` |
| **Enable KSM** | Enables the kernel to examine two or more already running programs and compare their memory. If any memory regions or pages are identical, KSM reduces multiple identical memory pages to a single page.<br>`Checked` |

The **Resilience** policy defines what should happen to a virtual machine should its host go offline. There are three resilience policies: **Migrate**, **Migrate only Highly Available**, and **Do Not Migrate**. Setting a virtual machine as "highly available" can be done per virtual machine (this setting will be seen later when creating virtual machines).

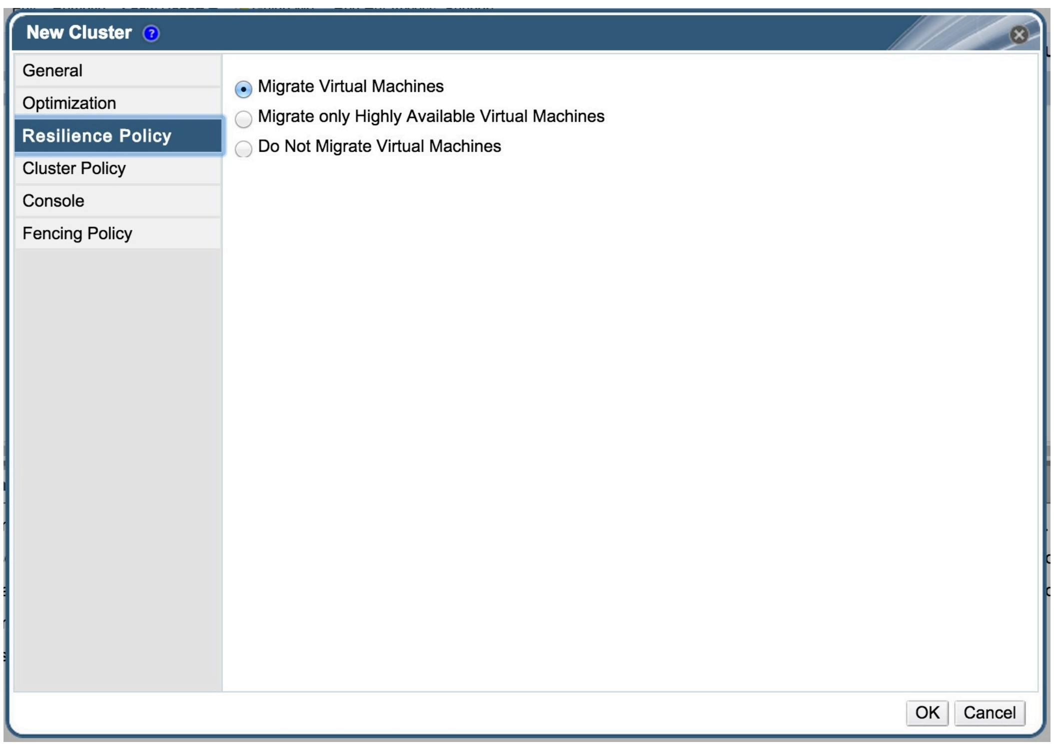


*Figure 4.6: Adding a cluster—resilience policy*

| Name | Description |
|---|---|
| Migrate Virtual Machines | Migrate all virtual machines if the host goes offline.<br>`Migrate Virtual Machines` |
| Migrate only Highly Available Virtual Machines | Selectively migrate (only highly available) virtual machines if the host goes offline. |
| Do Not Migrate Virtual Machines | Do not migrate virtual machines if the host goes offline. Virtual machines become unavailable. |

The **Cluster Policy** defines how virtual machines should be placed upon their creation. For each policy, it is possible to override the default values. Three policies are available.

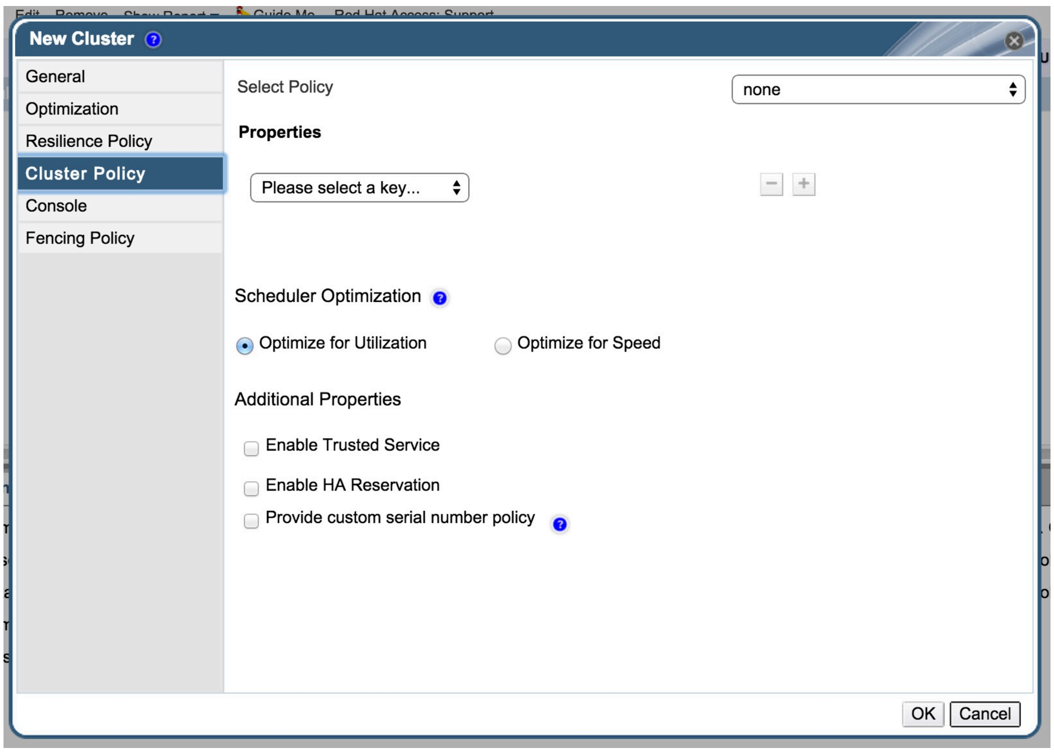


*Figure 4.7: Adding a cluster—cluster policy*

| Name | Description |
|---|---|
| Policy: power_saving | Host for running new virtual machines is selected according to lowest CPU utilization. Moreover, if a host reached the maximum service level and stayed there more than the set time, the virtual machines will be migrated one by one to the next host that has the lowest CPU utilization.<br>`Select Policy: power_saving` |
| Policy: evenly_distributed | If a host reached the maximum service level and stayed there more than the set time, then the virtual machines |

| Name | Description |
|---|---|
| | will be migrated one by one to the next host that has the lowest CPU utilization. |
| Policy: vm_evenly_distributed | Distributes virtual machines evenly between hosts based on a count of the virtual machines. |
| Scheduler optimization | Allows to either optimize the cluster for utilization, by weighting usage and picking the best suited host, or for speed, by checking the number of pending requests per host.<br>`Optimize for speed` |
| Enable Trusted Services | Enable integration with an **`OpenAttestation server`**. Before this can be enabled, the **`engine-config`** tool needs to be used in order to enter the **`OpenAttestation`** server's details.<br>`NOT checked` |
| Enable HA Reservation | Enables RHEV to monitor cluster capacity for highly available virtual machines. The engine ensures that appropriate capacity exists within a cluster for virtual machines designated as highly available to migrate in the event that their existing host fails unexpectedly.<br>`NOT checked` |
| Provide custom serial number policy | Allows specifying a serial number for the virtual machine.<br>`NOT checked` |

The **Console** tab allows administrators to define a SPICE proxy for the cluster.

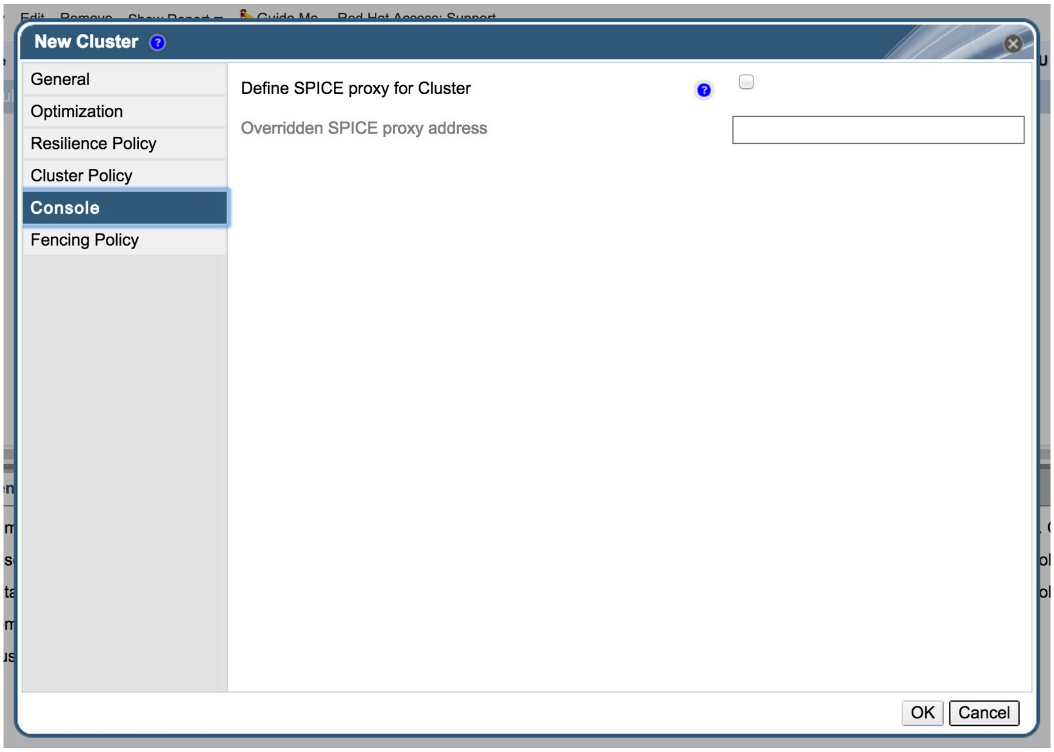


*Figure 4.8: Adding a cluster—cluster console*

The **Fencing Policy** allows administrators to set up fencing for the cluster. Fencing is the process of isolating a node of a computer cluster when a host appears to be malfunctioning.

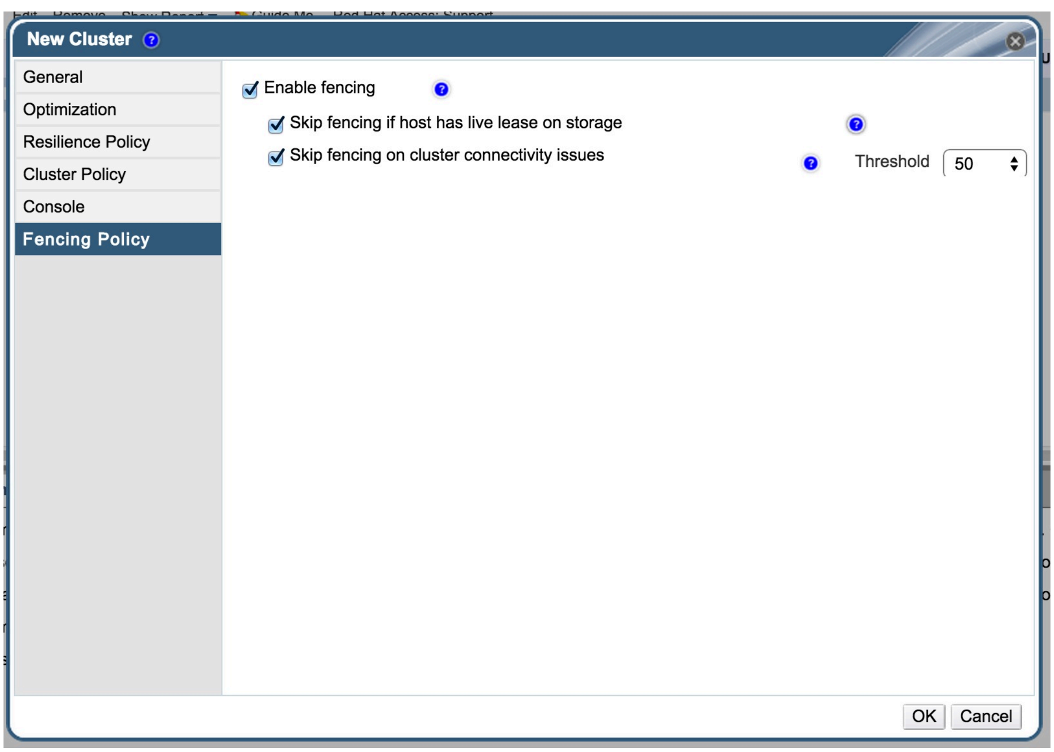

*Figure 4.9: Adding a cluster—cluster fencing*

| Name | Description |
| --- | --- |
| **Enable fencing** | Allows fencing for this cluster. Requires a fencing device in order to work.<br>`NOT checked` |
| **Skip fencing if host has live lease on storage** | Enabling this option will skip fencing for a host that has live lease on storage domains. |
| **Skip fencing on cluster connectivity issues** | Enabling this option will skip fencing if the percentage of cluster hosts with connectivity issues is greater than or equal to the defined threshold. |

3. While a **Guide Me** dialog is presented to step through configuring and choosing Hosts, select **Configure Later** as these steps will be performed separately.

4. Note that the new cluster is in the list of clusters.

5. Navigate to the **Hosts** tab.

6. If the host is not yet approved, right-click the host, choose **Approve**, then modify the datacenter and cluster to reflect where the host should be registered.

   If the host is already approved in another datacenter/cluster, right-click the host, and choose **Maintenance** to place the host in "maintenance mode". Right-click the host, choose **Edit**, then modify the datacenter and cluster to reflect where the host should be registered. Finally, right-click the host and choose **Activate** to bring the host online.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 4.1 Introduction to Clusters

Red Hat Enterprise Virtualization 3.5 Installation Guide

- Section 10.4. Clusters

# Practice: Creating Clusters

In this lab, you will create a new cluster named **`newclu`*X*** in datacenter **`newdc`*X*** for RHEV 3.5 with no memory optimization and a resilience policy to migrate with one host named **`servera`**.

**Outcome(s)**
You should be able to create a cluster with appropriate policies and migrate a host to it.

**Before you begin**
This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.pod*X*.example.com
- Administration Portal connects to rhevm.pod*X*.example.com
- An installed RHEV-H node

Perform the following steps:

1. Log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.

   Open a web browser on workstation and navigate to **`https://rhevm.pod`*X*`.example.com`**. Fill in the fields as listed previously.

2. Look Up CPU Type of Host
   Under the **Hosts** tab, select the **`servera.pod`*X*`.example.com`** host.

   In the lower pane, select the **Hardware Information** tab. Take note of the **CPU Type:** for use later.

3. Create a New Cluster
   3.1. Click the **Clusters** tab.

   3.2. Create a new cluster with the following values:

   - Data Center: **`newdc`*X*** (where *X* is your pod number).
   - Name: **`newclu`*X*** (where *X* is your pod number).
   - Description: **`New Cluster`**
   - CPU Name: *`Depends on classroom hardware`* (Use the value found in the earlier step).
   - Compatibility Version: **`3.5`**
   - For storage: **`Enable Virt Service`**
   - Memory Optimization: **None**
   - Resilience Policy: **`Migrate Virtual Machines`**

   Click the **New** button. The "New Cluster" dialog displays.

Complete the information using the previous values.

Click **OK**. The "Guide Me" dialog opens, displaying a list of configuration tasks.

Click **Configure Later** to close the dialog.

4. Add (Move) the RHEV-H Host (**`servera`**) to the New Cluster (newclu***X***)
   4.1. Navigate to **System**.

   4.2. Click the **Hosts** tab.

   4.3. Put the host **`servera.podX.example.com`** into **maintenance** mode.

   Select the host **`servera.podX.example.com`**, then click the **Maintenance** button. Click **OK** at the "Maintenance Host(s)" window that opens to confirm putting the host into maintenance mode.

   The host is put into *maintenance* mode so that we can edit its properties and move it from the **`Default`** cluster to the new **`newcluX`** cluster.

   4.4. Set the following options for the host properties:

   - Data Center: **`newdcX`**

   - Host Cluster: **`newcluX`**

   Highlight the host **`servera.podX.example.com`**, then click the **Edit** button. The "Edit Host" window opens.

   Modify the two pulldown menus with the previous settings.

   Click **OK** to save the settings. Click **OK** to ignore the "Power Management Configuration" warning.

   4.5. Activate the host **`servera.podX.example.com`** with the new settings.

   With the host **`servera.podX.example.com`** still highlighted, click the **Activate** button.

5. Verification of Cluster and Host from Cluster View
   5.1. Click the **Clusters** tab.

   5.2. Select the cluster **`newcluX`**.

   5.3. Click the *lower* **Hosts** tab and confirm that **`servera.podX.example.com`** appears in the list with a status of **`Up`**.

# Lab: Creating RHEV Datacenters and Clusters

In this lab, you will create a datacenter, **`testdc`*X***, populate it with a cluster, **`testclu`*X***, and a host, **`servera`**, in the cluster.

#### Outcome(s)

You should be able to establish an available datacenter, with a cluster, comprised of one host.

#### Before you begin

This lab requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.pod*X*.example.com
- Administration Portal connects to rhevm.pod*X*.example.com
- An installed RHEV-H node

#### Lab outline

Configure RHEV according to the following specifications:

- Create a datacenter, **`testdc`*X***, that will use shared storage for a RHEV 3.5 environment.
- Create a cluster in the datacenter named **`testclu`*X***.
- Move your host servera.pod*X*.example.com from the datacenter **newdc*X*** to the newly created datacenter.

1. Log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.
2. Create a New Datacenter
3. Create a New Cluster
4. Move the RHEV-H Host (**`servera`**) to the New Cluster (testclu***X***)
5. Verification of Cluster and Host from Cluster View

# Solution

In this lab, you will create a datacenter, **`testdcX`**, populate it with a cluster, **`testcluX`**, and a host, **`servera`**, in the cluster.

**Outcome(s)**

You should be able to establish an available datacenter, with a cluster, comprised of one host.

**Before you begin**

This lab requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.pod*X*.example.com
- Administration Portal connects to rhevm.pod*X*.example.com
- An installed RHEV-H node

**Lab outline**

Configure RHEV according to the following specifications:

- Create a datacenter, **`testdcX`**, that will use shared storage for a RHEV 3.5 environment.
- Create a cluster in the datacenter named **`testcluX`**.
- Move your host servera.pod*X*.example.com from the datacenter **newdc*X*** to the newly created datacenter.

1. Log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.

   Open a web browser on workstation and navigate to **`https://rhevm.podX.example.com`**. Fill in the fields as listed previously.

2. Create a New Datacenter

   2.1. Navigate to **Data Centers**

   Click the **Data Centers** tab.

   2.2. Create a new datacenter with the following values:

   - Name: **`testdcX`** (where *X* is your pod number).
   - Description: **`Test Datacenter`**
   - Storage Type: **`Shared`**
   - Compatibility Version: **`3.5`**
   - Quota Mode: **`Disabled`**

   Click the **New** button. The "New Data Center" dialog displays.

   Complete the information using the previous values.

   Click **OK**.

The "Guide Me" dialog opens, displaying a list of configuration tasks. Click **Configure Later** to close the dialog.

3. Create a New Cluster
   3.1. Navigate to **Clusters**

   Click the **Clusters** tab.

   3.2. Create a new cluster with the following values:

   - Data Center: **`testdc`*****`X`*** (where *X* is your pod number).
   - Name: **`testclu`*****`X`*** (where *X* is your pod number).
   - Description: **`Test Cluster`**
   - CPU Name: *`Depends on your hardware`*
   - Compatibility Version: **`3.5`**
   - For storage: **`Enable Virt Service`**
   - Memory Optimization: **`None`**
   - Resilience Policy: **`Migrate Virtual Machines`**

   Click the **New** button. The "New Cluster" dialog displays.

   Complete the information using the previous values.

   Click **OK**. The "Guide Me" dialog opens, displaying a list of configuration tasks.

   Click **Configure Later** to close the dialog.

4. Move the RHEV-H Host (**`servera`**) to the New Cluster (testclu***X***)
   4.1. Navigate to **System** from the left pane. Click the **Hosts** tab.

   4.2. Put the host **`servera.pod`*****`X`*****`.example.com`** into **Maintenance** mode.

   Select the host **`servera.pod`*****`X`*****`.example.com`**, then click the **Maintenance** button. Click **OK** at the "Maintenance Host(s)" window that opens to confirm putting the host into maintenance mode.

   The host is put into *maintenance* mode so its properties can be edited and it can be moved from **`newclu`*****`X`*** cluster to the new **`testclu`*****`X`*** cluster.

   4.3. Click the **System** top-level entry on the left pane, then click the **Hosts** tab. Set the following:

   - Data Center: **`testdc`*****`X`***
   - Host Cluster: **`testclu`*****`X`***

   With the host **`servera.pod`*****`X`*****`.example.com`** still highlighted, click the **Edit** button. The "Edit Host" window opens.

Modify the two pulldown menus with the previous settings.

Click **OK** to save the settings. Click **OK** to ignore the "Power Management Configuration" warning.

4.4. Activate the host **`servera.podX.example.com`** with the new settings.

With the host **`servera.podX.example.com`** still highlighted, click the **Activate** button.

5. Verification of Cluster and Host from Cluster View

5.1. Click the **Clusters** tab.

5.2. Select the cluster **`testcluX`**.

5.3. Click the *lower* **Hosts** tab and confirm that **`servera.podX.example.com`** appears in the list with a status of **`Up`**.

# Summary

### RHEV Resources

In this section, you learned:

- There are many resources in RHEV, including, but not limited to, datacenters, clusters, pools, and templates.
- The RHEV system scheduler manages the allocation of physical resources within the datacenter.

### Creating Datacenters

In this section, you learned:

- RHEV datacenters are comprised of clusters and networks, as well as virtual machines.
- It is possible to reinitialize a datacenter in order to reuse it from a clean state.
- A RHEV datacenter hosts the various RHEV logical resources.

### Creating Clusters

In this section, you learned:

- Running virtual machines requires you to define a cluster within a datacenter, and adjust settings accordingly.
- To ensure performance for a cluster, various settings related to memory optimization, CPU threads, memory balloon, and KSM control are available.
- In order to be able to create and run virtual machines, a host needs to be integrated into a cluster.

# CHAPTER 5

# CREATING AND MAINTAINING RHEV STORAGE

| Overview | |
|---|---|
| **Goal** | Create and administer **`data`**, **`iso`**, and **`export`** storage domains. |
| **Objectives** | • List the three types of storage domains and supported underlying storage technologies.<br>• Identify the role of the storage pool manager.<br>• Prepare, create and populate RHEV storage domains.<br>• List the storage integration options with Red Hat Enterprise Linux OpenStack Platform. |
| **Sections** | • Creating Storage Domains (and Practice)<br>• Integrating Storage with OpenStack (and Quiz) |
| **Lab** | • Creating and Maintaining RHEV Storage |

# Creating Storage Domains

## Objectives

After completing this section, students should be able to:

- Define the three types of storage domains that may be created.
- Define the storage pool manager.
- Create an NFS and iSCSI storage domain.

## Creating storage domains

A storage domain is a centrally accessed repository for disk images and metadata; they keep virtual machine data, metadata, ISO images, snapshots, and other data accessible to all hosts in a datacenter. Setting up storage is a **prerequisite** for a new datacenter: a datacenter cannot be initialized unless storage domains are attached and activated. In order to add a new storage domain (other than ISO added during the installation), administrators must have at least one host which is marked with an **Up** status in the infrastructure.

There are three types of storage domains that may be created:

- **`Data`** storage domain: Stores the hard disk images of all virtual machines. Disk images may contain an installed operating system or data stored or generated by a virtual machine.
- **`Export`** storage domain: Provides transitory storage for hard disk images and virtual machine templates being transferred between datacenters. Additionally, export storage domains store backed up copies of virtual machines.
- **`ISO`** storage domain: Stores ISO files, also called images. ISO files are representations of physical CDs or DVDs.

RHEV can use various types of storage, which include:

- Network File System (NFS).
- Other POSIX-compliant file systems.
- Gluster FS.
- Internet Small Computer System Interface (iSCSI).
- Local storage attached directly to the virtualization hosts.
- Fiber Channel Protocol (FCP).

Once the NFS, iSCSI, or Fibre Channel storage is configured, it can be added to RHEV through the RHEV-M interface. As has been mentioned previously, there must be at least one host configured to add a storage domain.

## The storage pool manager (SPM)

RHEV uses metadata to describe the internal structure of storage domains, and the hosts access the storage domain metadata in read-only mode, while there is only a single writer. The host that can make changes to the structure of the data domain is known as the storage pool manager (SPM). The SPM coordinates all metadata changes in the datacenter. All other hosts can only read storage domain structural metadata. Every host can be given an SPM priority, and can be manually selected as SPM; otherwise, it is being selected by RHEV.

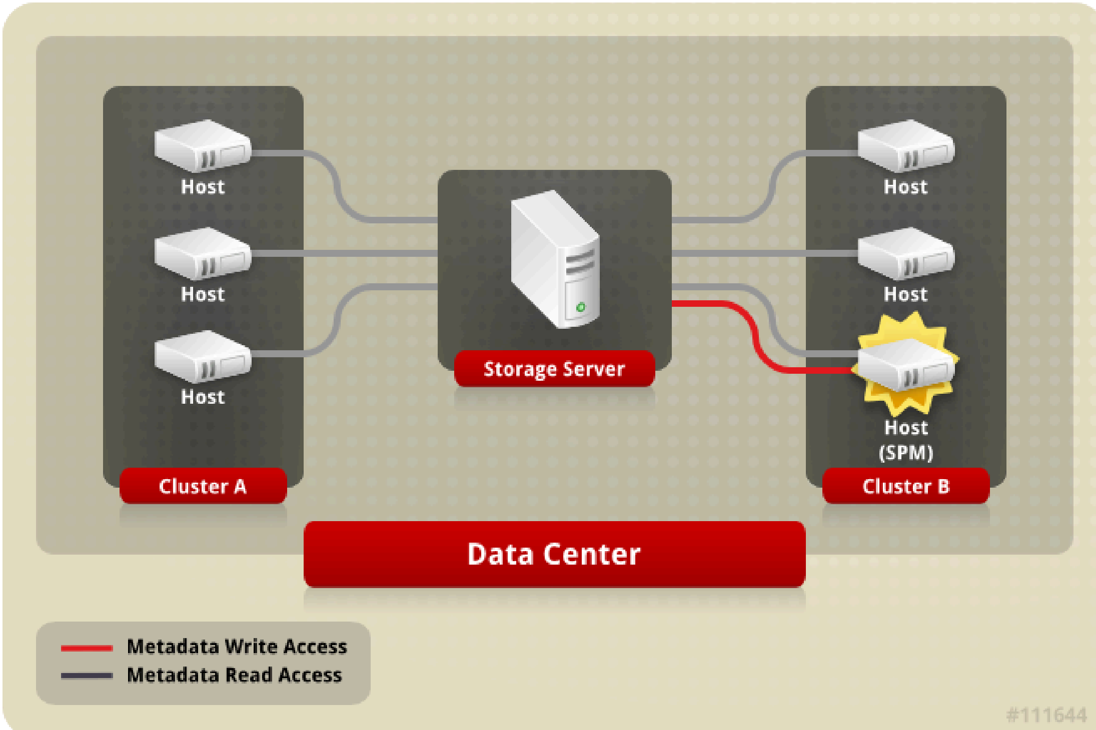


*Figure 5.1: The storage pool manager*

# Virtual disks

There are two types of virtual disks: **sparse**, and **preallocated**, and each works differently. The available formats for storage domains are either **`raw`** or **`qcow2`**.

- **Preallocated** or **sparse**: A preallocated virtual disk has reserved storage of the same size as the virtual disk itself. This results in better performance, as no storage allocation is required during runtime. On **SAN** (such as **`iSCSI`**, or **FCP**), this is achieved by creating a block device with the same size as the virtual disk. On NFS, this is achieved by filling the backing file with zeros and assuming that backing storage is not **`qcow2`** and does not deduplicate zeros.

  For the sparse format, virtual disks backing storage are not reserved and are allocated as needed during runtime. This allows for storage over commitment, under the assumption that most disks are not fully utilized and storage capacity can be utilized better. This requires the backing storage to monitor write requests, and can cause some performance issues. On NFS-backing storage it is achieved by using files. On SAN devices, this is achieved by creating a block device smaller than the virtual disk's defined size, and communicating with the hypervisor to monitor necessary allocations. This does not require support from the underlying storage devices.

- **Raw**: For raw virtual disks, the backing storage device (file/block device) is presented as is to the virtual machine, with no additional layering in between. This gives better performance but has several limitations. For possible combinations of storage types and formats, consider the following table.

| Storage | Format | Type | Description |
|---|---|---|---|
| **NFS** | Raw | Preallocated | A file whose initial size is the size defined for the virtual disk and has no formatting. |
| **NFS** | Raw | Sparse | A file whose initial size is close to zero and has no formatting. |
| **NFS** | Qcow2 | Preallocated | A file whose initial size is the size defined for the virtual disk and has qcow2 formatting. |
| **NFS** | Qcow2 | Sparse | A file whose initial size is close to zero and has qcow2 formatting. |
| **SAN** | Raw | Preallocated | A block device whose initial size is the size defined for the virtual disk and has no formatting. |
| **SAN** | Qcow2 | Preallocated | A block device whose initial size is the size defined for the virtual disk and has qcow2 formatting. Not useful, but possible. |
| **SAN** | Qcow2 | Sparse | A block device whose initial size is much smaller than the size defined for the VDisk (currently 1 GB) and has qcow2 formatting and for which space is allocated as needed (currently in 1-GB increments). |

## Preparing NFS storage

For the NFS ISO domain and any NFS data domains, administrators must ensure the appropriate file system is exported properly, with the correct file ownership, before attempting to add it as a storage domain in RHEV-M.

### Important

The storage server should *not* be one of the RHEV nodes. Ideally, this server should be highly reliable, as it is a single point of failure for the cluster it is attached to.

On the storage server, administrators need either to create or identify the file system that will be exported for use as a storage domain. If required, administrators can update the file **`/etc/exports`** on the storage server in order to ensure that the file system is exported in **`read-write`** to all RHEV hosts. For example, if **`/exports/iso0`** is being exported to nodes that are all on the **`172.25.0.0/24`** network, consider the following entry:

```
/exports/iso0  172.25.0.0/255.255.255.0(rw)
```

Administrators need also to make sure the needed daemons, depending on NFS version, are running, such as **rpcbind**, **nfs**, and **nfslock**. The command **`exportfs -r`** needs to be run to activate the changes.

The directory being exported must appear to be owned and writable by user **`vdsm`** and group **`kvm`**; this user and group might not exist on the storage server (they are created as part of the *rhevm* package scripts). They have **`UID 36`** and **`GID 36`**; the permission can be configured by running the following command:

```
[root@rhevm ~]# chown 36:36 /exports/iso0
```

To ensure that newly created directories have **`kvm`** group ownership, administrators need to set the **`user ID`**:

```
[root@rhevm ~]# chmod g+s /exports/iso0
```

> **Note**
>
> The **`engine-setup`** command does not offer the ability to create an ISO storage domain.

## Preparing iSCSI storage

RHEV supports iSCSI storage at two levels:

- **`VG level`**: The volume group level associates a set of predefined logical unit numbers (**`LUNs`**) to a volume group that is assigned to a storage domain. Volume groups cannot be shared between storage domains.
- **`LUN level`**: The LUN level allows an administrator to assign a set of LUNs to a storage domain. LUNs that are attached to one storage domain cannot be attached to another storage domain.

> **Note**
>
> For information regarding the setup and configuration of iSCSI on Red Hat Enterprise Linux, please refer to the Storage Administration Guide and the documentation files included in the *targetcli* package.

## Demonstration: Creating a new iSCSI data storage domain

In this example, observe the following steps as the instructor demonstrates the creation of a new iSCSI data storage domain. You will have an opportunity to create an NFS data storage domain in the performance checklist and an iSCSI data storage domain in the unit test.

1. Log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.
2. Navigate to the **Storage** tab in the RHEV-M Administration Portal.

3. Click the **New Domain** button to display the **New Domain** dialog.

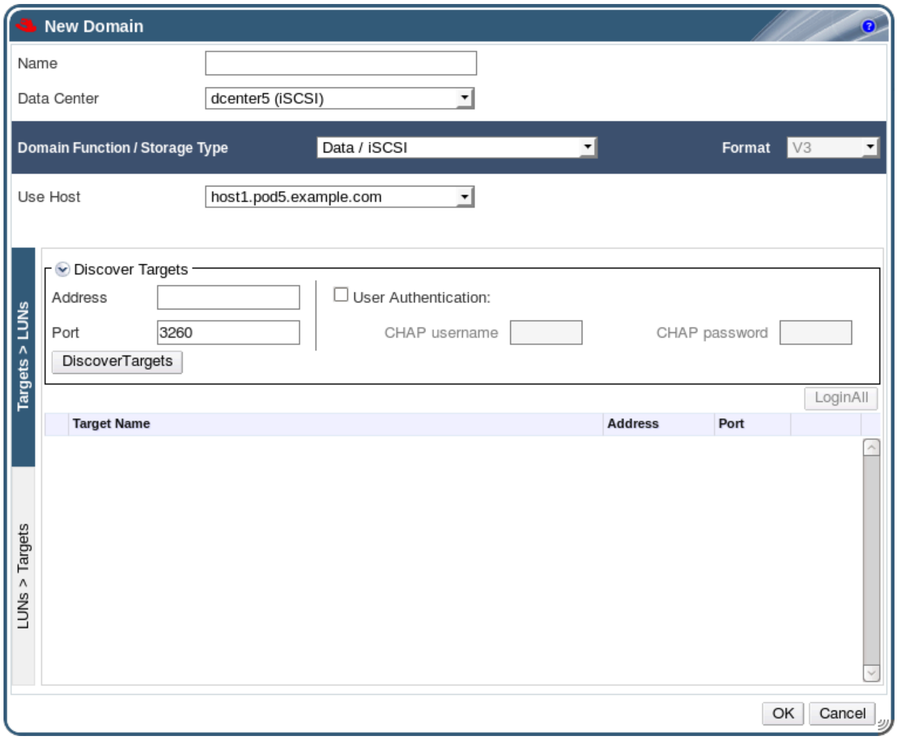


*Figure 5.2: Adding a data storage domain—iSCSI*

| Name | Description |
| --- | --- |
| **Name** | The unique name of the domain, which can be up to 40 characters.<br>`data0` |
| **Data Center** | The name of the datacenter to which this domain is to be registered.<br>`dcenter0` |
| **Domain Function/Storage Type** | The domain function will be **Data**, **ISO**, or **Export** coupled with a storage type of **NFS**, **iSCSI**, **FCP**, or **Local**, the choices being a product of the storage type specified for the datacenter.<br>`Data/iSCSI` |
| **Use Host** | A required active host is used to perform the communication with the storage.<br>`servera.pod`*`X`*`.example.com` |

4. With iSCSI, first *discover* the available targets from a node, so with the **IP Address:** and **Port:** fields filled in, click the **Discover Targets** button.

| **iSCSI settings** | |
|---|---|
| Discover Targets - Address | The IP address of the iSCSI target host.<br>`172.25.X.15` |
| Discover Targets - Port | The port the iSCSI target host is listening on, normally 3260.<br>`3260` |
| Target Name | After iSCSI discovery has been run, a list of possible targets are available.<br>`iqn.2011-10.com.example.rhevm.pod0:iscsi` |
| LUN ID | After **Login** to the target, a list of possible LUNs are available.<br>`Will vary` |

5. Click the newly available **Login** button to attach to that target.

6. Finally, expand the list of LUNs by clicking the newly available **+** ("plus" sign) next to the target, then check the box next to the available LUN.

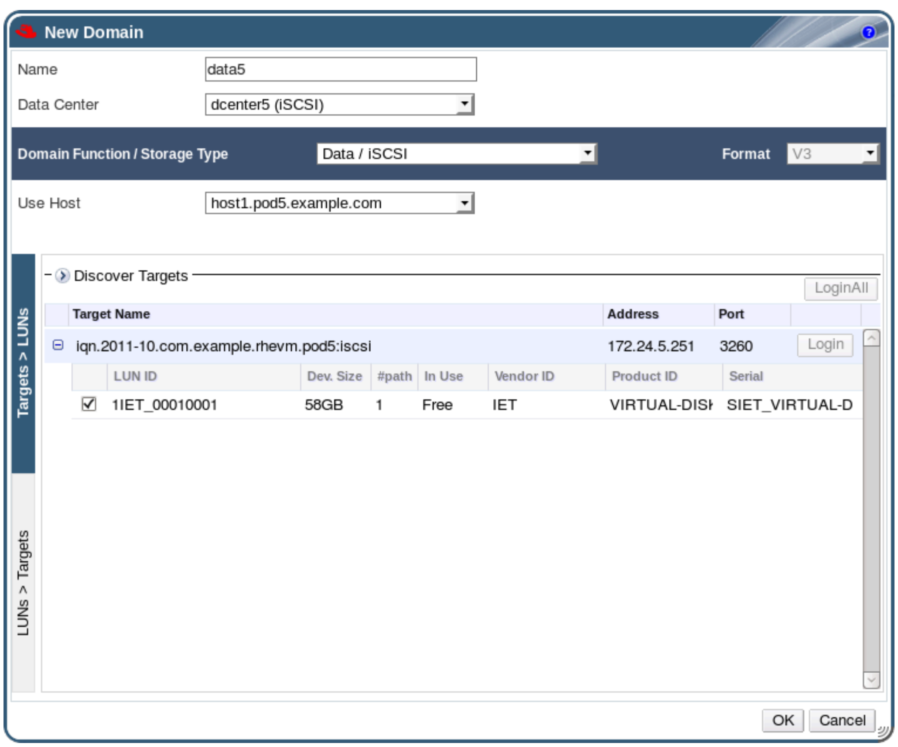

*Figure 5.3: Adding a data storage domain—iSCSI selected*

7. If this is the *first* data storage domain, it will be used as the *master* and will be activated automatically. Normally, the storage domain should appear on a line under the **Storage** tab in the RHEV-M console with a status of "Unattached" and a broken link icon to its left. Simply click the **Attach Domain** button to activate it inside the particular **Data Center**, **Storage** tab.

   Wait for the data storage domain to become active. It is not possible to create another storage domain until the first data domain is active.

# ISO library

*ISO domains* store ISO images (logical CD-ROMs and DVD-ROMs) and **VFDs** (virtual floppy drive images), that are used for the installation or rescue of virtual machines. An ISO domain can be shared across different datacenters. ISO storage domains may only be NFS-based.

# Demonstration: Attaching and populating a new ISO storage domain

In this example, observe the following steps as the instructor demonstrates the creation of a new ISO storage domain.

1. Configure the NFS Server to share the created iso directory to the local network with writable access. The entire **echo** command should be on a single line. Be careful to use a pair of > "greater-than" signs to append to the file, or use an editor to add the data inside the quoted text.

```
[root@rhevm ~]# echo "/exports/rhevisos
 172.25.0.0/255.255.0.0(rw,sync,no_root_squash)" >> /etc/exports

[root@rhevm ~]# exportfs -r
```

2. Log into the RHEV Administration Portal as user **rhevadmin** and password **redhat** in domain **example.com**.

3. Navigate to the **Storage** tab.

4. Click the existing **iso0** storage domain and select the lower tab **Data Center**.

5. Click the **Attach** button in the lower pane, check the **dcenter0** datacenter in the **Attach to Data Center** dialog, then click **OK**.

6. ISO Uploader is a Linux command-line tool that automatically puts files in the correct place and with the correct permissions. ISO Uploader can be slow to upload large ISOs because the image file is copied over NFS (it mounts the NFS share and copies the file to this share).

   ISO Uploader uses the **admin** user from the **internal** domain by default.

   List available ISO domains (though the one to use was just created):

```
[root@rhevm ~]# engine-iso-uploader list
```

```
Please provide the REST API password for the admin@internal RHEV-M user (CTRL+D to
 abort): redhat
ISO Storage Domain Name   | Datacenter                | ISO Domain Status
iso0                      | dcenter0                  | active
```

### Note

The **/etc/rhevm/isouploader.conf** configuration file can be used to include a username, password, or other such data. More information can be found in the **engine-iso-uploader**(8) man page.

7. Upload the following ISOs to the ISO storage domain from the directory available via **http://classroom.example.com/materials**:

   - **rhel-server-7.1-x86_64-boot.iso**

   Upload the ISO with **engine-iso-uploader**:

```
[root@rhevm ~]# wget http://classroom.example.com/materials/rhel-server-7.1-x86_64-
boot.iso
...
[root@rhevm ~]# engine-iso-uploader -i iso0 upload rhel-server-7.1-x86_64-boot.iso
... output omitted ...
```

   An alternate and faster technique than using ISO Uploader is to manually upload the ISO images to the NFS export directly. The part that makes this challenging is that each ISO domain is identified by a unique 128-bit UUID that is embedded into the path on the NFS export where the ISO files are stored. The correct directory at the top level of the ISO export will be something like: ***unique-UUID*/images/*11111111-1111-1111-1111-111111111111*/**. The easiest way to determine the *unique-UUID* is to simply look at the ISO domain prior to the copy. For instance, if the NFS export is **/exports/rhevisos/**, the following commands show the unique UUID (they might differ for other installations) and the path to the image directory:

```
[root@rhevm ~]# ls /exports/rhevisos/
56a00180-8301-4673-a1b5-b31c25686de6
[root@rhevm ~]# ls /exports/rhevisos/56a00180-8301-4673-a1b5-b31c25686de6/
images/11111111-1111-1111-1111-111111111111/
rhel-server-7.1-x86_64-boot.iso  virtio-win_amd64.vfd  virtio-win_x86.vfd
rhev-tools-setup.iso             virtio-win.iso
```

   Upload files directly to that directory. Once there, the files must be owned by user **vdsm** and group **kvm** (**UID 36**/**GID 36**). Once the copy is complete, the file should automatically appear in the RHEV-M interface.

```
[root@rhevm ~]# cd /exports/rhevisos/UUID/
images/11111111-1111-1111-1111-111111111111/
[root@rhevm 11111111-1111-1111-1111-111111111111]# wget http://content.example.com/
rhel7.1/x86_64/isos/rhel-server-7.1-x86_64-dvd.iso
--2015-04-08 13:39:33--  http://content.example.com/rhel7.1/x86_64/isos/rhel-
server-7.1-x86_64-dvd.iso
Resolving content.example.com... 172.25.254.254
Connecting to content.example.com|172.25.254.254|:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 3890216960 (3.6G) [application/octet-stream]
Saving to: “rhel-server-7.1-x86_64-dvd.iso”

100%[===================================>] 3,890,216,960 64.5M/s   in 53s

2015-04-08 13:40:26 (64.6 MB/s) - “rhel-server-7.1-x86_64-dvd.iso” saved
 [3890216960/3890216960]
```

Inspect the content, then use **chown** to update the files' ownership:

```
[root@rhevm 11111111-1111-1111-1111-111111111111]# ls -l
... output omitted ...

[root@rhevm 11111111-1111-1111-1111-111111111111]# chown 36:36 *.iso
```

# Demonstration: Creating a new export storage domain

In this example, observe the following steps as the instructor demonstrates the creation of a new export storage domain.

1. Configure the NFS server to share a created export directory to the local network with writable access. The entire echo command should be on a single line. Be careful to use a pair of > “greater-than” signs to append to the file, or use an editor to add the data inside the quoted text.

```
[root@rhevm ~]# mkdir /exports/exp
[root@rhevm ~]# chown 36:36 /exports/exp
[root@rhevm ~]# echo "/exports/exp 172.25.0.0/255.255.0.0(rw,sync,no_root_squash)"
 >> /etc/exports
[root@rhevm ~]# exportfs -r
```

2. Log into the RHEV Administration Portal as user **rhevadmin** and password **redhat** in domain **example.com**.

3. Navigate to the **Storage** tab.

4. Click the **New Domain** button to display the **New Domain** dialog.

| Name | Description |
|---|---|
| **Name** | The unique name of the domain, which can be up to 40 characters.<br>`exp0` |
| **Data Center** | The name of the datacenter to which this domain is to be registered.<br>`dcenter0` |
| **Domain Function/Storage Type** | The domain function will be **Data**, **ISO**, or **Export** coupled with a storage type of **NFS**, **iSCSI**, **FCP**, or **Local**, the |

| Name | Description |
|---|---|
| | choices being a product of the storage type specified for the datacenter. `Export/NFS` |
| Use Host | A required active host is used to perform the communication with the storage. `servera.podX.example.com` |
| Export Path | For NFS storage, enter the NFS server (IP address or resolvable host name) and exported path to which to connect. `rhevm.podX.example.com:/exports/exp` |

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 7.1. Understanding Storage Domains
- Section 7.3.2. Attaching NFS Storage
- Section 7.6.1. Importing Existing ISO or Export Storage Domains

# Practice: Creating a Data Storage Domain

In this lab, attach a new storage domain.

**Outcome(s)**

- You should be able to establish a data storage domain named **`testdataX`** in datacenter **`testdcX`** using NFS through **`servera`** pointing to **`rhevm.podX.example.com:/exports/testdata`**.
- You should be able to establish an export storage domain named **`testexportX`** in datacenter **`testdcX`** using NFS through **`servera`** pointing to **`rhevm.podX.example.com:/exports/testexport`**.

**Before you begin**

This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.
- Administration Portal connects to rhevm.
- An installed RHEV-H node.
- Datacenter named testdc*X*.
- Cluster named testclu*X* with a single RHEV-H host.

Perform the following steps:

1. Create the NFS exports that will be used for the RHEV storage domains.

   1.1. As **`root`** on **`rhevm`**, create a directory named **`/exports/testdata`**, as well as one named **`/exports/testexport`**, with ownership UID:GID of **`36:36`**.

   ```
   [root@rhevm ~]# mkdir /exports/testdata
   [root@rhevm ~]# mkdir /exports/testexport
   [root@rhevm ~]# chown 36:36 /exports/test*
   ```

   1.2. Configure the NFS server to share the new directories to the local network with writable access.

   The entire **echo** command should be on a single line. Be careful to use a *pair* of > “greater-than” signs to append to the file, or use an editor to add the data inside the quoted text.

   ```
   [root@rhevm ~]# echo "/exports/testdata
    172.25.0.0/255.255.0.0(rw,sync,no_root_squash)" >> /etc/exports

   [root@rhevm ~]# echo "/exports/testexport
    172.25.0.0/255.255.0.0(rw,sync,no_root_squash)" >> /etc/exports

   [root@rhevm ~]# exportfs -r
   ```

2. Create the **`data`** storage domain in your datacenter.

2.1. Log into the RHEV Administration Portal as user **rhevadmin** and password **redhat** in domain **example.com**.

2.2. Navigate to **System**, then click the **Storage** tab.

2.3. Create a new data storage domain with the following values:

- Name: **testdata*X*** (where *X* is your pod number).
- Data Center: **testdc*X*** (where *X* is your pod number).
- Domain Function/Storage Type: **Data / NFS**
- Use Host: **servera.pod*X*.example.com**
- Export path: **rhevm.pod*X*.example.com:/exports/testdata**

Click the **New Domain** button. The "New Domain" dialog displays.

Complete the information using the values mentioned previously.

Click **OK**.

**Patiently wait for the storage to transition to "Active" since this is the first storage domain and the "master".**

3. Create the **export** storage domain in your datacenter.

3.1. Still in **System**, click the **Storage** tab.

3.2. Create a new export storage domain with the following values:

- Name: **testexport*X*** (where *X* is your pod number).
- Data Center: **testdc*X*** (where *X* is your pod number).
- Domain Function/Storage Type: **Export / NFS**
- Use Host: **servera.pod*X*.example.com**
- Export path: **rhevm.pod*X*.example.com:/exports/testexport**

Click the **New Domain** button. The "New Domain" dialog displays.

Complete the information using the values mentioned previously.

Click **OK**.

**Patiently wait for the storage to transition to "Active".**

4. Ensure the storage domains become available for your cluster.

4.1. Click the **Data Centers** tab.

4.2. Select the datacenter **testdc*X***.

4.3. Click the *lower* **Storage** tab and confirm that **`testdataX`**, as well as **`testexportX`**, appear in the list with a status of **`Active`**.

# Integrating Storage with OpenStack

## Objectives

After completing this section, students should be able to:

- List the steps to connect the OpenStack Glance imaging service with RHEV.
- Describe the available features regarding the Glance integration.
- Describe the Cinder integration with the RHEV engine.

## Integrating the Glance image service

Present in Red Hat Enterprise Virtualization 3.5, administrators can interface an external OpenStack environment as an external provider. The OpenStack Glance imaging service is available as the **`OpenStack Image`** provider, and can be configured as an external storage pool. Once Glance is interfaced with RHEV, active Glance images can be retrieved, and imported into a RHEV data center. Upon import, images are available, and new virtual machines can be instantiated with the imported image, or transformed into a template. The Red Hat Enterprise Virtualization engine is also able to export images into Glance, allowing administrators to benefit Glance capabilities for their images' management. The current version already supports authentication and tokens management through the Keystone service.

The following procedure explains how to add an OpenStack image external provider:

1. Select the **External Providers** entry in the tree pane.
2. Click the **Add** button to open the **Add Provider** window.
3. Enter a name and description, then select **`OpenStack Image`** as the provider **Type**.
4. Enter the URL, or fully qualified domain name, of the machine on which the Glance instance is installed, in the **`Provider URL`** text field.
5. Optionally, select the **Requires Authentication** checkbox, and enter the credentials for Glance. Administrators must use the username and password for the Glance user registered in Keystone.

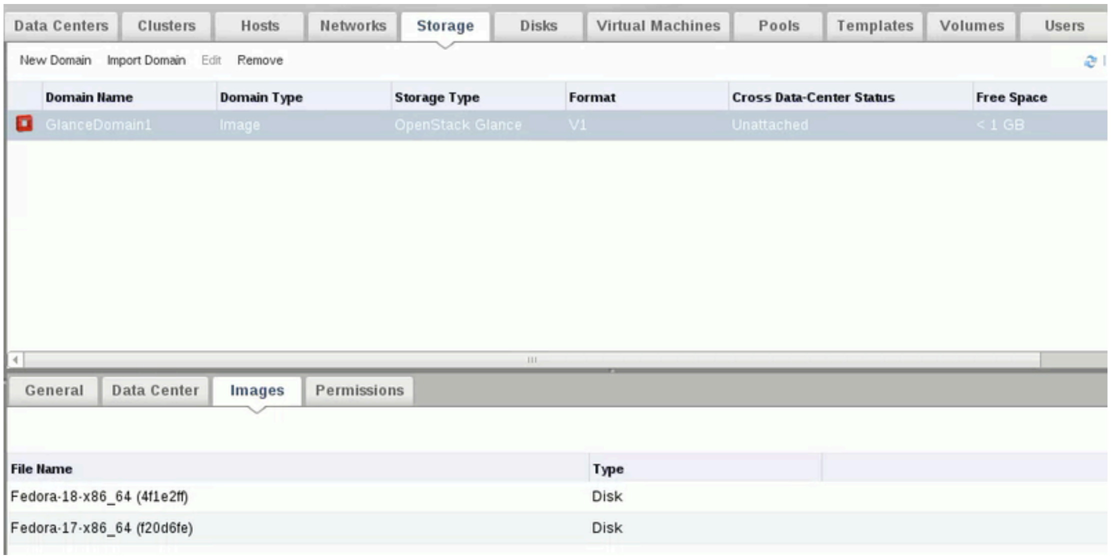


*Figure 5.4: The Glance image service*

# Glance current limitations

This tech preview still has some constraints and limitations, including:

- Only **`raw`** and **`qcow2`** images are supported.
- Over-SSL encrypted communications are not supported.
- It is not currently possible to export images with multiple volumes to Glance.
- Live export is not supported; the virtual machine that uses the disk needs to be shut down prior to the export.
- On file domains, sparse images are not supported for the **`raw`** format; imported images always need to be preallocated.
- It is not possible to pause the importing process and resume it later.
- Deleting images present in Glance is not possible.

# Integrating the Storage volume service

Present as a tech preview, administrators can interface RHEV with the OpenStack volume project, codename Cinder. Cinder is available as an external provider under the **`OpenStack Volume`** provider type. Upon integration, administrators can retrieve the Cinder volumes, and create new virtual disks in a Cinder domain. Created disks can be attached and detached to virtual machines, and support native RHEV features, such as marking disks as bootable or shareable. This current version is only able to manage volumes provisioned by a Ceph storage.

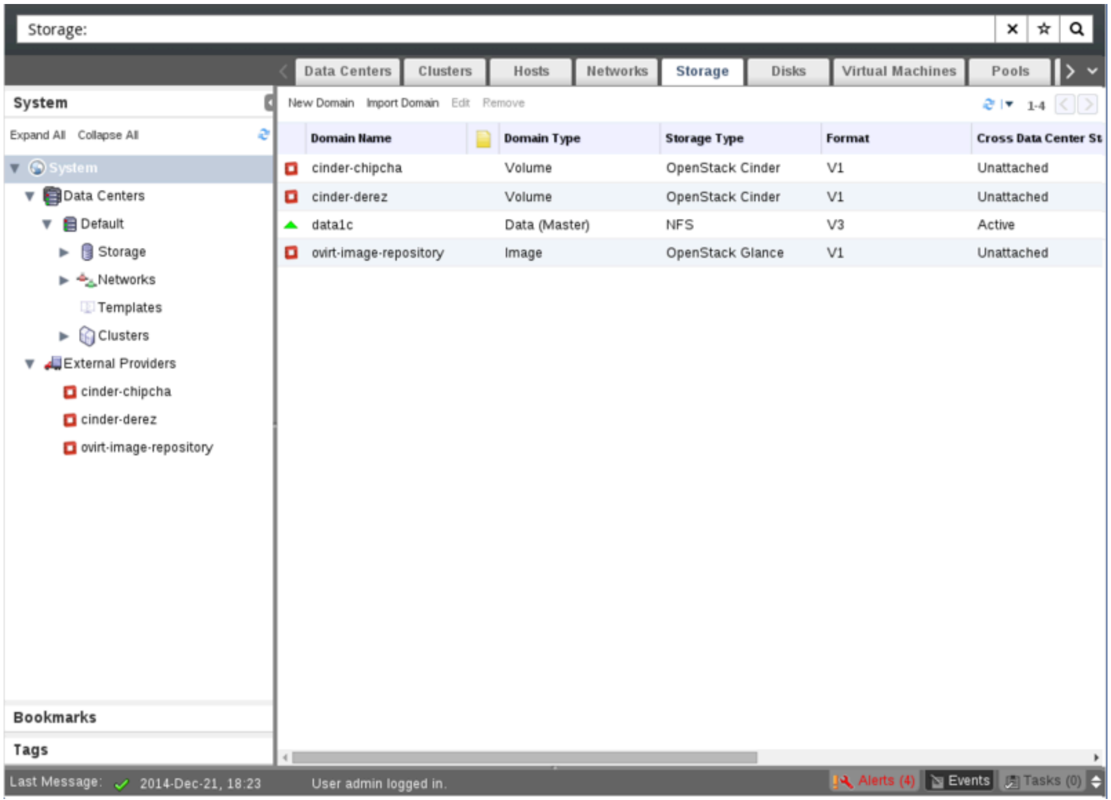


*Figure 5.5: The Cinder image service*

# Future work

The current implementation is actively under development. Some of the new features will include:

- Changing a volume type after its creation.
- Uploading a volume from Cinder to Glance.
- Running **CRUD** operations for volume types.
- Managing Cinder quotas.
- Importing and exporting virtual machines and templates.
- Creating live snapshots.
- Running live storage migrations.
- Synchronizing Cinder data with the RHEV engine database.
- Monitoring Cinder storage domains.
- Supporting multiple Cinder back ends, such as LVM.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 14.3. Adding external providers

> Features/Glance Integration
> http://www.ovirt.org/Features/Glance_Integration

> Features/Cinder Integration
> http://www.ovirt.org/Features/Cinder_Integration

# Quiz: Integrating Storage with OpenStack

Choose the correct answer to the following questions:

1. Name the provider type to use to integrate the Glance image service. (Choose one.)

    a. OpenStack Glance
    b. OpenStack Image
    c. External image provider
    d. OpenStack Image v2

2. Select the current limitations of the Glance integration with the RHEV engine. (Choose three.)

    a. Over-SSL encrypted communications are not supported.
    b. Only **vdmk** images are supported.
    c. Live export is not supported; the virtual machine that uses the disk needs to be shut down prior to the export.
    d. It is not possible to pause the importing process and resume it later.

3. Select the Glance disk formats the RHEV engine supports. (Choose two.)

    a. **iso**
    b. **vhd**
    c. **raw**
    d. **vdi**
    e. **qcow2**

4. Name some of the features that will be available regarding the Cinder integration with the RHEV engine. (Choose three.)

    a. Synchronizing Cinder data with the RHEV engine database.
    b. Running CRUD operations for volume types.
    c. Add support for reset-state on multiple volumes or snapshots at once.
    d. Supporting multiple Cinder back ends, such as LVM.
    e. Update action for snapshot metadata

# Solution

Choose the correct answer to the following questions:

1. Name the provider type to use to integrate the Glance image service. (Choose one.)

    a. OpenStack Glance
    **b. OpenStack Image**
    c. External image provider
    d. OpenStack Image v2

2. Select the current limitations of the Glance integration with the RHEV engine. (Choose three.)

    **a. Over-SSL encrypted communications are not supported.**
    b. Only **vdmk** images are supported.
    **c. Live export is not supported; the virtual machine that uses the disk needs to be shut down prior to the export.**
    **d. It is not possible to pause the importing process and resume it later.**

3. Select the Glance disk formats the RHEV engine supports. (Choose two.)

    a. `iso`
    b. `vhd`
    **c. `raw`**
    d. `vdi`
    **e. `qcow2`**

4. Name some of the features that will be available regarding the Cinder integration with the RHEV engine. (Choose three.)

    **a. Synchronizing Cinder data with the RHEV engine database.**
    **b. Running CRUD operations for volume types.**
    c. Add support for reset-state on multiple volumes or snapshots at once.
    **d. Supporting multiple Cinder back ends, such as LVM.**
    e. Update action for snapshot metadata

# Lab: Creating and Maintaining RHEV Storage

In this lab, you will create a storage domain, **`iscsidataX`**, in your existing cluster. You will also attach and populate the ISO domain created during installation.

### Outcome(s)

You should be able to establish a data storage domain named **`iscsidataX`** in datacenter **`testdcX`** using iSCSI as the storage type and work with an ISO domain.

### Before you begin

This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.
- Administration Portal connects to rhevm.
- An installed RHEV-H node.
- Datacenter named testdc*X*.
- Cluster named testclu*X* with a single RHEV-H host.

### Lab outline

Configure the iSCSI storage domain according to the following specifications:

- Name: **`iscsidataX`** (where *X* is your pod number).
- Data Center: **`testdcX`** (where *X* is your pod number).
- Domain Function/Storage Type: **`Data / iSCSI`**
- Use Host: **`servera.podX.example.com`**
- Target Address: **`rhevm.podX.example.com`**
- User Authentication: **`Disabled`**

1. Log into your RHEV manager from your workstation. Open Firefox, and go to rhevm.pod*X*.example.com. Use **`rhevadmin`**, and **`redhat`** as the password in domain **`example.com`**.
2. Create the **`iscsidataX`** storage domain in your datacenter.
3. Ensure the storage domains become available for your cluster.
4. Correct the NFS access permissions for the **`isoX`** so it is accessible from the classroom environment.
5. Attach the **`isoX`** domain to your datacenter.
6. Add the RHEL 7.1 boot iso available from **`http://classroom.example.com/materials/`** to the ISO domain.

# Solution

In this lab, you will create a storage domain, **`iscsidata`*`X`***, in your existing cluster. You will also attach and populate the ISO domain created during installation.

### Outcome(s)

You should be able to establish a data storage domain named **`iscsidata`*`X`*** in datacenter **`testdc`*`X`*** using iSCSI as the storage type and work with an ISO domain.

### Before you begin

This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.
- Administration Portal connects to rhevm.
- An installed RHEV-H node.
- Datacenter named testdc*X*.
- Cluster named testclu*X* with a single RHEV-H host.

### Lab outline

Configure the iSCSI storage domain according to the following specifications:

- Name: **`iscsidata`*`X`*** (where *X* is your pod number).
- Data Center: **`testdc`*`X`*** (where *X* is your pod number).
- Domain Function/Storage Type: **`Data / iSCSI`**
- Use Host: **`servera.pod`*`X`*`.example.com`**
- Target Address: **`rhevm.pod`*`X`*`.example.com`**
- User Authentication: **`Disabled`**

1. Log into your RHEV manager from your workstation. Open Firefox, and go to rhevm.pod*X*.example.com. Use **`rhevadmin`**, and **`redhat`** as the password in domain **`example.com`**.

2. Create the **`iscsidata`*`X`*** storage domain in your datacenter.

   2.1. Create the data storage domain using the following values:

   - Name: **`iscsidata`*`X`*** (where *X* is your pod number).
   - Data Center: **`testdc`*`X`*** (where *X* is your pod number).
   - Domain Function/Storage Type: **`Data / iSCSI`**
   - Use Host: **`servera.pod`*`X`*`.example.com`**
   - Target Address: **`rhevm.pod`*`X`*`.example.com`**
   - User Authentication: **`Disabled`**

Click the **New Domain** button. The "New Domain" dialog displays.

Complete the information using the values mentioned previously.

2.2. Click **Discover** to discover existing iSCSI targets. A new entry will appear:

```
iqn.2014-10.com.example.rhevm.podX:iscsi
```

Click the arrow, located on the same line as the entry, in order to log in.

2.3. Once logged in, click the tab **LUNs > Targets** in order to list all available targets. Check the only available entry.

2.4. Click **OK** to add the iSCSI storage.

**Patiently wait for the storage to transition to "Active".**

3. Ensure the storage domains become available for your cluster.

3.1. Click the **Data Centers** tab.

3.2. Select the datacenter **testdc*X***.

3.3. Click the *lower* **Storage** tab and confirm that **iscsidata*X*** appears in the list with a status of **Active** and type of **Data (Master)**.

4. Correct the NFS access permissions for the **iso*X*** so it is accessible from the classroom environment.

4.1. Adjust **/etc/exports** and update running NFS server.

```
[root@rhevm ~]# echo "/exports/rhevisos
 172.25.0.0/255.255.0.0(rw,sync,no_root_squash)" >> /etc/exports

[root@rhevm ~]# exportfs -r
```

5. Attach the **iso*X*** domain to your datacenter.

5.1. Click the domain **iso*X***. In the lower pane, select the **Data Center** tab.

5.2. Click the **Attach** link in order to attach the domain to the datacenter. In the window that appears, check the datacenter **testdc*X***. Click **OK** to save your changes.

6. Add the RHEL 7.1 boot iso available from **http://classroom.example.com/materials/** to the ISO domain.

6.1. Retrieve the ISO using **wget**:

```
[root@rhevm ~]# wget http://classroom.example.com/materials/rhel-server-7.1-
x86_64-boot.iso
```

6.2. Upload the image in the **iso*X*** datastore:

```
[root@rhevm ~]# engine-iso-uploader -i isoX upload rhel-server-7.1-x86_64-
boot.iso
Please provide the REST API password for the admin@internal oVirt Engine user
 (CTRL+D to abort): redhat
Uploading, please wait...
INFO: Start uploading rhel-server-7.1-x86_64-boot.iso
...
INFO: rhel-server-7.1-x86_64-boot.iso uploaded successfully
```

# Summary

**Creating Storage Domains**
In this section, you learned:

- RHEV defines three storage domains in order to host various elements, including:
  - Images
  - Snapshots
  - Templates
- Raw and qcow2 storage formats in raw or preallocated types can be defined for virtual disks.
- The storage pool manager (**`SPM`**) is used to manage storage metadata.
- NFS and iSCSI, as well as other protocols, can be used for storage domains.

**Integrating Storage with OpenStack**
In this section, you learned:

- Red Hat OpenStack Platform storage components of Glance and Cinder can be integrated with Red Hat Enterprise Virtualization.

# CHAPTER 6

# CREATING RHEV LOGICAL NETWORKS

| Overview | |
|---|---|
| **Goal** | Create RHEV Logical Networks in the RHEV Manager interface. |
| **Objectives** | • Plan for and create RHEV logical networks.<br>• Discuss network integration options and usage with Red Hat Enterprise Linux OpenStack Platform. |
| **Sections** | • Creating Logical Networks (and Practice)<br>• Integrating Networking with Openstack (and Quiz) |
| **Lab** | • Creating RHEV Logical Networks |

# Creating Logical Networks

## Objectives

After completing this section, students should be able to:

- Discuss the various RHEV networks.
- Manage system permissions for logical networks.
- Plan for and create RHEV logical networks.

## Logical networks

A **logical** network is a named set of global network connectivity properties in a datacenter. When a logical network is added to a host, it may be further configured with host-specific network parameters. Logical networks optimize network flow by grouping network traffic by usage, type, and requirements, and they allow both **connectivity** and **segregation**. Administrators can create logical networks for storage communication, in order to optimize network traffic between hosts and storage domains; a logical network specifically for all virtual machine traffic; or multiple logical networks to carry the traffic of groups of virtual machines. The default logical network in all datacenters is the management network called **rhevm**. The **rhevm** network carries all traffic, until another logical network is created. It is meant especially for management communication between the Red Hat Enterprise Virtualization Manager and hosts. A logical network is a datacenter-level resource; creating one in a data center makes it available to the clusters in a datacenter. A logical network that has been designated as required must be configured in all of a cluster's hosts before it is operational. Optional networks can be used by any host they have been added to.

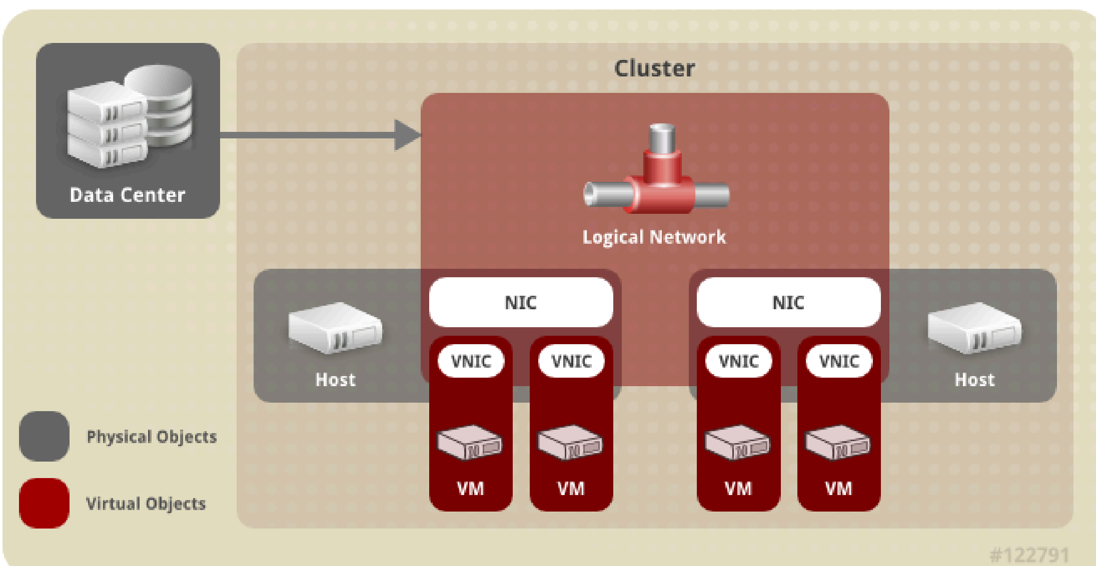


*Figure 6.1: RHEV logical networks*

The segregation in to logical networks is done based on the physical topology of the network and various functional requirements. There is no requirement to set up multiple logical networks; if administrators do not wish to segregate the traffic, they can allow all network traffic to use the default network configured for the datacenter at installation time. The predefined logical network is named **rhevm** and it is labeled with the description "Management Network".

> **Warning**
>
> Do not change networking in a datacenter or a cluster if any hosts are running, as this risks making the host unreachable.

A logical network may belong to multiple clusters and this is one way for virtual machines and nodes in multiple clusters to communicate with each other. Each cluster, in turn, may have a different set of logical networks available to it. However, all logical networks in use must be defined by the datacenter.

A common practice for defining logical networks would have administrators creating three logical networks with corresponding physical segregation:

1. A "public network" connecting the gateway router, the **RHEV-M** system, and the **RHEV-H** nodes.
2. A "management network" connecting the **RHEV-M** system to the **RHEV-H** nodes, but is not routed. For security reasons, one could restrict **ssh** logins to the management network.
3. A "storage network" connecting the **RHEV-H** nodes to the NFS or iSCSI storage domains and is also connected to the **RHEV-M** node. Unlike NFS and iSCSI, Fibre Channel storage uses a SAN and not the IP network for communication, so a separate network would not be necessary.

All hosts in a cluster must have the same network configuration. Otherwise, there would be a risk of network or communication errors when virtual machines migrate from one host to another.

# Port mirroring

Port **mirroring** copies layer 3 network traffic on a given logical network and host to a virtual interface on a virtual machine. This virtual machine can be used for network debugging and tuning, intrusion detection, and monitoring the behavior of other virtual machines on the same host and logical network. The only traffic copied is internal to one logical network on one host. There is no increase on traffic on the network external to the host; however, a virtual machine with port mirroring enabled uses more host CPU and RAM than other virtual machines. Port mirroring requires an IPv4 IP address, and **Hotplugging** profiles are not supported. As of Red Hat Enterprise Virtualization 3.4, port mirroring has been included in **vNIC** profiles. Port mirroring cannot be altered when the **vNIC** profile associated with port mirroring is attached to a virtual machine. To use port mirroring, administrators need to create a dedicated **vNIC** profile that has port mirroring enabled.

# Required networks, optional networks, and virtual machine networks

Red Hat Enterprise Virtualization 3.1 and higher distinguishes between *required* networks and *optional* networks.

- Required networks must be applied to all hosts in a cluster for the cluster and network to be operational. Logical networks are added to clusters as required networks by default. When a host's required network becomes nonoperational, virtual machines running on that host are migrated to another host; the extent of this migration is dependent upon the chosen cluster policy. This is beneficial if administrators have machines running mission-critical workloads. When a nonrequired network becomes non-operational, the virtual machines running on the network are not migrated to another host. This prevents unnecessary I/O overload caused by mass migrations.

- Optional networks are those logical networks that have not been explicitly declared required networks. Optional networks can be implemented on only the hosts that use them. The presence or absence of these networks does not affect the operational status of a host.

- Virtual machine networks (called a **`VM network`** in the user interface) are logical networks designated to carry only virtual machine network traffic. Virtual machine networks can be required or optional.

> **Note**
>
> A virtual machine with a network interface on an optional virtual machine network will not start on a host without the network.

# Managing system permissions for a network

As the **`SuperUser`**, the system administrator manages all aspects of the Administration Portal. More specific administrative roles can be assigned to other users. These restricted administrator roles are useful for granting a user administrative privileges that limit them to a specific resource. For example, a **`DataCenterAdmin`** role has administrator privileges only for the assigned datacenter with the exception of the storage for that datacenter, and a **`ClusterAdmin`** has administrator privileges only for the assigned cluster. A **`network administrator`** is a system administration role that can be applied for a specific network, or for all networks on a datacenter, cluster, host, virtual machine, or template. A network user can perform limited administration roles, such as viewing and attaching networks on a specific virtual machine or template. Administrators can use the **`Configure`** button in the header bar to assign a network administrator for all networks in the environment. The network administrator role permits the following actions:

- Create, edit and remove networks.

- Edit the configuration of the network, including configuring port mirroring.

- Attach and detach networks from resources including clusters and virtual machines.

> **Note**
>
> The user who creates a network is automatically assigned **`NetworkAdmin`** permissions on the created network. It is also possible to change the administrator of a network by removing the existing administrator and adding the new administrator.

# Demonstration: Creating a new logical network

In this example, observe the following steps as the instructor demonstrates how to create a new logical network.

1. Navigate to the **Data Centers** tab in the **RHEV-M** Administration Portal.

2. Select the datacenter you want to add the logical network to from the list of existing datacenters.

   ```
   dcenter0
   ```

3. Click the **Logical Networks** tab displayed in the bottom frame.

4. Click the **New** button to display the **New Logical Network** dialog (shown as follows).

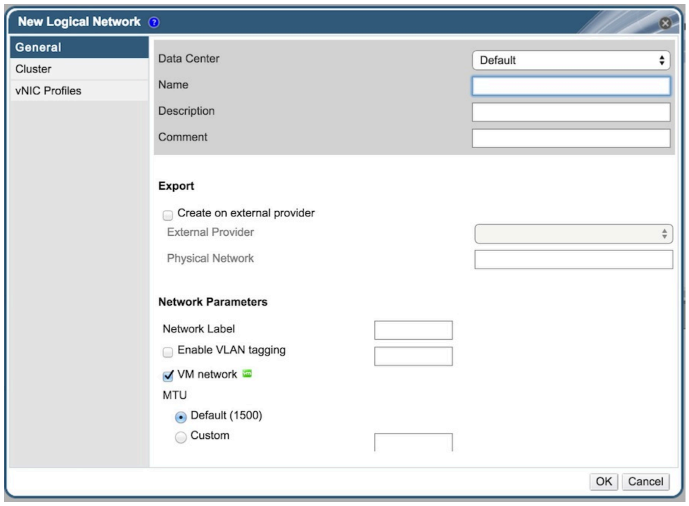


*Figure 6.2: Adding a logical network*

Use this dialog to assign a name and a more useful description to the logical network that you are creating. You can either select all of the clusters or you can select individual clusters in the datacenter to attach to the logical network.

Click the **Cancel** button.

# Demonstration: Configuring network MAC address pool

RHEV-M provides a prepopulated range of MAC addresses to assign to network cards provisioned to virtual machines. The default range of available MAC addresses will be chosen automatically at RHEV-M install time.

In this example, observe the following steps as the instructor demonstrates how to configure the range of available MAC addresses for virtual machines.

1. Open a terminal on rhevm.pod*X*.example.com, and find the RHEV setting for MAC address ranges:

   ```
   [root@rhevm ~]# engine-config -l | grep -i mac
   ...
   MacPoolRanges: "MAC Addresses Pool Ranges" (Value Type: String)
   ...
   ```

2. View the current settings.

```
[root@rhevm ~]# engine-config -g MacPoolRanges
MacPoolRanges: 00:1A:4A:01:00:00-00:1A:4A:01:00:FF version: general
```

3. The DHCP server is configured to provide IP addresses to machines based on the MAC addresses. Students must have the proper MAC address range to get the proper IP addresses for their network.

   Set the MAC range to **52:54:00:00:00:10-52:54:00:00:00:FF**.

```
[root@rhevm ~]# engine-config -s MacPoolRanges=52:54:00:00:00:10-52:54:00:00:00:FF
[root@rhevm ~]# service ovirt-engine restart
```

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 5.1. Introduction to Logical Networks
- Section 5.5. Logical Network Tasks

# Practice: Creating Logical Networks

In this lab, you will define a new logical network for your existing datacenter.

**Outcome(s)**
You should be able to create a new logical network, **`VMNet`**, for your datacenter **`testdcX`**.

**Before you begin**
This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.pod*X*.example.com.
- Administration Portal connects to rhevm.pod*X*.example.com.
- An installed RHEV-H node.

Perform the following steps:

1. Launch the RHEV-M interface, using Firefox on workstation, and access **`http://rhevm.podX.example.com/`**. Click the **Administration Portal** link. Log in using **`rhevadmin`** for the username, with a password of **`redhat`**, and using **`example.com`** for the domain.
2. From the left frame, navigate to **System** > **Data Centers** > **testdc*X***, then click the **Networks** entry.
3. From the right frame, click the tab **Networks** to access the network configuration for the datacenter. Click **New** to create a new network.
4. On the window that appears, fill in the fields using the following information:

| Tab | Field | Value |
|---|---|---|
| General | Name | **`VMNet`** |
| | Description | **`VM Network`** |
| | VM Network | **`Checked`** |
| | MTU | **`Default (1500)`** |
| Cluster | testclu*X* | **`Attached`** |
| | testclu*X* | **`Not Required`** |
| vNIC Profiles | Public | **`Checked`** |

5. Click **OK** to create the network.

# Integrating Networking with OpenStack

## Objectives

After completing this section, students should be able to:

- List the steps for integrating OpenStack Neutron as a network provider in RHEV.
- Discuss features and limitations of Neutron integration.

## OpenStack Neutron as a network provider

Currently available as a tech preview, administrators can connect OpenStack Neutron networks with a RHEV datacenter. Neutron can provide networking capabilities for consumption by RHEV hosts, as well as virtual machines. Once the Neutron provider has been added, RHEV is capable of discovering the available networks on the provider. Administrators can then decide to import the networks as a new network into a RHEV datacenter, or attach the network to an existing datacenter. Existing RHEV networks can also be exported into Neutron, giving administrators the possibility to create Neutron networks through RHEV, instead of using the Neutron API. When new virtual machines are created, administrators can choose to provision a new network interface on Neutron networks. Using Neutron as an external network provider provides various benefits, such as using the technologies Neutron provides for the networks, such as IP address management, L3 routing, and security groups. Moreover, Neutron integration leverages the use of technologies that are not natively supported in RHEV for VM networks.

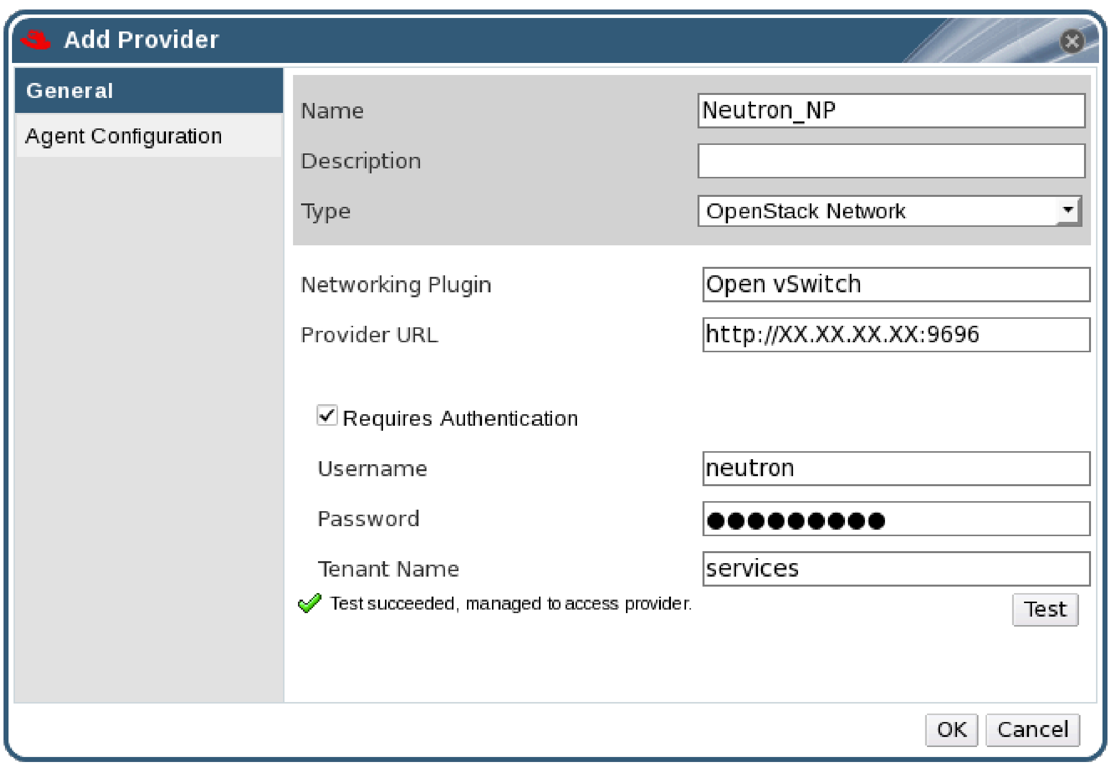


*Figure 6.3: Adding a Neutron external network*

Prior to the integration of a Neutron provider, administrators will have to:

- Ensure all Neutron base services are configured and running.
- Configure the Neutron message broker.
- Run the **engine-config** command, if Keystone is used:

```
[root@rhevm ~]# engine-config --set KeystoneAuthUrl=http://<host.fqdn>:35357/v2.0/
```

- Install the Linux bridging tools or Open vSwitch tools, depending on the network configuration, on the RHEV-H hosts.

The following procedure explains how to add a Neutron external provider:

1. Select the **External Providers** entry in the tree pane.
2. Click the **Add** button to open the **Add Provider** window.
3. Enter a name and description, then select **OpenStack Network** as the network **Type**.
4. Click the text field for **Networking Plugin** and select either **Linux Bridge** or **Open vSwitch**, depending on the plug-in set in the OpenStack environment.
5. Enter the URL, or fully qualified domain name, of the machine on which the OpenStack Networking instance is installed in the **Provider URL** text field, followed by the port number.
6. Optionally, select the **Requires Authentication** checkbox, and enter the credentials for the OpenStack Networking instance. Administrators must use the username and password for the OpenStack Networking user registered in Keystone.
7. In the **Agent Configuration** tab, fill in the QPID configuration.

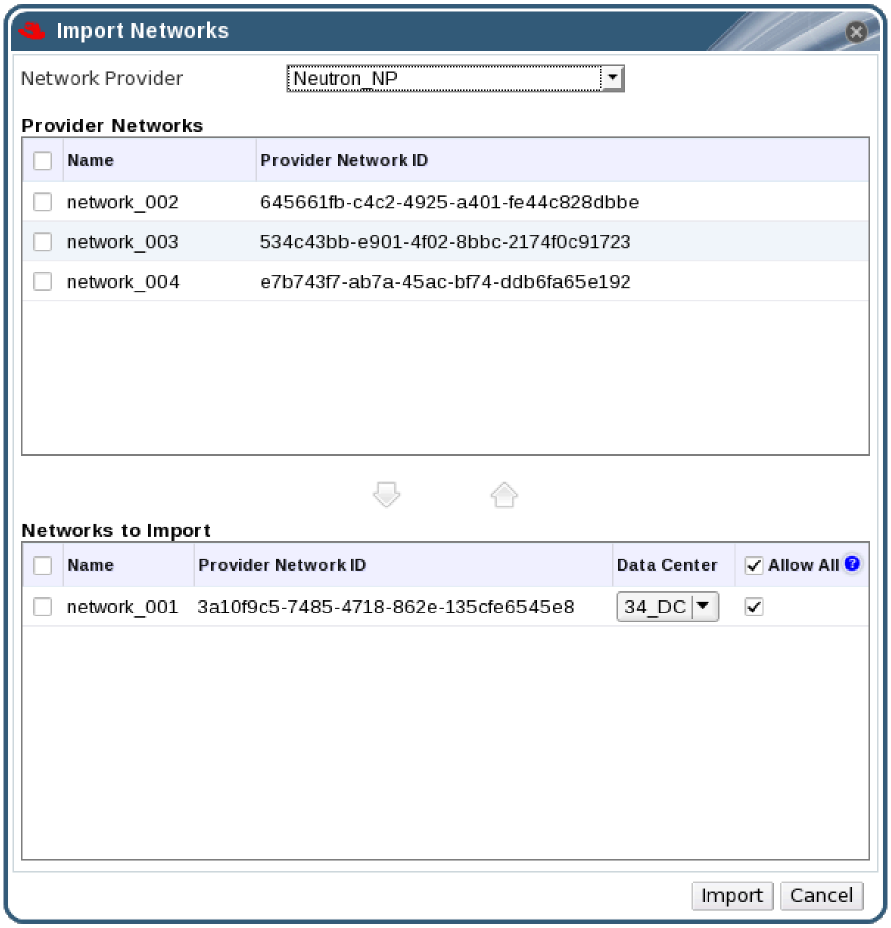


*Figure 6.4: Listing Neutron network*

# Current limitations and future work

The current implementation has several limitations administrators should take into account when working with Neutron providers:

- The same logical network can be imported more than once, but only to different datacenters.
- If a network has been externally provided by Neutron, administrators will not be able to edit the network in RHEV. To edit the details of a logical network offered by an external provider, administrators must edit the logical network directly from the OpenStack Networking instance that provides that logical network.
- Port mirroring is not available for virtual NICs that are connected to an external Neutron network.
- If a virtual machine uses a logical network offered by an external provider, that provider cannot be deleted from the manager while the logical network is still in use by the virtual machine.
- Networks offered by external providers are nonrequired. As such, scheduling for clusters in which such logical networks have been imported will not take those logical networks into account during host selection. Moreover, it is the responsibility of the user to ensure the availability of the logical network on hosts in clusters in which such logical networks have been imported.

The next implementation of the Neutron support will include:

- IPAM support.
- Better security groups support.
- Administrators will be able to delete Neutron networks using the RHEV-M.
- Neutron subnets will be shown to administrators, giving them the possibility to manage those subnets.
- Advanced features, such as VPN-as-a-Service and Load-Balancing-as-a-Service, will be available.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 6.5. External providers networks

> Ovirt Documentation - Detailed OSN Integration
> http://www.ovirt.org/Network_Provider

# Quiz: Integrating Networking with OpenStack

Choose the correct answer to the following questions:

1. Name the two plug-ins that are currently supported by RHEV for Neutron networks. (Choose two.)

    a. OPFlex
    b. Linux Bond
    c. Open vSwitch
    d. Linux Bridge

2. Using Neutron as a network external provider offers several benefits. Select the correct statement. (Choose one.)

    a. Administrators can use Neutron networks to setup advanced L3 routing between RHEV networks and Neutron networks.
    b. Native RHEV networks' high availability feature can be extended to Neutron networks.
    c. Neutron networks can be imported in order to simplify and unify Neutron networks management.
    d. Administrators can use the technologies Neutron provides for the networks, such as IP address management.

3. Prior to the integration of Neutron into the RHEV engine, which command needs to be run? (Choose one.)

    a. **engine-config --set KeystoneAuthUrl=http://<host.fqdn>:35357/**
    b. **engine-config --set KeystoneAuthUrl=http://<host.fqdn>:35357/v2.0/**
    c. **engine-config --set KeystoneAuthUrl=http://<host.fqdn>/v2.0/**
    d. **engine-config --set KeystoneAuthUrl=http://<host.fqdn>**

4. The current implementation has several limitations, such as: (Choose two.)

    a. If a network has been externally provided by Neutron, administrators will not be able to edit the network in RHEV.
    b. Discovered networks cannot be renamed.
    c. VPN-as-a-Service requires a specific VDSM hook for certificates management.
    d. Port mirroring is not available for virtual NICs that are connected to an external Neutron network.

# Solution

Choose the correct answer to the following questions:

1. Name the two plug-ins that are currently supported by RHEV for Neutron networks. (Choose two.)

   a. OPFlex
   b. Linux Bond
   **c. Open vSwitch**
   **d. Linux Bridge**

2. Using Neutron as a network external provider offers several benefits. Select the correct statement. (Choose one.)

   a. Administrators can use Neutron networks to setup advanced L3 routing between RHEV networks and Neutron networks.
   b. Native RHEV networks' high availability feature can be extended to Neutron networks.
   c. Neutron networks can be imported in order to simplify and unify Neutron networks management.
   **d. Administrators can use the technologies Neutron provides for the networks, such as IP address management.**

3. Prior to the integration of Neutron into the RHEV engine, which command needs to be run? (Choose one.)

   a. `engine-config --set KeystoneAuthUrl=http://<host.fqdn>:35357/`
   **b. `engine-config --set KeystoneAuthUrl=http://<host.fqdn>:35357/v2.0/`**
   c. `engine-config --set KeystoneAuthUrl=http://<host.fqdn>/v2.0/`
   d. `engine-config --set KeystoneAuthUrl=http://<host.fqdn>`

4. The current implementation has several limitations, such as: (Choose two.)

   **a. If a network has been externally provided by Neutron, administrators will not be able to edit the network in RHEV.**
   b. Discovered networks cannot be renamed.
   c. VPN-as-a-Service requires a specific VDSM hook for certificates management.
   **d. Port mirroring is not available for virtual NICs that are connected to an external Neutron network.**

# Lab: Creating Logical Networks

In this lab, you will create a new VM network, **vmnetwork**, in your existing cluster.

**Outcome(s)**

- You should be able to adjust the MAC pool range for virtual machines.
- You should be able to create a VM network named **vmnetwork** that will be available in the datacenter ***testdcX***.

**Before you begin**

This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.
- Administration Portal connects to rhevm.
- An installed RHEV-H node.
- Datacenter named testdc*X*.
- Cluster named testclu*X* with a single RHEV-H host.

**Lab outline**

Configure the network MAC pool range to correspond with addresses being offered by the DHCP server.

- Range: **52:54:00:00:XX:10-52:54:00:00:XX:FF** (where XX is the two-digit hexadecimal value of your pod number, pod 1 XX=01 ; pod 10 XX=0A ; pod 20 XX=14).

Configure the logical network according to the following specifications:

- Name: **vmnetwork**.
- Data Center: ***testdcX*** (where *X* is your pod number).
- MTU: **800**.
- Network **Attached**, but not **Required** for the datacenter.

1. Change the **MacPoolRanges** and restart **ovirt-engine**.
2. Log into the Administration Portal.
3. Access the Datacenter
   Access the datacenter ***testdcX*** in order to create the network within.
4. Create a New Network
   Define the network, according to the specifications.

# Solution

In this lab, you will create a new VM network, **vmnetwork**, in your existing cluster.

Outcome(s)

- You should be able to adjust the MAC pool range for virtual machines.
- You should be able to create a VM network named **vmnetwork** that will be available in the datacenter ***testdcX***.

**Before you begin**

This exercise requires the following components from earlier exercises:

- RHEV Manager installed on rhevm.
- Administration Portal connects to rhevm.
- An installed RHEV-H node.
- Datacenter named testdc*X*.
- Cluster named testclu*X* with a single RHEV-H host.

**Lab outline**

Configure the network MAC pool range to correspond with addresses being offered by the DHCP server.

- Range: **52:54:00:00:XX:10-52:54:00:00:XX:FF** (where XX is the two-digit hexadecimal value of your pod number, pod 1 XX=01 ; pod 10 XX=0A ; pod 20 XX=14).

Configure the logical network according to the following specifications:

- Name: **vmnetwork**.
- Data Center: ***testdcX*** (where *X* is your pod number).
- MTU: **800**.
- Network **Attached**, but not **Required** for the datacenter.

1. Change the **MacPoolRanges** and restart **ovirt-engine**.

```
[root@rhevm ~]# engine-config -s MacPoolRanges=52:54:00:00:XX:10-52:54:00:00:XX:FF
[root@rhevm ~]# service ovirt-engine restart
```

2. Log into the Administration Portal.

   Launch the RHEV-M interface, using Firefox on workstation, and access **http://rhevm.pod*X*.example.com/**. Click the Administration Portal link. Log in using **rhevadmin** for the username, with a password of **redhat**, and using **example.com** for the domain.

3. Access the Datacenter
   Access the datacenter ***testdcX*** in order to create the network within.

   From the left frame, navigate to System > Data Centers > testdc*X*, then click the Networks entry.

4. Create a New Network
   Define the network, according to the specifications.

   From the right frame, click the tab **Networks** to access the network configuration for the datacenter. Click **New** to create a new network. On the window that appears, fill in the fields using the following information:

| Tab | Field | Value |
| --- | --- | --- |
| General | Name | **vmnetwork** |
| | Description | **VM Network** |
| | VM Network | **Checked** |
| | MTU | **Custom**, with a value of **800** |
| Cluster | testclu*X* | **Attached** |
| | testclu*X* | **Not Required** |
| vNIC Profiles | Public | **Checked** |

   Finally, click **OK** to create the network.

# Summary

**Creating Logical Networks**

In this section, you learned:

- A logical network is a named set of global network connectivity properties in a datacenter. When a logical network is added to a host, it may be further configured with host-specific network parameters.
- Logical networks optimize network flow by grouping network traffic by usage, type, and requirements, and they allow both connectivity and segregation.
- A logical network may belong to multiple clusters, and this is one way for virtual machines and nodes in multiple clusters to communicate with each other.
- Red Hat Enterprise Virtualization 3.1 and higher distinguishes between required networks and optional networks.

**Integrating Networking with OpenStack**

In this section, you learned:

- Red Hat OpenStack Platform network component Neutron can be integrated with Red Hat Enterprise Virtualization.

redhat TRAINING

# CHAPTER 7

# DEPLOYING RHEV VIRTUAL MACHINES

| **Overview** | |
|---|---|
| **Goal** | Install and manage virtual machines with appropriate drivers and agents. |
| **Objectives** | • Install a new RHEL virtual machine.<br>• Start, stop, edit, and remove virtual machines.<br>• Install paravirtualized drivers and guest agents. |
| **Sections** | • Installing a New Virtual Server (and Practice)<br>• Managing Virtual Machines (and Practice)<br>• Installing Para-virtualized Drivers and Guest Agents (and Quiz) |
| **Lab** | • Deploying RHEV Virtual Machines |

# Installing a New Virtual Server

## Objectives

After completing this section, students should be able to:

- Install a new RHEL virtual machine

To create a new virtual server, log into the RHEV-M web interface and navigate to the **Virtual Machines** tab. In this tab, there is a **New VM** button. To provision a new server, click the **New VM** button. This will bring up a dialog to define the basic parameters of the virtual machine.

Once all the hardware for the new virtual server is configured, it is time to install the operating system. If the virtual machine were started normally, it may try to boot from the (empty) hard drive, so boot it in a different way. To do so, select the new virtual machine in the **Virtual Machines** then right-click it and select **Run Once**. This will bring up the **Run Virtual Machine(s)**, which allows pointing to an operating system installer, perhaps from the ISO library/storage domain.

## Demonstration: Installing a new Red Hat Enterprise Linux virtual machine

In this example, observe the following steps as the instructor demonstrates the creation of a new virtual machine.

1. In Firefox, navigate to **`http://classroom.example.com/materials/small.cfg`** to review the contents of a Kickstart file.

   A couple of things to note are the use of **`reboot`**, which intends to reboot the virtual machine upon the installation completing, and the lack of any specific **`rhevm-guest*`** packages.

2. In order to connect to a virtual machine console, use the Spice client. Install the client and restart Firefox if it is currently running.

   ```
   [root@workstation ~]# yum -y install spice-xpi
   ```

   Once it is installed, close down **Firefox** and relaunch.

3. On workstation, log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.

4. Navigate to the **Virtual Machines** tab. To provision a new server, click the **New VM** button.

   To configure a new virtual machine, at least all of the fields in the **General** tab should be filled out. A description of the settings can be found as follows.

| Name | Description |
|---|---|
| **Cluster** | The host cluster to be used for the new virtual server.<br>`cluster0/dcenter0` |

| Name | Description |
| --- | --- |
| **Based on Template** | Set this to **Blank** to start off with a fresh installation, or select an existing template for rapid deployments.<br>`Blank` |
| **Template Sub Version** | Set this to **base template (1)** to start off with a fresh installation, or select **latest** for stateless rapid deployments.<br>`base template (1)` |
| **Operating System** | The operating system to install on this virtual server. This setting is used both to determine what virtual hardware to make available to the new virtual server, as well as to select an icon for this virtual server in the User Portal.<br>`Red Hat Enterprise Linux 7.x x64` |
| **Instance Type** | The instance type on which the hardware configuration can be based. **Custom** means that the virtual machine is not connected to an instance type.<br>`Custom` |
| **Optimized for** | `Server` |
| **Name** | The name the new virtual machine should use in the RHEV-M interface. This does not have to be the host name for the new machine. This name is limited to a maximum of 64 characters and can only contain letters, numbers, underscores, and hyphens. No special characters or spaces are allowed.<br>`rhel0` |
| **Description** | A description of the new virtual server. This field is optional, but it can be used to find a specific virtual machine later on.<br>`First RHEL VM` |
| **Stateless**, **Start in Pause Mode**, and **Delete Protection** checkboxes | Leave at the default unchecked. |
| **nic1** | The logical network to associate the virtual machine's first NIC to. |

| Name | Description |
|---|---|
| | `rhevm` |

The **Console** tab should be left at the default of one monitor via a single PCI.

Eight additional tabs can be accessed by selecting the **Show Advanced Options** button. The other tabs in the **New Virtual Machine** dialog can be used to further tune the settings for the new virtual machine. The following table is a description of those tabs.

| Name | Description |
|---|---|
| **System** | This tab allows the user to specify memory size, CPU configuration, and the time zone. |
| **Initial Run** | This tab allows the user to specify **`cloud-init`** or **`sysprep`** configuration. |
| **Host** | Bind the new virtual machine to a specific host. The default is to allow any host in the selected cluster to run this virtual machine. |
| **High Availability** | Mark the virtual machine as being *highly available*. This means that when the host it is currently running on stops responding, the machine can be started on or migrated to another host automatically based on its priority. Using *high availability* requires all hosts in the cluster to have power management configured and working. |
| **Resource Allocation** | When using a template, use this tab to specify whether the virtual disk for this machine should be an overlay on top of the original template or a cloned copy of the template. The minimum amount of physical memory guaranteed for this virtual machine can also be set. |
| **Boot Options** | Specify an alternate boot order, attach CD images, or set a custom **`kernel`** and **`initrd`** to be used when booting this virtual machine, bypassing the normal boot loader. |
| **Random Generator** | Whether to enable the Random Number Generator (RNG) device. |
| **Custom Properties** | Define custom properties for this virtual machine. The names for custom properties need to be defined first using **`engine-config`** from the command line. Custom properties can be used in queries and reports, and VDSM hooks. |

5. Click **OK** to close the **New Virtual Machine** dialog and create the virtual machine. However, a **New Virtual Machine - Guide Me** dialog will popup. If this dialog is accidentally closed, bring it back by selecting the virtual machine to edit in the **Virtual Machines** tab of the RHEV-M web interface and clicking the **Guide Me** button.

   The new virtual machine is not yet ready for use. It has a processor, graphics card and memory, but it still lacks at least one hard drive.

6. Configure virtual server disks

   To add a disk to a virtual machine, either click the **Configure Virtual Disks** button from the **Guide Me** window, or select the virtual machine from the **Virtual Machines** tab in the RHEV-M web interface, navigate to the **Virtual Disks** tab in the lower pane, and click the **New** button. This will bring up the **New Virtual Disk** dialog.

   In this dialog, the following can be configured:

| Name | Description |
| --- | --- |
| **Internal** vs. **External (Direct Lun)** | Whether to use storage in a data storage domain (internal) or to directly attach to an iSCSI data source (external).<br>`Internal` |
| **Size(GB)** | The size of the new virtual disk as reported to the virtual machine.<br>`4` |
| **Alias** | The name of the disk.<br>`rhel0_Disk1` |
| **Description** | An optional description of the disk. |
| **Interface** | The type of disk interface that is presented to the virtual machine. This can be either **IDE** or **VirtIO**. VirtIO has better performance but requires special drivers to be installed in the virtual machine operating system. IDE does not require special drivers, but has poorer performance due to emulation overhead.<br>`VirtIO` |
| **Allocation Policy** | Whether the entire disk image should be created in advance (**Preallocated**) or whether the disk image should only be created as needed and grow along with the requirements (**Thin Provision**). Preallocated normally has better performance, but at the cost of using more disk space. Thin Provision tends to perform slightly worse, but saves disk space.<br>`Preallocated` |
| **Storage Domain** | The storage domain to use when creating this disk.<br>`data0` |
| **Disk Profile** | The storage domain to use when creating this disk. |

| Name | Description |
|---|---|
| | `data0` |
| **Activate**, **Wipe after delete**, **Bootable**, **Shareable**, and **Read Only** checkboxes | **Wipe after delete**—Whether the entire disk image should be zeroed out before removing this disk. If providing virtual machine hosting for different companies/departments, it might be required to check this option so as not to leak data unexpectedly.<br><br>**Bootable**—Whether the virtual machines can boot from this disk.<br><br>**Shareable**—Whether the disk can be attached to more than one virtual machine at a time.<br><br>`Check only` **Activate** `and` **Bootable** |

Click **OK** to create the new virtual disk, then select **Configure Later** to complete the virtual machine definition.

7. Installing a new virtual machine from the ISO library

   Using the ISO library is not mandatory. Since virtual machines in Red Hat Enterprise Virtualization have the ability to boot using PXE (Pre-eXecution Environment, boot from network), network-based deployments, such as automated Kickstart installations using a Red Hat Network Satellite Server with Cobbler, are also possible. To do so, simply make **Network (PXE)** the top priority in the **Boot Sequence** of a **Run Once** dialog, and follow the same procedure as for a bare-metal installation started from PXE.

   If a PXE-based deployment is not an option, minimize the size of the ISO library by simply adding the **`boot.iso`** and not adding the entire Red Hat Enterprise Linux DVD. The **`boot.iso`** can be used in conjunction with a network-based installation server like Red Hat Network Satellite.

   Now that all the hardware for the new virtual machine is configured, it is time to install the operating system. If the virtual machine were started normally, it may try to boot from the (empty) hard drive, so boot it in a different way. To do so, select the new virtual machine in the **Virtual Machines** then right-click it and select **Run Once**. This will bring up the **Run Virtual Machine(s)** dialog.

   In this dialog, the following choices can be made:

| Name | Description |
|---|---|
| **Boot Options** | **Attach Floppy**—Check this option and select which floppy image to use to add a virtual floppy drive (VFD) to the virtual machine.<br><br>**Attach CD**—Check this option and select which CD/DVD image to use to add a virtual CD-ROM/DVD-ROM drive to the virtual machine. |

| Name | Description |
|---|---|
| | **Boot Sequence**—Select the order in which the virtual machine will look for bootable devices. Highlight an entry and use the arrows on the right-hand side to change the ordering.<br><br>Checkboxes for **Enable boot menu**, **Run Stateless**, and **Start in Pause Mode**.<br><br>`Attach CD to rhel-server-7.1-x86_64-boot.iso` |
| **Linux Boot Options** | Use these fields to bypass the normal boot loader. |
| **Initial Run** | Use to override the choice of using **`cloud-init`**. |
| **Host** | Use to override the choice of what host to start running on. |
| **Display Protocol** | Change how connections are made to the virtual console. This can be useful, for example, when *Spice* drivers still need to be installed on a virtual machine. |
| **Custom Properties** | Define custom properties for this run (only). Custom properties can be used in queries and reports, and VDSM hooks. |

When clicking the **OK** button, the virtual machine will be started with the customized options set for this one run, allowing a normal installation.

## Important

Note that if the install reboots on its own, it will not "wipe" the BIOS settings from the **Run Once**, so it will appear to run multiple times with these settings. Once the virtual machine is stopped from the RHEV interface (or completely shut down from within the virtual machine), choosing **Run** from the RHEV interface reverts to the default boot settings.

8. Connect to the console of the virtual machine

   Connect to the console of **`rhel0`**. Hit **`Escape`** to be presented the **`boot:`** prompt. Point to a Kickstart file to automate the installation of Red Hat Enterprise Linux.

   ```
   boot: linux ks=http://classroom.example.com/materials/small.cfg
   ```

   To connect to the console of the new virtual machine, select the new virtual machine name in the **Virtual Machines** tab of the RHEV-M web interface and click the green **Console** button.

   From the console, start the installation and monitor its progress.

## References

Additional information may be available in the *Administration Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Installing a Red Hat Enterprise Linux Guest

In this lab, you will install a new Red Hat Enterprise Linux virtual machine.

| Resources | |
|---|---|
| **Files** | http://classroom.example.com/materials/small.cfg |
| **Application URL** | http://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to instantiate a new running Red Hat Enterprise Linux virtual machine.

**Before you begin**
Make sure the RHEV-M environment configured in the previous labs is still working (engine, hosts, and resources).

Perform the following steps:

1. Confirm Access to Kickstart File
   Use Firefox to access **`http://classroom.example.com/materials/small.cfg`** to review the contents of the Kickstart file.

2. Confirm that the Spice Plug-in Is Ready
   As root on workstation, install the Spice client and restart Firefox.

```
[root@workstation ~]# yum -y install spice-xpi
```

3. Log into the RHEV Administration Portal
   Log into the RHEV Administration Portal as user **`rhevadmin`** and password **`redhat`** in domain **`example.com`**.

4. Navigate to Virtual Machines
   Click the **Virtual Machines** tab.

5. Create a New Red Hat Enterprise Linux Server
   Create a new virtual machine with the following values:

   - Cluster: **`testcluX/testdcX`** (where *X* is your pod number).
   - Name: **`rhelX-test`** (where *X* is your pod number).
   - Description: **`First RHEL Guest`**
   - Operating System: **`Red Hat Enterprise Linux 7.x x64`**
   - Network Interface: *rhevm (rhevm)*

   a. Click the **New VM** button. The **New Virtual Machine** dialog displays.

b. Complete the information using the previous values, leaving the other values at their defaults.

c. Click **OK**. The **Guide Me** dialog opens, displaying a list of configuration tasks.

d. Click the **Configure Virtual Disks** button to add a 4 GB virtual disk with default values for other fields, then click **OK**.

e. Click **Configure Later** to close the dialog.

6. Install New Red Hat Enterprise Linux Server
   Back in the web console, verify the **`rhelX-test`** virtual machine has transitioned from **Image Locked** to **Down**.

   Launch the installer by modifying the following properties:

   - Attach CD: **`rhel-server-7.1-x86_64-boot.iso`**

   a. Right-click the **`rhelX-test`** virtual machine and choose **Run Once** to display the **Run Virtual Machine** dialog.

   b. Configure according to the previous properties, then click **OK** button to begin installation.

   c. Once the virtual machine has started and the console becomes available, click it to start the Red Hat Enterprise Linux installation.

   d. With **Install or upgrade an existing system** highlighted, press the **Tab** key to edit the installer options.

   e. Add, as a space-separated argument, **`ks=http://classroom.example.com/materials/small.cfg`** to the kernel command line. The cursor should be automatically positioned to allow you to add this at the end of the existing kernel arguments.

   f. After the installation completes, the virtual machine will automatically reboot from the CD. Interrupt the timer by pressing an arrow key.

   g. Close the console and power off the VM by right-clicking the VM entry and selecting **Power Off**. Confirm that you really want to power off the machine.

7. Test the Red Hat Server Virtual Machine
   Power on your new virtual machine by right-clicking the machine in the overview and selecting **Run**.

   Open the console by right-clicking the machine and selecting **Console**.

   After the machine is booted, log in with the **`student`** user account with a password of **`student`**.

# Managing Virtual Machines

## Objectives

After completing this section, students should be able to:

- Start, stop, edit, and remove virtual machines.

## Starting and stopping virtual machines

To start a virtual machine that is currently in the **Down** state, navigate to the **Virtual Machines** tab of the RHEV-M web interface and select the virtual machine to start. To start the virtual machine with its normal configuration, click the **Run** icon (green play button) or right-click the virtual machine and select **Run** from the menu.

To start a virtual machine with settings other than the defaults, right-click the virtual machine, and select **Run Once**. This will bring up the **Run Virtual machine(s)** dialog discussed in the previous section.

For shutting down a virtual machine there are three options:

- Shut it down from within the virtual machine itself, for example by using the **`poweroff`** command from within a Red Hat Enterprise Linux machine.
- Right-clicking a virtual machine and selecting **Shutdown**. This will send a virtual ACPI power button event to the virtual machine. Note that in certain cases, the operating system running on the virtual machine may ignore this event (e.g., Windows 7 displaying a login screen).
- Right-clicking the virtual machine in the RHEV-M web interface and selecting **Power Off** from the menu. This will effectively pull the virtual power cables from the machine, resulting in an ungraceful shutdown. Use this method sparingly and as a last resort only.

Virtual machines can also be paused by selecting them in the RHEV-M web interface and clicking the red **Pause** button. A paused virtual machine will effectively be put into *Hibernate* mode. Its memory and CPU state will be saved to disk and the machine will be suspended. To later resume a paused virtual machine, right-click the virtual machine and select **Run** from the menu.

## Editing a virtual machine

To permanently change the settings of a virtual machine, navigate to the **Virtual Machines** tab of the RHEV-M web interface and select the machine to edit, then right-click the virtual machine and select **Edit** from the menu. This will bring up the **Edit Virtual Server Machine** dialog shown as follows.

The **Edit Virtual Server Machine** dialog is nearly identical to the **New Server Virtual Machine** dialog discussed earlier, but some options are grayed out here. Please note that in order to change the amount of memory or the CPU configuration, the virtual machine must first be shut down.

### Important

Some settings (like changing the **Operating System**) can have a drastic effect on the virtual machine. Make sure that a recent backup is available before making changes.

To add extra network cards or disks to a virtual machine, shut it down first. Either use the **Guide Me** facility or the **Network Interfaces** and/or **Virtual Disks** tab in the lower pane to add, edit, or remove virtual network cards and disks.

### Important

Be careful when removing disks and network cards. When a disk is removed, all its contents will be lost forever. When a network card is removed and re-added, the MAC address of the card might change, forcing updates to both the **udev** rules and network configuration files on the machine.

## Removing virtual machines

To remove a virtual machine, first shut it down normally, then select it in the **Virtual Machines** tab of the RHEV-M web interface, then right-click the virtual machine and select **Remove** from the menu. Click **OK** in the confirmation dialog that appears to confirm removing this virtual machine.

### Important

When removing a virtual machine, all resources associated with the virtual machine will be removed as well. This includes all virtual disks and virtual network cards the machine had configured. Once a virtual disk is removed, both the disk and its contents are lost forever.

## Demonstration: Managing virtual machines

In this example, observe the following steps as the instructor demonstrates how to manage virtual machines.

1. Log into the RHEV-M Administration Portal as the user **`rhevadmin`** with a password of **`redhat`** in domain **`example.com`**. Navigate to the **Virtual Machines** tab.

2. Power down the **`rhel0`** virtual machine by selecting it in the overview, then right-clicking it and selecting **Shutdown**. Press **OK** to confirm shutting down the machine. Wait for the virtual machine to power down.

3. With the **`rhel0`** virtual machine selected, right-click the virtual machine and select **Edit** from the menu. Change the machine to match the following specifications:
   - Name: **`r0-modified`**.
   - Description: **`My First Modified Server`**
   - Memory Size (under System): **`768 MB`**

4. Add an additional 4 GB virtio-disk to the **r0-modified** virtual machine.

   With the **r0-modified** machine selected, click the **Disks** tab in the bottom pane. Click the **Add** button in the **Disks** pane. Set the size to **4** GB, and the **Allocation Policy** to **Thin Provision**; leave all the other settings at their defaults. Wait for the operation to complete before continuing.

5. Power on the **r0-modified** virtual machine by selecting it in the overview, then right-clicking it and selecting **Run** from the menu. Open a console by right-clicking the virtual machine in the overview and selecting **Console** from the menu.

6. Log into the virtual machine as **student**. Verify that the new disk appears:

```
[student@vmY ~]$ cat /proc/partitions
major minor  #blocks  name

 252        0    4194304 vda
 252        1     524288 vda1
 252        2    3668992 vda2
 252       16    4194304 vdb
```

   Finding **vdb** in the output means that the disk has been attached successfully.

7. Verify that the virtual machine now has 768 MB of memory by executing the following command on the virtual machine:

```
[student@vmY ~]$ grep MemTotal /proc/meminfo
MemTotal:         757460 kB
```

   If a size around **750000** kilobytes appears it means that the amount of memory in the virtual machine has been successfully extended.

8. Shut down the **r0-modified** virtual machine by running the **poweroff** command in the virtual machine. Wait for the machine to fully shut down before continuing.

9. Restore the name of the virtual machine to **rhel0**.

   Select the **r0-modified** virtual machine by right-clicking it in the overview pane, then choose **Edit** from the menu. Change the value of the **Name** field to **rhel0**, then click the **OK** button to confirm the change.

10. Permanently remove the additional disk that was added to the server earlier.

    With the **rhel0** virtual machine selected, click the **Disks** tab in the bottom pane. Change to the **All** view by selecting the radio button. Select the **r0-modified_Disk1** disk, then click the **Remove** button in the **Disks** pane.

    Check the **Remove permanently** checkbox and click the **OK** button to confirm the removal.

## References

Additional information may be available in the *Administration Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Managing Virtual Machines

In this lab, you will make changes to a Red Hat Enterprise Linux virtual machine.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to modify a Red Hat Enterprise Linux virtual machine.

**Before you begin**
Make sure the RHEV-M environment configured in the previous labs is still working (engine, hosts, and resources). You should also still have the **`rhelX-test`** virtual machine from the previous exercise.

Perform the following steps:

1. Log into the RHEV-M Administration Portal as the user **`rhevadmin`** with a password of **`redhat`** in domain **`example.com`**. Navigate to the **Virtual Machines** tab.

2. Power down your **`rhelX-test`** virtual machine by selecting it in the overview, then right-clicking it and selecting **Shutdown**. Press **OK** to confirm that you want to shut down your machine. Wait for your virtual machine to power down.

> ## Warning
>
> The GUI option of Shutdown will execute an ACPI event on **`rhelX-test`** that would trigger a setting in GNOME 3 that defaults to trying to suspend the system or simply gets ignored. How an operating system reacts to these ACPI events needs to be taken into account.

3. With your **`rhelX-test`** virtual machine selected, right-click the virtual machine and select **Edit** from the menu. Change the machine to match the following specifications:
   - Name: **`rhelX-modified`** (where *X* is your pod number)
   - Description: **`My First Modified Server`**
   - Memory Size (under System): **`768 MB`**

4. Add an additional 4 GB virtio-disk to your **`rhelX-modified`** virtual machine.

   With your **`rhelX-modified`** machine selected, click the **Disks** tab in the bottom pane. Click the **Add** button in the **Disks** pane. Set the size to **4** GB, and the **Allocation Policy** to **`Thin Provision`**; leave all the other settings at their defaults. Wait for the operation to complete before continuing.

5. Power on your **`rhelX-modified`** virtual machine by selecting it in the overview, then right-clicking it and selecting **Run** from the menu. Open a console by right-clicking the virtual machine in the overview and selecting **Console** from the menu.

6. Log into the virtual machine as **student**. Verify that you can see the new disk:

```
[student@vmY ~]$ cat /proc/partitions
major minor  #blocks  name

 252        0    4194304 vda
 252        1     524288 vda1
 252        2    3668992 vda2
 252       16    4194304 vdb
```

   Finding **vdb** in the output means that the disk has been attached successfully.

7. Verify that your virtual machine now has 768 MB of memory by executing the following command on your virtual machine:

```
[student@vmY ~]$ grep MemTotal /proc/meminfo
MemTotal:         757460 kB
```

   If you see a size around **750000** kilobytes, it means that you have successfully extended the amount of memory in your virtual machine.

8. Shut down your **rhel*X*-modified** virtual machine by running the **poweroff** command in your virtual machine. Wait for your machine to fully shut down before continuing.

9. Restore the name of your virtual machine to **rhel*X*-test**.

   Select the **rhel*X*-modified** virtual machine by right-clicking it in the overview pane, then choose **Edit** from the menu. Change the value of the **Name** field to **rhel*X*-test**, then click the **OK** button to confirm the change.

10. Permanently remove the additional disk that you added to the server earlier.

   With your **rhel*X*-test** virtual machine selected, click the **Disks** tab in the bottom pane. Change to the **All** view by selecting the radio button. Select the **rhel*X*-modified_Disk1** disk, then click the **Remove** button in the **Disks** pane.

   Check the **Remove permanently** checkbox and click the **OK** button to confirm the removal.

# Installing Paravirtualized Drivers and Guest Agents

## Objectives

After completing this section, students should be able to:

- Install paravirtualized drivers and guest agents.

## Paravirtualized drivers

One of the best ways to improve the performance of Microsoft Windows guests is to use paravirtualized devices and drivers for KVM in the guests. This provides close to bare-metal performance (up to 95%). These drivers are provided by the *virtio-win* RPM package in the RHEL Supplementary RHN channel, and can also be obtained from Microsoft (at **`www.windowsservercatalog.com`**).

- If administrators install the *virtio-win* package on a RHEL server, the directory **`/usr/share/virtio-win/`** will contain symlinks to two files: **`virtio-win-*.iso`** is a CD-ROM ISO driver disk image, and **`virtio-drivers-*.vfd`** is a floppy driver disk image for use when installing Windows. Both of these images can be uploaded to an ISO library for convenient use.

  If they add these images to the ISO library, they can attach the **`virtio-drivers-*.vfd`** image to the virtual floppy drive before manually installing a Windows guest.

- Alternatively, administrators can add the drivers to an existing Windows guest. First the ISO needs to be attached to the VM from the RHEV-M console so the installers can be used from within the Windows guest. There are installers for 32-bit and 64-bit versions of Windows (server and desktop) for the block and network drivers: **`RHEV-Block/RHEV-Block64`** and **`RHEV-Network/RHEV-Network64`**. The VM needs to be rebooted for the drivers to work, but overall it is a simple and fast process.

- Another way to install these drivers is to use the installer provided in the **`rhev-tools-setup.iso`** image found in the **`/usr/share/rhev-guest-tools-iso/`** directory and provided by the *rhev-guest-tools-iso* package in the **`rhevm`** RHN channel. Like the previous files, this ISO should be uploaded to the ISO library for easy use. This installer also provides Spice, USB, block, and network drivers plus the Management Agent and SSO (Single Sign On) in the same package.

### Note

Administrators need to make sure that the drivers are installed before they add any virtio-based devices or they will likely have trouble using them under Windows. (Editing the guest's system drive to use virtio *before* installing the paravirtualized drivers is a bad idea, for reasons which should be obvious,)

## RHEV-M guest agents

When a virtual machine is created in RHEV-M, RHEV attempts to contact the virtual machine to gather information (such as the IP address). RHEV also needs to communicate with the virtual machine when RHEV-M attempts to shut down the virtual machine.

On Red Hat Enterprise Linux, this communication is done by the *ovirt-guest-agent* service from the *rhevm-guest-agent-common* package. If installing a Red Hat Enterprise Linux virtual machine in RHEV, make sure to include the *rhevm-guest-agent-common* package.

On Windows, install the RHEV Agent as part of the RHEV Tools installation.

## Demonstration: Add RHEV-M guest agent to virtual machine

In this example, observe the following steps as the instructor demonstrates the addition of a guest agent on a new virtual machine.

1. Power on your rhel*X* virtual machine by selecting it in the overview, then right-clicking it and selecting **Run** from the menu. Open a console by right-clicking the virtual machine in the overview and selecting **Console** from the menu.

2. Log into the virtual machine as **`root`** with a password of **`redhat`**.

3. Add the appropriate repository/channel/subscription to obtain the *rhevm-guest-agent-common* package.

   ```
   [root@vmY ~]# wget http://classroom.example.com/materials/rheva.repo -P /etc/
   yum.repos.d/
   ```

4. List the repositories that were enabled.

   ```
   [root@vmY ~]# yum repolist
   ```

5. Install the package on the system.

   ```
   [root@vmY ~]# yum -y install rhevm-guest-agent-common
   ```

6. Reboot the system and investigate VM information reported in the Administration Portal.

   Look at the **Network Interfaces** tab in the lower pane of the virtual machine, then the subtab of **Guest Agent Data** to view the information about the interface.

## Selecting a console type for a virtual machine

When creating or editing a virtual machine, there is a choice between two (sometimes three) different console types:

| Type | Description |
|---|---|
| VNC | VNC is a well-known and widely supported protocol for connecting to a remote graphical environment. There are VNC clients available on almost every computing platform, but the capabilities of VNC are |

| Type | Description |
|---|---|
| | limited. Playing a video over a remote VNC connection, for example, might not give the smooth experience end users are looking for. |
| Spice | Spice is an open protocol developed especially for connecting to the consoles of virtual machines. Among the unique capabilities of Spice are driving up to four displays at once. Spice also supports command channels, making it possible to share USB devices between the client and the virtual machine being connected to. For performance, Spice selects where to do 3D rendering dynamically; either on the virtual machine, the hypervisor, or even on the client connecting to the virtual machine. This design makes it possible to have a Spice client show smooth full-screen video on up to four separate screens at once over a typical office connection. |
| RDP | RDP is the default remote desktop protocol for machines running Windows. RHEV can display a virtual machine console using RDP if the virtual machine being connected to is running Windows, but only if RDP access has been configured inside the operating system, and only after Windows has started. It is not possible to see the boot process of a Windows virtual machine using RDP. RDP can only be chosen as a console protocol for virtual machines running a version of Windows. |

Selecting the protocol to be used for connections to a virtual machine is a decision that should be made on a machine-by-machine basis. In most cases, using Spice is desired, but settings can be changed from the **Edit Server Virtual machine** and **Run Once** dialogs.

# Spice protocol

- Spice is an adaptive remote rendering protocol meant to provide high levels of performance and replace the use of VNC and RDP for virtual desktop environments. Spice can provide a user experience when connected to a guest console that is indistinguishable from the experience they have when logged into the physical hardware.

- Spice does this through an adaptive remote rendering solution. The local Spice client is designed to communicate with the graphical terminal's GPU for maximum performance. If the terminal has a sufficiently powerful video card, video processing and rendering will be offloaded to the terminal's GPU and performed locally. If not, Spice can fall back to using the host's GPU or even software rendering at a performance impact. Ideally, however, this offloads most of the graphic rendering work where it belongs: on the user's local machine.

- Bidirectional audio and video allows VoIP and video-conference calls from virtual desktops, while USB redirection allows connection of USB 1.0 or 2.0 devices to virtual desktops.

- In practice, **`qemu-kvm`** presents a paravirtualized QXL video device to the guest operating system. If the guest does not have special drivers, this device acts like a standard PCI-based VESA-compliant video card. For full functionality, the guest needs to install the QXL drivers for the video card.

- On the local graphical terminal side, to access the Spice console on the virtual machine, the end user can either launch a cross-platform Linux/Windows dedicated Spice client or they can launch a client from a web browser using ActiveX (for Internet Explorer) or a Mozilla plug-in (for Firefox). Note that the client connects to the host the guest is running on, so even if the guest is at a BSOD or panic, or is otherwise unresponsive, the console should be accessible.

## References

Additional information may be available in the *Administration Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Quiz: Installing Paravirtualized Drivers and Guest Agents

Choose the correct answer to the following questions:

1. What are the two RPM packages that provide Windows guest drivers and agents? (Choose two.)

    a. **`virtio-win`**
    b. **`rhev-agent-iso`**
    c. **`windows-drivers`**.
    d. **`rhev-guest-tools-iso`**

2. What RPM package contains the RHEV agent for RHEL? (Choose one.)

    a. **`ovirt-guest-agent`**
    b. **`rhevm-guest-agent-common`**
    c. **`rhel-rhev-guest-agent`**
    d. **`rhevm-agent-rhel6`**

3. What three types of consoles are supported by RHEV virtual machines? (Choose three.)

    a. Spice
    b. RDP
    c. VNC
    d. VirtIO

4. What protocol supports bidirectional audio and video along with USB redirection? (Choose one.)

    a. VNC
    b. TCP/IP
    c. Spice
    d. VirtIO

# Solution

Choose the correct answer to the following questions:

1. What are the two RPM packages that provide Windows guest drivers and agents? (Choose two.)

   a. **`virtio-win`**
   b. `rhev-agent-iso`
   c. `windows-drivers`.
   d. **`rhev-guest-tools-iso`**

2. What RPM package contains the RHEV agent for RHEL? (Choose one.)

   a. `ovirt-guest-agent`
   b. **`rhevm-guest-agent-common`**
   c. `rhel-rhev-guest-agent`
   d. `rhevm-agent-rhel6`

3. What three types of consoles are supported by RHEV virtual machines? (Choose three.)

   a. **Spice**
   b. **RDP**
   c. **VNC**
   d. VirtIO

4. What protocol supports bidirectional audio and video along with USB redirection? (Choose one.)

   a. VNC
   b. TCP/IP
   c. **Spice**
   d. VirtIO

# Lab: Deploying RHEV Virtual Machines

In this lab, you will install and configure a new small virtual server running Red Hat Enterprise Linux 7.

| Resources | |
|---|---|
| **Files** | http://classroom.example.com/materials/small-rhev.cfg |
| **Application URL** | http://rhevm.pod*X*.example.com |

**Outcome(s)**

You should be able to deploy a new virtual machine that will be running Red Hat Enterprise Linux 7 configured according to the following specifications:

- Minimal package set
- **`rhevm-guest-agent-common`** installed
- Power down when done

**Before you begin**

You should have an existing RHEV environment with functioning engine, hosts, and resources.

**Lab outline**

You have been tasked with creating and installing a new virtual machine running Red Hat Enterprise Linux 7. One of your colleagues has already created a Kickstart to be used, which is available at **`http://classroom.example.com/materials/small-rhev.cfg`**. The new machine should be configured according to the following specifications:

| Virtual machine specification | |
|---|---|
| Name | rhel*X* (where *X* is your pod-number) |
| Description | Basic RHEL Server |
| Operating system | Red Hat Enterprise Linux 7.x x64 |
| Primary disk | 4 GB VirtIO, Thin Provisioned |
| Installation method | PXE |
| Kickstart file | **`http://classroom.example.com/materials/small-rhev.cfg`** |

1. Verify Kickstart File
2. Create VM
3. Boot VM From Network
4. Point Installer to Kickstart File
5. Test VM

# Solution

In this lab, you will install and configure a new small virtual server running Red Hat Enterprise Linux 7.

| Resources | |
|---|---|
| **Files** | http://classroom.example.com/materials/small-rhev.cfg |
| **Application URL** | http://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to deploy a new virtual machine that will be running Red Hat Enterprise Linux 7 configured according to the following specifications:

- Minimal package set
- **`rhevm-guest-agent-common`** installed
- Power down when done

**Before you begin**
You should have an existing RHEV environment with functioning engine, hosts, and resources.

**Lab outline**

You have been tasked with creating and installing a new virtual machine running Red Hat Enterprise Linux 7. One of your colleagues has already created a Kickstart to be used, which is available at **`http://classroom.example.com/materials/small-rhev.cfg`**. The new machine should be configured according to the following specifications:

| Virtual machine specification | |
|---|---|
| Name | rhel*X* (where *X* is your pod-number) |
| Description | Basic RHEL Server |
| Operating system | Red Hat Enterprise Linux 7.x x64 |
| Primary disk | 4 GB VirtIO, Thin Provisioned |
| Installation method | PXE |
| Kickstart file | **`http://classroom.example.com/materials/small-rhev.cfg`** |

1. Verify Kickstart File
   In Firefox, navigate to **`http://classroom.example.com/materials/small-rhev.cfg`** to review the contents of a Kickstart file.

   A couple of things to note are the use of **`poweroff`**, which intends to shut down the virtual machine upon the installation completing, and the existence of **`repo`** entries and the **`rhevm-guest-agent-common`** package.

2. Create VM
   2.1. Log into the RHEV-M Administration Portal and navigate to the **Virtual Machines** tab.

   2.2. Click the **New VM** button to create a new virtual server machine called **`rhelX`** (where *X* is your pod number). Configure the machine as follows:

- Description: **`Basic RHEL Server`**
- Operating System: **`Red Hat Enterprise Linux 7.x x64`**
- nic1: **`rhevm`**

Any other settings should be left as is.

2.3. In the **Guide Me** window that pops up, click **Configure Virtual Disks**. Put a **4** in the **Size (GB):** field, select **`Thin Provision`** for the **Allocation Policy**, and use the default settings for all other fields. Click **OK** to complete the process.

Click **Configure Later** to close the **New Virtual Machine - Guide Me** dialog window.

3. Boot VM From Network
   Boot your new machine by right-clicking and selecting **Run Once**.

   Change the boot order to boot from PXE by selecting **Network (PXE)** and promoting it to the top of the **Boot Sequence** by using the up arrow button. Leave all other settings unchanged.

4. Point Installer to Kickstart File
   Open the console for your new virtual machine by right-clicking and selecting **Console**. If a pop-up appears asking you to install **RHEV SPICE Client**, run the installer and wait for the installation to finish before continuing. Open the console again.

   At the boot prompt, start a new installation using the **`small-rhev.cfg`** Kickstart by hitting **`Escape`**, then typing:

```
boot: linux ks=http://classroom.example.com/materials/small-rhev.cfg
```

   Wait for the installation to complete; this should only take a few minutes. When the machine powers off, it has finished installing.

5. Test VM
   Power on your new virtual machine by right-clicking the machine in the overview and selecting **Run**.

# Summary

### Installing a New Virtual Server

In this section, you learned:

- The **New Virtual Machine** dialog is used to define the hardware elements of a new virtual server.
- **kickstart** files can be used in order to install the operating system of a new RHEL virtual machine.

### Managing Virtual Machines

In this section, you learned:

- The **Run Once** option allows you to start virtual machines with one-time settings.
- Stopping a virtual machine can be accomplished in at least two ways: from inside the VM, or using Administration Portal.
- It is possible to temporarily or permanently edit virtual machine settings.
- Virtual machines that were created can also be deleted from RHEV.

### Installing Paravirtualized Drivers and Guest Agents

In this section, you learned:

- **Paravirtualized** drivers, in both RHEL and Windows, can be used to improve performance.
- A communication channel between the RHEV Manager and the guests with ovirt-agents can be opened.
- RHEV offers different console types for a virtual machine depending on support inside the VM.

# CHAPTER 8

# MANAGING VM SNAPSHOTS AND IMAGES

| Overview | |
|---|---|
| **Goal** | Manage VMs with snapshots and share images. |
| **Objectives** | • Create, restore, and delete virtual machine snapshots.<br>• Export and import virtual machine images. |
| **Sections** | • Creating and Using Image Snapshots (and Practice)<br>• Sharing and Editing Images (and Practice) |
| **Quiz** | • Managing VM Snapshots and Images |

# Creating and Using Image Snapshots

## Objectives

After completing this section, students should be able to:

- Create a virtual machine snapshot.
- Restore a virtual machine from a snapshot.
- Delete a virtual machine snapshot.

## Image management: Snapshots

A snapshot is a view of a virtual machine's operating system and applications on any or all available disks at a given point in time. Administrators can take snapshots of a virtual machine before making a change to it that may have unintended consequences. Administrators can use snapshots to return a virtual machine to a previous state. RHEV manages snapshots in the **`data`** domains so that two guests cannot access the same image at the same time.

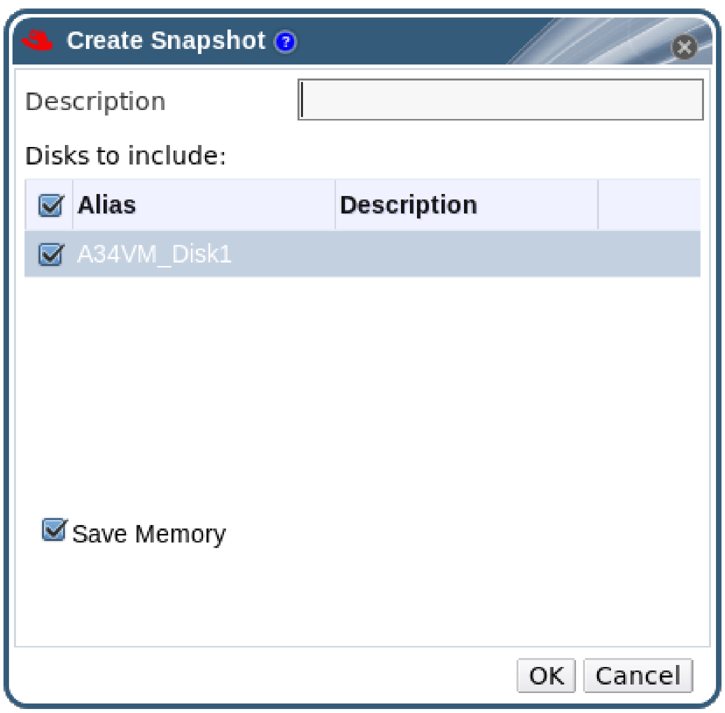


*Figure 8.1: Creating snapshots*

A snapshot can be taken of the state of a virtual machine's disks at a given point in time. Many snapshots of a machine's state can be taken, but only a single snapshot can be used at a time. Red Hat Enterprise Virtualization version 3.1 introduced support for "live" snapshots of running virtual machines. Virtual machines with a guest agent that supports **`quiescing`** can ensure file system consistency across snapshots. Red Hat Network-registered Red Hat Enterprise Linux guests can install the **`qemu-guest-agent`** package, then enable the **`qemu-ga`** service to enable **`quiescing`** before snapshots. Be aware that applications that leave their data in a "non-steady state" on disk may find themselves corrupted when trying to use the resulting snapshot.

RHEV administrators can temporarily preview any snapshot on virtual machine startup to make sure that it is the snapshot they want to use. An administrator can permanently roll the disk image back to that snapshot; all snapshots taken at later points in time are then permanently discarded.

# Cloning

Additionally, since Red Hat Enterprise Virtualization version 3.1, the ability to create a new virtual machine from a snapshot has been available, known as "cloning." However, a *clone* may have residual hardware-specific information, which *templates* were designed to address.

Introduced with Red Hat Enterprise Virtualization 3.5, we can now perform a direct clone of a virtual machine without needing a template or snapshot.

From the **Virtual Machines** tab, select the virtual machine to clone, then click the **Clone VM** button to open the **Clone Virtual Machine** window. Enter a name for the new virtual machine and click **OK**.

# Demonstration: Creating virtual machine snapshots

The procedure to create a snapshot is listed as follows. The virtual machine must be stopped before attempting to take a snapshot. A description can be added to the snapshot to describe why it was taken.

In this example, observe the following steps as the instructor demonstrates the creation of a virtual machine snapshot.

1. Log into your RHEV-M Administration Portal and navigate to the **Virtual Machines** tab.

2. Make sure the **`rhel0`** machine (or another RHEL desktop/server) is powered down. Select it and navigate to the **Snapshots** tab in the lower pane.

3. Click the **Create** button. In the dialog that pops up, enter **`Example Snapshot`** as the **Description**, then click **OK**.

4. Show that the virtual machine is now in the state **Locked**, and that the snapshot shows up in the **Snapshots** tab.

5. After the snapshot is complete, start your virtual machine.

6. Connect to the console of the virtual machine, and make an obvious change by appending the following line to **`/etc/issue`**:

   ```
   \d
   ```

# Demonstration: Restoring virtual machine snapshots

The process of reverting a virtual machine to an earlier snapshot is also relatively straightforward. Note carefully that once you commit to a snapshot, all later snapshots are permanently lost. The steps to restore a snapshot are outlined as follows:

In this example, observe the following steps as the instructor demonstrates restoring from a virtual machine snapshot.

- Power down the **`rhel0`** machine. Select this machine in the **Virtual Machines** tab and right-click to use the **Power Off** button to shut down the virtual machine to avoid the one-minute grace period RHEV uses.
- Navigate to the **Snapshots** tab in the lower pane.
- Select the **`Example Snapshot`** snapshot and click the **Preview**. The disk leaves the state **Locked** and the snapshot will show a status of **In Preview**. Start the virtual machine and open a console to show that the changes (**`/etc/issue`**) have been reverted.
- Power down the virtual machine again (with the **Power Off** button).
- In the **Snapshots** tab, select the snapshot and click the **Commit** button to permanently revert to this snapshot and throw away all later snapshots.

## Demonstration: Deleting virtual machine snapshots

Snapshots can take up significant space in the storage domain, and some older snapshots may become irrelevant. Individual snapshots can be removed in the RHEV-M administrative interface in order to free space in the storage domain. The steps to delete a snapshot are outlined as follows:

In this example, observe the following steps as the instructor demonstrates deleting a virtual machine snapshot.

1. Select the virtual machine from the **Virtual Machines** tab and make sure it is powered down.
2. Navigate to the **Snapshots** tab in the lower pane. Select the snapshot to delete (**`Example Snapshot`**).
3. Click the **Delete** button and confirm the deletion of this snapshot by clicking **OK**.

### Note

RHEV-M can take a long time to remove snapshots. It is not an instantaneous process.

4. Wait for the snapshot to be gone.

### References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 9.12.1. Creating a Snapshot of a Virtual Machine

# Practice: Creating and Using Image Snapshots

In this lab, you will take and restore virtual machines snapshots.

### Outcome(s)

- You should be able to create a virtual machine snapshot named **`Baseline Snapshot`**, then discard changes made to a virtual machine after the snapshot was created.

**Before you begin**

Make sure to have a working Red Hat Enterprise Linux virtual machine installed named **`rhelX`** in RHEV Manager.

Perform the following steps:

1. Log into the RHEV-M web administration portal as the user **`rhevadmin`** with a password of **`redhat`**.

2. Click the **Virtual Machines** tab and check the status of **`rhelX`**. If the virtual machine is not in the **Down** state, right-click the virtual machine, and select the **Shutdown** menu item. Click the **OK** button in the confirmation dialog box that appears.

3. Once **`rhelX`** is down, click the virtual machine name again to highlight it. In the lower pane for the virtual machine, select the **Snapshots** tab, and click the **Create** button.

4. In the **Create Snapshot** window that appears, enter **`Baseline Snapshot`** in the **Description** text box. Click **OK** to confirm.

5. When the snapshot is complete, right-click **`rhelX`** and select **Run** to boot the virtual machine, then open the **`rhelX`** console.

   To open the virtual machine console, right-click the **`rhelX`** virtual machine to select it, then choose **Console** from the menu that appears.

6. Log into your **`rhelX`** virtual machine through its console. Make some changes to it that have obvious effects: add text to **`/etc/issue`**, create a new user, and so on.

   There are a variety of things you can do to modify the system from its original state at this point. Log in as **`root`** with the password of **`redhat`** and edit **`/etc/issue`**. Modify the login banner and save your changes. Log out and confirm the banner has changed.

7. In the RHEV-M Administration Portal, on the **Virtual Machines** tab, click **`rhelX`** to highlight it. Shut it down.

   In the **Virtual Machines** tab, right-click the **`rhelX`** virtual host entry, then select **Shutdown**. Click the **OK** button in the confirmation dialog box that appears.

8. Once **`rhelX`** is down, select its **Snapshots** tab and click **Baseline Snapshot**. Click the **Preview** button.

9. When the preview is complete, run **`rhelX`**. Verify the changes you made earlier have been reverted (e.g., **`/etc/issue`** has been restored to its original state). Shut **`rhelX`** down again.

   Right-click the **`rhelX`** virtual host entry, then select **Run**.

All changes you made previously in this practice should be restored to their original state. For example, if you modified **/etc/issue**, you should notice that the login banner no longer displays the text you changed or added.

Right-click the **rhel*X*** virtual host entry, then select **Shutdown**. Click the **OK** button in the confirmation dialog box that appears.

10. So far you have only been previewing the old snapshot. Once the virtual machine is down, click the **Commit** button, which will permanently revert to that snapshot and remove all later snapshots.

11. Once the commit is complete, run **rhel*X*** again. Your changes should have been permanently reverted.

    Right-click the **rhel*X*** virtual host entry, then select **Run**.

    All changes you previously made in this lab exercise have been permanently discarded. **rhel*X*** has been restored to its original state when the snapshot was created.

# Sharing and Editing Images

## Objectives

After completing this section, students should be able to:

- Export/import virtual machines.
- Edit virtual machine images with **guestfish**.

## Moving images between datacenters

RHEV Manager can import and export virtual machines and templates in the industry-standard Open Virtual Machine Format (**OVF**). The **OVF** file format is a legal XML file representing either a virtual machine or a template in complete detail, enabling the virtual machine to be exported from one physical host and imported into another physical host. This import/export feature can be used to:

- Distribute virtual appliances.
- Move virtual resources to a different data center within the same installation of RHEV.
- Move virtual resources to a different installation of RHEV.
- Store full system backups of virtual machines for disaster recovery.

To export a virtual machine from RHEV, administrators first need an **export domain**, accessible via the **New Domain** option, in the **Storage** tab. The domain type needs to be set to **Export/NFS**. Once the domain is created, administrators can specify the cluster to attach the domain to.

### Note

An export domain can only be attached to one datacenter at a time. Likewise, a datacenter is only allowed to have one export domain attached simultaneously.

Moving virtual machines from one datacenter to another is a five-step process:

1. The **export** domain needs to be attached to the source datacenter:

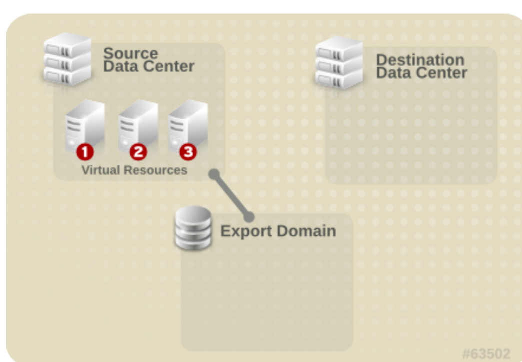


*Figure 8.2: Attaching the export domain*

2. The virtual machines can now be exported to the attached domain:

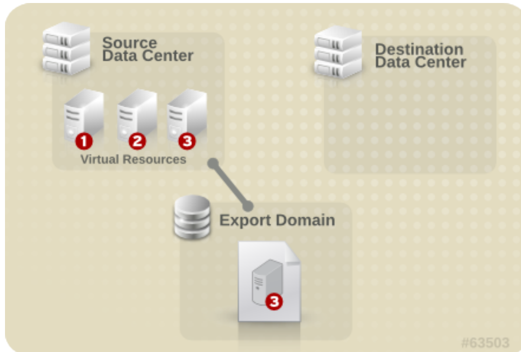


*Figure 8.3: Exporting the virtual machine*

3. Once the export is complete, the **`export`** domain can now be detached from the datacenter:

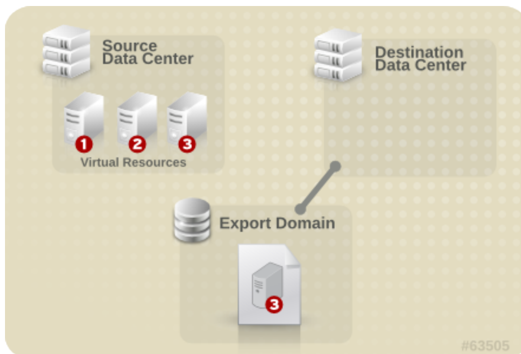


*Figure 8.4: Detaching the domain*

4. The export domain can be attached to the destination datacenter:

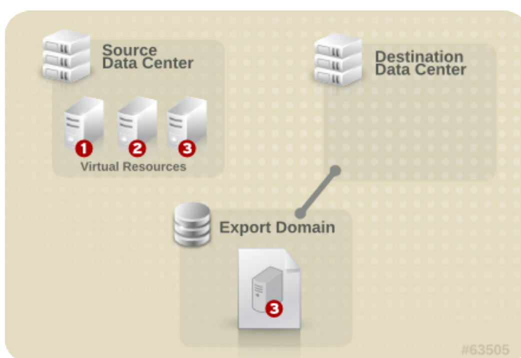


*Figure 8.5: Attaching the domain*

5. Once the **`export`** domain is attached, the virtual machine can be imported:

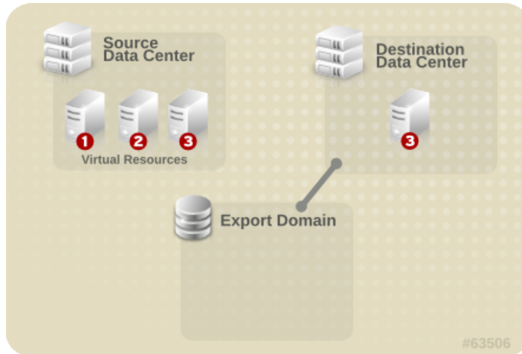


*Figure 8.6: Attaching the resource*

# Exporting virtual machines

To export a domain, administrators need to execute the following steps:

1. A virtual machine needs to be selected from the **Virtual Machines** tab in the RHEV-M administrative console.

2. Once selected, the virtual machine needs to be shut down.

3. Performing a right-click on the virtual machine will display the contextual menu. Clicking **Export** will display the **Export Virtual Machine** dialog.

4. By choosing **Force Override**, administrators can override existing images of the virtual machine that may already exist in the export domain. Choosing **Collapse Snapshots** will create a single export file per disk. Administrators can select the latter option if they wish to retain both the source and destination versions of the virtual machine.

5. Clicking **OK** will initiate the export.

Once validated, the export of the virtual machine begins, and the process can take some time, depending on the underlying architecture. The virtual machine that is being exported will be in the **Image Locked** state, while the export is running. The **Events** tab will display that the virtual machine has been exported.

Once the export is finished, administrators can navigate back to the **Data Centers** tab, and select the datacenter to which the **`export`** domain has been attached. Once the **`export`** domain is selected from the **Storage** tab in the lower pane, administrators can click the **Maintenance** button to put the export domain into **maintenance** mode. When the status for the export domain reaches the **Inactive** state, it is then possible to click the **Detach** button. The export domain can now be attached to another datacenter.

# Importing virtual machines

To import a virtual machine into a datacenter, it is required to attach it first to the destination datacenter. Follow the steps below to import the virtual machine into the destination datacenter:

1. Navigate to the **Data Centers** tab in the RHEV-M web interface and select the datacenter.

2. In the lower pane, navigate to the **Storage** tab and click the **Attach Export** button.

3. Select the export domain from the list and click **OK**.

4. Wait for the storage domain to reach the **Inactive** state, then select it and click the **Activate** button.

5. On the **Storage** tab, select the export data domain. The **Details** pane of the export data domain displays.

6. On the **Details** pane, select the **VM Import** tab.

7. Select the virtual machine that is to be imported.

8. Click the **Import** button. The **Import Virtual Machine** dialog displays.

9. Select the name of the virtual machine, then select the **Destination Cluster** and **Destination Storage** of the destination datacenter. If the original virtual machine has not been deleted in the source datacenter, select **Collapse Snapshots**.

10. Click **OK**.

The virtual machine is imported into the destination datacenter; depending on the underlying infrastructure, the process can take some time.

## Editing virtual machine images with `guestfish`

Red Hat Enterprise Linux 7 ships with a package called *guestfish*, based around the **`libguestfs`** library. The **`guestfish`** command allows administrators to mount and to use virtual machine disk images, without having to start the virtual machine itself. This can be useful for rescuing machines, as well as for modifying configurations before import of a virtual machine.

Consider the following **`guestfish`** sessions:

```
[root@rhevm ~]# guestfish -i --selinux -w -a /exports/exp/EXPORT-UUID/images/VM-
UUID/DISK-UUID
Welcome to guestfish, the libguestfs file system interactive shell for
editing virtual machine file systems.

Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quite the shell

Operating system: Red Hat Enterprise Linux Server release 6.2 (Santiago)
/dev/vda2 mounted on /
/dev/vda1 mounted on /boot

><fs> vi /etc/issue
><fs> quit
```

The options used previously include:

- **`-i`**: automatically mounts all file systems found within the virtual machine

- **`--selinux`**: enables SELinux support

- **`-w`**: mounts the image read-write (**`-r`** for read-only)

- **`-a`** ***`image-file`***: adds the *`image-file`* disk to the virtual machine, can be used multiple times

# Demonstration: Virtual machine exports and imports

In this example, observe the following steps as the instructor demonstrates the exporting and importing of a virtual machine.

1. Export the **rhel0** virtual machine and collapse all snapshots during the export.

   Browse to the **Virtual Machines** tab. Right-click the **rhel0** virtual machine and choose **Export**. Select the **Collapse Snapshots** checkbox and press **OK**. Wait for the export to complete.

2. Detach and remove the **exp0** export domain.

   Navigate to the **Data Centers** tab. Select the **dcenter0** datacenter. In the bottom pane, right-click the **exp0** export domain and choose **Maintenance**. Once the export domain is in maintenance mode, right-click the new export domain and choose **Detach**. Click **OK** to confirm.

   Navigate to the **Storage** tab. Right-click the **exp0** storage domain and select **Remove**. Click the **OK** button. Emphasize that the **Format Domain** box should *not* be checked.

3. Once the virtual machine has been successfully exported, proceed to remove the virtual machine.

   In the **Virtual Machines** tab, right-click the **rhel0** virtual machine and select **Remove**. Click **OK** to confirm.

4. Log in as **root** on the **rhevm.pod*X*.example.com** machine. Install the *libguestfs-tools* package.

   ```
   [root@rhevm ~]# yum -y install libguestfs-tools
   ```

5. Mount the virtual machine export image so it can be updated.

   ```
   [root@rhevm ~]# guestfish -i -w -a /exports/exp/EXPORT-UUID/images/VM-UUID/DISK-UUID
   Welcome to guestfish, the libguestfs filesystem interactive shell for
   editing virtual machine filesystems.

   Type: 'help' for help on commands
         'man' to read the manual
         'quit' to quit the shell

   Operating system: Red Hat Enterprise Linux Server release 7.1 (Maipo)
   /dev/sda2 mounted on /

   ><fs>
   ```

6. Edit the **/etc/issue** file and disconnect from the disk image.

   ```
   ><fs> vi /etc/issue
   ><fs> exit
   ```

7. Import the export domain.

Browse to the RHEV-M web interface, and choose the **Storage** tab. Click the **Import Domain** button. Verify the **Domain Function / Storage Type** is set to **`Export / NFS`**. For the **Export Path**, enter **`rhevm.podX.example.com:/exports/exp`** and click the **OK** button.

8. Activate the export domain.

   Navigate to the **Data Centers** tab and highlight the **`dcenter0`** datacenter. In the bottom pane, right-click the **`exp0`** export domain and choose **Activate**.

9. Import the virtual machine.

   Back in the **Storage** tab, highlight the **`exp0`** export domain. In the bottom pane, navigate to the **VM Import** tab. Right-click the **`rhel0`** virtual machine and select **Import**. Leave the settings at their defaults and press **OK** to confirm.

R

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 9.14. Importing and Exporting Virtual Machines

Red Hat Enterprise Linux 7 Virtualization Administration Guide

- Chapter 17. Guest virtual machine disk access with offline tools

The **`guestfish`**(1) man page

# Practice: Sharing and Editing Images

In this lab, you will export, edit, and import a virtual machine.

### Outcome(s)

- You should be able to export then delete the virtual machine **rhel*X***, modify the image with **guestfish**, then restore the virtual machine by importing the modified image.

**Before you begin**

Make sure to have a working Red Hat Enterprise Linux virtual machine installed named **rhel*X*** in RHEV Manager.

Perform the following steps:

1. From workstation, open Firefox, and log into the RHEV-M web interface, using **rhevadmin** as username and **redhat** as password. Once logged in, click System.

2. Shut down the **rhel*X*** virtual machine by browsing to the Virtual Machines tab. Right-click **rhel*X*** and choose Shutdown.

   When the machine has finished shutting down, export the **rhel*X*** virtual machine by right-clicking the **rhel*X*** virtual machine and choosing Export. Select the Collapse Snapshots checkbox and press OK. Wait for the export to complete.

3. Detach the **testexport*X*** export domain by navigating to the Data Centers tab. Select the **testdc*X*** datacenter. In the bottom pane, right-click the new export domain (**testexport*X***) and choose Maintenance. Once the export domain is in maintenance mode, right-click the new export domain and choose Detach. Click OK to confirm.

   To remove the **testexport*X*** export domain, navigate to the Storage tab. Right-click the **testexport*X*** storage domain, and select Remove. Click the OK button (do *not* check the Format Domain box).

4. Once the virtual machine has been successfully exported, proceed to remove the virtual machine. In the Virtual Machines tab, right-click the **rhel*X*** virtual machine and select Remove. Click OK to confirm.

5. Log in as **root** on your **rhevm.pod*X*.example.com** machine. Install the necessary packages to use the **guestfish** utility.

```
[root@rhevm ~]# yum -y install libguestfs-tools
```

6. Mount the virtual machine export image so it can be updated.

```
[root@rhevm ~]# guestfish -i -w -a /exports/testexport/EXPORT-UUID/images/VM-
UUID/DISK-UUID
Welcome to guestfish, the libguestfs filesystem interactive shell for
editing virtual machine filesystems.

Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell
```

```
Operating system: Red Hat Enterprise Linux Server release 6.4 (Santiago)
/dev/vda2 mounted on /
/dev/vda1 mounted on /boot

><fs>
```

7. Modify the **/etc/issue** file in the image so it contains a distinctive message that makes it clear that it has been modified. For instance, add a line to the bottom of the file that reads: **Testing guestfish**. When you are finished, disconnect from the disk image.

```
><fs> vi /etc/issue
><fs> exit
```

8. Import the export domain by browsing to the RHEV-M web interface, and choosing the **Storage** tab. Click the **Import Domain** button. Verify the **Domain Function / Storage Type** is set to **Export / NFS**. For the **Export Path**, enter **rhevm.pod*X*.example.com:/exports/testexport** and click the **OK** button.

9. To activate the export domain, navigate to the **Data Centers** tab and highlight the testdc*X* datacenter. In the bottom pane, right-click the **testexport*X*** export domain, and choose **Activate**.

10. Back in the **Storage** tab, highlight the **testexport*X*** export domain. In the bottom pane, navigate to the **VM Import** tab. Right-click the **rhel*X*** virtual machine and select **Import**. Leave the settings at their defaults and press **OK** to confirm.

11. Verify the import and edit worked. Once the import has completed, browse to the **Virtual Machines** tab. Right-click **rhel*X*** and select **Run** to start the virtual machine. Open a console and verify that the changes can be seen from the login prompt.

# Quiz: Managing VM Snapshots and Images

Choose the correct answer to the following questions:

1. Name the view of a virtual machine's operating system, and applications on any or all available disks, at a given point in time. (Choose one.)

   a. Clones
   b. Snapshots
   c. Templates
   d. Duplicate

2. Name the technology that pauses or alters the state of running processes on a computer, in order to guarantee a consistent and usable backup. (Choose one.)

   a. Quiescing
   b. Memory dumping
   c. Unallocating

3. Name the legal XML file format representing either a virtual machine or a template in complete detail. (Choose one.)

   a. Virtual Machine Template (**`VMT`**)
   b. Open Virtual Format (**`OVF`**)
   c. Virtual Machine Open Format (**`VMOF`**)
   d. Open Virtual Machine Format (**`OVMF`**)

4. Name the required domain to create for sharing images. (Choose one.)

   a. **`publish`**
   b. **`export`**
   c. **`data`**
   d. **`shares`**

5. Name the state for the virtual machine being exported. (Choose one.)

   a. Locked
   b. Paused
   c. Frozen
   d. Read-only

6. Name the tab to look at to monitor export progress. (Choose one.)

   a. **Activity**
   b. **Events**
   c. **Progress**
   d. **Tasks**

# Solution

Choose the correct answer to the following questions:

1. Name the view of a virtual machine's operating system, and applications on any or all available disks, at a given point in time. (Choose one.)

    a. Clones
    **b. Snapshots**
    c. Templates
    d. Duplicate

2. Name the technology that pauses or alters the state of running processes on a computer, in order to guarantee a consistent and usable backup. (Choose one.)

    **a. Quiescing**
    b. Memory dumping
    c. Unallocating

3. Name the legal XML file format representing either a virtual machine or a template in complete detail. (Choose one.)

    a. Virtual Machine Template (**VMT**)
    **b. Open Virtual Format (OVF)**
    c. Virtual Machine Open Format (**VMOF**)
    d. Open Virtual Machine Format (**OVMF**)

4. Name the required domain to create for sharing images. (Choose one.)

    a. **publish**
    **b. export**
    c. **data**
    d. **shares**

5. Name the state for the virtual machine being exported. (Choose one.)

    **a. Locked**
    b. Paused
    c. Frozen
    d. Read-only

6. Name the tab to look at to monitor export progress. (Choose one.)

    a. Activity
    **b. Events**
    c. Progress
    d. Tasks

# Summary

### Creating and Using Image Snapshots

In this section, you learned:

- Using an existing virtual machine, baseline snapshots can be created.
- Editing a virtual machine and checking the snapshot are methods used to ensure the snapshot integrity.

### Sharing and Editing Images

In this section, you learned:

- Use **guestfish** to mount a virtual machine disk image.
- RHEV-M offers the possibility to export and activate a domain.

# CHAPTER 9

# AUTOMATING VM DEPLOYMENT

| Overview | |
|---|---|
| **Goal** | Use templates, pools and **`cloud-init`** to automate deployment of VMs. |
| **Objectives** | • Describe the templating process.<br>• Seal a RHEL VM for template creation.<br>• Seal a Windows VM for template creation.<br>• Create and manage RHEL pools.<br>• Automate RHEL deployment with **`cloud-init`**. |
| **Sections** | • Templating Process (and Quiz)<br>• Creating RHEL VMs with Template Images (and Practice)<br>• Windows VMs with Template Images (and Quiz)<br>• Creating and Managing Pools (and Practice)<br>• Automating RHEL Deployment with **`cloud-init`** (and Practice) |

# Templating Process

## Objectives

After completing this section, students should be able to:

- Describe the templating process used by RHEV.

## Rapid deployments using templates

A template is a copy of a preconfigured virtual machine, used to simplify the subsequent, repeated creation of similar virtual machines. Templates capture the installed software, software configurations, and hardware configuration of the original virtual machine. When administrators need to deploy multiple (almost) identical machines, it can be beneficial to use *templates* instead of doing multiple installations. A *template* can be seen as a *bit-for-bit* copy of the original disk image and bears many resemblances to imaging a machine with tools like **Clonezilla** or **Ghost**. Just like an automated installation such as **`Kickstart`**, this will help maintain consistency across machines, making management and troubleshooting easier.

When working with virtual machines, basing machines on a template can also help reduce memory usage since Kernel Same-page Merging (**`KSM`**) will have a higher chance of finding duplicate memory pages in identical machines.

Using templates can also help reduce the amount of disk space needed. A template does not need to be cloned (but it can be cloned for performance reasons). Instead a small overlay can be put on top of the base image to store just the changes for one specific instance. Using this method of **Thin-Provisioning** is especially useful for desktop deployments and servers that won't see many changes to their disks (e.g., a web server in a hosting farm).

Templates are also necessary when administrators want to use **Pools** as discussed in the next chapter. Virtual machines in a pool will revert back to the base template when they are returned to the pool after use.

### Warning

One of the challenges involved in using templates is making sure that no identifying information, like references to MAC addresses or SSL certificates, are left behind since these may cause issues when multiple machines are deployed from the same template. This process is called sealing the image.

Consider the following list that describes the general steps necessary to create a template in RHEV:

1. Installing a fresh virtual machine to be the baseline for the template. It will be based off of the **Blank** template.

2. Removing all host-specific information from the new virtual machine (seal the machine). This includes virtual hardware information, such as MAC addresses, and unique system configuration, such as host name and static IP address.

3. Gracefully shutting down the sealed virtual machine.

4. Creating a template from the virtual machine.
5. Testing the template by creating another virtual machine based on the new template.

The following table describes the settings for the **New Template** window:

| Field | Description/action |
|---|---|
| Name | The name of the template. This is the name by which the template is listed in the **Templates** tab in the Administration Portal and is accessed via the REST API. This text field has a 40-character limit and must be a unique name with any combination of upper-case and lower-case letters, numbers, hyphens, and underscores. |
| Description | A description of the template. This field is recommended but not mandatory. |
| Comment | A field for adding plain text, human-readable comments regarding the template. |
| Cluster | The cluster with which the template is associated. This is the same as the original virtual machines by default. Any cluster in the datacenter can be selected. |
| Create as a Sub Template version | Specifies whether the template is created as a new version of an existing template. Administrators can select this checkbox to access the settings for configuring this option.<br>• **Root Template**: The template under which the subtemplate is added.<br>• **Sub Version Name**: The name of the template. This is the name by which the template is accessed when creating a new virtual machine based on the template. |
| Disks Allocation | • **Alias**: An alias for the virtual machine disk used by the template. By default, the alias is set to the same value as that of the source virtual machine.<br>• **Virtual Size**: The current actual size of the virtual disk used by the template. This value cannot be edited, and is provided for reference only.<br>• **Target**: The storage domain on which the virtual disk used by the template is stored. By default, the storage domain is set to the same value as that of the source virtual machine. Any storage domain in the cluster can be selected. |
| Allow all users to access this Template | Specifies whether a template is public or private. A public template can be accessed by all users, whereas a private template can only be accessed by users with the **`TemplateAdmin`** or **`SuperUser`** roles. |
| Copy VM permissions | Copies explicit permissions that have been set on the source virtual machine to the template. |

To create a template, administrators need to select the **Virtual Machines** tab, right-click the virtual machine they want to use for the template, then select **Make Template** from the menu. This will bring up a dialog like the one shown as follows.

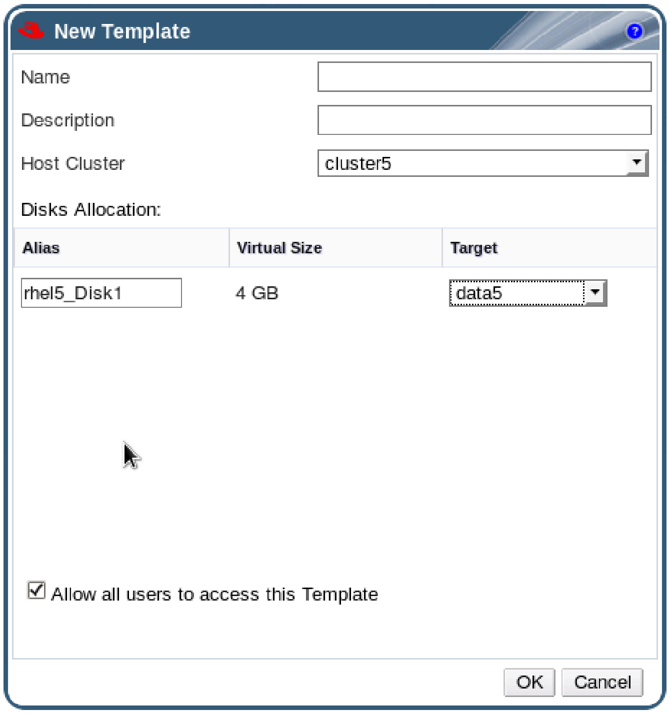


*Figure 9.1: The New Template dialog*

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 10.1. Introduction to Templates
- Section 10.3. Sealing Virtual Machines in Preparation for Deployment as Templates

# Quiz: Templating Process

The steps to establish a template are shown as follows. Indicate the order the steps should be taken.

__ a. Shut down the new virtual machine.

__ b. Install a new virtual machine from the **Blank** template.

__ c. Test the template by creating a virtual machine based on the template.

__ d. Create a template from the virtual machine.

__ e. Remove all identifying information from the new virtual machine (seal the machine).

# Solution

The steps to establish a template are shown as follows. Indicate the order the steps should be taken.

3 a. Shut down the new virtual machine.

1 b. Install a new virtual machine from the **Blank** template.

5 c. Test the template by creating a virtual machine based on the template.

4 d. Create a template from the virtual machine.

2 e. Remove all identifying information from the new virtual machine (seal the machine).

# Creating RHEL VMs with Template Images

## Objectives

After completing this section, students should be able to:

- Seal a Red Hat Enterprise Linux virtual machine.
- Create a new Red Hat Enterprise Linux-based template.
- Deploy a new Red Hat Enterprise Linux machine based on a template.

## Sealing Red Hat Enterprise Linux 6 for deployment

Normally, more than one virtual machine will be created from a single template, so all machine-specific information should be removed from the system to be used as a template. Ideally the machine-specific information is automatically detected or regenerated when the virtual machine boots without any needed operator intervention. This information includes, but is not limited to, the following parts (the table also shows how to remove this information from a system).

| Identifying information | How to remove |
|---|---|
| Static IP configurations | Make sure that **BOOTPROTO=dhcp** is set in all **/etc/sysconfig/network-scripts/ifcfg-*** files (except **ifcfg-lo**), or make sure that the system will reconfigure networking at first boot. |
| References to MAC addresses and other hardware-specific information. | Remove all **/etc/udev/rules.d/*-persistent-*.rules** files and remove all **HWADDR=** lines from **/etc/sysconfig/network-scripts/ifcfg-***. |
| Host names | Remove the **HOSTNAME=** line from **/etc/sysconfig/network**, or set **HOSTNAME=localhost.localdomain**. This should result in the network scripts setting the host name to the reverse-DNS lookup for a dynamic IP address. |
| SSH host keys | Remove the files **/etc/ssh/ssh_host*** and **/etc/ssh/moduli**. |
| SSL certificates | Make sure that there are no machine-specific certificates or keys left in **/etc/pki/tls/certs/**, **/etc/pki/tls/private/**, or any other location where applications might have stored keys. |
| Kerberos keytabs | Remove **/etc/krb5.keytab** and any application-specific Kerberos keytabs. |
| RHN System ID | Remove the file **/etc/sysconfig/rhn/systemid**. |

## Sealing Red Hat Enterprise Linux 7 for deployment

Red Hat Enterprise Linux 7 has shifted some identifying information to new places. The following table begins to list some of those changes (still consider the previous table as the starting point).

| Identifying information | How to remove |
|---|---|
| Host names | Most Red Hat Enterprise Linux 7 documentation will suggest removing **/etc/hostname**. Instead, set the host name to **localhost** to trigger the reverse DNS lookup when the system is initialized (a blank host name as suggested in the man page did not trigger the look up in RHEL 7.1).<br>`# hostnamectl set-hostname "localhost"` |
| Subscription | Use **subscription-manager unregister** to remove any RHSM configuration. |
| Machine ID | Consider whether **/etc/machine-id** needs to be unique in the environment. Review the man pages on **machine-id(5)**, **systemd-firstboot(1)**, and **systemd-machine-id-setup(1)** for possible approaches. |

## Sealing with virt-sysprep

With direct access to the disk image of the shut-down Red Hat Enterprise Linux virtual machine, administrators can use the **virt-sysprep** command to remove the machine-specific information. This tool is the recommended tool for sealing images.

```
[student@workstation ~]$ virt-sysprep --list-operations
abrt-data * Remove the crash data generated by ABRT
bash-history * Remove the bash history in the guest
blkid-tab * Remove blkid tab in the guest
ca-certificates   Remove CA certificates in the guest
crash-data * Remove the crash data generated by kexec-tools
cron-spool * Remove user at-jobs and cron-jobs
customize * Customize the guest
dhcp-client-state * Remove DHCP client leases
dhcp-server-state * Remove DHCP server leases
dovecot-data * Remove Dovecot (mail server) data
firewall-rules   Remove the firewall rules
flag-reconfiguration   Flag the system for reconfiguration
fs-uuids   Change filesystem UUIDs
kerberos-data   Remove Kerberos data in the guest
logfiles * Remove many log files from the guest
lvm-uuids * Change LVM2 PV and VG UUIDs
machine-id * Remove the local machine ID
mail-spool * Remove email from the local mail spool directory
net-hostname * Remove HOSTNAME in network interface configuration
net-hwaddr * Remove HWADDR (hard-coded MAC address) configuration
pacct-log * Remove the process accounting log files
package-manager-cache * Remove package manager cache
pam-data * Remove the PAM data in the guest
puppet-data-log * Remove the data and log files of puppet
rh-subscription-manager * Remove the RH subscription manager files
rhn-systemid * Remove the RHN system ID
rpm-db * Remove host-specific RPM database files
samba-db-log * Remove the database and log files of Samba
script * Run arbitrary scripts against the guest
smolt-uuid * Remove the Smolt hardware UUID
ssh-hostkeys * Remove the SSH host keys in the guest
ssh-userdir * Remove ".ssh" directories in the guest
sssd-db-log * Remove the database and log files of sssd
tmp-files * Remove temporary files
udev-persistent-net * Remove udev persistent net rules
```

```
user-account   Remove the user accounts in the guest
utmp * Remove the utmp file
yum-uuid * Remove the yum UUID
```

To remove the default information, administrators can use **virt-sysprep -a *disk.img***. The man page contains recommendations from the utility author. The **libguestfs-tools** can then be used to deposit some automation at first boot to avoid operator interaction.

# Demonstration: Sealing Red Hat Enterprise Linux 7 for deployment

In this example, observe the following steps as the instructor demonstrates how to manually seal a RHEL virtual machine for deployment.

1. Make sure **rhel0** is running, open its console, and log into the virtual machine as **root**.
2. Remove any SSH host keys by running:

   ```
   [root@vmY ~]# rm -f /etc/ssh/ssh_host_*
   ```

3. Remove **/etc/hostname**.

   ```
   [root@vmY ~]# hostnamectl set-hostname "localhost"
   ```

4. Make sure that there are no machine-specific certificates or keys left in **/etc/pki/tls/certs/**, **/etc/pki/tls/private/**, or any other location where applications might have stored keys.
5. Remove the values from the **UUID** and **HWADDR** lines in **/etc/sysconfig/network-scripts/ifcfg-eth***.

   ```
   [root@vmY ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
   ```

6. Delete all the logs from **/var/log/**, as well as build logs from **/root**.

   ```
   [root@vmY ~]# rm -rf /var/log/*
   ```

7. Finally, run **sys-unconfig** to complete the process and shut down the virtual machine.

# Sealing for manual deployment

The implication of a manual deployment is that the user who boots the resulting virtual machine will be prompted for basic configuration information, like network settings, to get the virtual machine running.

Although there is a tool called **sys-unconfig** that can be run on the running virtual machine, this utility will not do all of the above. **sys-unconfig** only removes the persistent **udev** rules and puts the system into a manual *reconfiguration* mode by creating the file **/.unconfigured**, as recommended in the listed references.

Another alternative method exists to manually put the system into *reconfigure* mode. In order to use this method, administrators can perform the following steps:

```
[root@vmY ~]# yum -y install firstboot
[root@vmY ~]# chkconfig firstboot on
[root@vmY ~]# sed -i 's/^RUN_FIRSTBOOT=.*/RUN_FIRSTBOOT=YES/' /etc/sysconfig/firstboot
[root@vmY ~]# touch /etc/reconfigSys
```

To recreate machine-specific information like Kerberos keytabs, administrators can use a configuration management system like Red Hat Network Satellite or write a custom firstboot module.

Power down the virtual machine when ready to create the template.

## Demonstration: Creating a Red Hat Enterprise Linux 7 template

The procedure for creating a template based on a Red Hat Enterprise Linux 7 virtual machine follows and presumes the virtual machine is already "sealed".

In this example, observe the following steps as the instructor demonstrates how to create a template based on a Red Hat Enterprise Linux 7 virtual machine.

1. Make sure the virtual machine is powered off and the state as reported in RHEV-M is **Down**.

2. Select the virtual machine in the **Virtual Machines** tab of the RHEV-M web interface and click **Make Template**.

3. In the **New Template** dialog that appears, enter the desired **Name** (**`rhel0-template`**) and **Description** (**`My template`**). Click **OK** to create the template.

4. Navigate to the **Templates** tab in the RHEV-M web interface and confirm the new template is there. Wait for the **Status** to change from **Locked** to **OK** before proceeding.

## Creating a new Red Hat Enterprise Linux 7 virtual server based on a template

Once the template for the virtual machine has been created, administrators can create new servers using this template.

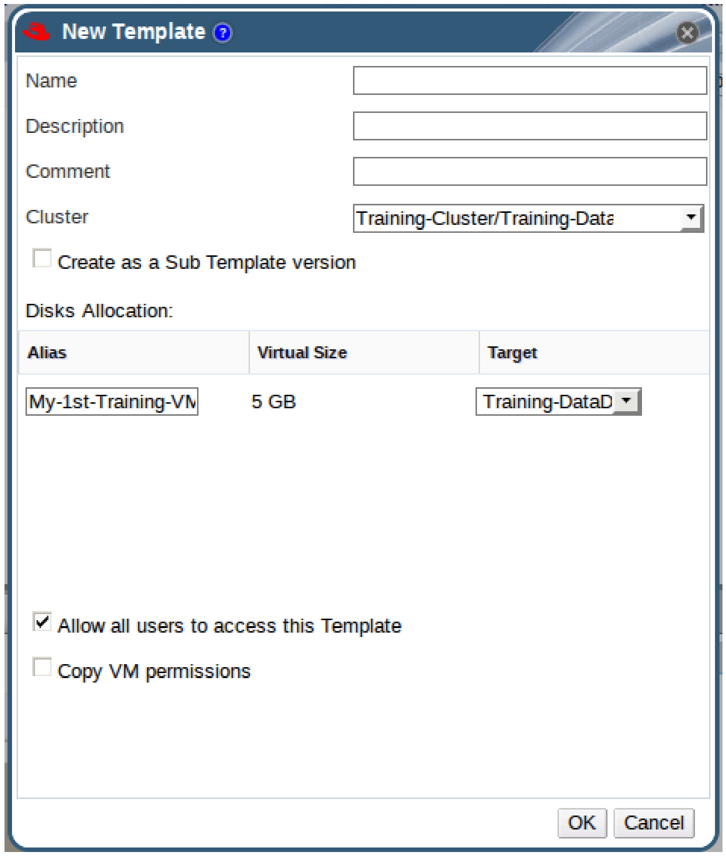


*Figure 9.2: New template dialog*

## Demonstration: Creating a new Red Hat Enterprise Linux 7 virtual server based on a template

In this example, observe the following steps as the instructor demonstrates how to create a new virtual server, based on the Red Hat Enterprise Linux 7 template that has been created previously.

1. To create a new virtual server based on a Red Hat Enterprise Linux template, log into the RHEV Administration Portal and navigate to the **Virtual Machines** tab. Click the **New VM** button to bring up the **New Virtual Machine** dialog.

2. In this dialog, enter the **Name** (**`rhel0-base`**) and **Description** (**`Template Child`**). In the **Based on Template** dropdown menu, select the earlier template (**`rhel0-template`**).

3. With the **New Virtual Machine** dialog still open, navigate to the **Resource Allocation** tab. Here, select how to provision the virtual disks for the new server. If **Provisioning** is set to **Thin**, an overlay will be used on top of the original template. If **Provisioning** is set to **Clone**, the original template will be cloned and the option to select (per disk) a **Preallocated** image or a **Thin Provision** disk becomes available.

4. Click **OK** to create the virtual machine. Wait for the **Status** to go from **Image Locked** to **Down** before proceeding.

5. Start the new virtual machine by right-clicking it in the **Virtual Machines** tab and selecting **Run** from the context menu.

6. Open a console to the new virtual machine by right-clicking it and selecting **Console** from the context menu.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 10.2. Template Tasks
- Section 10.3. Sealing Templates in Preparation for Deployment as Templates

Red Hat Enterprise Linux 7 Network Guide

- Section 8. Consistent Network Device Naming
- Section 1.9. Network Configuration Using Sysconfig Files

> FirstBoot/Modules - FedoraProject
> http://fedoraproject.org/wiki/FirstBoot/Modules

# Practice: Creating RHEL VMs with Template Images

In this lab, create a template from a Red Hat Enterprise Linux 7 virtual machine and use it to provision a new server.

### Outcome(s)

- You should be able to provision a new virtual server based on a newly created Red Hat Enterprise Linux 7 template.

**Before you begin**

Make sure that there is at least 10 GB available in your iSCSI data storage domain. Log into the RHEV Administration Portal, navigate to the **Storage** tab, and select the iSCSI data storage domain (iscsidata*X*). In the **General** tab in the bottom pane, view the **Available** space and ensure it is larger than 10 GB. If there is not enough space, remove some of the data stored in iSCSI.

Perform the following steps:

1. Log into the RHEV Administration Portal as **rhevadmin** and select the **Virtual Machines** tab.

2. Make sure your **rhel*X*** virtual machine is running.

   Once the system has finished booting, open the console and log in as **root** with a password of **redhat**.

3. To make the system ready for imaging, manually remove all host-specific information. Make sure to remove at least the following:
   - All references to MAC addresses.
   - All **SSH** keys.
   - Embedded host name.
   - The RHN **systemid** file associated with this system.

   To remove all references to MAC addresses, we must do two things: remove **/etc/udev/rules.d/*-persistent-*.rules** and remove all **HWADDR** and **UUID** lines from **/etc/sysconfig/network-scripts/ifcfg-***. To do so, execute the following commands:

   ```
   [root@vmY ~]# rm -f /etc/udev/rules.d/*-persistent-*.rules
   [root@vmY ~]# sed -i '/^HWADDR=/d' /etc/sysconfig/network-scripts/ifcfg-*
   [root@vmY ~]# sed -i '/^UUID=/d' /etc/sysconfig/network-scripts/ifcfg-*
   ```

   To remove all **SSH** keys, delete the **moduli** and **ssh_host*** files from **/etc/ssh/**:

   ```
   [root@vmY ~]# rm -f /etc/ssh/moduli /etc/ssh/ssh_host*
   ```

   To remove the embedded host name:

```
[root@vmY ~]# hostnamectl set-hostname "localhost"
```

To remove the RHN **systemid** associated with this system, run the following command:

```
[root@vmY ~]# rm -f /etc/sysconfig/rhn/systemid
```

4. Power down your **rhel*X*** machine by running the **poweroff** command.

5. In the **Virtual Machines** tab, select your **rhel*X*** machine and click **Make Template**. Use **rhel71-server** as the **Name** and **Basic RHEL7.1 Server** as the **Description**.

6. Wait for the template to be created. This should only take a couple of minutes.

7. Still in the **Virtual Machines** tab, click the **New VM** button. Select the **rhel71-server** template you created in the previous step. Use **One-More-Server** for the name and **Testing The Template** for the description. Click **OK** to complete the machine creation.

8. Start the new **One-More-Server** machine, and connect to the console. Verify that it has successfully established a network connection by logging in as **root** with a password of **redhat** and executing the following command:

```
[root@vmY ~]# ping -c3 172.25.254.254
```

9. Since there is limited disk space, remove the new VM.

   On the **Virtual Machines** tab, highlight the **One-More-Server** machine. Press the **Remove** button. Confirm the deletion.

# Windows VMs with Template Images

## Objectives

After completing this section, students should be able to:

- Seal a Windows machine for templating.
- Create a new Windows-based template.
- Deploy a new Windows machine based on a template.

## Sealing Windows for deployment

Sealing a Windows machine is slightly different from sealing a Red Hat Enterprise Linux machine. Before administrators can begin removing machine-specific information from a Windows Server 2008 or a Windows 7 host, they must make sure the Windows machine will look for configuration information in a file called **`sysprep.inf`** when it first boots. RHEV will provide the **`sysprep.inf`** file on a virtual floppy. These versions of Windows no longer automatically search on floppy drives, so they must be configured to search for this virtual floppy drive. Consider the following steps in order to seal a Windows machine:

1. Open a console to the Windows machine to template and log in.
2. Press **`Windows`**+**`R`** to launch a **Run** dialog. Enter **`regedit`** and press **`Enter`**. Click **OK** to authorize running **`regedit`** with system permissions.
3. In the **Registry Editor** window, use the left tree to navigate to **HKEY_LOCAL_MACHINE** > **SYSTEM** > **Setup**. In the right pane, right-click and select **New** > **String Value**. Enter **`UnattendFile`** as the key name. Double-click the new key and set the value to **`a:\sysprep.inf`**. This will allow Windows to find the **`sysprep.inf`** file that RHEV will make available to the new machine. Close the **Registry Editor** window. The following screenshot shows **`regedit`** with the **`UnattendFile`** key highlighted.

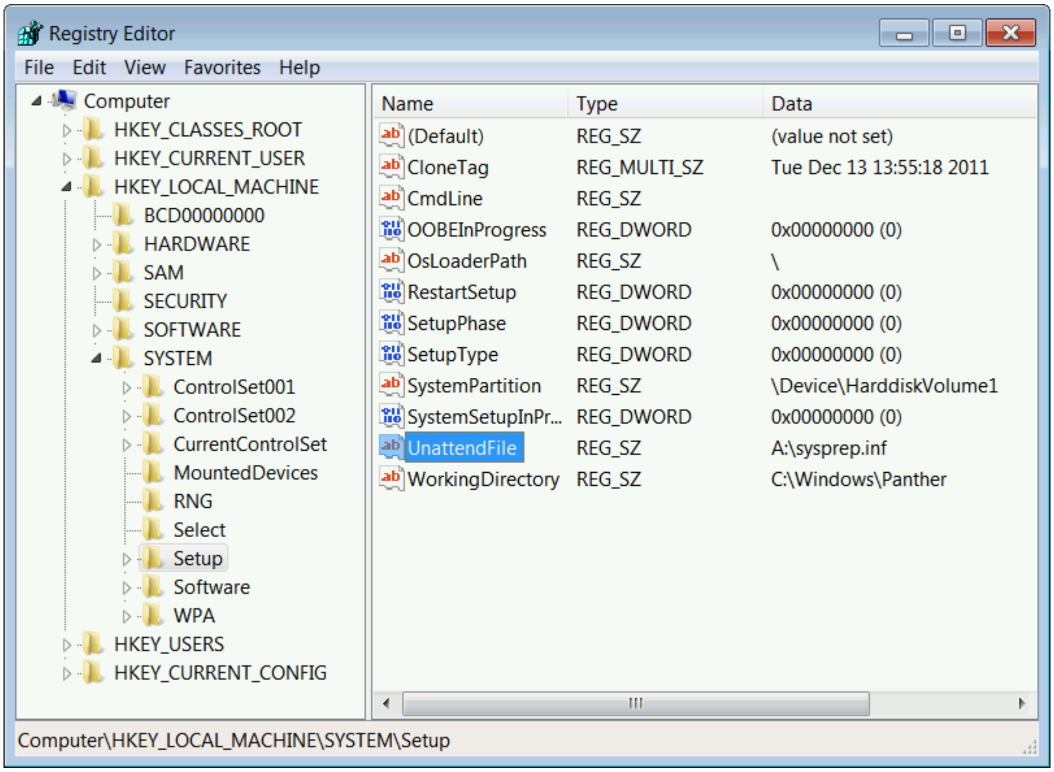

*Figure 9.3: The Windows Registry Editor*

Just like a Red Hat Enterprise Linux system, a Windows system will store machine-specific information like MAC addresses and security certificates. To remove all of this identifying information and force Windows into a *mini-setup* at the next reboot, administrators can use the **`sysprep.exe`** tool that ships with Windows. **`sysprep.exe`** can normally be found in **`C:\Windows\System32\sysprep\`** in Windows Server 2008 and Windows 7 machines. When **`sysprep.exe`** is started, a window will appear:

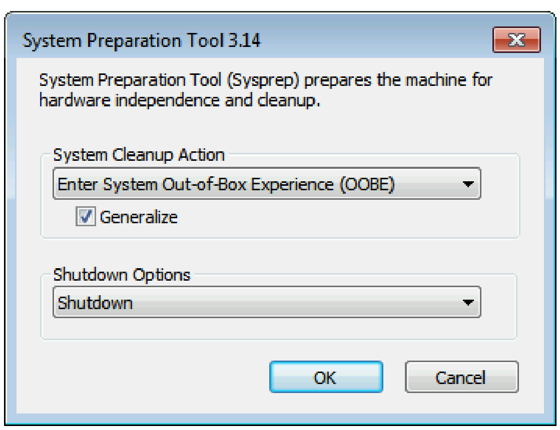

*Figure 9.4: Windows sysprep running on Windows 7*

In the **System Cleanup Action** dropdown menu, select **Enter System Out-of-Box Experience (OOBE)** and check **Generalize** to indicate that hardware-specific information should be stripped.

In **Shutdown Options**, select **Shutdown** to indicate that **`sysprep.exe`** should initiate a shutdown when it is finished.

# Creating a Windows template

Now that the virtual machine image has been sealed and the virtual machine shut down, administrators can create a template based on the virtual machine. Consider the following steps to create a Windows template in RHEV.

1. Log into the RHEV-M web interface and navigate to the **Virtual Machines** tab.
2. Select the virtual machine to use as the base for the template. Right-click the virtual machine and select **Make Template** from the context menu.
3. In the **New Template** dialog that appears, enter the **Name** and **Description** to use for the new template and click the **OK** button to initiate the template creation.
4. The virtual machine will transition to the **Image Locked** state while the template is being created. This can take a couple of minutes since the entire image will be copied.

# Create a new Windows virtual desktop based on a template

Once the Windows machine has been sealed, and the template created, administrators can create a new virtual server using the template. Consider the following steps to create a new virtual machine:

1. To create a new virtual desktop machine based on a template, administrators can log into the RHEV-M web interface and navigate to the **Virtual Machines** tab. The **New VM** button will bring up the **New Virtual Machine** dialog.
2. In this dialog, enter the **Name** and **Description** to use for the new virtual desktop. In the **Based on Template** dropdown menu, select the template to use as a basis for the virtual machine.
3. If the selected template has its **Operating System** set to a version of Windows, a new tab will appear in the **New Virtual Machine** dialog: **Windows Sysprep**. Navigate to this tab and select the Windows Active Directory domain from the **Domain** dropdown menu. The desired **Time Zone** for the new virtual desktop can be set here.

> **Note**
>
> Take care to choose a domain that is Active Directory-compatible for the Windows-based system to properly register with. Only domains that have been added with the **`rhevm-manage-domains`** utility will be shown in this list. If the desired domain is not listed, this utility must be used first in order to add the domain to RHEV-M.

4. With the **New Virtual Machine** dialog still open, navigate to the **Resource Allocation** tab. Here, choose whether the new virtual machine should run a cloned copy of the template (better performance, but requires more disk space), or whether the new virtual desktop should run from an overlay on top of the original template (slightly worse performance,

but more efficient use of available storage space). The default for a new virtual desktop is **Provisioning** set to **Thin** (overlay).

5. Once all of the settings are modified as desired, click **OK** to create the new virtual machine. The new virtual machine will appear with a status of **Image Locked**. Wait for the machine to report **Down** before proceeding.

6. The first time the new virtual desktop is started, it will need to be configured. To do so, select the virtual machine in the **Virtual Machines** tab of the RHEV-M interface, right-click the virtual machine, and select **Run Once** from the menu. In the **Run Virtual Machine(s)** dialog that appears, make sure that the **`[sysprep]`** floppy is attached and that **Domain** is set to the domain configured earlier. Click **OK** to start the virtual machine.

7. Once the virtual machine has started, open a console and wait for Windows to finish configuring the system. After the virtual machine reboots, a Windows login screen should appear.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 10.3.2. Sealing a Windows Virtual Machine for Deployment as a Template
- Section 10.3.2.1. Considerations when Sealing a Windows Template with Sysprep

# Quiz: Windows VMs with Template Images

Choose the correct answer to the following questions:

1. Which of the following will need to be removed from a Windows machine before templating? (Choose three.)

    a. All Security Identifiers (SIDs)
    b. All local users
    c. **`C:\Windows\System32\sysprep`**
    d. Hardware-specific information (e.g., MAC addresses)

2. What is the standard Windows tool to remove machine-specific information? (Choose one.)

    a. Ghost
    b. sysprep
    c. Clonezilla
    d. sys-unconfig

3. Which of the following will use more disk space? (Choose one.)

    a. A virtual machine based on a template using a clone.
    b. A virtual machine based on a template using a thin-provisioned overlay.

# Solution

Choose the correct answer to the following questions:

1. Which of the following will need to be removed from a Windows machine before templating? (Choose three.)

    **a. All Security Identifiers (SIDs)**
    **b. All local users**
    c. `C:\Windows\System32\sysprep`
    **d. Hardware-specific information (e.g., MAC addresses)**

2. What is the standard Windows tool to remove machine-specific information? (Choose one.)

    a. Ghost
    **b. sysprep**
    c. Clonezilla
    d. sys-unconfig

3. Which of the following will use more disk space? (Choose one.)

    **a. A virtual machine based on a template using a clone.**
    b. A virtual machine based on a template using a thin-provisioned overlay.

# Creating and Managing Pools

## Objectives

After completing this section, students should be able to:

- Create Red Hat Enterprise Virtualization pools.
- Edit an existing pool.

## Pools

A virtual machine pool is a group of virtual machines that are all clones of the same template and that can be used on demand by any user in a given group. Virtual machine pools allow administrators to rapidly configure a set of generalized virtual machines for users.

Users access a virtual machine pool by taking a virtual machine from the pool. When a user takes a virtual machine from a pool, they are provided with any one of the virtual machines in the pool if any are available. That virtual machine will have the same operating system and configuration as that of the template on which the pool was based, but users may not receive the same member of the pool each time they take a virtual machine. Users can also take multiple virtual machines from the same virtual machine pool, depending on the configuration of that pool.

Virtual machines in a virtual machine pool are stateless, meaning that data is not persistent across reboots. However, if a user configures console options for a virtual machine taken from a virtual machine pool, those options will be set as the default for that user for that virtual machine pool.

In principle, virtual machines in a pool are started when taken by a user, and shut down when the user is finished. However, virtual machine pools can also contain prestarted virtual machines. Prestarted virtual machines are kept in an up state, and remain idle until they are taken by a user. This allows users to start using such virtual machines immediately, but these virtual machines will consume system resources even while not in use due to being idle.

### Note

Virtual machines taken from a pool are not stateless when accessed from the Administration Portal. This is because administrators need to be able to write changes to the disk if necessary.

It is possible to have multiple pools, each with their own image. For example, you could have a separate pool for the Marketing department and another pool for the Finance department. A user will not always get the same virtual machine, but the user will get an available machine of the required type from the appropriate pool.

## Creating pools

Before a pool can be created, a template to use as a basis for the pool must exist. Once a template is built, administrators can log into the RHEV-M web interface and navigate to the **Pools** tab. Clicking **New** will bring up the dialog to define a new pool.

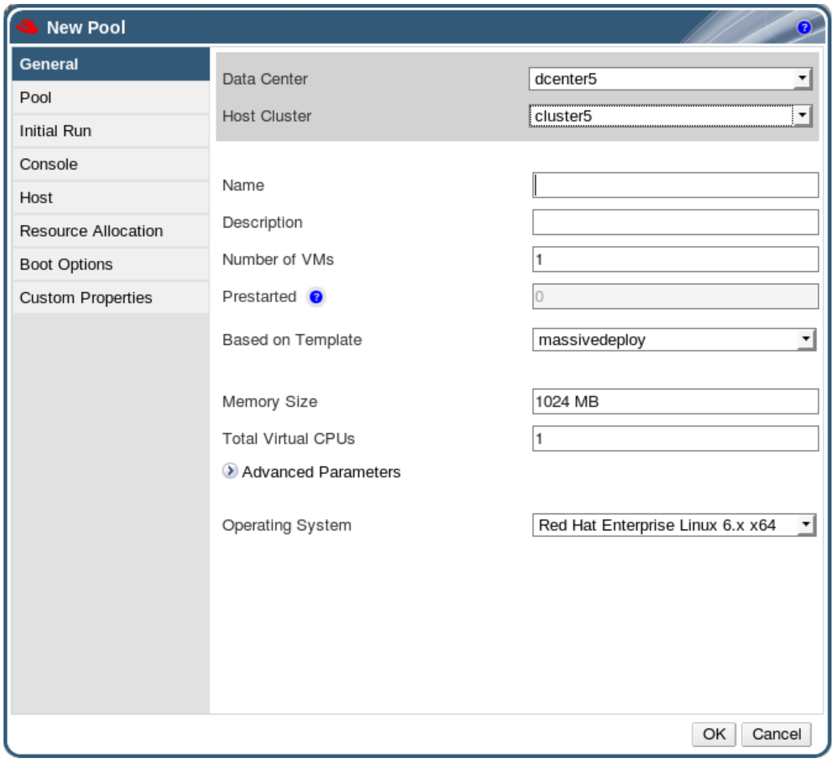


*Figure 9.5: The New Pool dialog*

In this dialog, administrators can set various options. The following table details the information required on the **General** tab of the **New Pool** window that is specific to virtual machine pools. All other settings are identical to those in the **New Virtual Machine** window.

| **General settings** | |
|---|---|
| **Number of VMs** | Allows administrators to specify the number of virtual machines in the virtual machine pool that will be created and made available to the virtual machine pool when that virtual machine pool is created. By default, the maximum number of virtual machines you can create in a pool is 1000. This value can be configured using the **`MaxVmsInPool`** key of the **`engine-config`** command. |
| **Maximum number of VMs per user** | Allows administrators to specify the maximum number of virtual machines a single user can take from the virtual machine pool at any one time. The value of this field must be between 1 and 32,767. |

The following table details the information required on the **Pool tab** of the **New Pool** window:

| **Console settings** | |
|---|---|
| **Pool Type** | This dropdown menu allows administrators to specify the type of the virtual machine pool. The following options are available:<br>• **Automatic**: After a user finishes using a virtual machine taken from a virtual machine pool, that virtual machine is automatically returned to the virtual machine pool. |

| Console settings | |
|---|---|
| | • **Manual**: After a user finishes using a virtual machine taken from a virtual machine pool, that virtual machine is only returned to the virtual machine pool when an administrator manually returns the virtual machine. |

The following table details the information required on the **Console** tab of the **New Pool** or **Edit Pool** window that is specific to virtual machine pools. All other settings are identical to those in the **New Virtual Machine** and **Edit Virtual Machine** windows:

| Console settings | |
|---|---|
| **Override Spice proxy** | Select this check box to enable overriding the Spice proxy defined in global configuration. This feature is useful in a case where the user (who is, for example, connecting via the User Portal) is outside of the network where the hypervisors reside. |
| **Overridden Spice proxy address** | The proxy by which the Spice client will connect to virtual machines. This proxy overrides both the global Spice proxy defined for the Red Hat Enterprise Virtualization environment and the Spice proxy defined for the cluster to which the virtual machine pool belongs, if any. The address must be in the following format: **`protocol://[host]:[port]`**. |

When administrators click **OK** to create the pool, both the pool and the virtual machines inside the pool will be created. The virtual machines will have names of *`Poolname`*-*`number`*. For example, if the pool is named **`marketing-desktops`** and three machines have been selected to be created, three virtual machines called **`marketing-desktops-1`**, **`marketing-desktops-2`**, and **`marketing-desktops-3`** will be created. It is possible to view these virtual machines either in the **Pools** tab by selecting the pool and navigating to the **Virtual Machines** tab in the lower pane, or by navigating to the **Virtual Machines** tab in the top pane. Virtual machines that belong to a pool will have a small **Pool** icon beside their names where administrators would normally see the **Desktop** or **Server** icon.

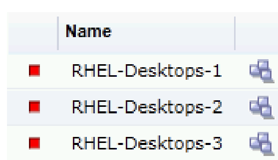


*Figure 9.6: Three virtual machines belonging to a pool*

> ### Important
>
> Pools created in this manner are not yet ready for use. In order for the pool to become fully usable, users will have to be assigned to the pool as well.

# Demonstration: Creating pools

In this example, observe the following steps as the instructor demonstrates the creation of a new pool.

1. Click the **Pools** tab.

2. Click the **New** button to open the **New Pool** window.

3. Use the dropdown list to select the cluster or use the selected default.

   - Use the **Based on Template** dropdown menu to select a template or use the selected default. If a template has been selected, optionally use the **Template Sub Version** dropdown menu to select a version of that template. A template provides standard settings for all the virtual machines in the pool.

   - Use the **Operating System** dropdown list to select an operating system or use the default provided by the template.

   - Use the **Optimized for** dropdown list to optimize virtual machines for either **Desktop** use or **Server** use.

4. Enter **`MyPool`** for the pool **Name**, and **`My First Pool`** for the pool **Description**. Optionally, enter a comment in the **Comments** field, as well as a number of virtual machines for the pool in the field **Number of VMs**.

5. Enter **2** for the **Maximum number of VMs per user** that a single user is allowed to run in a session. The minimum is one.

6. Optionally, click the **Show Advanced Options** button and perform the following steps:

   6.1. Select the **Console** tab. At the bottom of the tab window, select the **Override SPICE Proxy** checkbox to enable the **Overridden SPICE proxy address** text field and specify the address of a Spice proxy to override the global Spice proxy, if any.

   6.2. Click the **Type** tab and select a **Pool Type**:

   - **Manual**: The administrator is responsible for explicitly returning the virtual machine to the pool. The virtual machine reverts to the original base image after the administrator returns it to the pool.

   - **Automatic**: When the virtual machine is shut down, it automatically reverts to its base image and is returned to the virtual machine pool.

7. Click **OK** to create the pool.

# Managing pools

To add more machines to a pool, navigate to the **Pools** tab in the RHEV-M web interface and select the pool to manage. Click the **Edit** button to bring up the **Edit Pool** dialog.

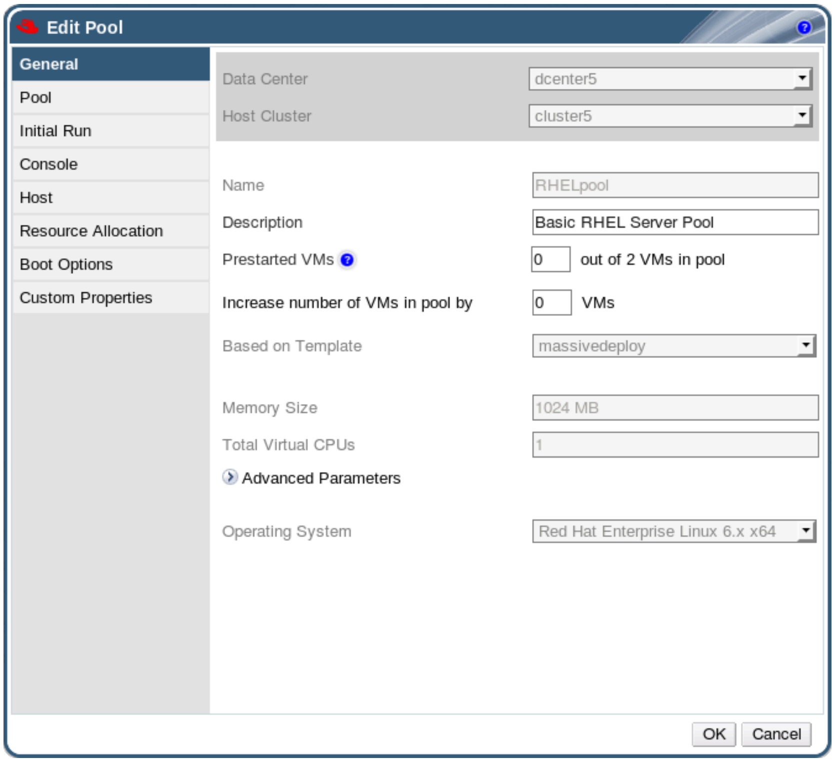


*Figure 9.7: The Edit Pool dialog*

While most settings will be initially locked, there are a few that can be edited:

| Name | Description |
| --- | --- |
| Name | The name and description of the pool. |
| Description | |
| **Prestarted VMs** | Allows administrators to specify the number of virtual machines in the virtual machine pool that will be started before they are taken and kept in that state to be taken by users. The value of this field must be between 0 and the total number of virtual machines in the virtual machine pool. |
| **Increase number of VMs in pool by** | Allows administrators to increase the number of virtual machines in the virtual machine pool by the specified number. |
| **Maximum number of VMs per user** | Allows administrators to specify the maximum number of virtual machines a single user can take from the virtual machine at any one time. The value of this field must be between 1 and 32,767. |

When administrators add more virtual machines to a pool, the **Memory Size**, **Total Cores**, **CPU Sockets**, and **Operating Systems** fields become accessible as well. Changing these fields will only affect new virtual machines being added to the pool.

# Demonstration: Editing a virtual pool

In this example, observe the following steps as the instructor demonstrates the editing of an existing pool.

1. Use the **Pools resource** tab (in tree mode), or the **`search`** function to find and select the virtual machine pool in the results list.

2. Click **Edit** to open the **Edit Pool** window.

3. Edit the properties of the virtual machine pool. Rename the existing pool to **`My Updated Pool`**.

4. Click **OK** to update the pool.

To detach a virtual machine from a pool, consider the following steps:

1. Navigate to the **Pools** tab in the RHEV-M web interface.

2. Select the pool to manage. In the bottom pane, navigate to the **Virtual Machines** tab.

3. Select the virtual machine(s) to detach from the pool, then click the **Detach** button.

4. Click **OK** in the confirmation dialog that appears to confirm detaching the virtual machine(s) from the pool.

### Warning

Once detached, a virtual machine cannot be added back into the pool.

5. After a virtual machine has been detached from the pool, it can be removed as normal by selecting it in the **Virtual Machines** tab then right-clicking and selecting **Remove**.

### Note

Virtual machines that are still part of a pool cannot be removed.

### References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 11.1. Introduction to Virtual Machine Pools
- Section 11.2.1. Creating a Virtual Machine Pool

# Practice: Creating and Managing Pools

In this lab, create a new pool of virtual machines.

**Outcome(s)**

- You should be able to establish a working pool that will consist of two Red Hat Enterprise Linux machines.

**Before you begin**

Make sure you still have the **`rhel71-server`** template created in an earlier lab.

Perform the following steps:

1. Log into the RHEV Administration Portal as the user **`rhevadmin`**, using the password **`redhat`**.

2. Navigate to the **Pools** tab. Click the **New** button to create a new pool. In the **General** tab of the dialog that appears, set the following values. Leave default values for all other settings.

| Name | Value |
|---|---|
| Name | RHELpool |
| Description | Basic RHEL Pool |
| Number of VMs | 2 |
| Based on Template | rhel71-server |

   Click **OK** to create the pool.

3. In the RHEV-M Administration Portal, navigate to the **Virtual Machines** tab. You should see two new virtual machines called *RHELpool-1* and *RHELpool-2*.

4. Switch back to the **Pools** tab and select the pool named **`RHELpool`**. In the bottom pane, switch to the **Virtual Machines** tab and select the machine named **`RHELpool-2`**. Right-click the machine name and choose **Detach** to detach this machine from the pool. Click **OK** in the dialog that pops up to confirm the operation.

5. Switch back to the top pane and select the **Virtual Machines** tab. Notice how **`RHELpool-2`** now has the normal desktop icon next to its name instead of the pool icon. Select **`RHELpool-2`**, then right-click and choose **Remove**. Click the **OK** button to confirm the removal. Wait for the removal to complete.

6. Switch to the **Pools** tab and select **`RHELpool`**. Click the **Edit** button to bring up the **Edit Pool** dialog. Change the **Increase number of VMs in pool by** value to **`1`** and click the **OK** button. You should see a new **`RHELpool-2`** machine being created in the **Virtual Machines** tab.

# Automating RHEL Deployment with `cloud-init`

## Objectives

After completing this section, students should be able to:

- Automate RHEL deployments using **`cloud-init`**.

## `cloud-init` overview

Red Hat provides for download a pre-installed image that can be used in both virtualization infrastructures and cloud deployments. While a raw KVM **`qcow2`** disk image can be downloaded from **`http://access.redhat.com`** on the same page as the installers, an **`rpm`** package is available from the **`Red Hat Enterprise Linux 7 Server - RH Common (RPMs)`** channel/repository. The **`cloud-init`** package is available in that same channel/repository and is pre-installed in the image.

**`cloud-init`** is a tool for automating the initial setup of virtual machines, such as configuring the host name, network interfaces, and authorized keys. It can be used when provisioning virtual machines that have been deployed based on a template to avoid conflicts on the network. To use this tool, the **`cloud-init`** package must first be installed on the virtual machine. Once installed, the **`cloud-init`** service starts during the boot process to search for instructions on what to configure. Administrators can then use options in the Run Once window, to provide these instructions one time only, or options in the New Virtual Machine, Edit Virtual Machine, as well as Edit Template windows, to provide these instructions every time the virtual machine starts.

**`cloud-init`** can be used to automate the configuration of virtual machines in a variety of scenarios. Several common scenarios are as follows:

- **Virtual machines created based on templates**: Administrators can use the **`cloud-init`** options in the Initial Run section of the Run Once window to initialize a virtual machine that was created based on a template. This allows administrators to customize the virtual machine the first time that virtual machine is started.

- **Virtual machine templates**: Administrators can use the Use Cloud-Init/Sysprep options in the Initial Run tab of the New Template and Edit Template windows to specify options for customizing virtual machines created based on that template.

- **Virtual machine pools**: Administrators can use the Use cloud-init/Sysprep options in the Initial Run tab of the New Pool window to specify options for customizing virtual machines taken from that virtual machine pool. This allows administrators to specify a set of standard settings that will be applied every time a virtual machine is taken from that virtual machine pool. It is possible to inherit or override the options specified for the template on which the virtual machine is based, or specify options for the virtual machine pool itself.

## Demonstration: Using `cloud-init` to initialize a virtual machine

In this example, observe the following steps as the instructor demonstrates the creation of a virtual machine, using **`cloud-init`**.

1. From workstation, SSH into rhevm.pod*X*.example.com:

```
[root@workstation ~]# ssh root@rhevm
```

2. Install the **rhel-guest-image** RPM file that contains the image:

```
[root@rhevm ~]# yum install http://content.example.com/rhel7.1/x86_64/extras/
Packages/rhel-guest-image-7-7.1-20150224.0.el7.noarch.rpm
...
Total size: 346 M
Installed size: 346 M
Is this ok [y/d/N]: Y
```

3. Using the **engine-image-uploader** command, upload the image in RHEV:

```
[root@rhevm ~]# cd /usr/share
[root@rhevm ~]# engine-image-uploader -e exp0 --name cloud-init upload rhel-guest-
image-7/rhel-guest-image-7.1-20150224.0.x86_64
Please provide the REST API password for the admin@internal oVirt Engine user (CTRL
+D to abort): redhat
```

4. From workstation, log into the RHEV Administration Portal.

5. Navigate to **System** > **Storage**, then click the **exp0** storage.

6. From the lower pane, click the **Template Import** tab to list all available images.

7. Select the **cloud-init** entry, then click the **Import** button.

8. From the pop-up, select **data0** for the **Default Storage Domain**, then click **OK** to initiate the import process. Do not check the **clone** box.

9. Navigate to **System** > **Data Centers** > **dcenter0**, then click **Templates**. Make sure the status is set to **OK** for the virtual machine **cloud-init**. If not, wait until the **Locked** state transitions to the **OK** state.

10. Still in the same datacenter, click the **VMs** entry, then click **New VM** in order to bring up the assistant.

11. Select **cloud-init** for the **Based on Template** option.

12. Enter **vm-cloudinit-demo** for the **Name**.

13. In the **nic1** option, click the dropdown to bring up all available network interfaces. Click **rhevm (rhevm)**.

14. Click **Show Advanced Options**, then select the **Initial Run** section. Check the **Use Cloud-Init/Sysprep** checkbox.

15. Enter a **VM hostname** of **vm-cloudinit-demo**.

16. Select the **Configure Time Zone** checkbox and select the time zone from the **Time Zone** dropdown menu.

17. Unfold the **Authentication** accordion. Enter a **User Name** of **`student`**, and a **Password** of **`redhat`**. Re-enter the password in the second field, **Verify Password**.

    Optionally, enter any SSH keys to be added to the authorized hosts file on the virtual machine in the **SSH Authorized Keys** text area.

    If SSH keys have been entered, select the **Regenerate SSH Keys** checkbox to regenerate SSH keys for the virtual machine; otherwise, leave the option unchecked.

18. Under the **Networks** section, enter the IP address **`172.25.254.254`** in the **DNS Servers** text field and **`podX.example.com`** in the **DNS Search Domains** text field.

19. Select the **Network** checkbox and use the **`+ Add New`** to add a network interface to the virtual machine. Click to edit that field to say **`eth0`**. Make sure **`DHCP`** is set for the **Boot Protocol** and check **`Start on Boot`**.

20. Finally, in the **Custom Script** text area, enter the following lines:

```
runcmd:
  - echo redhat | passwd --stdin root
  - echo "This VM has been provisioned using cloud-init" >> /etc/motd
```

### Note

**`cloud-init`** is particular about white-space indentation. The first line should not be indented and the additional lines need to indented the same number of spaces.

21. Click **OK** to initiate the creation of the virtual machine, using **`cloud-init`** as the provisioning method.

22. Once the virtual machine has been created, wait for it to be in a **`shutdown`** state. Click the green arrow to power it on.

23. Once the virtual machine is up and running, SSH into it, and verify that the following message appears on the screen:

```
This VM has been provisioned using cloud-init
```

    Optionally, launch a console to retrieve the state of the virtual machine.

### Note

It will appear that the machine is up, presenting a login prompt on the console even though **`cloud-init`** has not yet configured the system. If watching the console, wait for it to change the host name to be as specified.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 10.3.3. Using Cloud-Init to Automate the Configuration of Virtual Machines

# Practice: Automating RHEL Deployment with `cloud-init`

In this lab, you will create a new virtual machine, using **cloud-init** features.

### Outcome(s)

- You should be able to establish a new customized virtual server based on **cloud-init** features.

### Before you begin

- Make sure servera is up, and is attached to the **testdc*X*** datacenter.
- The datastore **testdata*X*** needs to be created and available.

Perform the following steps:

1. From workstation, SSH into rhevm.pod*X*.example.com:

```
[root@workstation ~]# ssh root@rhevm
```

2. Install the **rhel-guest-image** RPM file that contains the image:

```
[root@rhevm ~]# yum install http://content.example.com/rhel7.1/x86_64/extras/
Packages/rhel-guest-image-7-7.1-20150224.0.el7.noarch.rpm
...
Total size: 346 M
Installed size: 346 M
Is this ok [y/d/N]: Y
```

3. Using the **engine-image-uploader** command, upload the image in RHEV:

```
[root@rhevm ~]# cd /usr/share
[root@rhevm ~]# engine-image-uploader -e testexportX --name cloud-init upload rhel-
guest-image-7/rhel-guest-image-7.1-20150224.0.x86_64
Please provide the REST API password for the admin@internal oVirt Engine user (CTRL
+D to abort): redhat
```

4. From workstation, log into the RHEV-M.
5. Navigate to **System** > **Storage**, then click the **testexport*X*** storage.
6. From the lower pane, click the **Template Import** tab to list all available images.
7. Select the **cloud-init** entry, then click the **Import** button.
8. From the pop-up, select **iscsidata*X*** for the **Default Storage Domain**, then click **OK** to initiate the import process. Do not check the **clone** box.
9. Navigate to **System** > **Data Centers** > **testdc*X***, then click **Templates**. Make sure the status is set to **OK** for the virtual machine **cloud-init**. If not, wait until the **Locked** state transitions to the **OK** state.

10. Still in the same datacenter, click the **VMs** entry, then click **New VM** in order to bring up the assistant.

11. Select **`cloud-init`** for the **Based on Template** option.

12. Enter **`vm-cloudinit-demo`** for the **Name**.

13. In the **nic1** option, click the dropdown to bring up all available network interfaces. Click **`rhevm (rhevm)`**.

14. Click **Show Advanced Options**, then select the **Initial Run** section. Check the **Use Cloud-Init/Sysprep** checkbox.

15. Enter a **VM hostname** of **`vm-cloudinit-demo`**.

16. Select the **Configure Time Zone** checkbox and select the time zone from the **Time Zone** dropdown menu.

17. Unfold the **Authentication** accordion. Enter a **User Name** of **`student`**, and a **Password** of **`student`**. Re-enter the password in the second field, **Verify Password**.

18. Optionally, enter any SSH keys to be added to the authorized hosts file on the virtual machine in the **SSH Authorized Keys** text area.

    If SSH keys have been entered, select the **Regenerate SSH Keys** checkbox to regenerate SSH keys for the virtual machine; otherwise, leave the option unchecked.

19. Enter the IP address **`172.25.254.254`** for the DNS server in the **DNS Servers** text field in the **Networks** section.

    In the **DNS Search Domains** text field, enter **`pod`*`X`*`.example.com`**.

20. Select the **Network** checkbox and use the **`+ Add New`** to add a network interface to the virtual machine. Edit the field to **`eth0`**. Make sure **`DHCP`** is set for the **Boot Protocol**, and check **`Start on Boot`**.

21. Finally, in the **Custom Script** text area, enter the following lines:

```
runcmd:
  - echo redhat | passwd --stdin root
  - echo "This VM has been provisioned using cloud-init" >> /etc/motd
```

22. Click **OK** to initiate the creation of the virtual machine, using **`cloud-init`** as the provisioning method.

23. Once the virtual machine has been created, wait for it to be in a **`shutdown`** state. Click the green arrow to power it on.

24. Once the virtual machine is up and running, SSH into it, and verify that the following message appears on the screen:

```
This VM has been provisioned using cloud-init
```

Optionally, launch a console to retrieve the state of the virtual machine.

# Summary

### Templating Process

In this section, you learned:

- Templates are copies of virtual machines, used to simplify the subsequent creation of similar virtual machines.
- You need to remove machine-specific information before being able to create a template, such as the MAC address.

### Creating RHEL VMs with Template Images

In this section, you learned:

- Various configuration files need to be updated in order to seal the template, such as the **`RHN System ID`** configuration file.
- **`virt-sysprep`** allows for removing all machine-specific information.

### Windows VMs with Template Images

In this section, you learned:

- Sealing a Windows image is slightly different from sealing a RHEL image. The **`sysprep.inf`** file will be used upon server creation.

### Creating and Managing Pools

In this section, you learned:

- A pool is a group of virtual machines that are all clones of the same template.

### Automating RHEL Deployment with `cloud-init`

In this section, you learned:

- **Custom Script** is an option allowing the injection of extra **`cloud-init`** information in the virtual machine, such as **`MOTD`**s.

# CHAPTER 10

# MONITORING AND REPORTING OF RHEV

| Overview | |
| --- | --- |
| **Goal** | Locate information in the RHEV Administration Portal and generate reports of the RHEV environment. |
| **Objectives** | • Use the search bar and tags to query RHEV.<br>• Generate reports on RHEV with pre-built reports. |
| **Sections** | • Monitoring RHEV (and Practice)<br>• Generating Reports (and Practice) |
| **Lab** | • Monitoring and Reporting of RHEV |

# Monitoring RHEV

## Objectives

After completing this section, students should be able to:

- Use the search bar and tags to query RHEV.

## Using the search bar

- The RHEV Manager web administration interface has a powerful search bar that can be used to effectively find resources in the RHEV infrastructure. It supports both free-text searches and searches based around a formal syntax-based form. An autocomplete feature makes it easy for administrators who are unfamiliar with the search syntax to construct effective searches quickly. Frequently used searches can be bookmarked.

- The formal search syntax takes one required field: the type of resource to be returned in the search results. If no other parameters are provided, all resources will be returned. Search results can be narrowed to match a rich set of search criteria, which is useful when looking for a single virtual machine or host on a site managing thousands of guests on hundreds of hosts.

The following describes the syntax for search bar expressions:

```
Result-type: [Criteria...] [sortby Sort_spec]
```

- *`Result-type`* is a RHEV resource such as **`Vms`**, **`Datacenter`**, **`Hosts`**, **`Events`**, **`Cluster`**, **`Storage`**, or **`Template`**.

- Optional *`Criteria`* is an expression of the form: *`Property Operator Value`*.
  - *`Property`* is the thing being matched, like machine name.
  - *`Operator`* is how things are compared (= != > < >= <=).
  - *`Value`* is the value to match.

An optional **`sortby`** directive can be associated with a sorting specification to sort the results based on various criteria in ascending or descending order.

## Demonstration: Using the search bar

In this example, observe the following steps as the instructor demonstrates the use of the search bar.

A few examples of the search bar expression syntax are provided below. For more details on how to construct searches, consult the Red Hat Enterprise Virtualization Administration Guide.

1. Find all RHEV Hypervisor hosts that are up:

   ```
   Hosts: status = up
   ```

2. Find all RHEV Hypervisor hosts with VMs that are up:

```
Hosts: VMs.status = up
```

3. Find all virtual machines:

```
Vms:
```

4. Find all virtual machines with "rhel" in their name:

```
Vms: rhel
```

5. Find all virtual machines, sorted by uptime in descending order:

```
Vms: sortby uptime desc
```

6. Find all log events with a severity higher than **`Normal`**, sorted by time:

```
Events: severity > normal sortby time
```

7. Frequently used searches can be bookmarked.

   - Enter the search in the RHEV Manager search bar.
   - Click the star button to the right of the search bar.
   - Enter a name that clearly describes the search.
   - The bookmark will appear in the **Bookmarks** tab to the left of the main pane.

# Demonstration: Using tags

In this example, observe the following steps as the instructor demonstrates the use of tags.

Tags provide the ability to search for hosts or virtual machines according to predefined user tags, and filter the display according to specific tags. This is especially useful when there are many objects and concentrating on a specific set is desired. Tags can be arranged in a hierarchy that matches a set of needs.

1. Steps to create a tag

   1.a. Click the **Tags** button on the left pane to expand the **Tags** section.

   1.b. Select the node under which to create the tag. For example, to create it at the highest level, click the root node.

   1.c. Click the **New** button on the **Tags** pane. The **New Tag** dialog box displays.

   1.d. Enter the **Name** and **Description** of the new tag.

   1.e. Click **OK**.

2. Steps to edit a tag

    2.a. Click the **Tags** tab on the left pane.

    2.b. Select the tag to modify.

    2.c. Click **Edit**.

3. Steps to delete a tag

    3.a. Click the **Tags** tab.

    3.b. Select the tag to delete.

    3.c. Click **Remove**.

4. Adding tags to objects

    Tags can be attached to hosts and virtual machines. To add a tag to one or more objects, perform the following steps:

    4.a. Search for the object to tag.

    4.b. Select the object on the **Results** list.

    4.c. Click the **Assign Tags** button on the tool bar. A dialog box provides a list of tags.

    4.d. Select the checkbox to assign a tag to the object.

    4.e. Click **OK**.

5. Search for objects using tags

    To search for objects using tags, perform the following steps:

    5.a. Use the search bar and enter a search query of the object using **`tag`** as the property and the desired value as the criteria for the search.

    ```
    Vms: tag = nameoftag
    ```

    5.b. The objects tagged with the tag criteria specified will be listed in the results list.

## References

Additional information may be available in the *Administration Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Monitoring RHEV with the Search Bar and Tags

In this lab, search for RHEV resources using the search bar and tags.

| **Resources** | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

**Outcome(s)**
You should be able to perform various searches, create a search bookmark, and assign tags to the virtual machines.

**Before you begin**
You should have existing virtual machines installed and managed by your RHEV-M server.

Perform the following steps:

1. In the RHEV-M Administration Portal, click the **Virtual Machines** tab. Look at the **Search:** text box at the top of the window. It should contain the search **`Vms:`**, which means to display all virtual machines in the RHEV-M environment.

2. Replace **`Vms:`** with **`Vms: sortby uptime desc`**. Note the dropdown autocompletion that occurs as you type. Hit **`Enter`** if you have not already done so. This should again display all virtual machines in the RHEV environment, but this time it will sort them by descending uptime; machines up the shortest will be on the bottom of the list.

3. Now try another search. In the search bar, replace the current search string with **`Events: severity > normal sortby time`**. This will display all events with a severity greater than "normal" sorted in chronological order.

4. This seems like a search which you may want to run frequently, yet it's a handful to type. Set a bookmark for the search. Click the star icon to the right of the search bar. A window should open; for **Name:**, use **`important events`**. Leave the **Search string:** alone since it should already match what you have in the search bar. Click **OK**.

5. In the **Bookmarks** tab in the left pane of the RHEV-M interface, you should see **`important events`** displayed. In the search bar, replace the current search with **`Vms:`**. You should see a list of all your virtual machines. Now, test your bookmark. Click the **`important events`** bookmark under the **Bookmarks** tab. You should see the results of your bookmarked search in the main window.

6. Take a few moments to experiment with searches and search bookmarks.

7. Custom tags can be created, assigned to virtual machines, and used as a search criterion in the search bar. First, let's create the tags. Select the **Tags** tab. Create tags called **`server`** and **`desktop`** as children of the **`root`** tag. Then create a tag called **`rhel-srv`** as a child of the **`server`** tag and create a tag named **`rhel-pool`** as children of the **`desktop`** tag.

   Select the **Tags** tab, then select the **`root`** tag. Click the **New** button and type **`server`** for the **Name:**. Do the same for the **`desktop`** tag.

Select the **server** tag, then click the **New** button. Type **rhel-srv** in the **Name:** text box, then click **OK**. Select the **desktop** tag. Click the **New** button and type **rhel-pool** for the **Name:**.

8. Attach the tags to the relevant virtual machines. Tag the virtual machines according to the following:
   - Assign **rhel*X*-small** with the **rhel-srv** and **server** tags.
   - Assign **RHELpool-1** and **RHELpool-2** the **rhel-pool** and **desktop** tags.

   Select the **Virtual Machines** tab. Right-click the **rhel*X*-small** virtual machine and choose **Assign tags** from the menu. Select the checkbox by the **rhel-srv** and **server** tags and click the **OK** button to confirm.

   Right-click the **RHELpool-1** virtual machine and choose **Assign tags** from the menu. Select the checkbox by the **rhel-pool** and **desktop** tags and click the **OK** button to confirm.

   Right-click the **RHELpool-2** virtual machine and choose **Assign tags** from the menu. Select the checkbox by the **rhel-pool** and **desktop** tags and click the **OK** button to confirm.

9. Now perform a search of the virtual machines using tags. In the search bar, type **Vms: tag = desktop**, then click the **GO** button. You should see the virtual machines classified as desktops appear in the display window.

   Alternatively, for the **server** virtual machines, mouse over the **server** tag on the **Tags** tab. Notice that as you perform the mouse over, a pushpin icon appears next to the **server** tag. Click the pushpin. The **server** tag and the **Search:** field are both highlighted in green. Take note of the search that is autopopulated in the **Search:** field. To revert back to the original state, click the pushpin icon next to the **server** tag.

# Generating Reports

## Objectives

After completing this section, students should be able to:

- Generate reports on RHEV with prebuilt reports.

## Report Engine

- Report Engine is a built-in reporting tool for Red Hat Enterprise Virtualization Manager. It is comprised of JasperReports and JasperServer, which are open source reporting tools. JasperReports can produce reports that can be rendered to screen, printed, or exported to a variety of formats including PDF, CSV, Word, Excel, RTF, Flash, ODT and ODS.

- JasperServer allows reports built in JasperReports to be accessed via a web interface. Red Hat Enterprise Virtualization provides a customized implementation of JasperServer, which contains a range of preconfigured reports and dashboards, and provides the ability to create ad hoc reports.

- To access Report Engine, use a newer browser such as Firefox or Internet Explorer to access the following URL: **`https://`*`rhev-m.host.fqdn`*`/ovirt-engine--reports`** (the **Reports Portal** link on the main page). When prompted for a login, provide the username **`admin`** and the password assigned to the report administrator account when the **`engine-setup`** command was executed. Report user accounts are different than the RHEV users that administer the hypervisor hosts and virtual machines. Once logged into the Report Engine web interface, report users can be created and managed by hovering the mouse over the **Manage** tab, then selecting the **Users** menu item. There are two user roles that can be assigned to report users: **`ROLE_USER`** and **`ROLE_ADMINISTRATOR`**. Users with the **`ROLE_USER`** role can access standard reports and dashboards and create ad hoc reports. Users with the **`ROLE_ADMINISTRATOR`** role can additionally create and manage standard reports.

## The repository

An easy way to see which reports are available is to use the repository. To see which predefined reports are available in the repository, navigate to **View** in the top navigation bar and select **Repository**. Expand the **RHEVM Reports** folder to see the menus that are available. Current menus include:

- **Ad Hoc Components**
- **Dashboards**
- **Reports**
- **Resources**

The **Ad Hoc Components** menu lists **Domains** and **Topics** that are used as the basis for custom reports.

The **Dashboards** menu lists reports that give a high-level, executive view of system status. Typically each dashboard is a collection of reports that allows a variety of information to be displayed on a single screen. Dashboards can only be viewed on the screen. They cannot be exported in a variety of output formats in the way reports can.

The **Reports** menu has further submenus that include the following choices:

- **Executive**
- **Inventory**
- **Service Level**
- **Trend**

The **Executive** sub-menu contains high-level overviews of system status useful for capacity planning. The **Inventory** submenu reports list current system resources including physical hosts, storage domains, and virtual machines. The reports in the **Service Level** submenu display information about system performance and availability within user-defined thresholds. Finally the reports in the **Trend** submenu highlight changes in system utilization and availability, which are useful for identifying problems and bottlenecks in the system and for planning capacity expansion.

The **Resources** menu contains system configurations and resources that should not be modified. The other three selections are more useful for producing reports.

To create a custom report, hover over the **Create** tab and select **Ad Hoc View** from the pop-up menu that appears. Select either the **Topics** or **Domains** tab to determine the general type of information to display in the report. The shape of the report is selected by clicking the **Table**, **Chart**, or **Crosstab** below the selection tree. Select the corresponding option to open the editor that will be used to create the report.

## References

Additional information may be available in the *Administration Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at

> https://access.redhat.com/documentation/en-US/index.html

# Practice: Generating Reports (Prebuilt)

In this lab, sample some of the built-in reports provided by RHEV-M.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | http://rhevm.pod*X*.example.com |

### Outcome(s)

You should be able to generate a list of hosts that will be displayed to the screen and a list of RHEV virtual machines will be saved to a PDF.

**Before you begin**

You should have existing hosts and virtual machines installed and managed by your RHEV-M server.

Perform the following steps:

1. Launch the Firefox web browser on **`workstation`**. Go to the following link: **`https://rhevm.podX.example.com/ovirt-engine-reports`**. Authenticate with a user ID of **`admin`** and a password of **`redhat`**.
2. Select the View menu in the top navigation bar, and in the pop-up menu, choose Repository to open the repository.
3. Open the RHEVM Reports folder in the left pane and select the System Overview Dashboard in the Dashboards tree. Review the information displayed on the dashboard. When you are finished, navigate to the Library menu item in the top navigation bar.
4. Select the **`Hosts Inventory (BR1)`** report. Change the Data Center to be the active datacenter **`testdcX`** and click OK with the other default report parameters to display a list of RHEV-H hosts. When you are finished, navigate to the Library menu item in the top navigation bar.
5. Select the **`Virtual Machines Inventory (BR9)`** link. Ensure the Data Center field is set to the active datacenter **`testdcX`** and click OK with the other default report parameters to display a list of virtual machines. Hover the Export icon just to the right of the Save button. Choose As PDF in the dropdown menu to save the PDF report to a file and view it in the browser.

# Lab: Monitoring and Reporting of RHEV with Ad Hoc Reports

In this lab, you will create a custom RHEV Manager ad hoc report.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | https://rhevm.pod*X*.example.com |

Outcome(s)
You should be able to generate a custom report called **`Custom RH318 Report`** that displays a list of virtual machine attributes.

**Before you begin**
You should have existing virtual machines installed and managed by your RHEV-M server.

**Lab outline**

Create a custom report called **`Custom RH318 Report`** that displays a list of current virtual machine configuration information. It should have five columns with the following data and titles:

| Column title | Data to display |
|---|---|
| Name | RHEV-M name for the virtual machine |
| Description | Long name that displays in the **Description** field in the **General** tab of the virtual machine |
| VM Type | Type of virtual machine, i.e., **`Server`** or **`Desktop`** |
| Creation Date | Date the virtual machine was created |
| Update Date | Date the virtual machine was last changed |

The title displayed at the top of the report should read "Custom RH318 Report".

1. Log Into Report Engine
2. Create Report
3. Add Columns
4. Adjust Column Headers
5. Adjust Report Title
6. Save Report Definition
7. Test New Report

# Solution

In this lab, you will create a custom RHEV Manager ad hoc report.

| Resources | |
|---|---|
| **Files** | NA |
| **Application URL** | https://rhevm.pod*X*.example.com |

Outcome(s)

You should be able to generate a custom report called **`Custom RH318 Report`** that displays a list of virtual machine attributes.

**Before you begin**

You should have existing virtual machines installed and managed by your RHEV-M server.

**Lab outline**

Create a custom report called **`Custom RH318 Report`** that displays a list of current virtual machine configuration information. It should have five columns with the following data and titles:

| Column title | Data to display |
|---|---|
| Name | RHEV-M name for the virtual machine |
| Description | Long name that displays in the **Description** field in the **General** tab of the virtual machine |
| VM Type | Type of virtual machine, i.e., **`Server`** or **`Desktop`** |
| Creation Date | Date the virtual machine was created |
| Update Date | Date the virtual machine was last changed |

The title displayed at the top of the report should read “Custom RH318 Report”.

1. Log Into Report Engine

   Launch the **Firefox** web browser on **`rhevm`**. Go to the following link: **`https://rhevm.podX.example.com/ovirt-engine-reports`**. Authenticate with a user ID of **`admin`** and a password of **`redhat`**.

2. Create Report

   2.1. Hover over the **Create** tab and select **Ad Hoc View** from the pulldown menu.

   2.2. Select the **Topics** tab, then expand the **Ad Hoc Components** folder, then the **Topics** folder, then finally the **3.1** (yes, *3.1*) folder. Select the **Virtual Machine Inventory (Ad Hoc Reports)** topic and click the **Table** button.

3. Add Columns

   Expand the **Virtual Machines** folder in the left frame. Drag the following items from the left frame into the report body (in the following order):

   - **VM Name**
   - **VM Description**
   - **VM Type**

- **VM Create Date**
- **VM Update Date**

4. Adjust Column Headers
   Change the headers for the columns. For example, right-click the **VM Name** column and select **Edit Label** from the menu. Type **`Name`** in the text box. Change the headers for the relevant fields to the following values:
   - "Name"
   - "Description"
   - "Type"
   - "Creation Date"
   - "Update Date"

5. Adjust Report Title
   Click the **Click To Add a Title** field and type **`Custom RH318 Report`**.

6. Save Report Definition
   Position the mouse over the floppy disk icon and select **Save Ad Hoc View And Create Report** from the dropdown menu. In the dialog box that appears, type **`Custom RH318 Report`** in the **Data View Name (required):** text box and type some descriptive text in the **Data View Description:** text box. In the tree view, expand **RHEVM Reports**, then **Reports** and then select **Inventory**. Click **Save** to create the custom report and save the ad hoc view.

7. Test New Report
   Hover over the **View** tab and navigate to the **Repository**. Find your new custom report in the tree at **RHEVM Reports**, **Reports**, **Inventory** or in the **Library** tab on the top panel.

# Summary

### Monitoring RHEV

In this section, you learned:

- Using the search bar helps to more easily find resources on large deployments.
- Tags may be created to filter a list of hosts or virtual machines.

### Generating Reports

In this section, you learned:

- Use prebuilt reports in the **`Report Engine`** to monitor the RHEV environment.
- The Reports Portal can be used to generate custom ad hoc reports on RHEV.

redhat TRAINING

## CHAPTER 11

# ADVANCED RHEV TOPICS

| Overview | |
|---|---|
| **Goal** | Describe backing up and restoring RHEV, making RHEV highly available, and using command line and application programming interfaces. |
| **Objectives** | • Back up and restore RHEV databases and configuration.<br>• Configure RHEV manager for high availability.<br>• Access RHEV by utilizing various application programming interfaces. |
| **Sections** | • Backing up and Restoring RHEV (and Practice)<br>• Creating a Highly Available RHEV Manager (and Quiz)<br>• Exploring Application Programming Interfaces (and Quiz) |

# Backing Up and Restoring RHEV

## Objectives

After completing this section, students should be able to:

- Back up and restore a RHEV-M environment.

## Backing up Red Hat Enterprise Virtualization Manager

While taking complete backups of the machine on which the Red Hat Enterprise Virtualization Manager is installed is recommended whenever changing the configuration of that machine, a utility is provided for backing up only the key files related to the engine. This utility, **engine-backup**, can be used to rapidly back up the engine database and configuration files into a single file that can be easily stored.

The **engine-backup** command works in one of two basic modes:

```
engine-backup --mode=backup
```

and

```
engine-backup --mode=restore
```

These two modes are further extended by a set of parameters that allow administrators to specify the scope of the backup and different credentials for the engine database. A full list of parameters and their function is as follows:

Consider the following **basic** options:

| Flag | Description |
|---|---|
| **--mode** | Specifies whether the command will perform a **backup** operation or a **restore** operation. Two options are available —**backup**, and **restore**. This is a required parameter. |
| **--file** | Specifies the path and name of a file into which backups are to be taken in **backup** mode, and the path and name of a file from which to read backup data in **restore** mode. This is a required parameter in both **backup** mode and **restore** mode. |
| **--log** | Specifies the path and name of a file into which logs of the **backup** or **restore** operation are to be written. This parameter is required in both **backup** mode and **restore** mode. |
| **--scope** | Specifies the scope of the **backup** or **restore** operation. There are two options—**all**, which backs up both the engine database and configuration data, and **db**, which backs up only the engine database. |

Consider the following *databases* options. The following options are only available when using the **engine-backup** command in **restore** mode:

| Flag | Description |
|---|---|
| **--change-db-credentials** | Allows administrators to specify alternate credentials for restoring the engine database using credentials other than those stored in the backup itself. Specifying this parameter allows administrators to add the following parameters. |
| **--db-host** | Specifies the IP address or fully qualified domain name of the host on which the database resides. This is a required parameter. |
| **--db-port** | Specifies the port by which a connection to the database will be made. |
| **--db-user** | Specifies the name of the user by which a connection to the database will be made. This is a required parameter. |
| **--db-passfile** | Specifies a file containing the password by which a connection to the database will be made. Either this parameter or the **--db-password** parameter must be specified. |
| **--db-password** | Specifies the plain-text password by which a connection to the database will be made. Either this parameter or the **--db-passfile** parameter must be specified. |
| **--db-name** | Specifies the name of the database to which the database will be restored. This is a required parameter. |
| **--db-secured** | Specifies that the connection with the database is to be secured. |
| **--db-secured-validation** | Specifies that the connection with the host is to be validated. |

The Red Hat Enterprise Virtualization Manager uses multiple PostgreSQL databases. The **engine** database stores all information about resources (virtual machines, datacenters, clusters, etc.), users, and the current state of the RHEV-M environment. The **ovirt_engine_history** and **rhevmreports** databases store historical information and can be used for queries and reporting.

Apart from the databases, there are also a number of configuration files that will need to be backed up. For a complete list of these files and their functions, refer to Appendix D in the Red Hat Virtualization Administration Guide. This list includes files in **/etc/**, **/usr/share/ovirt-engine*/**, **/var/lib/ovirt-engine/**, **/root/**, and others. A normal full system backup should include these files.

### Warning

In order to prevent issues during restore, the **ovirt-engine-dwhd** service needs to be stopped. If it is not stopped, running **engine-setup** will fail. Administrators can run the following **UPDATE** statement, against the **engine** database, after making sure the service is no longer running:

```
engine=> \c engine
psql (8.4.20)
You are now connected to database "engine".
engine=> UPDATE dwh_history_timekeeping SET var_value=0 WHERE var_name
 ='DwhCurrentlyRunning';
```

The credentials for the **engine** user are located in **/etc/ovirt-engine-dwh/ovirt-engine-dwhd.conf.d/10-setup-database.conf**.

## Demonstration: Backing up Red Hat Enterprise Virtualization Manager

In this example, observe the following steps as the instructor demonstrates how to use the **engine-backup** command in order to back up the RHEV database.

1. From desktop.example.com, open a shell, and SSH into the RHEV-M server:

   ```
   [root@desktop.example.com ~]# ssh root@rhevm.podX.example.com
   ```

2. Before backing up, make sure to stop the **ovirt-engine-dwhd** daemon:

   ```
   [root@rhevm ~]# service ovirt-engine-dwhd stop
   ```

3. Once logged in, run the following command to create a full backup:

   ```
   [root@rhevm ~]# engine-backup --scope=all --mode=backup --log=/root/backup.log --
   file=/root/rhev-backup.tar
   ```

4. A **tar** file containing a backup of the engine database, or the engine database and the configuration data for the Red Hat Enterprise Virtualization Manager, is created using the path and file name provided.

   In order to perform a backup of RHEV-M, two databases will need to be backed up: one for RHEV-M, and one for RHEV-M reports (if reports are being used). The databases are called **engine** and **ovirt_engine_history** for RHEV-M, and **rhevmreports** for RHEV-M reports. Alternatively, it is possible to run the following command to back up only the engine database:

   ```
   [root@rhevm ~]# engine-backup --scope=db --mode=backup --log=[file name] --
   file=[file name]
   ```

# Restoring the Red Hat Enterprise Virtualization Manager database

While the process for restoring a backup using the **engine-backup** command is straightforward, it involves several additional steps in comparison to that for creating a backup, depending on the destination to which the backup is to be restored. For example, the **engine-backup** command can be used to restore backups to fresh installations of Red Hat Enterprise Virtualization, on top of existing installations of Red Hat Enterprise Virtualization, and using local or remote databases.

> ### Warning
>
> Backups can only be restored to environments of the same major release as that of the backup. For example, a backup of a Red Hat Enterprise Virtualization version 3.3 environment can only be restored to another Red Hat Enterprise Virtualization version 3.3 environment. To view the version of Red Hat Enterprise Virtualization contained in a backup file, administrators can unpack the backup file and read the value in the version file located in the root directory of the unpacked files.

The **engine-backup** command can be used to restore a backup to a fresh installation of the Red Hat Enterprise Virtualization Manager. The following procedure must be performed on a machine on which the base operating system has been installed and the required packages for the Red Hat Enterprise Virtualization Manager have been installed, but the **engine-setup** command has not yet been run. This procedure assumes that the backup file can be accessed from the machine on which the backup is to be restored.

> ### Note
>
> The **engine-backup** command does not handle the actual creation of the engine database or the initial configuration of the **postgresql** service. Therefore, these tasks must be performed manually as outlined as follows when restoring a backup to a fresh installation.

Consider the following steps in order to restore a backup:

1. Log onto the machine on which the Red Hat Enterprise Virtualization Manager is installed.

2. Manually create an empty database to which the database in the backup can be restored and configure the **postgresql** service:

   - Run the following commands to initialize the **postgresql** database, start the **postgresql** service, and ensure this service starts on boot:

     ```
     [root@rhevm ~]# service postgresql initdb
     [root@rhevm ~]# service postgresql start
     [root@rhevm ~]# chkconfig postgresql on
     ```

   - Run the following commands to enter the **postgresql** command line:

     ```
     [root@rhevm ~]# service postgresql initdb
     ```

```
[root@rhevm ~]# su postgres
[postgres@rhevm ~]$ psql
```

- Run the following command to create a new user:

```
postgres=# create role [user name] with login encrypted password '[password]';
```

- Run the following command to create the new database:

```
postgres=# create database [database name] owner [user name] template template0
 encoding 'UTF8' lc_collate 'en_US.UTF-8' lc_ctype 'en_US.UTF-8';
```

- Edit the **/var/lib/pgsql/data/pg_hba.conf** file as follows:
  - For local databases, replace the existing directives, in the section starting with **Local** at the bottom of the file, with the following directives:

```
host    [database name]    [user name]    0.0.0.0/0  md5
host    [database name]    [user name]    ::0/0      md5
```

  - For remote databases, add the following line immediately underneath the line starting with **Local** at the bottom of the file, replacing **X.X.X.X** with the IP address of the engine:

```
host    [database name]    [user name]    X.X.X.X/32   md5
```

- Run the following command to restart the **postgresql** service:

```
[root@rhevm ~]# service postgresql restart
```

3. Restore the backup using the **engine-backup** command with the **--change-db-credentials** parameter to pass the credentials of the new database:

```
[root@rhevm ~]# engine-backup --mode=restore --file=[file name] --log=[file name] --
change-db-credentials --db-host=[database location] --db-name=[database name] --db-
user=[user name] --db-password=[password]
```

   If successful, the following output displays:

```
You should now run engine-setup.
Done.
```

4. Run the following command and follow the prompts to configure the engine:

```
engine-setup
```

5. The engine database and configuration files for the Red Hat Enterprise Virtualization Manager have been restored to the version in the backup.

# RHEV-M reinstallation

For a complete restore, some additional steps are necessary, such as removing files and packages. Four steps that follow can be skipped if this is a restore on a fresh installation. The following procedure is also in the Student Guide and the Red Hat Enterprise Virtualization Administration Guide.

The following steps are to be used to restore the RHEV-M configuration files and databases.

1. ```
   [root@rhevm ~]# service ovirt-engine stop
   ```

2. ```
   [root@rhevm ~]# yum remove rhevm rhevm-dwh rhevm-reports
   ```

3. ```
   [root@rhevm ~]# rm -rf /etc/pki/ovirt-engine
   ```

4. ```
   [root@rhevm ~]# rm -rf /etc/ovirt-engine
   ```

5. ```
   [root@rhevm ~]# yum -y install rhevm rhevm-dwh rhevm-reports
   ```

6. Run **engine-setup**, using the **--config-append** flag, in order to use the answer file, thus speeding up the configuration process:

   ```
   [root@rhevm ~]# engine-setup --config-append=/root/answers.txt
   ```

If there is an existing installation, it is also possible to overwrite it, speeding up the restoration process by keeping installed packages. Consider the following steps in order to restoring a backup to overwrite an existing installation:

1. Log onto the machine on which the Red Hat Enterprise Virtualization Manager is installed.

2. Run the following command, and follow the prompts to remove the configuration files for and clean the database associated with the manager:

   ```
   [root@rhevm ~]# engine-cleanup
   ```

3. Restore the backup using the **engine-backup** command:

   ```
   [root@rhevm ~]# engine-backup --mode=restore --file=[file name] --log=[file name]
   ```

   If successful, the following output displays:

   ```
   You should now run engine-setup.
   Done.
   ```

4. Run the following command and ensure the **ovirt-engine** service is correctly configured:

   ```
   engine-setup
   ```
   ```

Optionally, use the **--config-append** argument in order to speed up the configuration process:

```
[root@rhevm ~]# engine-setup --config-append=/root/answers.txt
```

## Backing up virtual machines

Currently Red Hat Enterprise Virtualization has no built-in mechanism for creating backups of running virtual machines. For virtual machines that are shut down, administrators can use the **Export** functionality. If there is a need to create backups of running virtual machines, it is recommended to use the same methods and tools as the ones for a bare-metal machine.

### References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 15.1. Backing Up and Restoring the Red Hat Enterprise Virtualization Manager

Red Hat Customer Portal

> Backing up and restoring RHEV with engine-backup
> https://access.redhat.com/solutions/797463

# Practice: Backing Up and Restoring RHEV

In this lab, you will back up the RHEV installation and restore the backup.

**Outcome(s)**

- You should be able to re-establish a running RHEV installation using a previously created backup.

**Before you begin**

- RHEV Manager installed on rhevm.pod*X*.example.com
- Administration Portal connects to rhevm.pod*X*.example.com
- An installed RHEV-H node

Perform the following steps:

1. From workstation, log into your rhevm server:

```
[root@workstation ~]# ssh root@rhevm
```

2. Before backing up, make sure to stop the **ovirt-engine-dwhd** daemon:

```
[root@rhevm ~]# service ovirt-engine-dwhd stop
```

3. Using the command **engine-backup**, create a backup of the existing installation:

```
[root@rhevm ~]# engine-backup --scope=all --mode=backup --log=/root/rhevm-backup.log
 --file=/root/rhevm-backup.tar
```

4. Run the **engine-cleanup** command in order to remove the existing RHEV setup:

```
[root@rhevm ~]# engine-cleanup
[ INFO  ] Stage: Initializing
[ INFO  ] Stage: Environment setup
          Configuration files: ['/etc/ovirt-engine-setup.conf.d/10-packaging-
dwh.conf', '/etc/ovirt-engine-setup.conf.d/10-packaging-wsp.conf', '/etc/ovirt-
engine-setup.conf.d/10-packaging.conf', '/etc/ovirt-engine-setup.conf.d/20-
packaging-rhevm-reports.conf', '/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-
post.conf']
          Log file: /var/log/ovirt-engine/setup/ovirt-engine-remove-20150114091512-
hwz34u.log
          Version: otopi-1.3.0 (otopi-1.3.0-2.el6ev)
[ INFO  ] Stage: Environment packages setup
[ INFO  ] Stage: Programs detection
[ INFO  ] Stage: Environment customization
          Do you want to remove all components? (Yes, No) [Yes]: yes

          --== PRODUCT OPTIONS ==--

          Do you want to remove Engine database content? All data will be lost (Yes,
 No) [No]: yes
[ INFO  ] Stage: Setup validation
          During execution engine service will be stopped (OK, Cancel) [OK]: Enter
```

```
          All the installed ovirt components are about to be removed, data will be
 lost (OK, Cancel) [Cancel]: ok
[ INFO  ] Stage: Transaction setup
[ INFO  ] Stopping dwh service
[ INFO  ] Stopping reports service
[ INFO  ] Stopping engine service
[ INFO  ] Stopping ovirt-fence-kdump-listener service
[ INFO  ] Stopping websocket-proxy service
[ INFO  ] Stage: Misc configuration
[ INFO  ] Stage: Package installation
[ INFO  ] Stage: Misc configuration
[ INFO  ] Backing up PKI configuration and keys
[ INFO  ] Backing up database localhost:engine to '/var/lib/ovirt-engine/backups/
engine-20150114091529.zFdp73.dump'.
[ INFO  ] Clearing Engine database engine
[ INFO  ] Backing up database localhost:ovirt_engine_history to '/var/lib/ovirt-
engine-dwh/backups/dwh-20150114091544.tNHRUX.dump'.
[ INFO  ] Clearing DWH database ovirt_engine_history
[ INFO  ] Backing up database localhost:ovirt_engine_reports to '/var/lib/ovirt-
engine-reports/backups/reports-20150114091549.Av0Wcr.dump'.
[ INFO  ] Clearing Reports database ovirt_engine_reports
[ INFO  ] Removing files
[ INFO  ] Reverting changes to files
[ INFO  ] Stage: Transaction commit
[ INFO  ] Stage: Closing up

          --== SUMMARY ==--

          A backup of the Reports database is available at /var/lib/ovirt-engine-
reports/backups/reports-20150114091549.Av0Wcr.dump
          A backup of the DWH database is available at /var/lib/ovirt-engine-dwh/
backups/dwh-20150114091544.tNHRUX.dump
          A backup of the Engine database is available at /var/lib/ovirt-engine/
backups/engine-20150114091529.zFdp73.dump
          ovirt-engine has been removed
          A backup of PKI configuration and keys is available at /var/lib/ovirt-
engine/backups/engine-pki-201501140915292PiwI_.tar.gz
          Engine setup successfully cleaned up

          --== END OF SUMMARY ==--

[ INFO  ] Stage: Clean up
          Log file is located at /var/log/ovirt-engine/setup/ovirt-engine-
remove-20150114091512-hwz34u.log
[ INFO  ] Generating answer file '/var/lib/ovirt-engine/setup/
answers/20150114091556-cleanup.conf'
[ INFO  ] Stage: Pre-termination
[ INFO  ] Stage: Termination
[ INFO  ] Execution of cleanup completed successfully
```

5. Restore the backup, using the **engine-backup** command:

```
[root@rhevm ~]# engine-backup --mode=restore --file=/root/rhevm-backup.tar --log=/
root/rhevm-restore.log
Preparing to restore:
- Unpacking file '/root/rhevm-backup.tar'
Restoring:
- Files
- Engine database 'engine'
- DWH database 'ovirt_engine_history'
- Reports database 'ovirt_engine_reports'
You should now run engine-setup.
```

```
Done.
```

6. As indicated by the previous output, run the **engine-setup**. Use the answer file you generated in order to speed up the restore. When prompted, do not back up the existing database, as it is not necessary:

```
[root@rhevm ~]# engine-setup --config-append=/root/answers.txt
[ INFO  ] Stage: Initializing
[ INFO  ] Stage: Environment setup
          Configuration files: ['/etc/ovirt-engine-setup.conf.d/10-packaging-
dwh.conf', '/etc/ovirt-engine-setup.conf.d/10-packaging-wsp.conf', '/etc/ovirt-
engine-setup.conf.d/10-packaging.conf', '/etc/ovirt-engine-setup.conf.d/20-
packaging-rhevm-reports.conf', '/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-
post.conf', '/root/answers.txt']
          Log file: /var/log/ovirt-engine/setup/ovirt-engine-setup-20150114094424-
j5fq5q.log
          Version: otopi-1.3.0 (otopi-1.3.0-2.el6ev)
[ INFO  ] Stage: Environment packages setup
[ INFO  ] Stage: Programs detection
[ INFO  ] Stage: Environment setup
[ INFO  ] Stage: Environment customization

          Welcome to the RHEV 3.5 setup/upgrade.
          Please read the RHEV 3.5 install guide
          https://access.redhat.com/site/documentation/en-US/
Red_Hat_Enterprise_Virtualization/3.5/html/Installation_Guide/index.html.
          Please refer to the RHEV Upgrade Helper application
          https://access.redhat.com/labs/rhevupgradehelper/ which will guide you in
 the upgrading process.
          Would you like to proceed? (Yes, No) [Yes]: yes

          --== PRODUCT OPTIONS ==--


          --== PACKAGES ==--

[ INFO  ] Checking for product updates...
[ INFO  ] No product updates found

          --== ALL IN ONE CONFIGURATION ==--


          --== NETWORK CONFIGURATION ==--

[ INFO  ] iptables will be configured as firewall manager.

          --== DATABASE CONFIGURATION ==--

          The detected DWH database size is 24 MB.
          Setup can backup the existing database. The time and space required for
 the database backup depend on its size. This process takes time, and in some cases
 (for instance, when the size is few GBs) may take several hours to complete.
          If you choose to not back up the database, and Setup later fails for some
 reason, it will not be able to restore the database and all DWH data will be lost.
          Would you like to backup the existing database before upgrading it? (Yes,
 No) [Yes]: no
[WARNING] Are you sure you do not want to backup the DWH database?
          A positive reply makes sense only if you do not need the data in DWH, or
 have some other, external means to restore it to a working state.
          Are you sure you do not want to backup the DWH database?(Yes, No)
 [No]: yes
```

```
[WARNING] DWH Database will not be backed up. Rollback in case of failure will not
 be possible.
... Output omitted...
[ INFO ] Execution of setup completed successfully
```

7. Finally, run the following commands to join back the **example.com** domain:

   ```
   [root@rhevm ~]# engine-manage-domains add --domain=example.com --user=rhevadmin --
   provider=IPA
   oVirt Engine restart is required in order for the changes to take place (service
    ovirt-engine restart).
   Manage Domains completed successfully
   ```

   Use **redhat** when prompted for a password. After this command has finished, restart RHEV-M to activate the changes:

   ```
   [root@rhevm ~]# service ovirt-engine restart
   ```

8. From workstation, open Firefox, and access rhevm.pod*X*.example.com. Since the SSL certificate has been regenerated, confirm the security exception.

9. Using the domain **internal**, log in with the user **admin**, and a password of **redhat**.

10. Once logged in, you should see the existing cluster you created earlier.

# Creating a Highly Available RHEV Manager

## Objectives

After completing this section, students should be able to:

- Discuss the steps required to create a highly available RHEV-M installation.

## Setting up a highly available Red Hat Enterprise Virtualization Manager

To make RHEV-M highly available, administrators can configure it to run as a service in an HA cluster. Red Hat Cluster Suite (**RHCS**) high-availability clusters eliminate single points of failure, so if the node on which a service (which in this case includes resources needed by RHEV-M) is running should become inoperative, the service can start up again (using failover) to another cluster node with minimal interruption, and no data loss. Red Hat supports two options for making RHEV-M 3.5 highly available:

1. **RHEV-M as a highly available virtual machine**: This approach lets administrators configure a single RHEV-M as a virtual machine that is brought up on another host if the RHEV-M goes down. It offers simpler configuration and approach, but can result in a longer downtime of a few minutes when a VM goes down. Since this approach is the one supported by Red Hat, this unit will be focusing on this approach.

2. **RHEV-M as a highly available service**: This approach allows administrators to configure Red Hat Enterprise Virtualization Manager (RHEV-M) in a two-node, **RHCS** highly available (**HA**) cluster.

### Note

Although not strictly required, it is generally better to run at least a three-node cluster for the aforementioned setup. Besides offering extra resources, the additional node makes it less likely to end up in a *split-brain* condition, where both nodes believe they control the cluster.

There are many different ways of setting up a high-availability RHEV-M cluster. In the previous example, the following components would be used:

- **Two cluster nodes.** Install two machines with Red Hat Enterprise Linux 6 to act as cluster nodes.

- **A cluster web user interface**: A Red Hat Enterprise Linux system (not on either of the cluster nodes) running the **`luci`** web-based high-availability administration application. Administrators may want this running on a system outside the cluster, so if either node goes down, the management interface is not affected.

- **Network storage**: Shared network storage is required. This section shows how to use HA LVM from a RHEL 6 system, which is backed by iSCSI storage. (Fibre Channel and NFS are other technologies administrators could use instead of iSCSI.)

- **Red Hat products**: This section combines components from Red Hat Enterprise Linux, Red Hat Cluster Suite, Red Hat Enterprise Virtualization, and (optionally) Red Hat Enterprise Linux Server Resilient Storage.

## RHEV-M self-hosted engine

A self-hosted engine is a virtualized environment in which the engine, or **`Manager`**, runs on a virtual machine on the hosts managed by that engine. The virtual machine is created as part of the host configuration, and the engine is installed and configured in parallel to that host configuration process, referred to in these procedures as the deployment. The virtual machine running the engine is created to be highly available; if the host running the virtual machine goes into maintenance mode, or fails unexpectedly, the virtual machine will be migrated automatically to another host in the environment. The primary benefit of the self-hosted engine is that it requires less hardware to deploy an instance of Red Hat Enterprise Virtualization as the engine runs as a virtual machine, not on physical hardware. Additionally, the engine is configured to be highly available automatically, rather than requiring a separate cluster. The self-hosted engine currently only runs on Red Hat Enterprise Linux 6.5+ hosts. Red Hat Enterprise Virtualization Hypervisors and older versions of Red Hat Enterprise Linux are not recommended for use with a self-hosted engine.

> **Warning**
>
> The self-hosted engine currently only runs on current RHEL hosts. RHEV-H and older versions of RHEL are not recommended for use with a self-hosted engine.

At present, there are two main limitations for using the self-hosted engine configuration:

- An NFS storage domain is required for the configuration. Currently NFS is the only supported file system for the self-hosted engine.
- The host and hosted engine must use RHEL 6.6. RHEV-H and older versions of RHEL are not supported.

In order to deploy the self-hosted engine, consider the following requirements:

1. Subscribing to the appropriate Red Hat Network channels to install the packages. For Subscription Manager, these channels are:
   - **`rhel-6-server-rpms`**
   - **`rhel-6-server-supplementary-rpms`**
   - **`rhel-6-server-rhevm-3.5-rpms`**
   - **`jb-eap-6-for-rhel-6-server-rpms`**
   - **`rhel-6-server-rhev-mgmt-agent-rpms`**

> **Warning**
>
> While the **ovirt-hosted-engine-setup** package is provided by the Red Hat Enterprise Virtualization Manager channel, and can be installed using the standard channels for the **Manager**, the **vdsm** package is a dependency of the **ovirt-hosted-engine-setup** package, and is provided by the Red Hat Enterprise Virt Management Agent channel, which must be enabled. This channel is **rhel-6-server-rhev-mgmt-agent-rpms**, in Subscription Manager.

2. Upgrading the packages on the system.
3. Using **yum**, in order to install the **yum install ovirt-hosted-engine-setup** package.

# Configuring the self-hosted engine

When package installation is complete, the Red Hat Enterprise Virtualization Manager must be configured. The **hosted-engine** deployment script is provided to assist with this task. The script asks the administrators a series of questions, and configures the environment based on the answers provided. When the required values have been provided, the updated configuration is applied, and the Red Hat Enterprise Virtualization Manager services are started.

The **hosted-engine** deployment script guides the administrator through several distinct configuration stages. The script suggests possible configuration defaults in *square brackets*. Where these default values are acceptable, no additional input is required. This procedure requires a new Red Hat Enterprise Linux 6.5+ host with the **ovirt-hosted-engine-setup** package installed. This host is referred to as **rhev-he1**, with a fully qualified domain name (FQDN) of **rhev-he1.example.com** in this procedure. The hosted engine, the virtual machine created during configuration of **rhev-he1** to manage the environment, is referred to as **HostedEngine-VM**. Administrators will be prompted by the **hosted-engine** deployment script to access this virtual machine multiple times, in order to install an operating system and to configure the engine. All steps in this procedure are to be conducted as the **root** user for the specified machine.

Consider the following steps in order to configure the self-hosted engine:

1. **Initiating hosted engine deployment**: Begin configuration of the self-hosted environment by deploying the **hosted-engine** customization script on **rhev-he1**. To escape the script at any time, administrators can use the **CTRL+D** keyboard combination to abort deployment.

```
[root@rhev-he1 ~]# # hosted-engine --deploy
```

2. **Configuring storage**: Select the version of NFS and specify the full address, using either the **FQDN** or IP address, and path name of the shared storage domain. Choose the storage domain and storage datacenter names to be used in the environment.

```
During customization use CTRL-D to abort.
Please specify the storage you would like to use (nfs3, nfs4)[nfs3]:
Please specify the full shared storage connection path to use (example: host:/path):
 storage.example.com:/hosted_engine/nfs
[ INFO  ] Installing on first host
Please provide storage domain name. [hosted_storage]:
```

```
Local storage datacenter name is an internal name and currently will not be shown in
 engine's admin UI.Please enter local datacenter name [hosted_datacenter]:
```

3. **Configuring the network**: The script detects possible network interface controllers (**NICs**) to use as a management bridge for the environment. It then checks the firewall configuration and offers to modify it for console (SPICE or VNC) access to **HostedEngine-VM**. Provide a reachable gateway IP address to be used by the **ovirt-ha-agent** to help determine a host's suitability for running **HostedEngine-VM**.

```
Please indicate a nic to set rhevm bridge on: (eth1, eth0) [eth1]:
iptables was detected on your computer, do you wish setup to configure it? (Yes, No)
[Yes]:
Please indicate a pingable gateway IP address [X.X.X.X]:
```

4. **Configuring the virtual machine**: The script creates a virtual machine to be configured as the Red Hat Enterprise Virtualization Manager, the hosted engine referred to in this procedure as **HostedEngine-VM**. Specify the boot device and, if applicable, the path name of the installation media, the CPU type, the number of virtual CPUs, and the disk size. Specify a MAC address for the **HostedEngine-VM**, or accept a randomly generated one. The MAC address can be used to update the DHCP server prior to installing the operating system on the virtual machine. Specify memory size and console connection type for the creation of **HostedEngine-VM**.

```
Please specify the device to boot the VM from (cdrom, disk, pxe) [cdrom]:
The following CPU types are supported by this host:
          - model_Penryn: Intel Penryn Family
          - model_Conroe: Intel Conroe Family
Please specify the CPU type to be used by the VM [model_Penryn]:
Please specify the number of virtual CPUs for the VM [Defaults to minimum
 requirement: 2]:
Please specify the disk size of the VM in GB [Defaults to minimum requirement: 25]:
You may specify a MAC address for the VM or accept a randomly generated default
 [00:16:3e:77:b2:a4]:
Please specify the memory size of the VM in MB [Defaults to minimum requirement:
 4096]:
Please specify the console type you would like to use to connect to the VM (vnc,
 spice) [vnc]:
```

5. **Configuring the hosted engine**: Specify the name for **rhev-he11** to be identified in the Red Hat Enterprise Virtualization environment, and the password for the admin@internal user to access the Administration Portal. Provide the **FQDN** for **HostedEngine-VM**; this procedure uses the **FQDN** **HostedEngine-VM.example.com**. Finally, provide the name and TCP port number of the SMTP server, the email address used to send email notifications, and a comma-separated list of email addresses to receive these notifications.

6. **Reviewing configuration**: Before proceeding, the **hosted-engine** script displays the configuration values that were entered, and prompts for confirmation to proceed with these values.

```
Bridge interface                   : eth1
Engine FQDN                        : HostedEngine-VM.example.com
Bridge name                        : rhevm
SSH daemon port                    : 22
Firewall manager                   : iptables
Gateway address                    : X.X.X.X
```

```
Host name for web application      : rhev-he11
Host ID                            : 1
Image size GB                      : 25
Storage connection                 : storage.example.com:/hosted_engine/nfs
Console type                       : vnc
Memory size MB                     : 4096
MAC address                        : 00:16:3e:77:b2:a4
Boot type                          : pxe
Number of CPUs                     : 2
CPU Type                           : model_Penryn

Please confirm installation settings (Yes, No)[No]:
```

7. **Creating `HostedEngine-VM`**: The script creates a virtual machine to be **`HostedEngine-VM`**, and provides connection details. Administrators must install an operating system on **`HostedEngine-VM`** before the **`hosted-engine`** script can proceed on **`rhev-he1`**.

```
[ INFO  ] Stage: Closing up
[ INFO  ] Creating VM
You can now connect to the VM with the following command:
    /usr/bin/remote-viewer vnc://localhost:5900
Use temporary password "3042QHpX" to connect to vnc console.
Please note that in order to use remote-viewer you need to be able to run graphical
 applications.
This means that if you are using ssh you have to supply the -Y flag (enables trusted
 X11 forwarding).
Otherwise you can run the command from a terminal in your preferred desktop
 environment.
If you cannot run graphical applications you can connect to the graphic console from
 another host or connect to the console using the following command:
virsh -c qemu+tls://Test/system console HostedEngine
If you need to reboot the VM you will need to start it manually using the command:
hosted-engine --vm-start
You can then set a temporary password using the command:
hosted-engine --add-console-password
The VM has been started.  Install the OS and shut down or reboot it.  To continue
 please make a selection:

          (1) Continue setup - VM installation is complete
          (2) Reboot the VM and restart installation
          (3) Abort setup

          (1, 2, 3)[1]:
```

Using the naming convention of this procedure, administrators can connect to the virtual machine using VNC with the following command:

```
/usr/bin/remote-viewer vnc://rhev-he1.example.com:5900
```

8. **Installing the virtual machine operating system**: Connect to **`HostedEngine-VM`**, the virtual machine created by the **`hosted-engine`** script, and install a Red Hat Enterprise Linux 6.6 operating system. Ensure the machine is rebooted once installation has completed.

9. **Synchronizing the host and the virtual machine**: Return to **`rhev-he1`**, and continue the **`hosted-engine`** deployment script by selecting option **1**:

```
(1) Continue setup - VM installation is complete
```

```
 Waiting for VM to shut down...
[ INFO  ] Creating VM
You can now connect to the VM with the following command:
    /usr/bin/remote-viewer vnc://localhost:5900
Use temporary password "3042QHpX" to connect to vnc console.
Please note that in order to use remote-viewer you need to be able to run graphical
 applications.
This means that if you are using ssh you have to supply the -Y flag (enables trusted
 X11 forwarding).
Otherwise you can run the command from a terminal in your preferred desktop
 environment.
If you cannot run graphical applications you can connect to the graphic console from
 another host or connect to the console using the following command:
virsh -c qemu+tls://Test/system console HostedEngine
If you need to reboot the VM you will need to start it manually using the command:
hosted-engine --vm-start
You can then set a temporary password using the command:
hosted-engine --add-console-password
Please install and setup the engine in the VM.
You may also be interested in subscribing to "agent" RHN/Satellite channel and
 installing rhevm-guest-agent-common package in the VM.
To continue make a selection from the options below:
          (1) Continue setup - engine installation is complete
          (2) Power off and restart the VM
          (3) Abort setup
```

10. **Installing the manager**: Connect to **HostedEngine-VM**, subscribe to the appropriate Red Hat Enterprise Virtualization Manager channels, ensure that the most up-to-date versions of all installed packages are in use, and install the **rhevm packages**; after that, it will be possible to run the **engine-setup** command in order to configure the manager.

11. **Synchronizing the host and the manager**: Return to **rhev-he1**, and continue the **hosted-engine** deployment script by selecting option **1**:

```
(1) Continue setup - engine installation is complete
```

```
[ INFO  ] Engine replied: DB Up!Welcome to Health Status!
[ INFO  ] Waiting for the host to become operational in the engine. This may take
 several minutes...
[ INFO  ] Still waiting for VDSM host to become operational...
[ INFO  ] The VDSM Host is now operational
          Please shutdown the VM allowing the system to launch it as a monitored
 service.
          The system will wait until the VM is down.
```

12. **Shutting down HostedEngine-VM**: The virtual machine can be shut down, as the host **rhev-he1** will detect that the virtual machine is no longer running:

```
[ INFO  ] Enabling and starting HA services
          Hosted Engine successfully set up
[ INFO  ] Stage: Clean up
[ INFO  ] Stage: Pre-termination
[ INFO  ] Stage: Termination
```

When the **hosted-engine** deployment script completes successfully, the Red Hat Enterprise Virtualization Manager is configured and running on the server. In contrast to a bare-metal manager installation, the hosted-engine manager has already configured the datacenter, cluster, host (**rhev-he1**), storage domain, and virtual machine of the hosted engine (**HostedEngine-VM**). It is possible to log in as the admin@internal user to continue configuring the **Manager** and add further resources. Link your Red Hat Enterprise Virtualization Manager to a directory server, so it is possible to add additional users to the environment. Red Hat Enterprise Virtualization supports directory services from Red Hat Directory Services (**RHDS**), **IdM**, and Active Directory. It is possible to add a directory server to the environment, using the **engine-manage-domains** command.

The **ovirt-host-engine-setup** script also saves the answers that were given during configuration to a file to help with disaster recovery. If a destination is not specified using the **--generate-answer=<file>** argument, the answer file is generated at **/etc/ovirt-hosted-engine/answers.conf**.

As administrators would do with the traditional deployment, the self-hosted engine still allows:

- Installing additional hosts to a self-hosted environment.
- Upgrading the self-hosted engine.
- Upgrading additional hosts in a self-hosted environment.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 3.3. Installing the Self-Hosted Engine
- Section 3.6. Installing Additional Hosts to a Self-Hosted Environment

> Setting up a RHEV-M on a Highly Available Cluster (RHEV 3.1 and 3.0)
> https://access.redhat.com/articles/216973

> RHEV-M as a highly available virtual machine
> https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Cluster_Administration/s1-virt_machine_resources-ccs-CA.html

> Red Hat Enterprise Linux 6 Cluster Administration
> https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html-single/Cluster_Administration/

# Quiz: Creating a Highly Available RHEV Manager

Choose the correct answer to the following questions:

1. Name two of the options you can use in order to make RHEV-M highly available. (Choose two.)

   a. RHEV-M as a highly available virtual machine.
   b. RHEV-M as a highly available resource.
   c. RHEV-M as a highly available service.
   d. RHEV-M cluster suite.

2. Even though a two-nodes configuration is possible, name the main risk of running such a setup. (Choose one.)

   a. Having a broadcast storm, where the cluster overwhelms the network with continuous multicast or broadcast traffic.
   b. Having a **split-brain** condition, where each nodes considers itself to be the master.
   c. Preventing the **HA-VLM** failover from working.
   d. Having an unsupported configuration in the high-availability manager.

3. Name two benefits of using the self-hosted engine. (Choose two.)

   a. The self-hosted engine requires less hardware, as the engine runs as a virtual machine, not on physical hardware.
   b. The self-hosted engine uses hooks to achieve high availability, minimizing downtime.
   c. The self-hosted engine is able to re-create virtual machines on demand, minimizing downtime.
   d. The engine is configured to be highly available automatically, rather than requiring a separate cluster.

4. What is a mandatory step administrators need to follow before the **hosted-engine** script can proceed on the hypervisor? (Choose one.)

   a. Administrators need to run the **engine-setup** command.
   b. Administrators need to install an operating system on a virtual machine.
   c. Administrators need to install the **ovirt-ha-agent** package to help determine a host's suitability for running **HostedEngine-VM**.

5. What is the command to access the virtual machine console? (Choose one.)

   a. **/usr/bin/remote-viewer ssh://*hypervisor*.example.com:5900**
   b. **/usr/bin/rhev-viewer vnc://*hypervisor*.example.com:5900**
   c. **/usr/bin/remote-viewer vnc://*hypervisor*.example.com:5900**

# Solution

Choose the correct answer to the following questions:

1. Name two of the options you can use in order to make RHEV-M highly available. (Choose two.)

   a. **RHEV-M as a highly available virtual machine.**
   b. RHEV-M as a highly available resource.
   c. **RHEV-M as a highly available service.**
   d. RHEV-M cluster suite.

2. Even though a two-nodes configuration is possible, name the main risk of running such a setup. (Choose one.)

   a. Having a broadcast storm, where the cluster overwhelms the network with continuous multicast or broadcast traffic.
   b. **Having a `split-brain` condition, where each nodes considers itself to be the master.**
   c. Preventing the `HA-VLM` failover from working.
   d. Having an unsupported configuration in the high-availability manager.

3. Name two benefits of using the self-hosted engine. (Choose two.)

   a. **The self-hosted engine requires less hardware, as the engine runs as a virtual machine, not on physical hardware.**
   b. The self-hosted engine uses hooks to achieve high availability, minimizing downtime.
   c. The self-hosted engine is able to re-create virtual machines on demand, minimizing downtime.
   d. **The engine is configured to be highly available automatically, rather than requiring a separate cluster.**

4. What is a mandatory step administrators need to follow before the `hosted-engine` script can proceed on the hypervisor? (Choose one.)

   a. Administrators need to run the `engine-setup` command.
   b. **Administrators need to install an operating system on a virtual machine.**
   c. Administrators need to install the `ovirt-ha-agent` package to help determine a host's suitability for running `HostedEngine-VM`.

5. What is the command to access the virtual machine console? (Choose one.)

   a. `/usr/bin/remote-viewer ssh://hypervisor.example.com:5900`
   b. `/usr/bin/rhev-viewer vnc://hypervisor.example.com:5900`
   c. **`/usr/bin/remote-viewer vnc://hypervisor.example.com:5900`**

# Exploring Application Programming Interfaces

## Objectives

After completing this section, students should be able to:

- Communicate with RHEV-M utilizing the REST API.
- Modify a host configuration utilizing VDSM hooks.

## About the RHEV-M REST API

Red Hat Enterprise Virtualization Manager provides users with a *Representational State Transfer* (*REST*) application programming interface (API). It allows developers and system administrators to:

- Integrate RHEV-M with third-party virtualization software.
- Perform automated maintenance or error-checking tasks.
- Automate repetitive RHEV management tasks with scripts.

The REST API is completely stateless. Every operation is independent from any other operation, meaning the server does not have to keep state for the different clients. The RHEV-M REST API is based around the HTTP(S) protocol. This means that writing a client can be done in any programming language that has support for the HTTPS protocol. Information is exchanged using XML documents.

One of the other characteristics of the RHEV-M REST API is that it is built around the idea of resources. There are no function calls, just methods that can be called on a resource. This concept should feel familiar to those who have used an object-oriented programming language. All actions are performed using one of the four standard HTTP verbs: GET, POST, PUT, and DELETE.

## Using the RHEV-M REST API with `curl`

The easiest way to connect to the RHEV-M REST API is from the command line with a tool like **`wget`** or **`curl`**. Although this provides us with an easy way to connect and test the API, handling responses that come back from the server will be more difficult, as all answers are given as XML documents.

The API exposes several features, such as:

- Infrastructure configuration, such as host configuration and management, as well as network and storage configuration.
- VM configuration and management.
- Guest networking, virtual disks, and VM properties.
- User and accounting management.
- Daily maintenance and VM life-cycle management.
- Advanced operations not available using the GUI.

The API could be used in order to provide solutions for the following use cases:

- Scripting and utilities.
- Integrating with other software used in the organization.
- Automate administration tasks.
- Software development of specific add-ons or applications.
- Performance monitoring.
- Custom reporting.
- Provisioning life-cycle control.

Like any RESTful API, the RHEV-M API is based on the HTTP(S) protocol. As such, all actions are performed using the four standard HTTP verbs: GET, POST, PUT, and DELETE. All information interchange is done in XML.

For important deployments, administrators would probably consider a more advanced programming language such as Perl, Python, or Java. Any language that supports HTTP and XML can be used.

# Demonstration: Communicating with the **REST** API

In this example, observe the following steps as the instructor demonstrates how to communicate with the RHEV-M **REST** API:

1. Open **Firefox** on workstation.podX.example.com and browse to **`https://rhevm.podX.example.com/api`**. Enter a username of **`rhevadmin@example.com`** and a password of **`redhat`**.

   This will return the *entry point* of the API; all other elements can be found beneath this URL.

   Using an administrative account is no longer needed in the API. All RHEV users can view/change items in the API for which they have been given permission.

2. Switch to the command line to demonstrate the RHEV API with **`curl`**. First, get the CA certificate to allow secure connections.

```
[student@rhevm ~]$ wget http://rhevm.podX.example.com/ca.crt
```

3. Look at a basic **`curl`** command and the response from the server.

```
[student@rhevm ~]$ curl --cacert ca.crt❶ -u 'rhevadmin@example.com:redhat'❷\
https://rhevm.podX.example.com/api❸
```

❶ The **`--cacert ca.crt`** option tells **`curl`** to use the **`ca.crt`** certificate file as a certificate authority. This allows for proper SSL encryption.

❷ Since the RHEV-M REST API is stateless, authentication information must be sent with every single request. RHEV-M handles authentication for the REST API using HTTP Basic authentication. Any request that is not accompanied by correct credentials will result in an HTTP status code of **`HTTP/1.1 401 Authorization Required`**.

Credentials are specified with ***username@domain*:*password*** or other alternates shown later.

3. This is the HTTP resource being requested: **https://rhevm.pod*X*.example.com** is the server to talk to and **/api** is the *entry point* into the API. All other objects live beneath this point.

The response from the server will look something like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<api>
    <link rel="capabilities" href="/api/capabilities"/> 1
    <link rel="clusters" href="/api/clusters"/>
    <link rel="clusters/search" href="/api/clusters?search={query}"/>
    <link rel="datacenters" href="/api/datacenters"/>
    <link rel="datacenters/search" href="/api/datacenters?search={query}"/>
    <link rel="events" href="/api/events"/>
    <link rel="events/search" href="/api/events?search={query}&from={event_id}"/>
    <link rel="hosts" href="/api/hosts"/>
    <link rel="hosts/search" href="/api/hosts?search={query}"/>
    <link rel="networks" href="/api/networks"/>
    <link rel="roles" href="/api/roles"/>
    <link rel="storagedomains" href="/api/storagedomains"/>
    <link rel="storagedomains/search" href="/api/storagedomains?search={query}"/>
    <link rel="tags" href="/api/tags"/>
    <link rel="templates" href="/api/templates"/>
    <link rel="templates/search" href="/api/templates?search={query}"/>
    <link rel="users" href="/api/users"/>
    <link rel="users/search" href="/api/users?search={query}"/>
    <link rel="groups" href="/api/groups"/>
    <link rel="groups/search" href="/api/groups?search={query}"/>
    <link rel="domains" href="/api/domains"/>
    <link rel="vmpools" href="/api/vmpools"/>
    <link rel="vmpools/search" href="/api/vmpools?search={query}"/>
    <link rel="vms" href="/api/vms"/>
    <link rel="vms/search" href="/api/vms?search={query}"/>
    <special_objects> 2
        <link rel="templates/blank" href="/api/
templates/00000000-0000-0000-0000-000000000000"/>
        <link rel="tags/root" href="/api/tags/00000000-0000-0000-0000-000000000000"/
>
    </special_objects>
    <product_info> 3
        <name>Red Hat Enterprise Virtualization</name>
        <vendor>Red Hat</vendor>
        <version revision="0" build="0" minor="0" major="3"/>
    </product_info>
    <summary> 4
        <vms>
            <total>4</total>
            <active>2</active>
        </vms>
        <hosts>
            <total>1</total>
            <active>1</active>
        </hosts>
        <users>
            <total>6</total>
            <active>2</active>
        </users>
        <storage_domains>
            <total>3</total>
```

```
            <active>3</active>
        </storage_domains>
    </summary>
</api>
```

1. This list contains entry points for all the resource collections in the RHEV environment. Every **<link>** tag has a **rel** attribute pointing to the type of resource, and an **href** attribute detailing the URL (minus the host name) that should be requested to query that collection. If a **rel** attribute includes **/search** at the end, administrators can use the **href** attribute to perform a search using the same syntax as they would in the search bar in the RHEV-M web interface. Just replace **{query}** in the **href** with a URL-encoded version of the search string.
2. This section contains *Special Objects*, typically the **Blank** template and the **Root** tag. Both of these can also be found in the **/api/templates** and the **/api/tags** collections, respectively.
3. This section contains detailed version information.
4. This list contains a brief overview of the number of active and total virtual machines, hosts, users, and storage domains.

4. Using a password on the command line is a security risk, so try running the command without the password:

```
[student@rhevm ~]$ curl --cacert ca.crt -u 'rhevadmin@example.com'\
https://rhevm.podX.example.com/api
Enter host password for user 'rhevadmin@example.com': redhat
```

5. Requiring user intervention is not very efficient for scripting, so create an authentication file and tell **curl** to use it for authentication:

```
[student@rhevm ~]$ echo 'machine rhevm.podX.example.com login rhevadmin@example.com
 password redhat' > ~/.netrc
[student@rhevm ~]$ chmod 600 ~/.netrc
[student@rhevm ~]$ curl -n --cacert ca.crt \
https://rhevm.podX.example.com/api
```

It is possible to add the **-n** option to the previous **curl** command; this format will be used for the rest of the demonstration.

6. Now look at the representation of a virtual machine. Use the following command to query for all virtual machines with a name matching ***rhel***:

```
[student@rhevm ~]$ curl -n --cacert ca.crt \
https://rhevm.podX.example.com/api/vms?search=*rhel*
```

The only thing that differs from the previous incarnation is the last part of the URL. It was not necessary to URL-encode the asterisks in the URL since **curl** will do that. This command results in the following output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vms>
    <vm id="b56a9645-f275-4698-9cad-68e3f0cd7026" href="/api/vms/b56a9645-
f275-4698-9cad-68e3f0cd7026"> ❶
```

```
        <name>rhelX</name>
        <description>Basic RHEL Server</description>
        <actions> ❷
            <link rel="shutdown" href="/api/vms/b56a9645-
f275-4698-9cad-68e3f0cd7026/shutdown"/>
            <link rel="start" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
start"/>
            <link rel="stop" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
stop"/>
            <link rel="suspend" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
suspend"/>
            <link rel="detach" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
detach"/>
            <link rel="export" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
export"/>
            <link rel="move" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
move"/>
            <link rel="ticket" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
ticket"/>
            <link rel="migrate" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
migrate"/>
        </actions>
        <link rel="disks" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
disks"/> ❸
        <link rel="nics" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/nics"/>
        <link rel="cdroms" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
cdroms"/>
        <link rel="snapshots" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
snapshots"/>
        <link rel="tags" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/tags"/>
        <link rel="permissions" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
permissions"/>
        <link rel="statistics" href="/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/
statistics"/>
        <type>server</type>
        <status>
            <state>up</state>
        </status>
        <memory>1073741824</memory>
        <cpu>
            <topology cores="1" sockets="1"/>
        </cpu>
        <os type="rhel_6x64"> ❹
            <boot dev="hd"/>
        </os>
        <high_availability>
            <enabled>false</enabled>
            <priority>0</priority>
        </high_availability>
        <display>
            <type>spice</type>
            <address>192.168.0.21</address>
            <port>5900</port>
            <secure_port>5901</secure_port>
            <monitors>1</monitors>
        </display>
        <host id="bfe3bd64-273a-11e1-b2f4-0010182b986b" href="/api/hosts/
bfe3bd64-273a-11e1-b2f4-0010182b986b"/> ❺
        <cluster id="a8ed512c-1ff8-11e1-ae09-0010182b986b" href="/api/clusters/
a8ed512c-1ff8-11e1-ae09-0010182b986b"/>
        <template id="00000000-0000-0000-0000-000000000000" href="/api/
templates/00000000-0000-0000-0000-000000000000"/>
```

```
        <start_time>2011-12-19T09:58:51.378Z</start_time>
        <creation_time>2011-12-19T10:56:59.934+01:00</creation_time>
        <origin>rhev</origin>
        <stateless>false</stateless>
        <placement_policy>
            <affinity>migratable</affinity>
        </placement_policy>
        <memory_policy>
            <guaranteed>715128832</guaranteed>
        </memory_policy>
        <usb>
            <enabled>true</enabled>
        </usb>
    </vm>
</vms>
```

1. Every virtual machine (actually every resource) will have a unique identifier in RHEV, identified here by the **id** attribute. When working with the REST API, use those unique identifiers to point to the resources to control. The **href** attribute here gives a direct link to this, and only this, virtual machine.
2. The **<actions>** block here lists all the *Actions* that can be performed on this virtual machine. To use an action, use the appropriate HTTP request (typically **POST** or **PUT**) with the **href** specified in this block.
3. This set of **<link>** tags points to the URLs to use to query and modify the resources associated with this virtual machine, like disks, network cards, and snapshots.
4. This block shows the operating system associated with a virtual machine, and the boot device that should be used by default to start it.
5. These links show with which host a virtual machine is associated, which cluster, and from what template it was defined.

7. Run the **curl** command using the request and header options. The default request is **GET**, so it has not been necessary. The RHEV REST API returns only XML currently, but the API may support other formats in the future.

```
[student@rhevm ~]$ curl -n --cacert ca.crt -X GET -H 'Accept: application/xml'\
https://rhevm.podX.example.com/api/vms?search=*rhel*
```

**-X GET -H 'Accept: application/xml'** specifies that **application/xml** answers from the server will be accepted. The RHEV REST API returns only XML currently, but the API may support other formats in the future.

8. It is also possible to modify virtual machine settings by using an HTTP PUT request on the resource URL. The following command will modify the **Description** of the **rhel*X*** virtual machine to read "API Example". This command will return the XML representation of the virtual machine, so administrators can check that the changes have been applied.

Make sure to replace the UUID in the following example with the UUID for the used virtual machine. This should be displayed from the previous command. Administrators do not have to send the entire virtual machine configuration, but just the parts they wish to change.

```
[student@rhevm ~]$ curl -n --cacert ca.crt -X PUT \ (1)
-H 'Accept: application/xml' \

-H 'Content-Type: application/xml' \ (2)
```

```
-d '<vm><description>API Example</description></vm>' \
https://rhevm.podX.example.com/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026
```

1. Here, instruct **curl** to use an HTTP PUT request instead of the normal GET request.
2. Whenever sending data to a server, it is necessary to tell the server what kind of data is being sent. In this case, XML data is being sent.
3. Here it is possible to specify the data administrators want to send to RHEV-M. Only values that need to be changed have to be specified; in this case, only the outer **<vm>** tag needs to be specified, as well as the **<description>** tag, which is what needs to be modified.

> ### Note
>
> RHEV-M will not let administrators make changes to a currently running machine. If they do try to make changes to a running machine, a normal HTTP error code will be sent back, along with a short description of what went wrong.

9. To start a virtual machine, use the **start** action with an HTTP POST request. The POST request must have a body consisting of an **<action>** tag containing any options administrators wish to use for this start. In this example, make the virtual machine start in **Paused** mode.

   Make sure to replace the UUID in the following example with the UUID for the existing virtual machine. This requires a POST operation; if administrators do not wish to pass in extra options, the **<action>** tag can be empty like **<action/>**.

```
[student@rhevm ~]$ curl -n -X POST \
-H 'Accept: application/xml' \
-H 'Content-Type: application/xml' \
-d '<action><pause>true</pause></action>' \
https://rhevm.podX.example.com/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/start
```

   The **shutdown** and **stop** actions are much the same, except that they do not take any arguments save for an empty **<action />** tag.

```
[student@rhevm ~]$ curl -n --cacert ca.crt -X POST \
-H 'Accept: application/xml' \
-H 'Content-Type: application/xml' \
-d '<action />' \
https://rhevm.podX.example.com/api/vms/b56a9645-f275-4698-9cad-68e3f0cd7026/shutdown
```

There are many other ways to use the RHEV REST API for managing RHEV. For instance, **Perl** or **Python** can be used to create scripts like the commands that were shown with **curl**. The Red Hat Enterprise Virtualization Developer Guide shows many examples using **Python**. Some Python examples can be found in the script located in the classroom at **http://classroom.example.com/pub/materials/storage_query.py**.

# Using VDSM hooks

VDSM hooks are a feature of Red Hat Enterprise Virtualization allowing a hypervisor to perform actions when certain events (such as starting or stopping a virtual machine) occur. A list of all events can be found in the Red Hat Enterprise Virtualization Administration Guide.

Among the use cases for custom VDSM hooks are sending customized notifications, influencing CPU pinning of a freshly started virtual machine, and adding custom hardware like SmartCard readers. The Red Hat CloudForms Configuration Server (Audrey) product uses VDSM hooks to customize virtual machines after boot.

> **Warning**
>
> Be careful when writing hook scripts. An incorrect hook script can cause virtual machines to crash or render virtual machines unable to start. The VDSM hook API may change in the future, and as such, hook scripts should be rigorously tested before initial deployment and after every update to RHEV.

Hook scripts are stored on a managed Red Hat Enterprise Linux host in the **`/usr/libexec/vdsm/hooks/`*`eventname`*`/`** directory. When an event occurs, all scripts in the relevant event directory will be parsed in lexicographic (alphanumeric) order. To keep sane, it is recommended that administrators start the names of the hook scripts with two digits to indicate the order they should be executed like **`01-execute_me_first.sh`** and **`90-execute_me_later.py`**.

All hook scripts will be run as the user **`vdsm`**, with the exception of the **`before_vdsm_start`** hook and the **`after_vdsm_stop`** hook, which will be executed as **`root`**.

Whenever a custom hook script is started, two variables will be added to the environment for that script. **`vmID`** will contain the UUID of the affected virtual machine, and **`_hook_domxml`** will have the file name of the **`libvirt`** domain XML file. If this is an event triggered by the starting of a virtual machine, the file referenced by **`_hook_domxml`** can be modified in order to change how the machine will be started.

If any *Custom Properties* have been set for a virtual machine, those will be passed into the script as environment variables as well. These **Custom Properties** variables could be used to further customize the VDSM hook scripts.

If administrators decide to write their custom hook scripts in Python, there is a helper library available called **`hooking`**. This library lives in the **`/usr/share/vdsm`** directory, which is automatically added to the **`PYTHONPATH`** environment variable for hook scripts. The two main functions in this library are **`read_domxml`**, which will return an **`xml.dom`** object containing the **`libvirt`** XML definition of the virtual machine, and **`write_domxml`** that will take an **`xml.dom`** object representing a virtual machine and write it back to disk.

Hook scripts use their **`exit`** status to inform RHEV-M if they were successful or not. Returning **`0`** indicates that everything went successfully, **`1`** indicates the script failed but other scripts should still be processed, and **`2`** indicates that the script failed and processing should stop completely. Any other exit codes are reserved and should not be used.

In order to deploy hook scripts, administrators will consider the following steps:

1. First, they need to either obtain or write a hook. Administrators can download ready-to-use hooks on the oVirt web site (*http://www.ovirt.org*).

2. The target hosts need to be put into maintenance mode.

3. The hook needs to be manually deployed by copying the file to the target location or via RPM on all hosts belonging to the cluster.

4. The host can be activated.

5. The hook is now displayed under the **Host Hooks** tab.

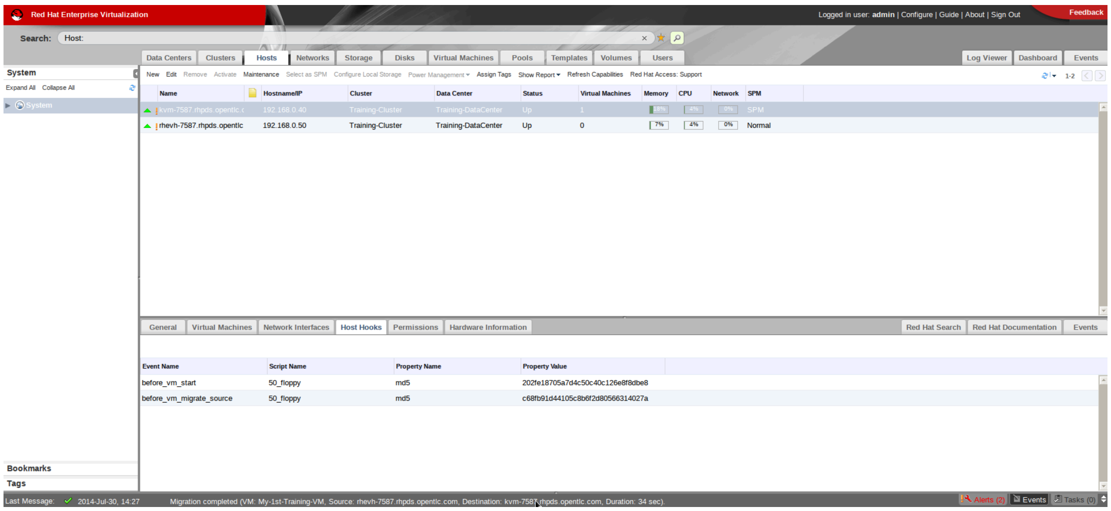


*Figure 11.1: Host hooks tab*

R

## References

Red Hat Enterprise Virtualization 3.5 Technical Guide

- Section 21.58. Hook States

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Appendix B. VDSM and Hooks
- Appendix B.4. Supported VDSM Events
- Appendix B.7. Defining Custom Properties
- Appendix B.10. Using the VDSM Hooking Module

> libvirt: Domain XML format
> http://libvirt.org/formatdomain.html

> Xpath Syntax
> http://www.w3schools.com/xpath/xpath_syntax.asp

# Quiz: Exploring Application Programming Interfaces

Choose the correct answer to the following questions.

1. Which of the following statements about the RHEV-M REST API are true? (Choose three.)

    a. The API is built around the concept of resources.
    b. All communication takes place over an HTTPS connection.
    c. Windows PowerShell is required to use the API.
    d. The API is completely stateless.
    e. You can only use the API from a Red Hat Enterprise Linux system.

2. Only users with an Administrator role can connect to the API. (Choose one.)

    a. True. RHEV users cannot connect to the API and view or change items, even the ones that they have been given permission to change.
    b. False. RHEV users can connect to the API and view or change items that they have been given permission to change.

3. Name the language that can be used to write Red Hat Enterprise Virtualization REST API scripts. (Choose one.)

    a. Perl
    b. Bash
    c. Python
    d. Perl, Bash, and Python

4. Which of the following statements are true? (Choose two.)

    a. RHEV-M will not let administrators make changes to a currently running machine.
    b. In the XML **`representation`**, every virtual machine will have a unique identifier in RHEV, identified by the **`vm_id`** attribute.
    c. It is not possible to modify virtual machine settings by using an HTTP **PUT** request on the resource URL.
    d. VDSM hooks can only be used on RHEV hypervisors.

# Solution

Choose the correct answer to the following questions.

1. Which of the following statements about the RHEV-M REST API are true? (Choose three.)

    a. **The API is built around the concept of resources.**
    b. **All communication takes place over an HTTPS connection.**
    c. Windows PowerShell is required to use the API.
    d. **The API is completely stateless.**
    e. You can only use the API from a Red Hat Enterprise Linux system.

2. Only users with an Administrator role can connect to the API. (Choose one.)

    a. True. RHEV users cannot connect to the API and view or change items, even the ones that they have been given permission to change.
    b. **False. RHEV users can connect to the API and view or change items that they have been given permission to change.**

3. Name the language that can be used to write Red Hat Enterprise Virtualization REST API scripts. (Choose one.)

    a. Perl
    b. Bash
    c. Python
    d. **Perl, Bash, and Python**

4. Which of the following statements are true? (Choose two.)

    a. **RHEV-M will not let administrators make changes to a currently running machine.**
    b. In the XML `representation`, every virtual machine will have a unique identifier in RHEV, identified by the `vm_id` attribute.
    c. It is not possible to modify virtual machine settings by using an HTTP `PUT` request on the resource URL.
    d. **VDSM hooks can only be used on RHEV hypervisors.**

# Summary

### Backing Up and Restoring RHEV

In this section, you learned:

- The **`engine-backup`** can be used to back up an existing RHEV installation, as well as restore an installation. Various arguments can be used against the utility.
- Apart from the databases, there are also a number of configuration files that will need to be backed up, such as the **`ovirt-engine`** configuration files.
- Backups can only be restored to environments of the same major RHEV release as that of the backup.

### Creating a Highly Available RHEV Manager

In this section, you learned:

- RHEV-M can be configured as a highly available virtual machine, giving administrators the possibility to configure a single RHEV-M as a virtual machine that is brought up on another host if the RHEV-M goes down.
- It is possible to manually or script file synchronization across all nodes in the cluster. However, using a configuration management system improves reliability.
- Even though a two-nodes configuration is possible, such setup can suffer from a split-brain condition, where each nodes considers itself to be the master.

### Exploring Application Programming Interfaces

In this section, you learned:

- The REST API is completely stateless. Every operation is independent from any other operation, meaning the server does not have to keep state for the different clients.
- One of the other characteristics of the RHEV-M REST API is that it is built around the idea of resources. There are no function calls, just methods that can be called on a resource.
- RHEV-M will not let administrators make changes to a currently running machine. A normal HTTP error code will be sent back, along with a short description of what went wrong.

CHAPTER 12

# INSTALLING AND CONFIGURING RHEL HOSTS

| Overview | |
|---|---|
| **Goal** | Install and configure RHEL Hosts. |
| **Objectives** | • Install, configure, and approve a RHEL Host. |
| **Sections** | • Converting RHEL to be a RHEV Host (and Practice) |
| **Quiz** | • Installing and Configuring RHEL Hosts |

# Converting RHEL to Be a RHEV Host

## Objectives

After completing this section, students should be able to:

- Install and configure a RHEL hypervisor host.

There are some circumstances where it is desirable to use a full Red Hat Enterprise Linux 7 server to act as a hypervisor for Red Hat Enterprise Virtualization. For example, a site may want to install additional monitoring software or exercise tighter control over performance-tuning variables.

Several steps are needed to prepare a Red Hat Enterprise Linux 7 host to become a RHEV-managed hypervisor. Install a Red Hat Enterprise Linux system with the **base** package set and any other packages desired.

Subscribe the system to the correct Red Hat Network Channels. The required channel is called **Red Hat Enterprise Virtualization Management Agent (v.7 for x86_64)** in the Red Hat Network web interface and **rhel-7-server-rhev-mgmt-agent-rpms** from the command line.

## Demonstration: Prepare a Red Hat Enterprise Linux host

In this example, observe the following steps as the instructor demonstrates how to prepare a Red Hat Enterprise Linux hypervisor.

1. First, make sure the system is installed with the **x86_64** version of Red Hat Enterprise Linux 7.0 or higher. Only the **base** package is required. Any other packages can be removed.

   > Important
   >
   > Only the **x86_64** version of Red Hat Enterprise Linux 7 is supported. Make sure the CPU reports both the **vmx/svm** and **nx** flags in **/proc/cpuinfo**.

   The virtual machine was installed previously, so check the release:

   ```
   [root@serverb ~]# cat /etc/redhat-release
   Red Hat Enterprise Linux Server release 7.1 (Maipo)
   ```

2. Make sure the **lm**, **vmx/svm** and **nx** flags are present in **/proc/cpuinfo**.

   ```
   [root@serverb ~]# egrep --color 'lm|vmx|svm|nx' /proc/cpuinfo
   flags  : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush
    dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon
    pebs bts rep_good aperfmperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr
    pdcm dca lahf_lm dts tpr_shadow vnmi flexpriority
   ```

3. Subscribe the system to the **Red Hat Enterprise Virtualization Management Agent (v7 x86_64)** channel on RHN, or, in the classroom, download the **rhevh.repo**

file from **ftp://classroom.example.com/materials/rhevh.repo** to **/etc/yum.repos.d/**.

```
[root@serverb ~]# wget http://classroom.example.com/materials/rhevh.repo -P /etc/
yum.repos.d
```

4. Install the *vdsm* package using **yum**.

```
[root@serverb ~]# yum install -y vdsm
```

5. Add an entry for the RHEV-M machine to **/etc/hosts** so the hypervisor can always find the RHEV-M machine, even when DNS is down.

> **Note**
>
> In order to use *live-migration* (discussed later) there must be both *forward* and *reverse* DNS records for the machine.

```
[root@serverb ~]# echo '172.25.X.15 rhevm.podX.example.com rhevm' >> /etc/hosts
```

6. Disable the **NetworkManager** service. **vdsmd** will configure the bridged networking and is not compatible (yet) with **NetworkManager**.

```
[root@serverb ~]# systemctl stop NetworkManager
[root@serverb ~]# systemctl disable NetworkManager
[root@serverb ~]# systemctl start network
[root@serverb ~]# systemctl enable network
```

7. Make sure that **root** can log in over SSH with password authentication. Do not forget to open port **22 TCP** on the firewall if you need to. Once the host is registered with RHEV-M, password authentication can be turned off again.

```
[root@workstation ~]# ssh root@serverb.podX.example.com
```

# Demonstration: Adding a Red Hat Enterprise Linux host to RHEV-M

In this example, observe the following steps as the instructor demonstrates how to integrate a new Red Hat Enterprise Linux hypervisor into RHEV-M.

Unlike RHEV-H nodes, a Red Hat Enterprise Linux 7 hypervisor node will not add itself to RHEV-M automatically. Instead, an administrator must add the host from within the RHEV-M web interface. Follow these steps to manually add a hypervisor node.

1. Log into the RHEV-M web interface and navigate to the **Hosts** tab.

2. Click the **New** button to bring up the **New Host** dialog.

3.

| General tab | |
|---|---|
| Data Center | The datacenter and cluster to which this host will be added. <br> `Data Center: dcenter0` <br> `Host Cluster: cluster0` |
| Host Cluster | |
| Name | The name for this host as it should appear in the RHEV-M interface. <br> `serverb.podX.example.com` |
| Address | The IP address for this host. Do not use a host name in this field, as SPICE connections will not work if one is present. <br> `172.25.`*`X`*`.11` |
| Root Password | The root password for this host. RHEV-M uses this during initial setup to log into the machine via SSH. <br> `redhat` |
| Automatically configure host firewall | When this setting is checked RHEV-M will update the firewall rules on the host to allow for all the connections needed. If a firewall is not currently configured (or one is not wanted), this box can be unchecked safely. <br> `Deselect` |

| Power Management tab | |
|---|---|
| Enable Power Management | Check this box to allow RHEV-M to be able to remotely reboot this hypervisor. This is necessary for *High Availability* configurations (discussed later). **The machine must have compatible hardware.** |
| Address | Fill out these fields to match the settings and type of the remote management card. |
| Username | |
| Password | |
| Type | |
| Options | |
| Test | Use this button to test power management on the new host. |

| SPM tab | |
|---|---|
| SPM Priority | The likelihood that the host will be given the role of Storage Pool Manager (SPM). Choices are Low, Normal, and High. |

4. Click the **OK** button to start installing the host. RHEV-M will connect to the host over SSH and install all the needed packages, as well as make any configurations changes necessary. If there were any errors reported during this phase, remove the host from the RHEV-M interface, fix the problem reported, and start over.

5. Navigate to the **Hosts** tab in the RHEV-M web interface and wait for the new host to appear. Initially it will be reported with a status of **Installing**, but it should move through **Non Responsive** to **Up** after a moment.

## Removing a managed RHEL host

The procedure for removing a managed Red Hat Enterprise Linux 7 host is the same as the procedure for removing a RHEV-H node.

1. Navigate to the **Hosts** tab in RHEV-M.

2. Put the managed Red Hat Enterprise Linux 7 host in maintenance mode by selecting it, right-clicking it, then selecting **Maintenance**. Click **OK** to confirm.

   When the host changes to maintenance mode, any virtual machines running on this hypervisor will be moved to a different hypervisor automatically (this will be demonstrated later). If this host was the *Storage Pool Manager (SPM)*, a new election will be held for a new SPM.

> ### Important
>
> If this is the only host in a cluster and there are running virtual machines inside the cluster, the host cannot be put into maintenance mode. In order to put this host into maintenance mode, first shut down all virtual machines running on this host manually.

3. Once the host reports its status as **Maintenance**, remove the hypervisor by selecting it, right-clicking it, then selecting **Remove**. Confirm the action by clicking **OK**.

   The freshly removed host can be moved to a different datacenter or cluster or be used for any other purpose.

> ### References
>
> Additional information may be available in the *Installation Guide* for Red Hat Enterprise Virtualization 3.5, which can be found at
>
> https://access.redhat.com/documentation/en-US/index.html

# Practice: Converting RHEL to Be a RHEV Host

In this lab, you will configure a Red Hat Enterprise Linux 7 server to act as a hypervisor for Red Hat Enterprise Virtualization.

| Resources | |
|---|---|
| **Files** | http://classroom.example.com/materials/rhevh.repo |
| **Application URL** | https://rhevm.pod*X*.example.com |

### Outcome(s)

You should be able to migrate a RHEL system into a hypervisor host managed by your RHEV-M Engine.

**Before you begin**

Make sure the RHEV-M environment configured in a previous lab is still working.

Perform the following steps:

1. **ssh** into the serverb.podX.example.com system from the workstation Terminal windows as **root**.

2. To install the packages needed to turn this system into a Red Hat Enterprise Virtualization Host, you need to subscribe it to a new RHN channel or **yum** repository. On a production system, you would subscribe to the **Red Hat Enterprise Virt Management Agent** channel on Red Hat Network, but here in the classroom, we have a custom **yum** repository. To activate this repository, download the file **rhevh.repo** from **http://classroom.example.com/materials/rhevh.repo** and place it in the directory **/etc/yum.repos.d/**.

```
[root@serverb ~]# wget http://classroom.example.com/materials/rhevh.repo -P /etc/
yum.repos.d/
```

3. Now that the repository is set up, install the *vdsm* package. Normally, RHEV-M provides this package, but for this practice, perform this process.

```
[root@serverb ~]# yum -y install vdsm
```

4. This step is only necessary if you are running this in an online learning environment.

> **Important**
>
> This step is only necessary if you are running this in an online learning environment since the nested virtualization of the cloud is missing some information needed to run RHEV-H.
>
> Open the virtual keyboard to send **F2** to the system to open the Rescue Shell. Select **OK** to be presented with a prompt.
>
> ```
> [root@serverb ~]# cd /usr/share/libvirt
> [root@serverb libvirt]# mv cpu_map.xml cpu_map.xml-orig
> [root@serverb libvirt]# wget http://classroom.example.com/materials/
> cpu_map.xml
> ```

5. Make sure that your Red Hat Enterprise Linux host can always find your RHEV-M machine, even when DNS is down. Add an entry for your rhevm machine to **/etc/hosts** on your serverb machine.

   ```
   [root@serverb ~]# echo "172.25.X.15 rhevm.podX.example.com rhevm" >> /etc/hosts
   ```

6. Since **vdsmd** will configure network bridges, make certain that **NetworkManager** is disabled, or not installed. To do so, execute the following commands:

   ```
   [root@serverb ~]# systemctl stop NetworkManager
   [root@serverb ~]# systemctl disable NetworkManager
   [root@serverb ~]# systemctl start network
   [root@serverb ~]# systemctl enable network
   ```

7. Back in the RHEV-M GUI, navigate to the **Hosts** tab and click the **New** button. In the dialog that opens, use **testdc*X*** and **testclu*X*** for the data center and cluster, **serverb.podX.example.com** for the **Name**, **172.25.*X*.11** for the **Address**, and enter **redhat** as a password. Ensure the firewall box is deselected. Click **OK** to initiate the installation. Click **OK** again in the confirmation dialog that appears to confirm running without power management.

8. RHEV-M will now install and configure packages on your machine. Follow this process either by running **top** on your hypervisor or by looking at the event log in the **Event** tab of the RHEV-M interface.

9. Once your new hypervisor has joined, wait until the status has changed to **Up** and the **SPM** column has gone from **Contending** to **Normal**.

# Quiz: Installing and Configuring RHEL Hosts

Match the following items to their counterparts in the table.

No eXecute (NX)

Red Hat Enterprise Virtualization Management Agent (v.7 for x86_64)

Virtual Desktop Server Manager (VDSM)

| Definition | Term |
|---|---|
| CPU flag required by RHEV but not RHEL KVM for execute space protection | |
| Management module that serves as a RHEV Manager (RHEV-M) agent on RHEV hypervisors (RHEV-H) or Red Hat Enterprise Linux hosts | |
| Required channel/ repository to obtain packages for a Red Hat Enterprise Linux host to be managed by RHEV-M | |

# Solution

Match the following items to their counterparts in the table.

| Definition | Term |
| --- | --- |
| CPU flag required by RHEV but not RHEL KVM for execute space protection | No eXecute (NX) |
| Management module that serves as a RHEV Manager (RHEV-M) agent on RHEV hypervisors (RHEV-H) or Red Hat Enterprise Linux hosts | Virtual Desktop Server Manager (VDSM) |
| Required channel/repository to obtain packages for a Red Hat Enterprise Linux host to be managed by RHEV-M | Red Hat Enterprise Virtualization Management Agent (v.7 for x86_64) |

# Summary

**Converting RHEL to Be a RHEV Host**

In this section, you learned:

- A channel subscription must be established before a Red Hat Enterprise Linux server can function as a hypervisor host.
- RHEV-M Administration Portal is used to add or register the prepared Red Hat Enterprise Linux host.
- RHEV-M Administration Portal can also be used to remove a Red Hat Enterprise Linux host.

CHAPTER 13

# MIGRATING VMS AND CONFIGURING HIGH AVAILABILITY

| Overview | |
|---|---|
| **Goal** | Migrate VMs manually and automatically and configure high availability with live migration. |
| **Objectives** | • Manually migrate a virtual machine.<br>• Automate migration with cluster policy.<br>• Configure high availability and live migration. |
| **Sections** | • Migrating a Virtual Machine (and Practice)<br>• Automating Migration (and Practice)<br>• Configuring High Availability (and Practice) |
| **Quiz** | • Migrating VMs and Configuring High Availability |

# Migrating a Virtual Machine

## Objectives

After completing this section, students should be able to:

- Manage virtual machine migrations.
- Configure cluster policies.
- Configure automated migration.
- Implement high availability with live migration.

## Running a virtual machine on a single host

There are cases when a virtual machine is required to run on a single host. For instance, administrators may want a virtual machine to run on a host that has the best CPU processing power, or one that has the most RAM. The following steps can be used to restrict a virtual machine to a single host:

## Demonstration: Running a virtual machine on a single host

In this example, observe the following steps as the instructor demonstrates how to force a virtual machine to run on a single host.

1. Browse to the **Virtual Machines** tab in RHEV-M.
2. Right-click the **`rhel`*`X`***, and choose **Edit**.
3. Choose the **Host** tab.
4. In the right pane, under **Start Running On:**, select **Specific** and choose serverb.podX.example.com. Click the **OK** button.

> **Note**
>
> This operation can be performed while the virtual machine is running. Changes will be applied on the virtual machine next start.

5. To verify, start the rhel*X* virtual machine and verify that it runs on **`serverb.podX.example.com`**.

## Virtual machine migration

Virtual machine migration refers to the process of moving a running virtual machine between different physical machines, without disconnecting the client, by keeping network and system connectivity. Memory, storage, and network connectivity of the virtual machine are transferred from the original host machine to the destination. Migration support is integrated into Red Hat Enterprise Virtualization. A virtual machine may only migrate to hypervisor hosts that are members of its cluster. This helps to ensure that virtual machines do not migrate between machines that support a different set of processor features. Virtual machines can be migrated from one host to another while running. This is called *live migration*. Clients communicating with the virtual machine should notice no more than a network pause of a few milliseconds as the transfer completes.

Live migration is transparent to the end user; the virtual machine remains powered on and user applications continue to run while the virtual machine is relocated to a new physical host. Live migration is used to seamlessly move virtual machines to support a number of common maintenance tasks. Administrators must ensure their Red Hat Enterprise Virtualization environment is correctly configured to support live migration well in advance of using it. At a minimum, for successful live migration of virtual machines to be possible:

- The source and destination host must both be members of the same cluster, ensuring CPU compatibility between them.
- The source and destination host must have a status of **Up**.
- The source and destination host must have access to the same virtual networks and VLANs.
- The source and destination host must have access to the data storage domain on which the virtual machine resides.
- There must be enough CPU capacity on the destination host to support the virtual machine's requirements.
- There must be enough RAM on the destination host that is not in use to support the virtual machine's requirements.
- The migrating virtual machine must not have the **cache!=none** custom property set.

In addition, for best performance, the storage and management networks should be split to avoid network saturation. Virtual machine migration involves transferring large amounts of data between hosts. Live migration is performed using the management network. Each live migration event is limited to a maximum transfer speed of 30 MBps, and the number of concurrent migrations supported is also limited by default.

### Warning

Despite these measures, concurrent migrations have the potential to saturate the management network. It is recommended that separate logical networks are created for storage, display, and virtual machine data to minimize the risk of network saturation.

The following steps demonstrate how to use RHEV-M to migrate virtual machines. Note that as the virtual machine migrates, it will briefly show up in the VM count on both the original host and on the host it is migrating to. The virtual machine status will indicate that a migration is in progress.

## Demonstration: Migrate a virtual machine

1. Red Hat Enterprise Virtualization is capable of *live migration*; that is, moving a running virtual machine from one hypervisor to another without interrupting the virtual machine. In

order for live migration to work, there must be at least two hypervisors running in the same cluster.

Find and select the **rhel*X*** virtual machine under the **Virtual Machines** tab.

2. Open a console to the virtual machine to be used during the migration. Right-click the **rhel*X*** virtual machine and choose **Console**.

   After migration beings, it is not possible to use the console.

3. Right-click the virtual machine and choose **Migrate**.

4. Choose **Select Destination Host** and pick the servera.pod*X*.example.com host in the dropdown menu. Click **OK**.

### Note

The VM will continue to run on the original host until its memory state has completely transferred to the new host.

5. While the virtual machine is migrating, bring up the console and use the **ping** command to view the network response.

```
[root@rhelX ~]# ping 172.25.254.254
```

   A few packets should be seen as being dropped as the network changes to the new hypervisor.

   Find and select the **rhel*X*** virtual machine under the **Virtual Machines** tab.

6. Using the same procedure, move back the virtual machine from servera.pod*X*.example.com to serverb.podX.example.com, in order to have the same configuration as students for the following guided exercise.

### References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 9.15. Migrating Virtual Machines Between Hosts

Red Hat Enterprise Virtualization 3.5 Technical Guide

- Section 5.9. Migration

# Practice: Migrating a Virtual Machine

In this lab, you will migrate a virtual machine from one host (servera.pod*X*.example.com) to another (serverb.podX.example.com).

### Outcome(s)

- You should be able to migrate a virtual machine to another RHEV host without downtime.

**Before you begin**

You must have RHEV-M running with two hosts in a cluster. **`rhelX`** must be installed and running in the cluster on the servera.pod*X*.example.com host.

Perform the following steps:

1. Log into RHEV-M running on workstation and browse to the **Virtual Machines** tab.

2. Right-click the **`rhelX`** virtual machine and select **Console**. Leave the console up so you can connect to the machine during the migration.

3. In RHEV-M, right-click the **`rhelX`** virtual machine and select **Migrate**.

4. Choose the **Select Destination Host** radio button and choose the serverb.podX.example.com host. Press **OK**.

5. While you are waiting for the migration to complete, bring up the console and log into the **`rhelX`** virtual machine. Run the **`ping`** command to watch the status as it migrates:

```
[root@rhelX ~]# ping 172.25.254.254
```

6. Once the migration is complete, verify it is running on the new host. In the **Virtual Machines** tab, the **`rhelX`** should show **`serverb.podX.example.com`** in the **Host** column.

# Automating Migration

## Objectives

After completing this section, students should be able to:

- Manage a resilience policy for a cluster.
- Manage a cluster (distribution) policy.
- Migrate machines based on the cluster policies.
- Place a host in maintenance.

## Automated migration and cluster policy

The resilience policy sets the virtual machine migration policy in the event of host failure. Virtual machines running on a host that unexpectedly shuts down or is put into maintenance mode are migrated to other hosts in the cluster, and this migration is dependent upon the **cluster policy**.

> **Note**
>
> Virtual machine migration is a network-intensive operation. For instance, on a setup where a host is running 10 or more virtual machines, migrating all of them can be a long and resource-consuming process. Therefore, administrators must be sure to select the policy action that best suits their setup.

Cluster *policies* allow administrators to specify the usage and distribution of virtual machines between available hosts. Defining the cluster policy enables automatic load balancing across the hosts in a cluster. A host that exceeds the **`Maximum Service Level`** will send virtual machines (one at a time) to other hosts in an attempt to lighten the CPU load. A host that does not exceed the **`Minimum Service Level`** will migrate all of its virtual machines to other hosts and power down until such time as it is required again (thus saving power). Many common maintenance tasks, including network configuration and deployment of software updates, require that hosts be placed into *maintenance* mode. When a host is placed into maintenance mode, the Red Hat Enterprise Virtualization Manager attempts to migrate all running virtual machines to alternative hosts. The normal prerequisites for live migration apply; in particular, there must be at least one active host in the cluster with capacity to run the migrated virtual machines.

Red Hat Enterprise Virtualization Manager automatically initiates live migration of all virtual machines running on a host when the host is moved into maintenance mode. The destination host for each virtual machine is assessed as the virtual machine is migrated in order to spread the load across the cluster. The **`Manager`** automatically initiates live migration of virtual machines in order to maintain load balancing or power-saving levels in line with cluster policy. While no cluster policy is defined by default, administrators should specify the cluster policy that best suits the needs of the environment. It is also possible to disable automatic, or even manual, live migration of specific virtual machines where required.

Red Hat Enterprise Virtualization Manager allows administrators to disable automatic migration of virtual machines. It is possible to disable manual migration of virtual machines by setting the virtual machine to run only on a specific host. The ability to disable automatic migration and require a virtual machine to run on a particular host is useful when using application high-availability products, such as Red Hat High Availability or Cluster Suite.

# Demonstration: Setting cluster policy and host maintenance mode

In this example, observe the following steps as the instructor demonstrates how to set a cluster policy, and set a host into maintenance mode.

1. Browse to the **Clusters** tab in RHEV-M. Right-click the **`testclu`*X*** cluster and choose **Edit**. In the left pane, click the **Resilience Policy** tab.

   This screen is the same whether administrators are creating a new cluster or editing an existing cluster (as here). The resilience policy can be configured to choose which virtual machines are automatically migrated when a host fails.

   - **Migrate Virtual Machines** migrates all virtual machines in order of the defined priority (high-availability priorities shown later in this chapter).
   - **Migrate only Highly Available Virtual Machines** migrates only server virtual machines marked as **Highly Available** (also shown later). All other virtual machines will be shut down.
   - **Do Not Migrate Virtual Machines** will not migrate any virtual machines. All virtual machines will be shut down and remain so until manually started.

2. Click the **Cluster Policy** tab. From the **Select Policy** option, **`power_saving`** should be selected. Select the **`evenly_distributed`** policy from the dropdown list.

   **`Even Distribution`** allows a host to live-migrate virtual machines to other hosts in the cluster if the CPU load exceeds the **Maximum Service Level**. If the virtual machines on a host in this cluster use more than 90% of the CPU resources, RHEV will begin to migrate virtual machines to another available host. Additional virtual machines attached to a host will not start if that host has reached the defined **Maximum Service Level**.

3. Edit the policy properties using the following values:

| Key | Value |
|---|---|
| CpuOverCommitDuration | **`3`** |
| HighUtilization | **`90`** |

   **`Power Saving`** mode will distribute CPU processing load across a subset of available hosts to reduce power consumption on underutilized hosts. Hosts with a CPU load *below* the minimum service level past the defined time interval will migrate all virtual machines to other hosts so that it can be powered down. As with **`Even Distribution`**, if the CPU load exceeds the maximum service level, RHEV Manager will move virtual machines to other available hosts in the cluster. In the example, if virtual machines on a host use less than 10% of the CPU for five minutes, RHEV Manager will migrate virtual machines off of the host to another available host in the cluster. If the virtual machines on a host use more than 75% of the CPU for five minutes, RHEV Manager will migrate virtual machines off of the host to another available host in the cluster.

4. In the left pane, choose the **Resilience Policy** tab. Ensure the **Migrate Virtual Machines** policy is set. Click **OK**.

5. Browse to the **Hosts** tab in RHEV-M. Right-click the **`serverb.podX.example.com`** host and choose **Maintenance**.

6. All the running virtual machines should be visibly migrating to the servera.pod*X*.example.com host.

7. Once the migration is complete, the **`serverb.podX.example.com`** host status will go from **`Preparing for Maintenance`** to **`Maintenance`**.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 6.7. Host Resilience
- Section 4.2.4. Setting Load and Power Management Policies for Hosts in a Cluster
- Section 5.5.2. Explanation of Settings and Controls in the New Cluster and Edit Cluster Window
- Section 6.5.9. Moving a Host to Maintenance Mode
- Section 9.15. Migrating Virtual Machines Between Hosts

Red Hat Enterprise Virtualization 3.1 Evaluation Guide

- Section 5.9. Migration

# Practice: Moving Host into Maintenance Mode

In this lab, you will set the cluster policy and place the servera.pod*X*.example.com host into maintenance mode.

### Outcome(s)

- A cluster policy will be enabled, updated, and tested.
- **servera** will be put in maintenance mode; the virtual machines running on it will be migrated to **serverb**.

**Before you begin**

You must have RHEV-M running with two hosts in a cluster.

Perform the following steps:

1. Browse to the **Clusters** tab in RHEV-M. Right-click the **clusterX** cluster and choose **Edit**.
2. Click the **Cluster Policy** tab. From the **Select Policy** option, **evenly_distributed** should be selected. Select the **evenly_distributed** policy from the dropdown list.
3. Edit the policy properties using the following values:

| Key | Value |
|---|---|
| CpuOverCommitDuration | **2** |
| HighUtilization | **90** |

4. In the left pane, choose the **Resilience Policy** tab. Ensure the **Migrate Virtual Machines** policy is set. Click **OK**.
5. **ssh** to the servera.pod*X*.example.com machine, using **root** for the username and **redhat** as the password. Find out how many processors are available:

```
[student@workstation ~]# ssh root@servera.podX.example.com
[root@servera ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
...
```

This example shows a system with eight CPUs.

**ssh** to the serverb.podX.example.com machine, using **root** for the username and **redhat** as the password. Find out how many processors are available:

```
[student@workstation ~]# ssh root@serverb.podX.example.com
[root@serverb ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
```

```
...
```

This example shows a system with two CPUs.

> **Important**
>
> If the two hosts have a different number of CPUs, use the smaller value in the next step.

6. Shut down the **rhel*X*** virtual machine and configure it to use the same number of CPUs that you found previously.
   a. In the **Virtual Machines** tab, right-click the **rhel*X*** virtual machine and choose **Power Off**. Wait for the virtual machine to shut down.
   b. Once the **rhel*X*** virtual machine has shut down, right-click the virtual machine and choose **Edit**. Change the **Total Virtual CPUs** to the number found previously. Press **OK**.
   c. Right-click the **rhel*X*** virtual machine and choose **Run**. If the **rhel*X*** virtual machine did not start running on the serverb.podX.example.com host, migrate the virtual machine to the serverb.podX.example.com host:
      c.i. Right-click the **rhel*X*** virtual machine and choose **Migrate**.
      c.ii. Click **Select Destination Host** and choose **serverb.podX.example.com**. Click **OK**.
      c.iii. Wait for the virtual machine to finish the migration.
7. Right-click the **rhel*X*** virtual machine and choose **Console**. Log into the virtual machine (**root**:**redhat**). Run the following command at least two times, or once for each CPU:

```
[root@vmY ~]# dd if=/dev/zero of=/dev/null &
```

   (In the example, we would run this command four times.)
8. In RHEV-M, open the **Virtual Machines** tab and watch as the CPU usage for the **rhel*X*** virtual machine approaches 100%. After the timeout period, you should see it migrate to the servera.pod*X*.example.com host.
9. Once it has moved to servera.pod*X*.example.com, go to the **rhel*X*** console and run the following command to clean up:

```
[root@vmY ~]# kill `pidof dd`
```

10. Browse to the **Hosts** tab in RHEV-M. Right-click the **servera.pod*X*.example.com** host and choose **Maintenance**.
11. The **rhel*X*** virtual machine will automatically migrate from the **servera.pod*X*.example.com** host to the **serverb.podX.example.com** host.

12. Once the migration is complete, the **servera.pod*X*.example.com** host status will go from **Preparing for Maintenance** to **Maintenance**.

# Configuring High Availability

## Objectives

After completing this section, students should be able to:

- Implement high availability with live migration.

## Virtual machine high availability

High availability means that a virtual machine will be automatically restarted if its process is interrupted. This happens if the virtual machine is terminated by methods other than powering off from within the guest or sending the shutdown command from the **Manager**. When these events occur, the highly available virtual machine is automatically restarted, either on its original host or another host in the cluster. High availability is possible because the Red Hat Enterprise Virtualization Manager constantly monitors the hosts and storage, and automatically detects hardware failure. If host failure is detected, any virtual machine configured to be highly available is automatically restarted on another host in the cluster.

With high availability, interruption to service is minimal because virtual machines are restarted within seconds with no user intervention required. High availability keeps the resources balanced by restarting guests on a host with low current resource utilization, or based on any workload balancing or power saving policies that are configured. This ensures that there is sufficient capacity to restart virtual machines at all times.

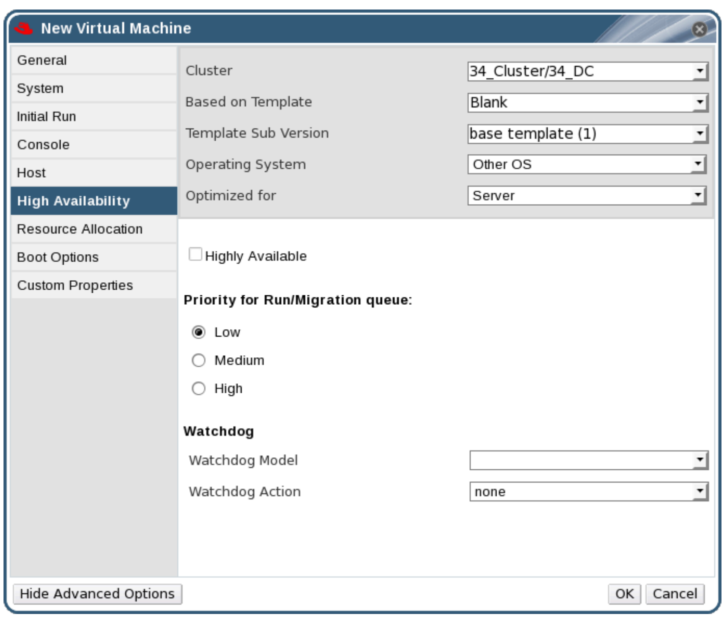

*Figure 13.1: High-availability policy*

High availability is recommended for virtual machines running critical workloads. High availability can ensure that virtual machines are restarted in the following scenarios:

- When a host becomes nonoperational due to hardware failure.
- When a host is put into maintenance mode for scheduled downtime.

- When a host becomes unavailable because it has lost communication with an external storage resource.

> **Note**
>
> A high-availability virtual machine is automatically restarted, either on its original host or another host in the cluster.

Guests may be configured to automatically migrate from one host to another if the host becomes unresponsive or the virtual machine unexpectedly crashes. This is accomplished by checking the **Highly Available** property at creation, or through the virtual machine **Edit** button. In order to use this feature, all hosts in the cluster must support power management; they must have an out-of-band management agent such as an **iLO**, **DRAC**, or **RSA** port, and it must be configured properly.

RHEV must make absolutely certain that the virtual machine is never running on two hosts at the same time, or its disk image is likely to become corrupt. In order to ensure this does not happen, RHEV will use the out-of-band management agent to *fence* the host; that is, to kill its power and ensure that it and its virtual machines are down. Only then will it reboot the virtual machine on a new host.

*Multilevel priority intelligent failover*: High availability ensures that high-priority virtual machines are automatically restarted on failure of the virtual machine itself or the host on which it resides. Multiple levels of priority give highest restart priority to the most important virtual machines.

To enable the migration of highly available virtual machines, the following requirements must be met:

1. Power management must be configured for the hosts running the highly available virtual machines.
2. The host running the highly available virtual machine must be part of a cluster that has other available hosts.
3. The destination host must be running.
4. The source and destination host must have access to the data domain on which the virtual machine resides.
5. The source and destination host must have access to the same virtual networks and VLANs.
6. There must be enough CPUs on the destination host that are not in use to support the virtual machine's requirements.
7. There must be enough RAM on the destination host that is not in use to support the virtual machine's requirements.

> **Note**
>
> Administrators can only set high availability for virtual servers. They cannot set high availability for virtual desktops. The **High Availability** tab only appears in the server's configuration.

# Demonstration: Configuring high availability

In this example, observe the following steps as the instructor demonstrates how to activate high availability for a virtual machine.

**Virtual servers** (not desktops) can be marked as *Highly Available* either at creation time or later on when editing by going to the **High Availability** tab. Highly available virtual servers will be automatically restarted (on another host) if they crash or if the underlying hypervisor stops responding.

For high availability to work, all hosts in a cluster must have *power management* configured. This means that RHEV-M can forcefully shut down or reboot a hypervisor by using a remote management card on that hypervisor. Explain that this is called *fencing*. This is necessary to make sure that a hypervisor, and all the virtual machines running on that hypervisor, are down before we start the virtual machines on another hypervisor. Ask your students if they can imagine the destruction that would occur if a virtual machine somehow ended up running on two hypervisors at the same time while using the same disk image.

1. *Creating a new server*: Click the **New VM** in the **Virtual Machines** tab. Navigate to the **High Availability** tab. Check the **Highly Available** button and choose the priority.

   If more than one virtual machine needs to be started on a different hypervisor due to a high-availability failover, RHEV will place all machines in a queue. The failover will first start migrating all machines with a priority of **High**, then those marked **Medium**, and finally those marked as **Low** priority.

2. *Editing an existing server*: In the **Virtual Machines** tab, right-click the **rhel*X*** virtual machine and choose **Edit**. Navigate to the **High Availability** tab. Check the **Highly Available** button and choose the priority.

## References

Red Hat Enterprise Virtualization 3.5 Administration Guide

- Section 4.2.2.3. Resilience Policy Settings Explained
- Section 9.5.6. Virtual Machine High Availability Settings Explained
- Section 9.15. Migrating Virtual Machines Between Hosts
- Section 9.16. Improving Uptime with Virtual Machine High Availability

# Practice: Configuring High Availability

In this lab, you will enable high-availability capabilities for a running virtual machine.

**Outcome(s)**

- The virtual machine rhel*X* will have high availability enabled.

**Before you begin**

You must have an existing virtual machine named rhel*X*.

Perform the following steps:

1. Log into RHEV-M running on desktop and browse to the **Virtual Machines** tab.
2. Right-click the **`rhelX`** virtual machine and select **Edit**.
3. Navigate to the **High Availability** tab. Check the **Highly Available** button and set the priority to **High**.
4. Click **OK** in order to save your changes.

# Quiz: Migrating VMs and Configuring High Availability

Choose the correct answer to the following questions.

1. Which of the following requirements have to be met for successful migration of virtual machines? (Choose three.)

    a. The source and destination host must both be members of the same cluster.
    b. The migrating virtual machine must have the **`cache!=none`** custom property set.
    c. The source and destination host must have a status of *Up*.
    d. The source and destination host must have access to the same virtual networks and VLANs.
    e. The source and destination host must have the same amount of RAM.

2. When migrating, what is the potential bottleneck that might be encountered? (Choose one.)

    a. Saturating the CPU.
    b. Saturating the management network.
    c. Saturating I/Os.

3. Name the policy which sets the virtual machine migration policy in the event of host failure. (Choose one.)

    a. Cluster policy.
    b. Resilience policy.
    c. Host policy.
    d. Migration policy.

4. Which of the following requirements have to be met for high availability to work? (Choose three.)

    a. Power management must be configured for the hosts running the highly available virtual machines.
    b. The source and destination host must have the **`fencing=on`** custom property set.
    c. The source and destination host must have access to the data domain on which the virtual machine resides.
    d. The destination host must be running.
    e. The destination host can be put in maintenance mode, as long as power management is configured on it.

# Solution

Choose the correct answer to the following questions.

1. Which of the following requirements have to be met for successful migration of virtual machines? (Choose three.)

   a. **The source and destination host must both be members of the same cluster.**
   b. The migrating virtual machine must have the **cache!=none** custom property set.
   c. **The source and destination host must have a status of *Up*.**
   d. **The source and destination host must have access to the same virtual networks and VLANs.**
   e. The source and destination host must have the same amount of RAM.

2. When migrating, what is the potential bottleneck that might be encountered? (Choose one.)

   a. Saturating the CPU.
   b. **Saturating the management network.**
   c. Saturating I/Os.

3. Name the policy which sets the virtual machine migration policy in the event of host failure. (Choose one.)

   a. Cluster policy.
   b. **Resilience policy.**
   c. Host policy.
   d. Migration policy.

4. Which of the following requirements have to be met for high availability to work? (Choose three.)

   a. **Power management must be configured for the hosts running the highly available virtual machines.**
   b. The source and destination host must have the **fencing=on** custom property set.
   c. **The source and destination host must have access to the data domain on which the virtual machine resides.**
   d. **The destination host must be running.**
   e. The destination host can be put in maintenance mode, as long as power management is configured on it.

# Summary

### Migrating a Virtual Machine

In this section, you learned:

- For specific needs, it is possible to force the host on which the virtual machine will run.
- Before migrating a virtual machine, an environment has to meet specific criteria, such as host status or enough CPU capacity.
- Migration may potentially saturate the network bandwidth; this is why is it recommended to have a dedicated network for migration.

### Automating Migration

In this section, you learned:

- Automating migration can be accomplished by setting cluster policies and resiliency.
- Cluster policies allow for specifying usage and distribution of virtual machines between available hosts.
- Red Hat Enterprise Virtualization Manager automatically initiates live migration of all virtual machines running on a host when the host is moved into maintenance mode.

### Configuring High Availability

In this section, you learned:

- High availability keeps the resources balanced by restarting guests on a host with low current resource utilization, or based on any workload balancing or power-saving policies that are configured.
- For high availability to work correctly, an environment has to met specific criteria, such as power-management or running hosts.
- Fencing may be used in order to prevent data corruption.

redhat TRAINING

# CHAPTER 14

# COMPREHENSIVE REVIEW

| Overview | |
|---|---|
| **Goal** | Practice the skills attained throughout the course. |
| **Objectives** | • Review the objectives from the course. |
| **Sections** | • Comprehensive Review (and Lab) |

# Comprehensive Review

## Objectives

After completing this section, students should be able to:

- Apply the performance objectives from throughout the course.

## Installation Review

The basic steps for building a Red hat Enterprise Virtualization environment are as follows:

1. Verify hardware requirements of engine and host systems.
2. Subscribe the RHEV-M machine to the appropriate channels.
3. Install the *rhevm*, *rhevm-dwh*, and *rhevm-reports* packages.
4. Configure RHEV-M with **engine-setup** and **engine-config**.
5. Bind to an authentication domain using **engine-manage-domains**.
6. Configure your web browser (if needed).
7. Configure a datacenter.
8. Configure a cluster.
9. Install, configure, and approve a RHEV-H hypervisor host.
10. Configure the NFS exports on rhevm.
11. Create a data domain.
12. Create an export domain.
13. Create and populate an ISO domain.
14. Create a VM network.
15. Install a virtual machine.

## Troubleshooting review

1. Authentication
   - Make sure that the RHEV-M instance is tied to all the domains to use.
   - Make sure that the time on all machines is set correctly and that NTP is configured on all hosts.
2. Web interface
   - Make sure the **ovirt-engine** service is running on the **rhevm.pod*X*.example.com** machine.

- Make sure that the RHEV-M CA certificate has been properly imported into the web browser.

3. Storage

   - Make sure the datacenter has an active domain.
   - Make sure your datacenter has at least one active cluster.
   - Make sure your cluster has at least one active host.
   - Make sure the host is active (not in **Maintenance** or **Contending**) mode.

4. RHEV-H

   - Make sure to be pointing to the correct RHEV-M host.
   - Make sure that networking is configured correctly.
   - Make sure that all storage being used is accessible from the RHEV-H host.
   - Make sure that all *Logical Networks* configured in the cluster are configured on the host.

# Management review

Some of the management tasks that have been done are as follows:

1. Upgrade a RHEV-H hypervisor host.
2. Create a snapshot.
3. Export a VM image.
4. Create a template.
5. Create a pool.
6. Create search tags.
7. Create an ad hoc report.
8. Create and test a backup.

# Comprehensive Review: Installation

In this comprehensive review, you will build a Red Hat Enterprise Virtualization environment.

**Outcome(s)**

- You should be able to create and configure a RHEV environment. You should be able to test the environment by creating a virtual machine.

**Before you begin**

Reset your virtual systems, **workstation**, **rhevm**, **servera**, and **serverb**.

> **Note**
>
> If running a physical **servera**, perform the steps to uninstall it (see earlier unit lab for details).

Perform the following steps:

1. Verify hardware requirements of engine (**rhevm.pod*X*.example.com**) and host (**servera.pod*X*.example.com**) systems.

2. Subscribe the RHEV-M machine to the appropriate channels.

   Use the provided **http://classroom.example.com/materials/rhevm.repo** file.

3. Install the *rhevm*, *rhevm-dwh*, and *rhevm-reports* packages.

4. Configure RHEV-M with **engine-setup**. Generate an answer file named **/root/answers.txt**.

| Prompt | Answer |
|---|---|
| **Configure Engine on this host** | yes |
| **Configure Data Warehouse on this host** | yes |
| **Configure Reports on this host** | yes |
| **WebSocket Proxy** | yes |
| **Do you want Setup to configure the firewall?** | no |
| **Host fully qualified DNS name of this server** | rhevm.pod*X*.example.com |
| **Where is the Reports database located?** | local |
| **Would you like Setup to automatically configure postgresql and create Reports database, or prefer to perform that manually? (Automatic, Manual)** | automatic |
| **Where is the DWH database located?** | local |
| **Would you like Setup to automatically configure postgresql and create DWH database, or prefer to perform that manually?** | automatic |

| Prompt | Answer |
|---|---|
| **Where is the Engine database located? (Local, Remote) [Local]:** | local |
| **Would you like Setup to automatically configure postgresql and create Engine database, or prefer to perform that manually?** | automatic |
| **Engine admin password (confirm)** | redhat |
| **Use weak password?** | yes |
| **Application mode (Virt, Gluster, Both)** | both |
| **Organization name for certificate** | pod*X*.example.com |
| **Do you wish to set the application as the default page of the web server?** | yes |
| **Do you wish Setup to configure that, or prefer to perform that manually?** | automatic |
| **Configure an NFS share on this server to be used as an ISO Domain?** | yes |
| **Local ISO domain path** | /exports/rhevisos |
| **Local ISO domain ACL** | rhevm.pod*X*.example.com(rw) |
| **Local ISO domain name (make sure to replace X by your station number)** | iso*X* |
| **Reports power users password (confirm the password)** | redhat |
| **Use weak password?** | yes |
| **Would you like transactions from the Red Hat Access Plugin sent from the RHEV Manager to be brokered through a proxy server?** | no |

5. Bind to the IdM authentication domain, **example.com**, using **rhevadmin** and **redhat** as credentials.

6. Configure your web browser certificate.

7. Create a new datacenter with the following values:

   - Name: **testdc*X*** (where *X* is your pod number)
   - Description: **Test Datacenter**
   - Storage Type: **Shared**
   - Compatibility Version: **3.5**
   - Quota Mode: **Disabled**

8. Create a new cluster with the following values:

   - Data Center: **testdc*X*** (where *X* is your pod number)
   - Name: **testclu*X*** (where *X* is your pod number)

- Description: **Test Cluster**
- CPU Name: *Depends on classroom hardware*
- Compatibility Version: **3.5**
- For storage: **Enable Virt Service**
- Memory Optimization: **None**
- Resilience Policy: **Migrate Virtual Machines**

9. Install, configure, and approve a RHEV-H host on **servera**.
    - RHEV-H host administrator password: **redhat123**
    - Network, Hostname: **servera.pod*X*.example.com**; DNS Server 1: **172.25.254.254**; NTP Server 1: **172.25.254.254**
    - Interface: **Static**; IP Address: **172.25.*X*.10**; Netmask: **255.255.255.0**; Gateway: **172.25.*X*.254**
    - Enable ssh pasword authentication
    - Register to **rhevm.pod*X*.example.com**
10. Configure NFS Exports of **/exports/testdata**, **/exports/testexport**, and **/exports/rhevisos** on RHEV-M
11. Create a data domain with the following values:
    - Name: **testdata*X*** (where *X* is your pod number)
    - Data Center: **testdc*X*** (where *X* is your pod number)
    - Domain Function/Storage Type: **Data / NFS**
    - Use Host: **servera.pod*X*.example.com**
    - Export path: **rhevm.pod*X*.example.com:/exports/testdata**
12. Create an export domain with the following values.
    - Name: **testexport*X*** (where *X* is your pod number)
    - Data Center: **testdc*X*** (where *X* is your pod number)
    - Domain Function/Storage Type: **Export / NFS**
    - Use Host: **servera.pod*X*.example.com**
    - Export path: **rhevm.pod*X*.example.com:/exports/testexport**
13. Attach the ISO domain (**iso*X***) and upload the RHEL 7.1 boot iso available at **http://classroom.example.com/materials/**.

14. Create a VM network with the following values:

| Tab | Field | Value |
|---|---|---|
| General | Name | **VMNet** |
| | Description | **VM Network** |
| | VM Network | **Checked** |
| | MTU | **Default (1500)** |
| Cluster | testclu*X* | **Attached** |
| | testclu*X* | **Not Required** |
| vNIC Profiles | Public | **Checked** |

15. Define the range of MAC addresses for the virtual machines that is being offered by the DHCP server: **52:54:00:00:XX:10-52:54:00:00:XX:FF** (where **XX** is the two-digit hexadecimal value of your pod number, pod 1 XX=01; pod 10 XX=0A; pod 20 XX=14).

16. Install and test a virtual machine using the following values:

    - Cluster **testclu*X*/testdc*X*** (where *X* is your pod number)
    - Name: **rhel*X*** (where *X* is your pod number)
    - Description: **First RHEL Guest**
    - Memory Size: **1 GB**
    - Operating System: **Red Hat Enterprise Linux 7.x x64**
    - Network Interface: *rhevm (rhevm)*
    - Attach CD: **rhel-server-7.1-x86_64-boot.iso**
    - Boot Sequence: **CD-ROM (first)**
    - Display Protocol: **Spice**
    - Kickstart file: **http://classroom.example.com/materials/small-rhev.cfg**

# Solution

In this comprehensive review, you will build a Red Hat Enterprise Virtualization environment.

**Outcome(s)**

- You should be able to create and configure a RHEV environment. You should be able to test the environment by creating a virtual machine.

**Before you begin**

Reset your virtual systems, **workstation**, **rhevm**, **servera**, and **serverb**.

> ## Note
>
> If running a physical **servera**, perform the steps to uninstall it (see earlier unit lab for details).

Perform the following steps:

1. Verify hardware requirements of engine (**rhevm.pod*X*.example.com**) and host (**servera.pod*X*.example.com**) systems.

   1.1. From workstation, SSH into rhevm, and run the following commands to ensure hardware requirements are met:

```
[root@rhevm ~]# free -m
             total       used       free     shared    buffers     cached
Mem:          3832       3396        436         35         44        930
-/+ buffers/cache:       2421       1410
Swap:            0          0          0

[root@rhevm ~]# ip addr show
...
2: eth0:
    inet 172.25.X.15/24 brd 172.25.0.255 scope global eth0
    inet6 fe80::5054:ff:fe00:f/64 scope link
...
```

   Run the **hostname** command to see what your machine's host name is:

```
[root@rhevm ~]# hostname
rhevm.podX.example.comm
```

   Type **exit** to exit the shell.

   1.2. Since servera is no longer currently installed, we will hope for the best.

2. Subscribe the RHEV-M machine to the appropriate channels.

   Use the provided **http://classroom.example.com/materials/rhevm.repo** file.

   2.1. From rhevm, retrieve the repository file that contains the same package as the ones channels deliver:

```
[root@rhevm ~]#  wget http://classroom.example.com/materials/rhevm.repo -P /etc/
 yum.repos.d/
```

2.2. Update the system, then reboot if necessary:

```
[root@rhevm ~]# yum update -y
```

Reboot the system if there was a new kernel.

3. Install the *rhevm*, *rhevm-dwh*, and *rhevm-reports* packages.

```
[root@rhevm ~]# yum -y install rhevm rhevm-dwh rhevm-reports
```

4. Configure RHEV-M with **engine-setup**. Generate an answer file named **/root/answers.txt**.

| Prompt | Answer |
|---|---|
| **Configure Engine on this host** | yes |
| **Configure Data Warehouse on this host** | yes |
| **Configure Reports on this host** | yes |
| **WebSocket Proxy** | yes |
| **Do you want Setup to configure the firewall?** | no |
| **Host fully qualified DNS name of this server** | rhevm.pod*X*.example.com |
| **Where is the Reports database located?** | local |
| **Would you like Setup to automatically configure postgresql and create Reports database, or prefer to perform that manually? (Automatic, Manual)** | automatic |
| **Where is the DWH database located?** | local |
| **Would you like Setup to automatically configure postgresql and create DWH database, or prefer to perform that manually?** | automatic |
| **Where is the Engine database located? (Local, Remote) [Local]:** | local |
| **Would you like Setup to automatically configure postgresql and create Engine database, or prefer to perform that manually?** | automatic |
| **Engine admin password (confirm)** | redhat |
| **Use weak password?** | yes |
| **Application mode (Virt, Gluster, Both)** | both |
| **Organization name for certificate** | pod*X*.example.com |
| **Do you wish to set the application as the default page of the web server?** | yes |

| Prompt | Answer |
|---|---|
| **Do you wish Setup to configure that, or prefer to perform that manually?** | automatic |
| **Configure an NFS share on this server to be used as an ISO Domain?** | yes |
| **Local ISO domain path** | /exports/rhevisos |
| **Local ISO domain ACL** | rhevm.pod*X*.example.com(rw) |
| **Local ISO domain name (make sure to replace X by your station number)** | iso*X* |
| **Reports power users password (confirm the password)** | redhat |
| **Use weak password?** | yes |
| **Would you like transactions from the Red Hat Access Plugin sent from the RHEV Manager to be brokered through a proxy server?** | no |

4.1.

```
[root@rhevm ~]# engine-setup --generate-answer=/root/answers.txt
```

5. Bind to the IdM authentication domain, **example.com**, using **rhevadmin** and **redhat** as credentials.

5.1.

```
[root@rhevm ~]# engine-manage-domains add --domain=example.com --user=rhevadmin
 --provider=IPA
oVirt Engine restart is required in order for the changes to take place (service
 ovirt-engine restart).
Manage Domains completed successfully
```

```
[root@rhevm ~]# service ovirt-engine restart
```

```
[root@rhevm ~]# engine-manage-domains list
Domain: example.com
        User: rhevadmin@EXAMPLE.COM
Manage Domains completed successfully
```

6. Configure your web browser certificate.

6.1. Start a new **Firefox** instance on **workstation.podX.example.com** and navigate to **https://rhevm.pod*X*.example.com**. Click **Administration Portal** to navigate to the RHEV-M Administration Portal. On the error page presented by **Firefox**:

- Click **I Understand the Risks** to expand the certificate exception dialog in Firefox.
- Select **Add Exception...**.
- In the resulting pop-up, click the **Get Certificate** button.
- Select **Confirm Security Exception** at the bottom of the pop-up window.

6.2. You should be automatically redirected to the RHEV **Administration Portal**.

7. Create a new datacenter with the following values:

   - Name: **`testdc`*`X`*** (where *X* is your pod number)
   - Description: **`Test Datacenter`**
   - Storage Type: **`Shared`**
   - Compatibility Version: **`3.5`**
   - Quota Mode: **`Disabled`**

   7.1. Navigate to Datacenters

   Click the **Data Centers** tab.

   7.2. Click the **New** button. The “New Data Center” dialog displays.

   Complete the information using the previous values.

   Click **OK**.

   If the “Guide Me” dialog opens, displaying a list of configuration tasks, click **Configure Later** to close the dialog.

8. Create a new cluster with the following values:

   - Data Center: **`testdc`*`X`*** (where *X* is your pod number)
   - Name: **`testclu`*`X`*** (where *X* is your pod number)
   - Description: **`Test Cluster`**
   - CPU Name: *`Depends on classroom hardware`*
   - Compatibility Version: **`3.5`**
   - For storage: **`Enable Virt Service`**
   - Memory Optimization: **`None`**
   - Resilience Policy: **`Migrate Virtual Machines`**

   8.1. Click the **Clusters** tab.

   8.2. Click the **New** button. The “New Cluster” dialog displays.

   Complete the information using the previous values.

   Click **OK**. The “Guide Me” dialog opens, displaying a list of configuration tasks.

   Click **Configure Later** to close the dialog.

9. Install, configure, and approve a RHEV-H host on **`servera`**.

   - RHEV-H host administrator password: **`redhat123`**

- Network, Hostname: **servera.pod*X*.example.com**; DNS Server 1: **172.25.254.254**; NTP Server 1: **172.25.254.254**
- Interface: **Static**; IP Address: **172.25.*X*.10**; Netmask: **255.255.255.0**; Gateway: **172.25.*X*.254**
- Enable ssh pasword authentication
- Register to **rhevm.pod*X*.example.com**

9.1. Reboot **servera** and select the PXE boot option when the BIOS screens appear. When the PXE menu appears, type **rhevh** at the **boot:** prompt.

```
boot: rhevh
```

9.2. After the RHEV Hypervisor installer boots, select **Install Hypervisor 7.1-*version***, then hit **Enter**.

Select your keyboard layout and press **Enter**.

The internal hard disk drive should be selected when the installer asks for the boot device. Confirm the internal hard drive is selected, then choose **<Continue>** and press **Enter**.

The internal hard disk drive should be selected when the installer asks for the installation device. Confirm the internal hard drive is selected, then choose **<Continue>** and press **Enter**. Accept the default partition sizes, then choose **<Continue>**

9.3. The RHEV-H installer will prompt for the RHEV-H host administrator password. Type **redhat123** in the **Password:** and **Confirm Password:** fields, choose **<Install>**, then press **Enter**.

At this point the hard disk drive will be formatted, followed by the installation of the RHEV Hypervisor. Once the installation finishes, press **Enter** at the **<Reboot>** prompt. Now that the RHEV-H software is installed, it is time to configure the system.

9.4. Once the RHEV Hypervisor system boots, log into the console as **admin** with a password of **redhat123**. A main menu of status and configuration items will appear at the left side of the screen. Each menu item can be selected by using the up and down arrow keys to highlight a choice followed by **Enter**.

9.5. Select **Network** from the main menu. When the network configuration screen appears, set the host name and DNS and NTP settings. Provide the following values:

- **Hostname: servera.pod*X*.example.com**
- **DNS Server 1: 172.25.254.254**
- **NTP Server 1: 172.25.254.254**

Select **<Save>** to confirm your choices.

Next, configure the network interface. Use the **`Tab`** key to move the cursor to the network devices and make sure **`eth0`** is highlighted, then press **`Enter`**.

When the network interface configuration screen appears, choose the **`Static`** radio button for **`IPv4 Settings:`**, then provide the following values:

- **`IP Address: 172.25.`*`X`*`.10`**
- **`Netmask: 255.255.255.0`**
- **`Gateway: 172.25.`*`X`*`.254`**

Select **`<Save>`** to confirm your choices. When the confirmation box appears, select **`Close`**.

9.6. This step is only necessary if you are running this in an online learning environment.

## Important

This step is only necessary if you are running this in an online learning environment since the nested virtualization of the cloud is missing some information needed to run RHEV-H.

Open the virtual keyboard to send **`F2`** to the system to open the Rescue Shell. Select **`OK`** to be presented with a prompt.

```
[root@servera admin]# mkdir -p /config/usr/share/libvirt
[root@servera admin]# wget http://classroom.example.com/materials/
cpu_map.xml -P /config/usr/share/libvirt
[root@servera admin]# echo "/usr/share/libvirt/cpu_map.xml" >> /config/
files
[root@servera admin]# uuidgen -r > /etc/vdsm/vdsm.id
[root@servera admin]# reboot
```

After it finishes booting, log back into the RHEV-H configuration interface.

9.7. Select **`Security`** from the main menu. Navigate to the **`Enable ssh password authentication`** field and hit **`Space`** to toggle it on. Select **`<Save>`**, then select **`Close`** to confirm your settings.

9.8. Select **`RHEV-M`** from the main menu. Specify the values that will allow your RHEV-H host to communicate with your RHEV-M server.

- **`Management Server: rhevm.pod`*`X`*`.example.com`**
- **`Management Server Port: 443`**

Fill in the fields with the provided information. Select **`< Save & Register <`** to save your changes.

When the RHEV-H host contacts the RHEV-M server, it should display a digital fingerprint. Select **`<Accept>`** to continue the RHEV-M registration. Select **`Close`** once the registration succeeds.

Select **Status** from the main menu, then select **<Log Off>** to exit the configuration tool and get back to the RHEV-H login prompt.

9.9. Log into your RHEV-M web administration interface as **admin** with a password of **redhat** in the **internal** domain. Select the **Hosts** tab. The host should appear with a status of **Pending Approval**. Click **servera.pod*X*.example.com** to highlight it and then click the **Approve** button. Review the **Edit & Approve Host** window. Click **OK**.

On the resulting **Power Management Configuration** window, select **OK** to skip configuring power management.

The host should change state to **Installing**, then quickly progress through **Unassigned** to **Up**.

If the RHEV-H host enters a **Non-Operational state**, reboot the RHEV-H host and check its BIOS settings. Make sure the "Execute Disable Memory Protection Technology" is enabled.

10. Configure NFS Exports of **/exports/testdata**, **/exports/testexport**, and **/exports/rhevisos** on RHEV-M

10.1. As **root** on **rhevm**, create a directory named **/exports/testdata**, as well as one named **/exports/testexport**, with ownership UID:GID of **36:36**.

```
[root@rhevm ~]# su -
Password: redhat
[root@rhevm ~]# mkdir /exports/testdata
[root@rhevm ~]# mkdir /exports/testexport
[root@rhevm ~]# chown 36:36 /exports/test*
```

10.2. Configure the NFS server to share the new directories to the local network with writable access.

The entire **echo** command should be on a single line. Be careful to use a *pair* of > "greater-than" signs to append to the file, or use an editor to add the data inside the quoted text.

```
[root@rhevm ~]# echo "/exports/testdata
 172.25.0.0/255.255.0.0(rw,sync,no_root_squash)
 172.25.0.0/255.255.255.0(rw,sync,no_root_squash)" >> /etc/exports

[root@rhevm ~]# echo "/exports/testexport
 172.25.0.0/255.255.0.0(rw,sync,no_root_squash)
 172.25.0.0/255.255.255.0(rw,sync,no_root_squash)" >> /etc/exports

[root@rhevm ~]# echo "/exports/rhevisos
 172.25.0.0/255.255.0.0(rw,sync,no_root_squash)
 172.25.0.0/255.255.255.0(rw,sync,no_root_squash)" >> /etc/exports

[root@rhevm ~]# exportfs -r
```

11. Create a data domain with the following values:

- Name: **testdata*X*** (where *X* is your pod number)

- Data Center: **`testdcX`** (where *X* is your pod number)
- Domain Function/Storage Type: **`Data / NFS`**
- Use Host: **`servera.podX.example.com`**
- Export path: **`rhevm.podX.example.com:/exports/testdata`**

11.1. Create a new data storage domain with the preceding values.

Click the **New Domain** button. The "New Domain" dialog displays.

Complete the information using the values mentioned previously.

Click **OK**.

**Patiently wait for the storage to transition to "Active" since this is the first storage domain and the "Master".**

12. Create an export domain with the following values.

- Name: **`testexportX`** (where *X* is your pod number)
- Data Center: **`testdcX`** (where *X* is your pod number)
- Domain Function/Storage Type: **`Export / NFS`**
- Use Host: **`servera.podX.example.com`**
- Export path: **`rhevm.podX.example.com:/exports/testexport`**

12.1. Still in **System**, click the **Storage** tab.

12.2. Click the **New Domain** button. The "New Domain" dialog displays.

Complete the information using the values mentioned previously.

Click **OK**.

**Patiently wait for the storage to transition to "Active".**

13. Attach the ISO domain (**`isoX`**) and upload the RHEL 7.1 boot iso available at **`http://classroom.example.com/materials/`**.

13.1. Navigate to **System**, then click the **Storage** tab.

13.2. Click the domain **`isoX`**. In the lower pane, select the **Data Center** tab.

13.3. Click the **Attach** link in order to attach the domain to the datacenter. In the window that appears, check the datacenter **`testdcX`**. Click **OK** to save your changes.

13.4. Using the **`engine-iso-upload`** command, upload a new ISO within RHEV.

a. From workstation, SSH into rhevm, using **`redhat`** for the password:

```
[root@workstation ~]# ssh root@rhevm.podX.example.com
```

b. Retrieve the ISO using **wget**:

```
[root@rhevm ~]# wget http://classroom.example.com/materials/rhel-
server-7.1-x86_64-boot.iso
```

c. Upload the image in the ***isoX*** datastore:

```
[root@rhevm ~]# engine-iso-uploader -i isoX upload rhel-server-7.1-x86_64-
boot.iso
Please provide the REST API password for the admin@internal oVirt Engine
 user (CTRL+D to abort): redhat
Uploading, please wait...
INFO: Start uploading rhel-server-7.1-x86_64-boot.iso
...
INFO: rhel-server-7.1-x86_64-boot.iso uploaded successfully
```

14. Create a VM network with the following values:

| Tab | Field | Value |
|---|---|---|
| General | Name | **VMNet** |
| | Description | **VM Network** |
| | VM Network | **Checked** |
| | MTU | **Default (1500)** |
| Cluster | testclu*X* | **Attached** |
| | testclu*X* | **Not Required** |
| vNIC Profiles | Public | **Checked** |

14.1. From the left frame, navigate to **System** > **Data Centers** > **testdc*X***, then click the **Networks** entry.

14.2. From the right frame, click the tab **Networks** to access the network configuration for the datacenter. Click **New** to create a new network.

14.3. On the window that appears, fill in the fields using the preceding table.

14.4. Click **OK** to create the network.

15. Define the range of MAC addresses for the virtual machines that is being offered by the DHCP server: **52:54:00:00:XX:10-52:54:00:00:XX:FF** (where **XX** is the two-digit hexadecimal value of your pod number, pod 1 XX=01; pod 10 XX=0A; pod 20 XX=14).

15.1. Change the **MacPoolRange**.

```
[root@rhevm ~]# engine-config -s
 MacPoolRanges=52:54:00:00:XX:10-52:54:00:00:XX:FF
```

15.2. Restart **ovirt-engine**.

```
[root@rhevm ~]# service ovirt-engine restart
```

16. Install and test a virtual machine using the following values:

    - Cluster **testcluX/testdcX** (where *X* is your pod number)
    - Name: **rhelX** (where *X* is your pod number)
    - Description: **First RHEL Guest**
    - Memory Size: **1 GB**
    - Operating System: **Red Hat Enterprise Linux 7.x x64**
    - Network Interface: *rhevm (rhevm)*
    - Attach CD: **rhel-server-7.1-x86_64-boot.iso**
    - Boot Sequence: **CD-ROM (first)**
    - Display Protocol: **Spice**
    - Kickstart file: **http://classroom.example.com/materials/small-rhev.cfg**

    16.1. Navigate to Virtual Machines
    Click the **Virtual Machines** tab.

    16.2. Create a New Red Hat Enterprise Linux VM

    a. Click the **New VM** button. The **New Virtual Machine** dialog displays. For the **Memory Size**, click the **Show Advanced Options** button, then the **System** tab.

    b. Complete the information using the previous values, leaving the other values at their defaults.

    c. Click **OK**. The **Guide Me** dialog opens, displaying a list of configuration tasks.

    d. Click **Configure Virtual Disks** button to add a 5 GB virtual disk with default values for other fields, then click **OK**.

    e. Click **Configure Later** to close the dialog.

    16.3. Install New Red Hat Enterprise Linux VM
    Back in the web console, verify the **rhelX** virtual machine has transitioned from **Image Locked** to **Down**.

    Launch the installer by modifying the following properties:

    a. Right-click the **rhelX** virtual machine and choose **Run Once** to display the **Run Virtual Machine** dialog.

    b. Configure according to the previous properties, then click **OK** button to begin installation.

c. Once the virtual machine has started and the console becomes available, click it to start the Red Hat Enterprise Linux installation.

d. With **Install or upgrade an existing system** highlighted, press the **Tab** key to edit the installer options.

e. Add, as a space-separated argument, **`ks=http://classroom.example.com/materials/small-rhev.cfg`** to the kernel command line. The cursor should be automatically positioned to allow you to add this at the end of the existing kernel arguments.

f. After the installation completes, the virtual machine will automatically power off. Using the RHEV Administration Portal, start the machine.

16.4. Test the Red Hat Desktop Virtual Machine

After the machine is booted, log in with the **`student`** user account with a password of **`student`**.

# Comprehensive Review: Troubleshooting

In this comprehensive review, you will troubleshoot various RHEV components.

**Outcome(s)**

- You should be able to check time on the engine, as well as the hypervisor; check the status of the storage domains; and review logical networks.

**Before you begin**

Make sure you completed all the steps from the previous unit before moving forward.

Perform the following steps:

1. Authentication
   1.1. Make sure that the RHEV-M instance is tied to all the domains you wish to use.

   1.2. Make sure that the time on all machines is set correctly and that NTP is configured on all hosts.

2. Web Interface
   2.1. Make sure the **`ovirt-engine`** service is running on your **`rhevm.podX.example.com`** machine.

   2.2. Make sure that the RHEV-M CA certificate has been properly imported into the web browser.

3. Storage
   3.1. Make sure your datacenter has active domains.

   3.2. Make sure your datacenter has at least one active cluster.

   3.3. Make sure your cluster has at least one active host.

   3.4. Make sure the host is active (not in **Maintenance** or **Contending** mode).

4. RHEV-H
   4.1. Make sure that you are pointing to the correct RHEV-M host.

   4.2. Make sure that networking is configured correctly.

   4.3. Make sure that all storage you are using is accessible from the RHEV-H host.

   4.4. Make sure that all *Logical Networks* configured in your cluster are configured on your host.

# Solution

In this comprehensive review, you will troubleshoot various RHEV components.

**Outcome(s)**

- You should be able to check time on the engine, as well as the hypervisor; check the status of the storage domains; and review logical networks.

**Before you begin**

Make sure you completed all the steps from the previous unit before moving forward.

Perform the following steps:

1. Authentication

   1.1. Make sure that the RHEV-M instance is tied to all the domains you wish to use.

```
[root@rhevm ~]# engine-manage-domains list
```

   1.2. Make sure that the time on all machines is set correctly and that NTP is configured on all hosts.

   From rhevm, as well as servera, run the following commands:

   - To verify time:

```
[root@rhevm ~]# date
```

```
[root@servera ~]# date
```

   - To verify the NTP server from servera, open a terminal from workstation, then log into the server, using **admin** as username and **redhat123** as password. Once logged in, navigate to **Network** to ensure the NTP server is set **172.25.254.254**.

   - Navigate to **Status** in order to be able to **Log Off**.

2. Web Interface

   2.1. Make sure the **ovirt-engine** service is running on your **rhevm.pod*X*.example.com** machine.

   From workstation, SSH into rhevm, then run the following command. Pay attention to the output:

```
[root@rhevm ~]# service ovirt-engine status
ovirt-engine (pid  7387) is running...
```

   2.2. Make sure that the RHEV-M CA certificate has been properly imported into the web browser.

   Still on workstation, open Firefox, then navigate to **Edit** > **Preferences** > **Advanced** > **Certificates**, in order to list all installed certificates. Click **View Certificates**.

Click the **Others** button, and make sure certificates for **`rhevm.podX.example.com`** are listed.

3. Storage
   3.1. Make sure your datacenter has active domains.

   From the RHEV-M web interface, navigate to **System** > **testdc*X*** > **Storage**. On the right panel, make sure the following domains are listed, and that their status is set to **`Active`**:

   - **`isoX`**
   - **`testdataX`**
   - **`testexportX`**

   3.2. Make sure your datacenter has at least one active cluster.

   Click the **Clusters** link from the left panel. Make sure the cluster **`testcluX`** is listed.

   3.3. Make sure your cluster has at least one active host.

   Still in the **Clusters** tab, click the cluster. From the lower panel, click the **Hosts** tab to list all attached hosts. **`servera.podX.example.com`** should be listed.

   3.4. Make sure the host is active (not in **Maintenance** or **Contending** mode).

   From the left panel, navigate to **Clusters** > **testclu*X*** > **Hosts**, then click the host **`servera.podX.example.com`**.

   From the right frame, make sure the **`Status`** for the host is set to **`Up`**.

4. RHEV-H
   4.1. Make sure that you are pointing to the correct RHEV-M host.

   From workstation, open a terminal and SSH into servera, using **`admin`** as username and **`redhat123`** as password. Once logged in, select the entry **`RHEV-M`** from the console-based interface.

   Make sure the **Management Server** is set to **`rhevm.podX.example.com`**, and the port set to **`443`**. Hit **`Esc`** on your keyboard to exit the console.

   4.2. Make sure that networking is configured correctly.

   From the RHEV-M web-interface, navigate to **System** > **Hosts**, then click the host **`servera.podX.example.com`**. From the lower panel, click the **Network Interfaces** tab, and make sure the interface **`eth0`** is **`Up`**, and has the IP address **`172.25.X.10`** for the **`rhevm`** network.

   4.3. Make sure that all storage you are using is accessible from the RHEV-H host.

   From the RHEV-M web interface, click **System** from the left panel. Select the **Storage** tab in order to list all available storage domains.

For each domain listed, make sure the **Use Host** is set to **rhevm.pod*X*.example.com**. To do so, click **Edit** to edit the selected domain. Repeat this step for the **data** domain and the **iso** domain, as well as the **export** domain.

4.4. Make sure that all *Logical Networks* configured in your cluster are configured on your host.

Still in the RHEV-M web interface, navigate to **System** > **Networks** to list all available networks. Select the network **rhevm**, attached to the data center **testdc*X***. Click **Hosts** from the lower panel in order to view the attachment. Ensure the network is attached to **eth0** for the **servera.pod*X*.example.com**, with a status of **Up**. Click the network **VMNet**; from the **Hosts** tab, ensure the network is attached to **eth1** for the **servera.pod*X*.example.com** with a status of **Up**.

# Comprehensive Review: Management

In this comprehensive review, you will troubleshoot various RHEV components.

**Outcome(s)**

- You should be able to perform various tasks against virtual machines, such as creating a template or a virtual machine pool.

**Before you begin**

Make sure you completed all the steps from the previous unit before moving forward.

Perform the following steps:

1. Upgrade a RHEV-H hypervisor host to the latest available.

2. Create a snapshot of the **rhel*X*** VM.

3. Export the **rhel*X*** VM image, then modify **/etc/issue** with **guestfish** by appending a line:

   ```
   Testing guestfish
   ```

   Import the resulting image and test that the edit succeeded.

4. Create a template called **rhel71-server** based on a new, sealed VM called **rhel*X*-base** (Memory: 1024M; Disk: 4GB, Thin; Network: **rhevm**; Kickstart: **http://classroom.example.com/materials/small-rhev.cfg**).

5. Create and test a pool using the following values:

| Name | Value |
| --- | --- |
| Name | RHELpool |
| Description | Basic RHEL Pool |
| Number of VMs | 2 |
| Based on Template | rhel71-server |

6. Create search bookmark named **important events** that displays events with a severity above normal sorted by time.

7. Save a VM inventory report as a pdf.

8. Create and test a backup.

# Solution

In this comprehensive review, you will troubleshoot various RHEV components.

### Outcome(s)

- You should be able to perform various tasks against virtual machines, such as creating a template or a virtual machine pool.

**Before you begin**

Make sure you completed all the steps from the previous unit before moving forward.

Perform the following steps:

1. Upgrade a RHEV-H hypervisor host to the latest available.

   1.1. Use **ssh** to connect to the **rhevm** virtual machine as **root**.

   1.2. Install the newest *rhev-hypervisor7* package available.

   ```
   [root@rhevm ~]# yum -y install rhev-hypervisor7
   ```

   1.3. In the RHEV-M management console, select the **Hosts** tab and select the host **servera.pod*X*.example.com**. Click the **Maintenance** button, then click **OK**.

   1.4. Right-click the host, then select the option **Upgrade**.

   An **Install Host** window will open. The new hypervisor ISO file should be listed in the pulldown menu next to the **RHEV-H-ISO Name** option. Select the **rhevh-7.1-*latest-version*.iso**, then click **OK**.

   The host **Status** moves from **Maintenance** to **Installing**, and after some time changes to **Reboot**. When the status finally changes to **Up**, the upgrade process has finished.

> **Note**
>
> It takes a considerable time after a RHEV-H host reboots for RHEV-M to update its status and show it in the **Up** state.
>
> The default time that RHEV-M waits for the host to reboot is 300 seconds (as seen in **/var/log/ovirt-engine/engine.log**). This can be changed using the **engine-config** command to change the **ServerRebootTimeout** key.

   1.5. When the host has finished, new information about the operating system, kernel, etc. will be updated in the **Details** pane below the **General** tab.

2. Create a snapshot of the **rhel*X*** VM.

   2.1. Log into the RHEV-M web administration portal as the user **rhevadmin** with a password of **redhat**.

In the login dialog box that appears, type **`rhevadmin`** in the **User Name** box, type **`redhat`** in the **Password** box, and make sure **`example.com`** is selected in the **Domain** box. Click the **Login** button to confirm your choices and authenticate.

2.2. From the left panel, click **System**. From the right panel, click the **Virtual Machines** tab and check the status of **`rhelX`**. If the virtual machine is not in the **Down** state, right-click the virtual machine, and select the **Shutdown** menu item. Click the **OK** button in the confirmation dialog box that appears.

2.3. Once **`rhelX`** is down, click the virtual machine name again to highlight it. In the lower pane for the virtual machine, select the **Snapshots** tab, and click the **Create** button.

2.4. In the **Create Snapshot** window that appears, enter **`Baseline Snapshot`** in the **Description** text box. Click **OK** to confirm.

2.5. When the snapshot is complete, right-click **`rhelX`** and select **Run** to boot the virtual machine.

3. Export the **`rhelX`** VM image, then modify **`/etc/issue`** with **`guestfish`** by appending a line:

```
Testing guestfish
```

Import the resulting image and test that the edit succeeded.

3.1. From workstation, open Firefox, and log into the RHEV-M web interface using **`admin`** as username, and **`redhat123`** as password. Once logged in, click **System**.

3.2. Shut down the **`rhelX`** virtual machine by browsing to the **Virtual Machines** tab. Right-click **`rhelX`** and choose **Shutdown**.

When the machine has finished shutting down, export the **`rhelX`** virtual machine by right-clicking the **`rhelX`** virtual machine and choosing **Export**. Select the **Collapse Snapshots** checkbox and press **OK**. Wait for the export to complete.

3.3. Detach the **`testexportX`** export domain by navigating to the **Data Centers** tab. Select the **`testdcX`** datacenter. In the bottom pane, right-click the new export domain (**`testexportX`**) and choose **Maintenance**. Once the export domain is in maintenance mode, right-click the new export domain and choose **Detach**. Click **OK** to confirm.

To remove the **`testexportX`** export domain, navigate to the **Storage** tab. Right-click the **`testexportX`** storage domain, and select **Remove**. Click the **OK** button (do *not* check the **Format Domain** box).

3.4. Once the virtual machine has been successfully exported, proceed to remove the virtual machine. In the **Virtual Machines** tab, right-click the **`rhelX`** virtual machine and select **Remove**. Click **OK** to confirm.

3.5. Log in as **`root`** on your **`rhevm.podX.example.com`** machine. Install the necessary packages to use the **`guestfish`** utility.

```
[root@rhevm ~]# yum -y install libguestfs-tools
```

3.6. Mount the virtual machine export image so it can be updated.

```
[root@rhevm ~]# guestfish -i -w -a /exports/testexport/EXPORT-UUID/images/VM-UUID/DISK-UUID
Welcome to guestfish, the libguestfs filesystem interactive shell for
editing virtual machine filesystems.

Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell

Operating system: Red Hat Enterprise Linux Server release 6.4 (Santiago)
/dev/vda2 mounted on /
/dev/vda1 mounted on /boot

><fs>
```

3.7. Modify the **/etc/issue** file in the image so it contains a distinctive message that makes it clear that it has been modified. For instance, add a line to the bottom of the file that reads: **Testing guestfish**. When you are finished, disconnect from the disk image.

```
><fs> vi /etc/issue
><fs> exit
```

3.8. Import the export domain by browsing to the RHEV-M web interface, and choosing the **Storage** tab. Click the **Import Domain** button. Verify the **Domain Function / Storage Type** is set to **Export / NFS**. For the **Export Path**, enter **rhevm.pod*X*.example.com:/exports/testexport** and click the **OK** button.

3.9. To activate the export domain, navigate to the **Data Centers** tab and highlight the testdc*X* datacenter. In the bottom pane, right-click the **testexport*X*** export domain, and choose **Activate**.

3.10. Back in the **Storage** tab, highlight the **testexport*X*** export domain. In the bottom pane, navigate to the **VM Import** tab. Right-click the **rhel*X*** virtual machine and select **Import**. Leave the settings at their defaults and press **OK** to confirm.

3.11. Verify the import and edit worked. Once the import has completed, browse to the **Virtual Machines** tab. Right-click **rhel*X*** and select **Run** to start the virtual machine. Open a console and verify that the changes can be seen from the login prompt.

4. Create a template called **rhel71-server** based on a new, sealed VM called **rhel*X*-base** (Memory: 1024M; Disk: 4GB, Thin; Network: **rhevm**; Kickstart: **http://classroom.example.com/materials/small-rhev.cfg**).

4.1. Still in the RHEV-M web interface, select the **Virtual Machines** tab.

4.2. Click the **New VM** button to create a new virtual machine. Set the **Name** to **rhel*X*-base**, the **Description** to **RHEL base**, the **Memory Size** to **1024 MB**, the **Operating System** to **Red Hat Enterprise Linux 7.x x64**, and the **Network** to **rhevm**. Keep all the other settings at their defaults. Click **OK** to continue.

4.3. In the **Guide Me** window that appears, add one 4 GB thin-provisioned **VirtIO** disk. Keep all settings at their defaults. Select **Configure Later** to continue.

4.4. Start your new **rhel*X*-base** machine, and open the console. Since the hard disk is still empty, the system will come up to a PXE boot. Initiate a **kickstart** installation by typing:

```
boot: linux ks=http://classroom.example.com/materials/small-rhev.cfg
```

4.5. Wait for the install to complete. This should only take a few minutes.

4.6. Once the system has finished rebooting and is at a login prompt, log in as **root** with a password of **redhat**.

4.7. To make the system ready for imaging, manually remove all host-specific information. Make sure to remove at least the following:

- All references to MAC addresses.
- All **SSH** keys.
- Change the host name to **localhost** so that it can be set automatically by the networking scripts.
- The RHN **systemid** file associated with this system.

To remove all references to MAC addresses, we must do two things: remove **/etc/udev/rules.d/*-persistent-*.rules** and remove all **HWADDR** lines from **/etc/sysconfig/network-scripts/ifcfg-***. To do so, execute the following commands:

```
[root@vmY ~]# rm -f /etc/udev/rules.d/*-persistent-*.rules
[root@vmY ~]# sed -i '/^HWADDR=/d' /etc/sysconfig/network-scripts/ifcfg-*
```

To remove all **SSH** keys, we need to delete the **moduli** and **ssh_host*** files from **/etc/ssh/**:

```
[root@vmY ~]# rm -f /etc/ssh/moduli /etc/ssh/ssh_host*
```

To revert the host name back to **localhost** we can run the following command:

```
[root@vmY ~]# hostnamectl set-hostname "localhost"
```

To remove the RHN **systemid** associated with this system, run the following command:

```
[root@vmY ~]# rm -f /etc/sysconfig/rhn/systemid
```

4.8. Power down your **rhel*X*-base** machine by running the **poweroff** command.

4.9. In the **Virtual Machines** tab of the RHEV-M GUI, select your **rhel*X*-base** machine and click **Make Template**. Use **rhel71-server** as the **Name** and **Basic RHEL7.1 Server** as the **Description**.

4.10. Wait for the template to be created. This should only take a couple of minutes.

4.11. Still in the **Virtual Machines** tab, click the **New VM** button. Select the **`rhel71-server`** template you created in the previous step. Use **`One-More-Server`** for the name and **`Testing The Template`** for the description. Click **OK** to complete the machine creation.

4.12. Start your new **`One-More-Server`** machine, and connect to the console. Verify that it has successfully established a network connection by logging in as **`root`** with a password of **`redhat`** and executing the following command:

```
[root@vmY ~]# ping -c3 172.25.254.254
```

4.13.

### Warning

Clean up by shutting down and deleting both **`rhelX-base`** and **`One-More-Server`**. Keep the **`rhel71-server`** template for the next step.

On the **Virtual Machines** tab, highlight the **`rhelX-base`** machine. Press the **Remove** button. Confirm the deletion. Do the same for **`One-More-Server`**.

5. Create and test a pool using the following values:

| Name | Value |
|---|---|
| Name | RHELpool |
| Description | Basic RHEL Pool |
| Number of VMs | 2 |
| Based on Template | rhel71-server |

5.1. Still in the RHEV-M Administration Portal, click **System**.

5.2. Remove the **rhel*X*-base** virtual machine, if it exists. Navigate to the **Virtual Machines** tab. If **rhel*X*-base** is still running, right-click it and select **Power Off**. Click **OK** on the confirmation pop-up. Once the machine is shut down, right-click **rhel*X*-base** and **Remove** it. Answer **OK** to the confirmation pop-up.

5.3. Remove the **rhel*X*** virtual machine. If **rhel*X*** is still running, right-click it and select **Power Off**. Click **OK** on the confirmation pop-up. Once the machine is shut down, right-click **rhel*X*** and **Remove** it. Answer **OK** to the confirmation pop-up.

5.4. Navigate to the **Pools** tab. Click the **New** button to create a new pool. In the **General** tab of the dialog that appears, set the preceding values. Leave default values for all other settings.

Click **OK** to create the pool.

5.5. In the RHEV-M Administration Portal, navigate to the **Virtual Machines** tab. You should see two new virtual machines called *RHELpool-1* and *RHELpool-2*.

5.6. Switch back to the **Pools** tab and select the pool named **`RHELpool`**. In the bottom pane, switch to the **Virtual Machines** tab and select the machine named **`RHELpool-2`**. Right-click the machine name and choose **Detach** to detach this machine from the pool. Click **OK** in the dialog that pops up to confirm the operation.

5.7. Switch back to the top pane and select the **Virtual Machines** tab. Notice how **`RHELpool-2`** now has the normal desktop icon next to its name instead of the pool icon. Select **`RHELpool-2`**, then right-click and choose **Remove**. Click the **OK** button to confirm the removal. Wait for the removal to complete.

5.8. Switch to the **Pools** tab and select **`RHELpool`**. Click the **Edit** button to bring up the **Edit Pool** dialog. Change the **Increase number of VMs in pool by** value to **`1`** and click the **OK** button. You should see a new **`RHELpool-2`** machine being created in the **Virtual Machines** tab.

6. Create search bookmark named **`important events`** that displays events with a severity above normal sorted by time.

6.1. In the RHEV-M Administration Portal, click the **Virtual Machines** tab. Look at the **Search:** text box at the top of the window. It should contain the search **`Vms:`**, which means to display all virtual machines in the RHEV-M environment.

6.2. Replace **`Vms:`** with **`Vms: sortby uptime desc`**. Note the dropdown autocompletion that occurs as you type. Hit **`Enter`** if you have not already done so. This should again display all virtual machines in the RHEV environment, but this time it will sort them by descending uptime; machines up the shortest will be on the bottom of the list.

6.3. Now try another search. In the search bar, replace the current search string with **`Events: severity > normal sortby time`**. This will display all events with a severity greater than "normal" sorted in chronological order.

6.4. This seems like a search that you may want to run frequently, yet it is a handful to type. Set a bookmark for the search. Click the star icon to the right of the search bar. A window should open; for **Name:**, use **`important events`**. Leave the **Search string:** alone since it should already match what you have in the search bar. Click **OK**.

6.5. In the **Bookmarks** tab in the left pane of the RHEV-M interface, you should see **`important events`** displayed. In the search bar, replace the current search with **`Vms:`**. You should see a list of all your virtual machines. Now, test your bookmark. Click the **`important events`** bookmark under the **Bookmarks** tab. You should see the results of your bookmarked search in the main window.

7. Save a VM inventory report as a pdf.

7.1. Launch the **Firefox** web browser on **`workstation`**. Go to the following link: **`https://rhevm.podX.example.com/rhevm-reports`**. Authenticate with a user ID of **`rhevadmin`** and a password of **`redhat`**.

7.2. Select **View** menu in the top navigation bar, and in the pop-up menu, choose **Repository** to open the repository.

7.3. Open the **RHEVM Reports** folder in the left pane and select the **System Overview Dashboard** in the **Dashboards** tree. Review the information displayed on the dashboard. When you are finished, navigate to the **Library** menu item in the top navigation bar.

7.4. Select the **`Hosts Inventory (BR1)`** report and click **OK** with the default report parameters to display a list of RHEV-H hosts. When you are finished, navigate to the **Library** menu item in the top navigation bar.

7.5. Select the **`Virtual Machines Inventory (BR9)`** link and click **OK** with the default report parameters to display a list of virtual machines. Hover the **Export** icon just to the right of the **Save** button. Choose **As PDF** in the dropdown menu to save the PDF report to a file and view it with the browser.

8. Create and test a backup.

8.1. From workstation, log into your rhevm server:

```
[root@workstation ~]# ssh root@rhevm
```

8.2. Before backing up, make sure to stop the **`ovirt-engine-dwhd`** daemon:

```
[root@rhevm ~]# /etc/init.d/ovirt-engine-dwhd stop
```

8.3. Using the command **`engine-backup`**, create a backup of the existing installation:

```
[root@rhevm ~]# engine-backup --scope=all --mode=backup --log=/root/rhevm-
backup.log --file=/root/rhevm-backup.tar
```

8.4. Run the **`engine-cleanup`** command in order to remove the existing RHEV setup:

```
[root@rhevm ~]# engine-cleanup
[ INFO  ] Stage: Initializing
... Output Omitted ...

[ INFO  ] Stage: Termination
[ INFO  ] Execution of cleanup completed successfully
```

8.5. Restore the backup, using the **`engine-backup`** command:

```
[root@rhevm ~]# engine-backup --mode=restore --file=/root/rhevm-backup.tar --
log=/root/rhevm-restore.log
Preparing to restore:
- Unpacking file '/root/rhevm-backup.tar'
Restoring:
- Files
- Engine database 'engine'
- DWH database 'ovirt_engine_history'
- Reports database 'ovirt_engine_reports'
You should now run engine-setup.
Done.
```

8.6. As indicated by the previous output, run the **engine-setup**. Use the answer file you generated in order to speed up the restore. When prompted, do not back up the existing database, as it is not necessary:

```
[root@rhevm ~]# engine-setup --config-append=/root/answers.txt
[ INFO  ] Stage: Initializing
... Output omitted...
[ INFO ] Execution of setup completed successfully
```

8.7. Optionally, you can run the following commands to join back the **example.com** domain:

```
[root@rhevm ~]# engine-manage-domains add --domain=example.com --user=rhevadmin
 --provider=IPA
oVirt Engine restart is required in order for the changes to take place (service
 ovirt-engine restart).
Manage Domains completed successfully
```

Use **redhat** when prompted for a password. After this command has finished, restart RHEV-M to activate the changes:

```
[root@rhevm ~]# service ovirt-engine restart
```

8.8. From workstation, open Firefox, and access rhevm.pod*X*.example.com. Since the SSL certificate has been regenerated, confirm the security exception.

8.9. Using the domain **internal**, log in with the user **admin** and a password of **redhat**.

8.10. Once logged in, you should see the existing cluster you created earlier.

# Summary

**Comprehensive Review: Installation**

In this section, you learned:

- Ensuring the hardware meets the requirements is the first step before deploying RHEV.
- Various storage domains can be defined, depending on the need: saving images, hosting virtual machines, or exporting data.
- RHEV offers a console-based interface in order to deploy RHEV-H.

**Comprehensive Review: Troubleshooting**

In this section, you learned:

- You can use the console-based interface available for the RHEV hosts in order to configure as well as diagnose a hypervisor.
- It is possible to verify the states of many RHEV resources, such as clusters, datacenters, and storage domains. Various status include **`Up`**, **`Maintenance`**, or **`Active`**.

**Comprehensive Review: Management**

In this section, you learned:

- It is possible to use a running virtual machine as a base for a snapshot. Snapshots can be used to restore a previous state for the virtual machine.
- Templates allows you to publish a virtual machine in order to speed up delivery.
- A virtual machine pool is a group of virtual machines that are all clones of the same template and that can be used on demand by any user in a given group.