



# **Red Hat Enterprise Linux 8.0 Beta**

## **Configuring and managing virtualization**

A guide to managing virtualization on Red Hat Enterprise Linux 8.0 Beta



# Red Hat Enterprise Linux 8.0 Beta Configuring and managing virtualization

---

A guide to managing virtualization on Red Hat Enterprise Linux 8.0 Beta

## Legal Notice

Copyright © 2018 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## Abstract

This document describes how to manage virtualization on Red Hat Enterprise Linux 8.0 Beta. In addition to general information about virtualization, it describes how to manage virtualization with command-line tools, as well as with the Cockpit web interface.

# Table of Contents

<b>THIS IS A BETA VERSION!</b> .....	<b>6</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>7</b>
<b>CHAPTER 1. VIRTUALIZATION IN RHEL 8 - AN OVERVIEW</b> .....	<b>8</b>
1.1. VIRTUALIZATION IN RHEL 8 - THE BASICS	8
1.2. ADVANTAGES OF VIRTUALIZATION	8
1.3. VIRTUALIZATION IN RHEL 8 - SOFTWARE COMPONENTS	9
1.4. TOOLS AND INTERFACES FOR VIRTUALIZATION MANAGEMENT IN RHEL 8	9
1.5. RED HAT VIRTUALIZATION SOLUTIONS	11
<b>CHAPTER 2. GETTING STARTED WITH VIRTUALIZATION IN RHEL 8</b> .....	<b>12</b>
2.1. ENABLING VIRTUALIZATION IN RHEL 8	12
Prerequisites	12
Procedure	13
Additional information	13
2.2. CREATING VIRTUAL MACHINES	13
2.2.1. Prerequisites	13
2.2.2. Creating a guest virtual machine using the command-line interface	13
Prerequisites	14
Procedure	14
2.2.3. Creating virtual machines using Cockpit	15
Prerequisites	15
Procedure	15
Additional resources	16
2.3. STARTING VIRTUAL MACHINES	16
2.3.1. Prerequisites	16
2.3.2. Starting a guest virtual machine using the command-line interface	17
2.3.3. Powering up virtual machines in Cockpit	17
Prerequisites	17
Procedure	17
Additional resources	17
2.4. CONNECTING TO VIRTUAL MACHINES	17
2.4.1. Prerequisites	17
2.4.2. Connecting to a guest virtual machine using the command-line interface	17
2.4.3. Viewing the graphical console in Cockpit	18
Prerequisites	18
Procedure	18
Additional Resources	19
2.5. SHUTTING DOWN VIRTUAL MACHINES	19
2.5.1. Shutting down a guest virtual machine using the command-line interface	19
2.5.2. Powering down virtual machines in Cockpit	20
Prerequisites	20
Procedure	20
Additional resources	20
<b>CHAPTER 3. USING COCKPIT FOR MANAGING VIRTUAL MACHINES</b> .....	<b>21</b>
3.1. VIRTUAL MACHINE MANAGEMENT IN COCKPIT	21
3.2. SETTING UP COCKPIT TO MANAGE VIRTUAL MACHINES	21
Prerequisites	21
Procedure	21
3.3. CREATING VIRTUAL MACHINES AND INSTALLING OPERATING SYSTEMS USING COCKPIT	22

3.3.1. Creating virtual machines using Cockpit	22
Prerequisites	22
Procedure	23
Additional resources	23
3.3.2. Installing operating systems using Cockpit	24
Prerequisites	24
Procedure	24
3.3.3. Deleting virtual machines using Cockpit	24
Prerequisites	24
Procedure	24
3.4. POWERING UP AND POWERING DOWN VIRTUAL MACHINES USING COCKPIT	25
3.4.1. Powering up virtual machines in Cockpit	25
Prerequisites	25
Procedure	25
Additional resources	25
3.4.2. Powering down virtual machines in Cockpit	26
Prerequisites	26
Procedure	26
Additional resources	26
3.4.3. Restarting virtual machines using Cockpit	26
Prerequisites	26
Procedure	26
3.4.3.1. Additional resources	27
3.4.4. Sending non-maskable interrupts to virtual machines using Cockpit	27
Prerequisites	27
Procedure	27
3.4.4.1. Additional resources	27
3.5. VIEWING VIRTUAL MACHINE INFORMATION USING COCKPIT	28
3.5.1. Viewing virtual machine information in Cockpit	28
Prerequisites	28
Procedure	28
3.5.1.1. Additional resources	29
3.5.2. Viewing basic virtual machine information in Cockpit	29
Prerequisites	29
Procedure	29
3.5.2.1. Additional resources	30
3.5.3. Viewing virtual machine resource usage in Cockpit	31
Prerequisites	31
Procedure	31
3.5.3.1. Additional resources	31
3.5.4. Viewing virtual machine disk information in Cockpit	32
Prerequisites	32
Procedure	32
3.5.4.1. Additional resources	33
3.5.5. Viewing virtual NIC information in Cockpit	33
Prerequisites	34
Procedure	34
3.5.5.1. Additional resources	35
3.6. MANAGING VIRTUAL CPUS USING COCKPIT	35
3.6.1. Configuring virtual CPUs for virtual machines using Cockpit	35
Prerequisites	35
Procedure	35
3.7. MANAGING VIRTUAL MACHINE DISKS USING COCKPIT	36

3.7.1. Viewing virtual machine disk information in Cockpit	37
Prerequisites	37
Procedure	37
3.7.1.1. Additional resources	38
3.7.2. Adding new disks to virtual machines using Cockpit	38
Prerequisites	38
Procedure	38
Additional resources	40
3.7.3. Attaching existing disks to virtual machines using Cockpit	40
Prerequisites	40
Procedure	40
Additional resources	42
3.7.4. Detaching disks from virtual machines	42
Prerequisites	42
Procedure	43
Additional resources	43
3.8. USING COCKPIT FOR MANAGING VIRTUAL MACHINE VNICS	44
3.8.1. Viewing virtual NIC information in Cockpit	44
Prerequisites	44
Procedure	44
3.8.1.1. Additional resources	45
3.8.2. Connecting virtual NICs in Cockpit	45
Prerequisites	45
Procedure	45
3.8.3. Disconnecting virtual NICs in Cockpit	46
Prerequisites	46
Procedure	46
3.9. VIEWING VIRTUAL MACHINE CONSOLES USING COCKPIT	47
3.9.1. Viewing the virtual machine graphical console in Cockpit	47
Prerequisites	47
Procedure	47
Additional Resources	48
3.9.2. Viewing virtual machine consoles in remote viewers using Cockpit	48
3.9.2.1. Viewing the graphical console in a remote viewer	48
Prerequisites	48
Procedure	49
Additional Resources	51
3.9.2.2. Viewing the graphical console in a remote viewer connecting manually	51
Prerequisites	51
Procedure	52
Additional Resources	52
3.9.3. Viewing the virtual machine serial console in Cockpit	52
Prerequisites	52
Procedure	52
Additional Resources	53
<b>CHAPTER 4. CREATING VIRTUAL MACHINE SNAPSHOTS IN RHEL 8</b>	<b>54</b>
4.1. UNDERSTANDING VIRTUAL MACHINE SNAPSHOTS	54
4.2. CREATING VIRTUAL MACHINE SNAPSHOTS FROM XML FILES	54
4.2.1. Creating virtual machine snapshots using XML files	55
Procedure	55
Additional resources	55
4.3. CREATING VIRTUAL MACHINE SNAPSHOTS ENTERING METADATA ON THE COMMAND LINE	55

4.3.1. Creating virtual machine snapshots entering metadata on the command line	55
Procedure	56
Additional resources	56
4.4. CLONING VIRTUAL MACHINE SNAPSHOTS	56
4.4.1. Cloning virtual machine snapshots	56
Procedure	56
4.4.2. Related information	57
4.5. RELATED INFORMATION	57
<b>CHAPTER 5. UNDERSTANDING VIRTUAL NETWORKING</b> .....	<b>58</b>
5.1. VIRTUAL NETWORKING IN ROUTED MODE	59
Common topologies	59
DMZ	59
Virtual server hosting	60
5.2. VIRTUAL NETWORKING IN BRIDGED MODE	60
Common scenarios	61
Additional resources	62
5.3. VIRTUAL NETWORKING ISOLATED MODE	62
5.4. VIRTUAL NETWORKING NETWORK ADDRESS TRANSLATION	62
5.5. VIRTUAL NETWORKING DNS AND DHCP	63
5.6. VIRTUAL NETWORKING DEFAULT CONFIGURATION	63
<b>CHAPTER 6. FEATURE SUPPORT AND LIMITATIONS IN RHEL 8 VIRTUALIZATION</b> .....	<b>65</b>
6.1. RHEL 8 VIRTUALIZATION SUPPORT - THE BASICS	65
6.2. RECOMMENDED FEATURES IN RHEL 8 VIRTUALIZATION	65
Additional resources	66
6.3. UNSUPPORTED FEATURES IN RHEL 8 VIRTUALIZATION	66
Additional resources	67
6.4. RESOURCE ALLOCATION LIMITS IN RHEL 8 VIRTUALIZATION	67





## **THIS IS A BETA VERSION!**

Thank you for your interest in Red Hat Enterprise Linux 8.0 Beta. Be aware that:

- Beta code should not be used with production data or on production systems.
- Beta does not include a guarantee of support.
- Feedback and bug reports are welcome. Discussions with your account representative, partner contact, and Technical Account Manager (TAM) are also welcome.
- Upgrades to or from a Beta are not supported or recommended.

## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the [Bugzilla](#) website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

# CHAPTER 1. VIRTUALIZATION IN RHEL 8 - AN OVERVIEW

This document provides a general overview of virtualization in RHEL 8: its basics, advantages, components, and other possible virtualization solutions provided by Red Hat.

## 1.1. VIRTUALIZATION IN RHEL 8 - THE BASICS

Red Hat Enterprise Linux 8 (RHEL 8) provides the *virtualization* functionality. This means that with the help of virtualization software, a machine running RHEL 8 can *host* multiple virtual machines (VMs), also referred to as *guests*. Guests use the host's physical hardware and computing resources to run a separate, virtualized operating system as a user-space process on the host's operating system.

In other words, virtualization makes it possible to have operating systems within operating systems.

For more information on what virtualization is, see [the Red Hat Customer Portal](#).

Virtual machines enable you to safely test software configurations and features, run legacy software, or optimize the workload efficiency of your hardware. For more information on the benefits, see [Section 1.2, “Advantages of virtualization”](#).

To try out virtualization in RHEL 8, see [Chapter 2, \*Getting started with virtualization in RHEL 8\*](#).

Note that in addition to RHEL 8 virtualization, Red Hat offers a number of specialized virtualization solutions, each with a different user focus and features. For more information, see [Section 1.5, “Red Hat virtualization solutions”](#)

## 1.2. ADVANTAGES OF VIRTUALIZATION

Using virtual machines has the following benefits in comparison to using physical machines:

- **Flexible and fine-grained allocation of resources**

A virtual guest runs on a host machine, which is usually physical, and physical hardware can also be assigned for the guest to use. However, the allocation of physical resources to the guest is done in software, and is therefore very flexible. A guest uses a configurable fraction of the host memory, CPUs, or storage space, and that configuration can specify very fine-grained resource requests.

For example, what the guest sees as its disk can be represented as a file on the host file system, and the size of that disk is much less constrained than the available sizes for physical disks.

- **Software-controlled configurations**

All of the guest's configuration is saved as data on the host, and under software control. Therefore, a guest can easily be created, removed, cloned, migrated, operated remotely, or connected to remote storage.

In addition, the current state of the guest can be backed up as a snapshot at any time. A snapshot can then be loaded to restore the system to the saved state.

- **Separation from the host**

A guest runs on a virtualized kernel, separate from the host OS. This means that any OS can be installed on a guest, and that even if a guest becomes unstable or is compromised, the host is not affected in any way.

**NOTE**

Not all operating systems are supported as a guest OS in a RHEL 8 host. For details, see [Section 6.2, “Recommended features in RHEL 8 virtualization”](#).

- **Space and cost efficiency**

A single physical machine can host a large number of virtual guests. Therefore, it avoids the need for multiple physical machines to do the same tasks, and thus lowers the space, power, and maintenance requirements associated with physical hardware.

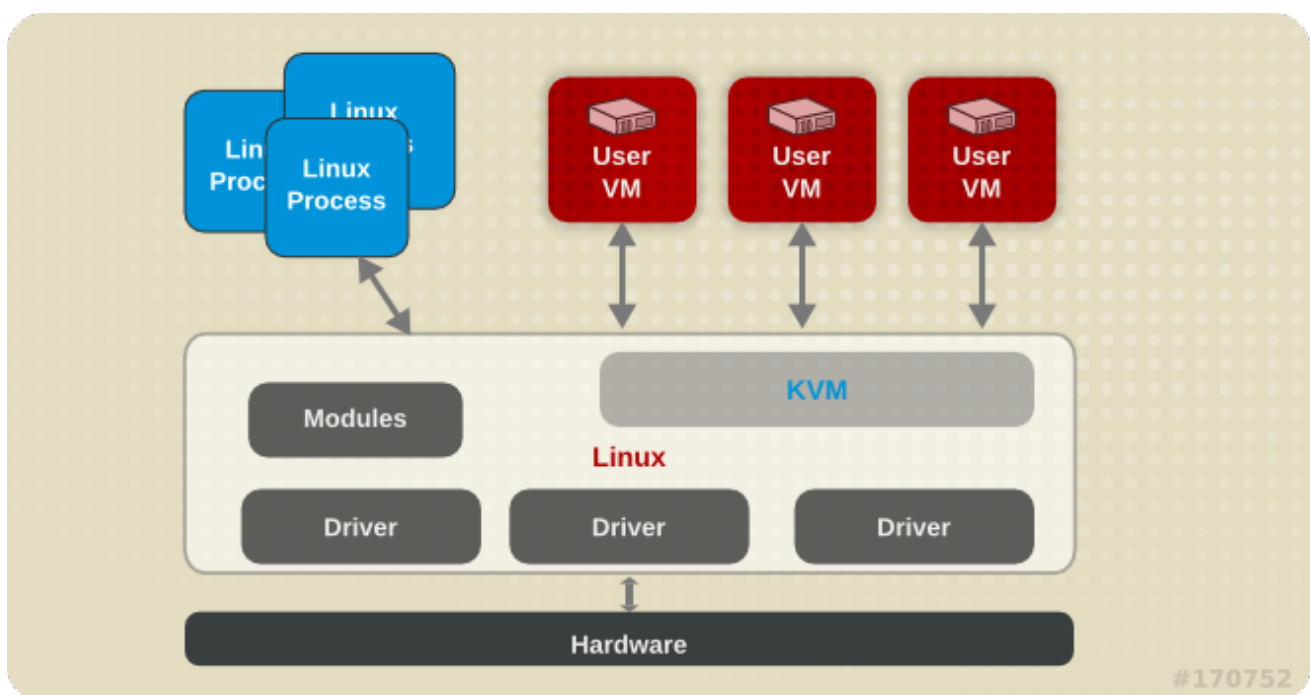
### 1.3. VIRTUALIZATION IN RHEL 8 - SOFTWARE COMPONENTS

The basis of virtualization in RHEL 8 is the **Kernel-based Virtual Machine (KVM)** kernel module and virtualization kernel drivers, such as **virtio** and **vfio**. These components ensure that the Linux kernel on the host machine provides resources for virtualization to user-space software.

At the user-space level, the **QEMU** emulator simulates a complete virtualized hardware platform that the guest operating system can run in, and manages how resources are allocated on the host and presented to the guest. In addition, the **libvirt** software suite serves as a management and communication layer, making QEMU easier to interact with, enforcing security rules, and providing a number of additional tools for configuring and running guests.

As such, libvirt translates input from specialized virtualization tools, such as the **virsh**, **virt-install**, and **guestfish** commands or the Cockpit GUI, into instructions for QEMU. QEMU communicates the instructions to KVM, which ensures that the kernel appropriately assigns the resources necessary to carry out the instructions. As a result, QEMU can execute the corresponding user-space changes, such as creating or modifying a guest, or performing an action in the guest's operating system.

While QEMU is an essential component of the architecture, it is not intended to be used directly on RHEL 8 systems, due to security concerns. Therefore, using **qemu-\*** commands is not supported by Red Hat, and it is highly recommended to interact with QEMU using libvirt.



### 1.4. TOOLS AND INTERFACES FOR VIRTUALIZATION MANAGEMENT IN RHEL 8

The following briefly describes the available methods of managing virtualization in Red Hat Enterprise Linux 8 (RHEL 8).

- The command-line interface (CLI) is the most powerful method of managing virtualization in RHEL 8. Prominent CLI commands for virtualization management include:
  - **virsh** - A versatile virtualization command-line utility and shell with a great variety of purposes, depending on the provided arguments. For example:
    - Starting and shutting down a virtual machine - **virsh start** and **virsh shutdown**
    - Listing available virtual machines - **virsh list**
    - Creating a virtual machine from a configuration file - **virsh create**
    - Entering a virtualization shell - **virsh**

For more information, see the **virsh(1)** man page.

- **virt-install** - A CLI utility for creating new virtual machines. For more information, see the **virt-install(1)** man page.
- **virt-xml** - A utility for editing the configuration of a virtual machine.
- **guestfish** - A utility for examining and modifying virtual machine disk images. For more information, see the **guestfish(1)** man page.

For instructions on basic virtualization management with CLI, see [Chapter 2, Getting started with virtualization in RHEL 8](#).

- Several graphical interfaces can also be used to manage virtualization in RHEL 8:
  - The **Cockpit** web-based application provides a remotely accessible and easy to use graphical user interface for managing virtual machines and virtualization hosts. For instructions on basic virtualization management with Cockpit, see [Chapter 3, Using Cockpit for managing virtual machines](#).
  - The **Virtual Machine Manager** (virt-manager) application provides a specialized GUI for managing virtual machines and virtualization hosts.



### IMPORTANT

Although still supported in RHEL 8, **virt-manager** has been deprecated. Cockpit is intended to become its replacement in a subsequent release. It is, therefore, recommended that you get familiar with Cockpit for managing virtualization in a GUI. However, in RHEL 8, some features may only be accessible from either virt-manager or the command line.

- The **Gnome Boxes** application is a lightweight graphical interface to view and access virtual machines and remote systems. Gnome Boxes is primarily designed for use on desktop systems.



## IMPORTANT

**Gnome Boxes** is provided as a part of the GNOME desktop environment and is supported on RHEL 8, but Red Hat recommends that you use **Cockpit** for managing virtualization in a GUI.

## 1.5. RED HAT VIRTUALIZATION SOLUTIONS

The following Red Hat products are built on top of RHEL 8 virtualization features and expand the KVM virtualization capabilities available in RHEL 8. In addition, many [limitations of RHEL 8 virtualization](#) do not apply to these products:

- **Red Hat Virtualization (RHV)**

RHV is designed for enterprise-class scalability and performance, and enables management of your entire virtual infrastructure, including hosts, virtual machines, networks, storage, and users from a centralized graphical interface.

For information about the differences between virtualization in Red Hat Enterprise Linux and Red Hat Virtualization, see [the Red Hat Customer Portal](#).

Red Hat Virtualization can be used by enterprises running large deployments or mission-critical applications. Examples of large deployments suited to Red Hat Virtualization include databases, trading platforms, and messaging systems that must run continuously without any downtime.

For more information about Red Hat Virtualization, see [the Red Hat Customer Portal](#) or the [Red Hat Virtualization documentation suite](#).

To download a fully supported 60-day evaluation version of Red Hat Virtualization, see <https://access.redhat.com/products/red-hat-virtualization/evaluation>

- **Red Hat OpenStack Platform (RHOSP)**

Red Hat OpenStack Platform offers an integrated foundation to create, deploy, and scale a secure and reliable public or private [OpenStack](#) cloud.

For more information about Red Hat OpenStack Platform, see [the Red Hat Customer Portal](#) or the [Red Hat OpenStack Platform documentation suite](#).

To download a fully supported 60-day evaluation version of Red Hat OpenStack Platform, see <https://access.redhat.com/products/red-hat-openstack-platform/evaluation>

In addition, specific Red Hat products provide operating-system-level virtualization, also known as **containerization**:

- Containers are isolated instances of the host OS and operate on top of an existing OS kernel. For more information on containers, see the [Red Hat Customer Portal](#).
- Containers do not have the versatility of KVM virtualization, but are more lightweight and flexible to handle. For a more detailed comparison, see the [Introduction to Linux Containers](#).

## CHAPTER 2. GETTING STARTED WITH VIRTUALIZATION IN RHEL 8

To start using [virtualization in RHEL 8](#), use the following steps. The default method for this is the command-line interface (CLI), but for user convenience, some of the steps can be completed in the [Cockpit GUI](#).

1. Enable the virtualization module and install the virtualization packages - see [Section 2.1, “Enabling virtualization in RHEL 8”](#).
2. Create a guest virtual machine:
  - For CLI, see [Section 2.2.2, “Creating a guest virtual machine using the command-line interface”](#).
  - For GUI, see [Section 2.2.3, “Creating virtual machines using Cockpit”](#).
3. Start the guest virtual machine:
  - For CLI, see [Section 2.3.2, “Starting a guest virtual machine using the command-line interface”](#).
  - For GUI, see [Section 2.3.3, “Powering up virtual machines in Cockpit”](#).
4. Connect to the guest virtual machine:
  - For CLI, see [Section 2.4.2, “Connecting to a guest virtual machine using the command-line interface”](#).
  - For GUI, see [Section 2.4.3, “Viewing the graphical console in Cockpit”](#).



### NOTE

Cockpit currently provides only a subset of guest management functions for virtual machines, so using the command line is recommended for advanced use of virtualization in RHEL 8.

### 2.1. ENABLING VIRTUALIZATION IN RHEL 8

To use virtualization in RHEL 8, you must enable the virtualization module, install virtualization packages, and ensure your system is configured to host virtual machines.

#### Prerequisites

- Red Hat Enterprise Linux 8 must be installed and registered on your host machine.
- Your system must meet the following hardware requirements to work as a virtualization host:
  - The architecture of your host machine [supports KVM virtualization](#).
  - The following system resources are available, or more:
    - 6 GB free disk space for the host, plus another 6 GB for each intended guest
    - 2 GB of RAM for the host, plus another 2 GB for each intended guest



## Procedure

1. Install the packages in the virtualization module:

```
# yum module install virt
```

2. Use the **virt-host-validate** command to verify that your system is prepared to be a virtualization host. For example:

```
# virt-host-validate
[...]
QEMU: Checking for device assignment IOMMU support      : PASS
QEMU: Checking if IOMMU is enabled by kernel            : WARN
(IOMMU appears to be disabled in kernel. Add intel_iommu=on to
kernel cmdline arguments)
LXC: Checking for Linux >= 2.6.26                      : PASS
[...]
LXC: Checking for cgroup 'blkio' controller mount-point : PASS
LXC: Checking if device /sys/fs/fuse/connections exists : FAIL
(Load the 'fuse' module to enable /proc/ overrides)
```

3. If all **virt-host-validate** checks return a **PASS** value, your system is prepared for [creating virtual machines](#).

If any of the checks return a **FAIL** value, follow the displayed instructions to fix the problem.

If any of the checks return a **WARN** value, consider following the displayed instructions to improve virtualization capabilities.

## Additional information

- Note that if virtualization is not supported by your host CPU, **virt-host-validate** will generate the following output:

```
QEMU: Checking for hardware virtualization: FAIL (Only emulated CPUs
are available, performance will be significantly limited)
```

However, attempting to create virtual machines on such a host system will fail, rather than have performance problems.

## 2.2. CREATING VIRTUAL MACHINES

To create a virtual machine (VM) in RHEL 8, use the [command line interface](#) or the [Cockpit GUI](#).

### 2.2.1. Prerequisites

- Virtualization must be [installed and enabled](#) on your system.
- Prior to creating virtual machines (VMs) on your system, consider the amount of system resources you need to allocate to your VMs, such as disk space, RAM, or CPUs. The recommended values may vary significantly depending on the intended tasks and workload of the VMs.

### 2.2.2. Creating a guest virtual machine using the command-line interface

## Prerequisites

- An operating system (OS) installation source, which can be one of the following, and can be available locally or on a network:
  - A physical installation medium, such as a DVD
  - An ISO image of an installation medium
  - A disk image of an existing guest installation
- Optionally, a Kickstart file can also be provided for faster and easier configuration of the installation.

## Procedure

To create a guest and start its OS installation, use the **virt-install** command. In its arguments, specify at minimum:

- The name of the new machine
- The amount of allocated memory
- The number of allocated virtual CPUs (vCPUs)
- The type and size of allocated storage
- The type and location of the OS installation source

Based on the chosen installation method, the necessary options and values can vary. See below for examples:

- The following creates a virtual machine named **demo-guest1** that installs the Fedora 29 OS from an ISO image locally stored in the **/home/username/Downloads/Fedora29.iso** file. This guest is also allocated with 2048 MiB of RAM, 2 vCPUs, and a 8 GiB qcow2 virtual disk is automatically configured for the guest.

```
# virt-install --name demo-guest1 --memory 2048 --vcpus 2 --disk  
size=8 --os-variant fedora29 --cdrom  
/home/username/Downloads/Fedora29.iso
```

- The following creates a virtual machine named **demo-guest2** that uses the **/home/username/Downloads/Fedora29.iso** image to run an OS from a live CD. No disk space is assigned to this guest, so changes made during the session will not be preserved. In addition, the guest is allocated with 4096 MiB of RAM and 4 vCPUs.

```
# virt-install --name demo-guest2 --memory 4096 --vcpus 4 --disk  
none --livecd --os-variant fedora29 --cdrom  
/home/username/Downloads/Fedora29.iso
```

- The following creates a RHEL 8 virtual machine named **demo-guest3** connects to an existing disk image, **/home/username/backup/disk.qcow2**. This is similar to physically moving a hard drive between machines, so the OS and data available to demo-guest3 are determined by how the image was handled previously. In addition, this guest is allocated with 2048 MiB of RAM and 2 vCPUs.

```
# virt-install --name demo-guest3 --memory 2048 --vcpus 2 --os-  
variant rhel8 --import --disk /home/username/backup/disk.qcow2
```



Note that the **--os-variant** option is highly recommended when importing a disk image. If it is not provided, the performance of the created VM will be suboptimal.

- The following creates a virtual machine named **demo-guest4** that installs from the <http://example.com/OS-install> URL. For the installation to start successfully, the URL must contain a working OS installation tree. In addition, the OS is automatically configured using the **/home/username/ks.cfg** kickstart file. This guest is also allocated with 2048 MiB of RAM, 2 vCPUs, and a 16 GiB qcow2 virtual disk.

```
# virt-install --name demo-guest4 --memory 2048 --vcpus 2 --disk
size=16 --os-variant fedora29 --location http://example.com/OS-
install --initrd-inject /home/username/ks.cfg --extra-
args="ks=file:/ks.cfg console=tty0 console=ttyS0,115200n8"
```

- The following creates a virtual machine named **demo-guest5** that installs from a **RHEL8.iso** file in text-only mode, without graphics. It connects the guest console to the serial console. The guest has 16384 MiB of memory, 16 vCPUs, and 280 GiB disk. This kind of installation is useful when connecting to a host over a slow network link.

```
# virt-install --name demo-guest5 --memory 16384 --vcpus 16 --disk
size=280 --os-variant rhel8 --location RHEL8.iso --no-graphics --
extra-args='console=ttyS0'
```



#### NOTE

A number of other options can be specified for **virt-install** to further configure the guest and its OS installation. For details, see the **virt-install** man page.

### 2.2.3. Creating virtual machines using Cockpit

You can create virtual machines using Cockpit.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

You can create a virtual machine on the machine to which Cockpit is connected.

1. Click **Create New VM**.  
The Create New Virtual Machine dialog appears.

2. Enter the basic configuration of the virtual machine you want to create.

- **Name** - The name of the virtual machine.
- **Installation Source Type** - The type of the installation source: Filesystem, URL
- **Installation Source** - The path or URL that points to the installation source.
- **OS Vendor** - The vendor of the virtual machine's operating system.
- **Operating System** - The virtual machine's operating system.
- **Memory** - The amount of memory with which to configure the virtual machine.
- **Storage Size** - The amount of storage space with which to configure the virtual machine.
- **Immediately Start VM** - Whether or not the virtual machine will start immediately after it is created.

3. Click **Create**.

The virtual machine is created and will start immediately if the **Immediately Start VM** checkbox is selected.

You must install the operating system the first time the virtual machine is run.

#### Additional resources

- For information on installing an operating system on a virtual machine, see [Section 3.3.2, “Installing operating systems using Cockpit”](#).

## 2.3. STARTING VIRTUAL MACHINES

To start an virtual machine in RHEL 8, you can use [the command line interface](#) or [the Cockpit GUI](#).

### 2.3.1. Prerequisites

- Before a VM can be started, it must be created and ideally also installed with an OS. For instruction to do so, see [Section 2.2, “Creating virtual machines”](#).

### 2.3.2. Starting a guest virtual machine using the command-line interface

To start a guest from the command-line interface, use the **virsh start** command:

```
# virsh start demo-guest1
Domain demo-guest1 started
```

### 2.3.3. Powering up virtual machines in Cockpit

You can run a selected virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

If a virtual machine is in the **shut off** state, you can start it.

1. Click a row with the name of the virtual machine you want to start.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Run**.  
The virtual machine starts.

#### Additional resources

- For information on shutting down a virtual machine, see [Section 3.4.2, “Powering down virtual machines in Cockpit”](#).
- For information on restarting a virtual machine, see [Section 3.4.3, “Restarting virtual machines using Cockpit”](#).
- For information on sending a non-maskable interrupt to a virtual machine, see [Section 3.4.4, “Sending non-maskable interrupts to virtual machines using Cockpit”](#).

## 2.4. CONNECTING TO VIRTUAL MACHINES

To interact with the OS installed in a virtual machine (VM) running on a Red Hat Enterprise Linux 8 host, you need to connect to the VM. To do so, you can use the command line interface or the Cockpit GUI.

### 2.4.1. Prerequisites

- Before connecting to a VM, it must be created and ideally also installed with an OS. For instruction to do so, see [Section 2.2, “Creating virtual machines”](#)

### 2.4.2. Connecting to a guest virtual machine using the command-line interface

To connect to a running guest using the command-line interface:

1. Use the **virsh domifaddr** command to obtain the guest's IP address:

```
# virsh domifaddr demoguest1
Name          MAC address          Protocol    Address
-----
vnet0         52:54:00:12:34:56      ipv4        192.168.123.456/24
```

2. Use SSH to log in to the guest:

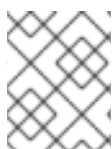
```
# ssh root@192.168.123.456
root@192.168.123.456's password:
```

### 2.4.3. Viewing the graphical console in Cockpit

You can view the graphical console of a selected virtual machine in Cockpit. The virtual machine console shows the graphical output of the virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.

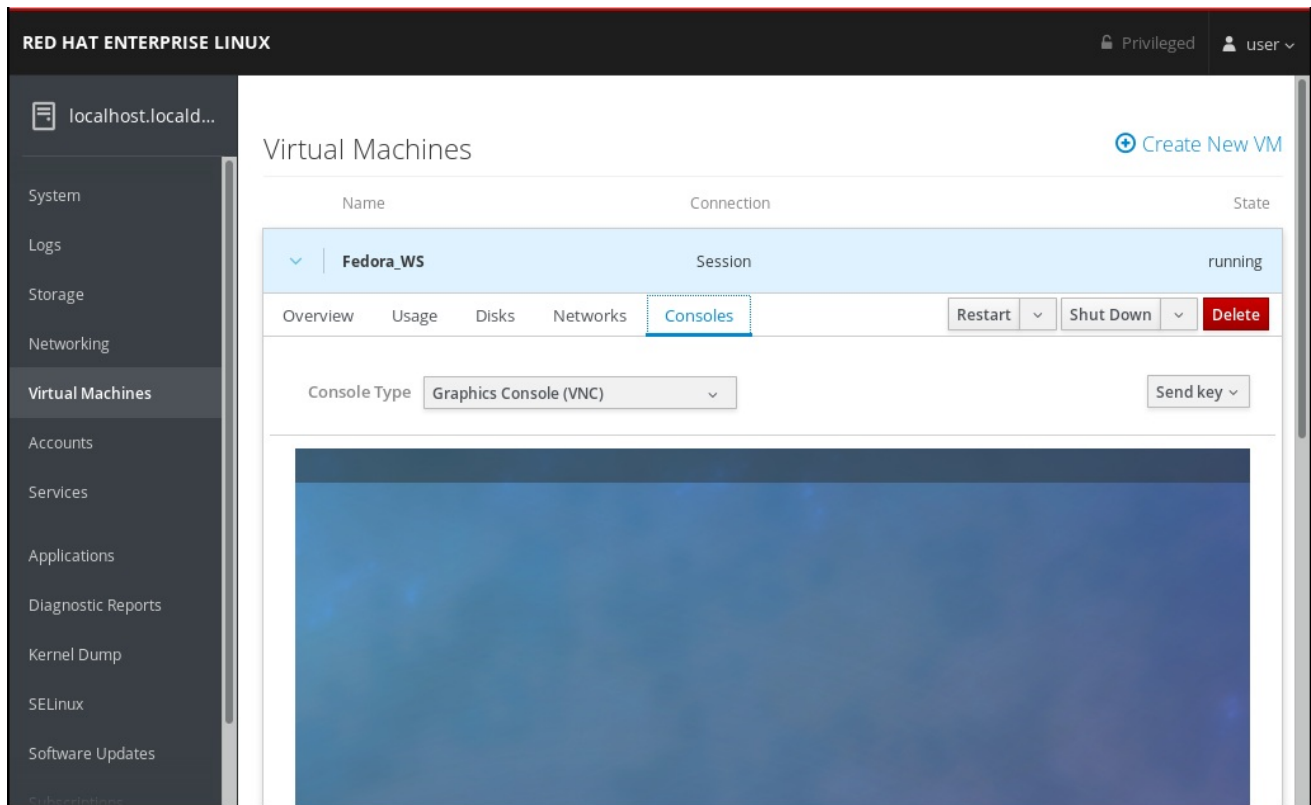


#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

1. Click a row with the name of the virtual machine whose graphical console you want to view.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Consoles**.  
The graphical console appears in the Cockpit pane.



You can interact with the virtual machine console using the mouse and keyboard in the same manner you interact with a real machine. The display in the virtual machine console reflects the activities being performed on the virtual machine.

The server on which Cockpit is running can intercept specific key combinations preventing them from being sent to the virtual machine. To send such key combinations, click the **Send key** menu and select the key sequence to send. For example, to send the **Ctrl+Alt+Del** combination to the virtual machine, click on the **Ctrl+Alt+Del** menu entry.

### Additional Resources

- For details on viewing the graphical console in a remote viewer, see [Section 3.9.2, “Viewing virtual machine consoles in remote viewers using Cockpit”](#).
- For details on viewing the serial console in Cockpit, see [Section 3.9.3, “Viewing the virtual machine serial console in Cockpit”](#).

## 2.5. SHUTTING DOWN VIRTUAL MACHINES

To shut down a running virtual machine in Red Hat Enterprise Linux 8, use [the command line interface](#) or [the Cockpit GUI](#).

### 2.5.1. Shutting down a guest virtual machine using the command-line interface

To shut down a guest under normal circumstances, do one of the following:

- use a shutdown command appropriate to the guest OS while [connected to the guest](#)
- use the **virsh shutdown** command on the host:

```
# virsh shutdown demo-guest1
Domain demo-guest1 is being shutdown
```

To force a guest to shut down, for example if it has become unresponsive, use the **virsh destroy** command on the host:

```
# virsh destroy demo-guest1
Domain demo-guest1 destroyed
```



#### NOTE

The **virsh destroy** command does not actually delete or remove the guest configuration or disk images. It only destroys the running guest instance. However, in rare cases, this command may cause corruption of the guest's file system, so using **virsh destroy** is only recommended if all other shutdown methods have failed.

## 2.5.2. Powering down virtual machines in Cockpit

You can shut down a selected virtual machine.

### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

### Procedure

If a virtual machine is in the **running** state, you can shut it down.

1. Click a row with the name of the virtual machine you want to shut down.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Shut Down**.  
The virtual machine shuts down.



#### NOTE

If the virtual machine does not shut down, click the arrow next to the **Shut Down** button and select **Force Shut Down**.

### Additional resources

- For information on starting a virtual machine, see [Section 3.4.1, “Powering up virtual machines in Cockpit”](#).
- For information on restarting a virtual machine, see [Section 3.4.3, “Restarting virtual machines using Cockpit”](#).
- For information on sending a non-maskable interrupt to a virtual machine, see [Section 3.4.4, “Sending non-maskable interrupts to virtual machines using Cockpit”](#).



## CHAPTER 3. USING COCKPIT FOR MANAGING VIRTUAL MACHINES

You can use the [Cockpit](#) interface to manage virtual machines. The following sections describe Cockpit's virtualization management capabilities and provide instructions for using them.

### 3.1. VIRTUAL MACHINE MANAGEMENT IN COCKPIT

Cockpit is a web-based interface for administering servers. It provides a graphical view of the virtual machines on the servers to which Cockpit can connect, and allows monitoring system resources and adjusting configuration with ease. With the installation of a Cockpit plug-in, Cockpit can be used to manage virtual machines on the servers to which Cockpit can connect.

Using Cockpit for virtual machine management, you can do the following:

- Create and delete virtual machines
- Install operating systems on virtual machines
- Run and shut down virtual machines
- View information about virtual machines
- Create and attach disks to virtual machines
- Configure virtual CPU settings for virtual machines
- Manage virtual network interfaces
- Interact with virtual machines using virtual machine consoles



#### NOTE

In RHEL 8, Cockpit is the default graphical management tool for virtual machines, because the Virtual Machine Manager has been deprecated. However, the Virtual Machine Manager is still supported in RHEL 8.

### 3.2. SETTING UP COCKPIT TO MANAGE VIRTUAL MACHINES

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.

#### Prerequisites

- Ensure that Cockpit is installed on your machine.

```
$ yum info cockpit
Installed Packages
Name           : cockpit
[...]
```

#### Procedure

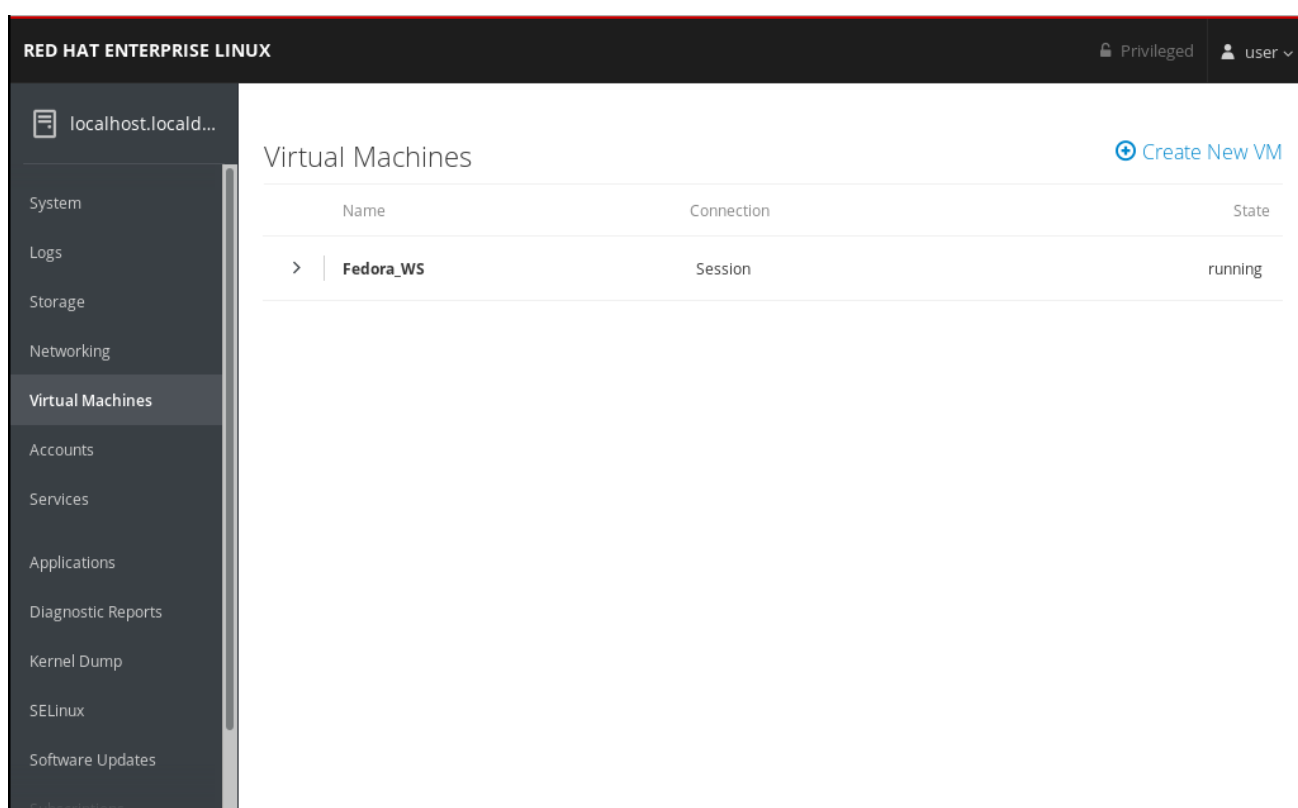
1. Install the **libvirt-dbus** package.

```
# yum install libvirt-dbus
```

2. Install the **cockpit-machines** plug-in.

```
# yum install cockpit-machines
```

If the installation is successful, **Virtual Machines** appears in the Cockpit side menu.



### 3.3. CREATING VIRTUAL MACHINES AND INSTALLING OPERATING SYSTEMS USING COCKPIT

Using Cockpit, you can create virtual machines and install operating systems on virtual machines. You can also delete virtual machines.

#### 3.3.1. Creating virtual machines using Cockpit

You can create virtual machines using Cockpit.

##### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



## NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

## Procedure

You can create a virtual machine on the machine to which Cockpit is connected.

1. Click **Create New VM**.

The Create New Virtual Machine dialog appears.

2. Enter the basic configuration of the virtual machine you want to create.

- **Name** - The name of the virtual machine.
- **Installation Source Type** - The type of the installation source: Filesystem, URL
- **Installation Source** - The path or URL that points to the installation source.
- **OS Vendor** - The vendor of the virtual machine's operating system.
- **Operating System** - The virtual machine's operating system.
- **Memory** - The amount of memory with which to configure the virtual machine.
- **Storage Size** - The amount of storage space with which to configure the virtual machine.
- **Immediately Start VM** - Whether or not the virtual machine will start immediately after it is created.

3. Click **Create**.

The virtual machine is created and will start immediately if the **Immediately Start VM** checkbox is selected.

You must install the operating system the first time the virtual machine is run.

## Additional resources

- For information on installing an operating system on a virtual machine, see [Section 3.3.2, “Installing operating systems using Cockpit”](#).

### 3.3.2. Installing operating systems using Cockpit

You can install an operating system on virtual machines using Cockpit.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

The first time a virtual machine loads, you must install an operating system on the virtual machine.

- Click **Install**.  
The installation routine of the operating system runs in the virtual machine console.



#### NOTE

If the *Immediately Start VM* checkbox in the Create New Virtual Machine dialog is checked, the installation routine of the operating system starts automatically when the virtual machine is created.



#### NOTE

If the installation routine fails, the virtual machine must be deleted and recreated.

### 3.3.3. Deleting virtual machines using Cockpit

You can delete virtual machines and their associated storage files.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

You can delete a virtual machine from the host to which Cockpit is connected.

1. Click a row with the name of the virtual machine you want to delete.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Delete**.  
A confirmation dialog appears.

Confirm deletion of Fed

The VM is running and will be forced off before deletion.

Delete associated storage files:

<input checked="" type="checkbox"/>	/home/images/Fed.qcow2	vda
-------------------------------------	------------------------	-----

Cancel

Delete

- If you want to delete all or some of the storage files associated with the virtual machine, select the checkboxes next to the storage files you want to delete.
- Click **Delete**.  
The virtual machine and any selected associated storage files are deleted.

## 3.4. POWERING UP AND POWERING DOWN VIRTUAL MACHINES USING COCKPIT

Using Cockpit, you can run, shut down, and restart virtual machines. You can also send a non-maskable interrupt to a virtual machine that is unresponsive.

### 3.4.1. Powering up virtual machines in Cockpit

You can run a selected virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

If a virtual machine is in the **shut off** state, you can start it.

- Click a row with the name of the virtual machine you want to start.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
- Click **Run**.  
The virtual machine starts.

#### Additional resources

- For information on shutting down a virtual machine, see [Section 3.4.2, “Powering down virtual machines in Cockpit”](#).
- For information on restarting a virtual machine, see [Section 3.4.3, “Restarting virtual machines using Cockpit”](#).

- For information on sending a non-maskable interrupt to a virtual machine, see [Section 3.4.4, “Sending non-maskable interrupts to virtual machines using Cockpit”](#).

### 3.4.2. Powering down virtual machines in Cockpit

You can shut down a selected virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

If a virtual machine is in the **running** state, you can shut it down.

1. Click a row with the name of the virtual machine you want to shut down.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Shut Down**.  
The virtual machine shuts down.



#### NOTE

If the virtual machine does not shut down, click the arrow next to the **Shut Down** button and select **Force Shut Down**.

#### Additional resources

- For information on starting a virtual machine, see [Section 3.4.1, “Powering up virtual machines in Cockpit”](#).
- For information on restarting a virtual machine, see [Section 3.4.3, “Restarting virtual machines using Cockpit”](#).
- For information on sending a non-maskable interrupt to a virtual machine, see [Section 3.4.4, “Sending non-maskable interrupts to virtual machines using Cockpit”](#).

### 3.4.3. Restarting virtual machines using Cockpit

You can restart a selected virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

If a virtual machine is in the **running** state, you can restart it.

1. Click a row with the name of the virtual machine you want to restart.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Restart**.  
The virtual machine shuts down and restarts.



#### NOTE

If the virtual machine does not restart, click the arrow next to the **Restart** button and select **Force Restart**.

#### 3.4.3.1. Additional resources

- For information on starting a virtual machine, see [Section 3.4.1, “Powering up virtual machines in Cockpit”](#).
- For information on shutting down a virtual machine, see [Section 3.4.2, “Powering down virtual machines in Cockpit”](#).
- For information on sending a non-maskable interrupt to a virtual machine, see [Section 3.4.4, “Sending non-maskable interrupts to virtual machines using Cockpit”](#).

#### 3.4.4. Sending non-maskable interrupts to virtual machines using Cockpit

You can send a non-maskable interrupt (NMI) to a selected virtual machine that becomes unresponsive.

##### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

##### Procedure

If a virtual machine is in the **running** state but is unresponsive, you can send it an NMI.

1. Click a row with the name of the virtual machine to which you want to send an NMI.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click the arrow next to the **Shut Down** button and select **Send Non-Maskable Interrupt**.  
An NMI is sent to the virtual machine.

#### 3.4.4.1. Additional resources

- For information on starting a virtual machine, see [Section 3.4.1, “Powering up virtual machines in Cockpit”](#).
- For information on restarting a virtual machine, see [Section 3.4.3, “Restarting virtual machines using Cockpit”](#).

- For information on shutting down a virtual machine, see [Section 3.4.2, “Powering down virtual machines in Cockpit”](#).

## 3.5. VIEWING VIRTUAL MACHINE INFORMATION USING COCKPIT

Using Cockpit, you can view information about the virtual machines to which Cockpit is connected.

### 3.5.1. Viewing virtual machine information in Cockpit

The following describes how to view state and connection information about the virtual machines to which the Cockpit session is connected.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



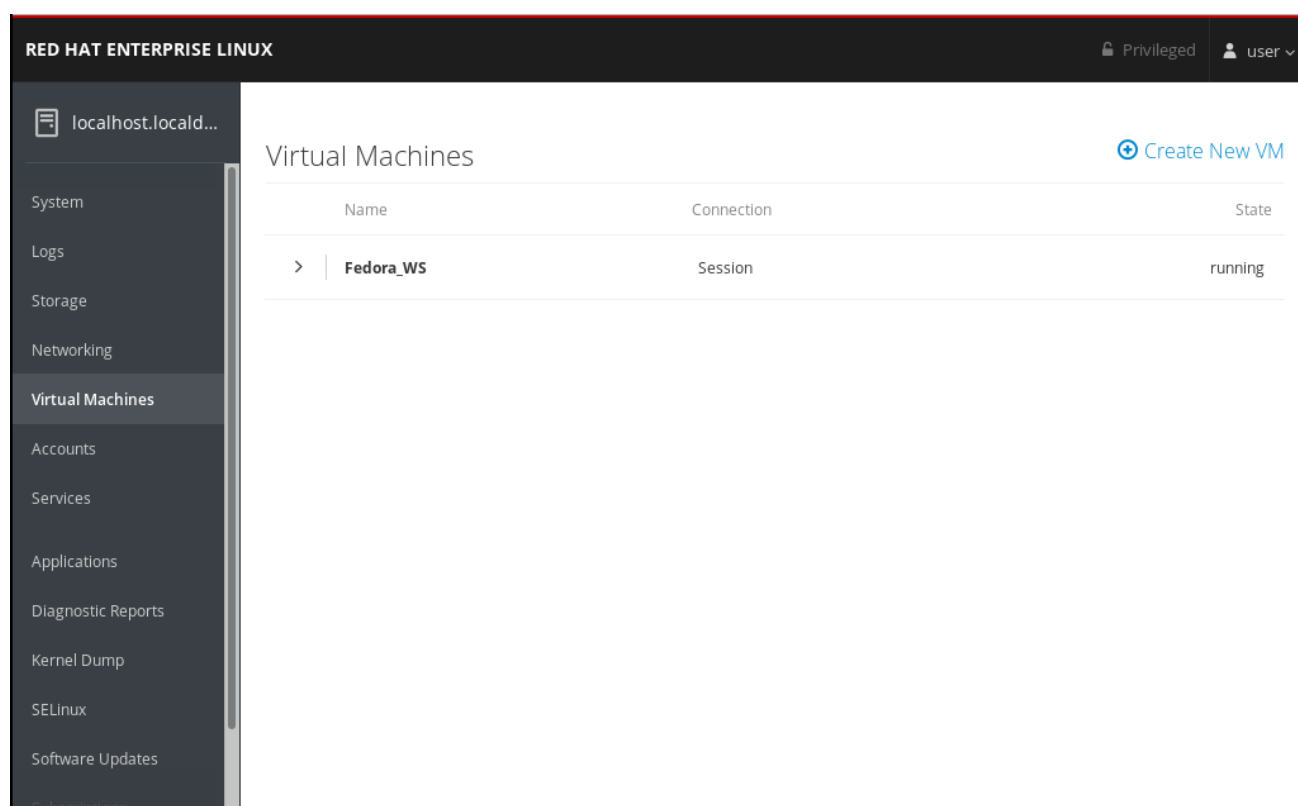
#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

To view connection and state information for the virtual machines to which Cockpit is attached.

- Click **Virtual Machines** in the Cockpit side menu.  
Information about the virtual machines to which the Cockpit session is connected appears.



The information includes the following:

- **Name** - The name of the virtual machine.
- **Connection** - The type of libvirt connection, system or session.



- **State** - The state of the virtual machine.

### 3.5.1.1. Additional resources

- For information on viewing basic information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.2, “Viewing basic virtual machine information in Cockpit”](#).
- For information on viewing resource usage for a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.3, “Viewing virtual machine resource usage in Cockpit”](#).
- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.4, “Viewing virtual machine disk information in Cockpit”](#).
- For information on viewing virtual network interface card information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.5, “Viewing virtual NIC information in Cockpit”](#).

## 3.5.2. Viewing basic virtual machine information in Cockpit

The following describes how to view basic information about a selected virtual machine to which the Cockpit session is connected.

### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

### Procedure

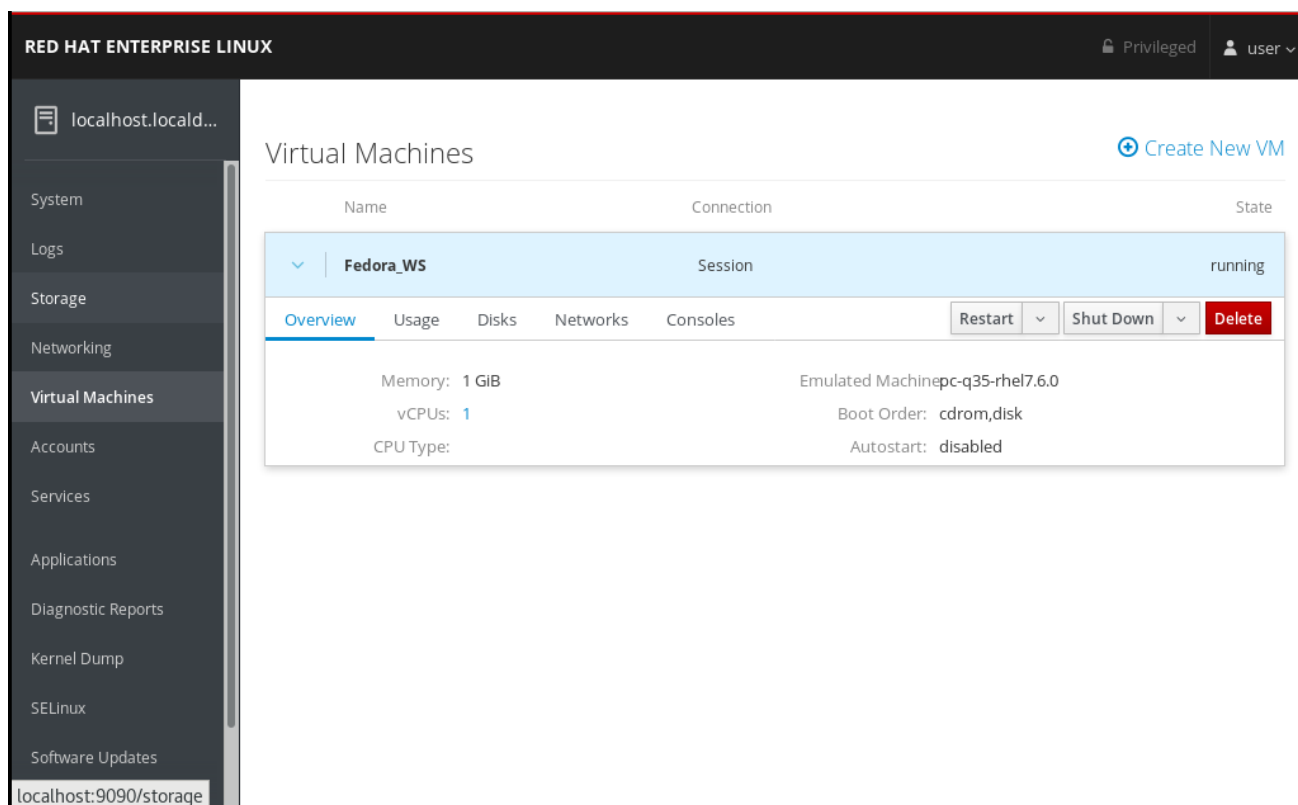
To view basic information about a selected virtual machine.

- Click a row with the name of the virtual machine whose information you want to see. The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.



### NOTE

If another tab is selected, click **Overview**.



The information includes the following:

- **Memory** - The amount of memory assigned to the virtual machine.
- **Emulated Machine** - The machine type emulated by the virtual machine.
- **vCPUs** - The number of virtual CPUs configured for the virtual machine.



#### NOTE

To see more detailed virtual CPU information and configure the virtual CPUs configured for a virtual machine, see [Section 3.6.1, “Configuring virtual CPUs for virtual machines using Cockpit”](#).

- **Boot Order** - The boot order configured for the virtual machine.
- **CPU Type** - The architecture of the virtual CPUs configured for the virtual machine.
- **Autostart** - Whether or not autostart is enabled for the virtual machine.

#### 3.5.2.1. Additional resources

- For information on viewing information about all of the virtual machines to which the Cockpit session is connected, see [Section 3.5.1, “Viewing virtual machine information in Cockpit”](#).
- For information on viewing resource usage for a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.3, “Viewing virtual machine resource usage in Cockpit”](#).
- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.4, “Viewing virtual machine disk information in Cockpit”](#).

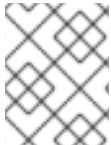
- For information on viewing virtual network interface card information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.5, “Viewing virtual NIC information in Cockpit”](#).

### 3.5.3. Viewing virtual machine resource usage in Cockpit

The following describes how to view resource usage information about a selected virtual machine to which the Cockpit session is connected.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

To view information about the memory and virtual CPU usage of a selected virtual machine.

1. Click a row with the name of the virtual machine whose information you want to see.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Usage**.  
The Usage pane appears with information about the memory and virtual CPU usage of the virtual machine.

The screenshot shows the Cockpit interface for managing virtual machines. The left sidebar contains a navigation menu with options like System, Logs, Storage, Networking, Virtual Machines (selected), Accounts, Services, Applications, Diagnostic Reports, Kernel Dump, SELinux, and Software Updates. The main panel is titled 'Virtual Machines' and shows a table with columns for Name, Connection, and State. A row for 'Fedora\_WS' is selected, showing a 'Session' connection and a 'running' state. Below the table, there are tabs for Overview, Usage (selected), Disks, Networks, and Consoles. The Usage tab displays two circular progress indicators: one for memory usage showing 1.4 GiB used from a 1 GiB limit, and another for vCPU usage showing 0.0% used from 1 vCPU. Action buttons for Restart, Shut Down, and Delete are visible on the right side of the Usage pane.

#### 3.5.3.1. Additional resources

- For information on viewing information about all of the virtual machines to which the Cockpit session is connected, see [Section 3.5.1, “Viewing virtual machine information in Cockpit”](#).
- For information on viewing basic information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.2, “Viewing basic virtual machine information in Cockpit”](#).
- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.4, “Viewing virtual machine disk information in Cockpit”](#).
- For information on viewing virtual network interface card information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.5, “Viewing virtual NIC information in Cockpit”](#).

### 3.5.4. Viewing virtual machine disk information in Cockpit

The following describes how to view disk information about a virtual machine to which the Cockpit session is connected.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



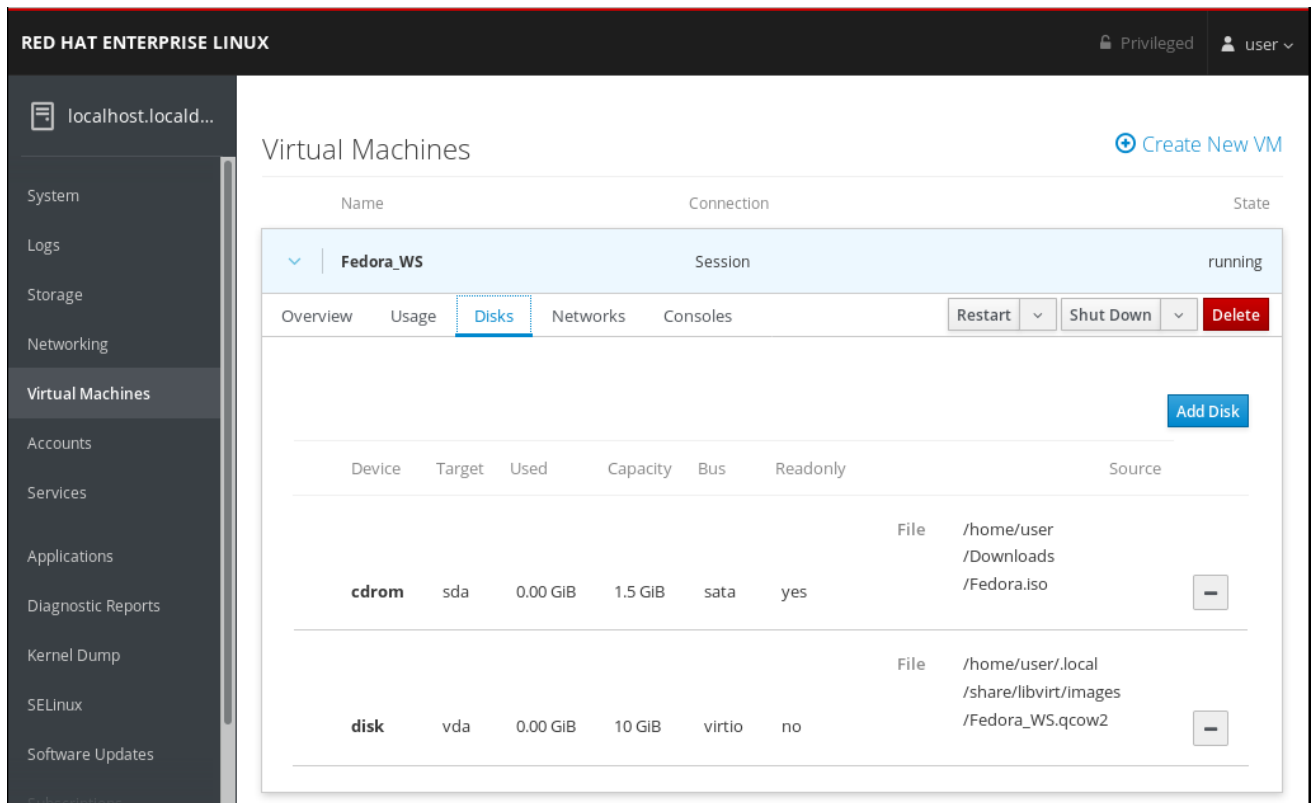
#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

To view disk information about a selected virtual machine.

1. Click a row with the name of the virtual machine whose information you want to see.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Disks**.  
The Disks pane appears with information about the disks assigned to the virtual machine.



The information includes the following:

- **Device** - The device type of the disk.
- **Target** - The controller type of the disk.
- **Used** - The amount of the disk that is used.
- **Capacity** - The size of the disk.
- **Bus** - The bus type of the disk.
- **Readonly** - Whether or not the disk is read-only.
- **Source** - The disk device or file.

#### 3.5.4.1. Additional resources

- For information on viewing information about all of the virtual machines to which the Cockpit session is connected, see [Section 3.5.1, “Viewing virtual machine information in Cockpit”](#).
- For information on viewing basic information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.2, “Viewing basic virtual machine information in Cockpit”](#).
- For information on viewing resource usage for a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.3, “Viewing virtual machine resource usage in Cockpit”](#).
- For information on viewing virtual network interface card information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.5, “Viewing virtual NIC information in Cockpit”](#).

#### 3.5.5. Viewing virtual NIC information in Cockpit

The following describes how to view information about the virtual network interface cards (vNICs) on a selected virtual machine:

## Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



## NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

## Procedure

To view information about the virtual network interface cards (NICs) on a selected virtual machine.

1. Click a row with the name of the virtual machine whose information you want to see.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Networks**.  
The Networks pane appears with information about the virtual NICs configured for the virtual machine.

RED HAT ENTERPRISE LINUX

Privileged user

localhost.locald...

System  
Logs  
Storage  
Networking  
**Virtual Machines**  
Accounts  
Services  
Applications  
Diagnostic Reports  
Kernel Dump  
SELinux  
Software Updates

### Virtual Machines

Create New VM

Name	Connection	State
Fedora_WS	Session	running

Overview Usage Disks **Networks** Consoles Restart Shut Down Delete

Type	Model type	MAC Address	Source	State
user	virtio	52:54:00:96:48:5e		down

Plug Edit

The information includes the following:

- **Type** - The type of network interface for the virtual machine. Types include direct, network, bridge, ethernet, hostdev, mcast, user, and server.
- **Model type** - The model of the virtual NIC.
- **MAC Address** - The MAC address of the virtual NIC.
- **Source** - The source of the network interface. This is dependent on the network type.

- **State** - The state of the virtual NIC.

### 3.5.5.1. Additional resources

- For information on viewing information about all of the virtual machines to which the Cockpit session is connected, see [Section 3.5.1, “Viewing virtual machine information in Cockpit”](#).
- For information on viewing basic information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.2, “Viewing basic virtual machine information in Cockpit”](#).
- For information on viewing resource usage for a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.3, “Viewing virtual machine resource usage in Cockpit”](#).
- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.4, “Viewing virtual machine disk information in Cockpit”](#).

## 3.6. MANAGING VIRTUAL CPUS USING COCKPIT

Using Cockpit, you can manage the virtual CPUs configured for the virtual machines to which Cockpit is connected. You can view information about the virtual machines. You can also configure the virtual CPUs for virtual machines.

### 3.6.1. Configuring virtual CPUs for virtual machines using Cockpit

You can configure the number of virtual CPUs and other virtual CPU parameters for a virtual machine using Cockpit.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure


1. Click a row with the name of the virtual machine for which you want to view and configure virtual CPU parameters.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine, including the number of virtual CPUs, and controls for shutting down and deleting the virtual machine.
2. Click the number of vCPUs in the Overview pane.  
The vCPU Details dialog appears.

**Fed vCPU Details**

vCPU Count  ⓘ Sockets  ⓘ

vCPU Maximum  ⓘ Cores per socket  ⓘ

Threads per cores  ⓘ

 All changes will take effect only after stopping and starting the VM.

3. Configure the virtual CPUs for the selected virtual machine.

- **vCPU Count** - Enter the number of virtual CPUs for the virtual machine.



#### NOTE

The vCPU count cannot be greater than the vCPU Maximum.

- **vCPU Maximum** - Enter the maximum number of virtual CPUs that can be configured for the virtual machine.
- **Sockets** - Select the number of sockets to expose to the virtual machine.
- **Cores per socket** - Select the number of cores for each socket to expose to the virtual machine.
- **Threads per core** - Select the number of threads for each core to expose to the virtual machine.

4. Click **Apply**.

The virtual CPUs for the virtual machine are configured.



#### NOTE

Changes to the virtual CPU settings only take effect after the virtual machine is stopped and restarted.

## 3.7. MANAGING VIRTUAL MACHINE DISKS USING COCKPIT

Using Cockpit, you can manage the disks configured for the virtual machines to which Cockpit is connected.

You can:

- [View information about disks.](#)
- [Create and attach new virtual disks to virtual machines.](#)



- [Attach existing virtual disks to virtual machines.](#)
- [Detach virtual disks from virtual machines.](#)

### 3.7.1. Viewing virtual machine disk information in Cockpit

The following describes how to view disk information about a virtual machine to which the Cockpit session is connected.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

To view disk information about a selected virtual machine.

1. Click a row with the name of the virtual machine whose information you want to see.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Disks**.  
The Disks pane appears with information about the disks assigned to the virtual machine.

The screenshot shows the Cockpit interface for a Red Hat Enterprise Linux system. The left sidebar contains a navigation menu with options like System, Logs, Storage, Networking, Virtual Machines (selected), Accounts, Services, Applications, Diagnostic Reports, Kernel Dump, SELinux, and Software Updates. The main area is titled 'Virtual Machines' and shows a table of virtual machines. The 'Fedora\_WS' VM is selected, and its 'Disks' pane is expanded. The 'Disks' pane shows a table of disks with columns: Device, Target, Used, Capacity, Bus, Readonly, and Source. Two disks are listed: 'cdrom' and 'disk'.

Device	Target	Used	Capacity	Bus	Readonly	Source
cdrom	sda	0.00 GiB	1.5 GiB	sata	yes	File /home/user/Downloads/Fedora.iso
disk	vda	0.00 GiB	10 GiB	virtio	no	File /home/user/.local/share/libvirt/images/Fedora_WS.qcow2

The information includes the following:

- **Device** - The device type of the disk.
- **Target** - The controller type of the disk.

- **Used** - The amount of the disk that is used.
- **Capacity** - The size of the disk.
- **Bus** - The bus type of the disk.
- **Readonly** - Whether or not the disk is read-only.
- **Source** - The disk device or file.

#### 3.7.1.1. Additional resources

- For information on viewing information about all of the virtual machines to which the Cockpit session is connected, see [Section 3.5.1, “Viewing virtual machine information in Cockpit”](#).
- For information on viewing basic information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.2, “Viewing basic virtual machine information in Cockpit”](#).
- For information on viewing resource usage for a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.3, “Viewing virtual machine resource usage in Cockpit”](#).
- For information on viewing virtual network interface card information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.5, “Viewing virtual NIC information in Cockpit”](#).

#### 3.7.2. Adding new disks to virtual machines using Cockpit

You can add new disks to virtual machines by creating a new disk (storage pool) and attaching it to a virtual machine using Cockpit.



##### NOTE

You can only use directory-type storage pools when creating new disks for virtual machines using Cockpit.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.

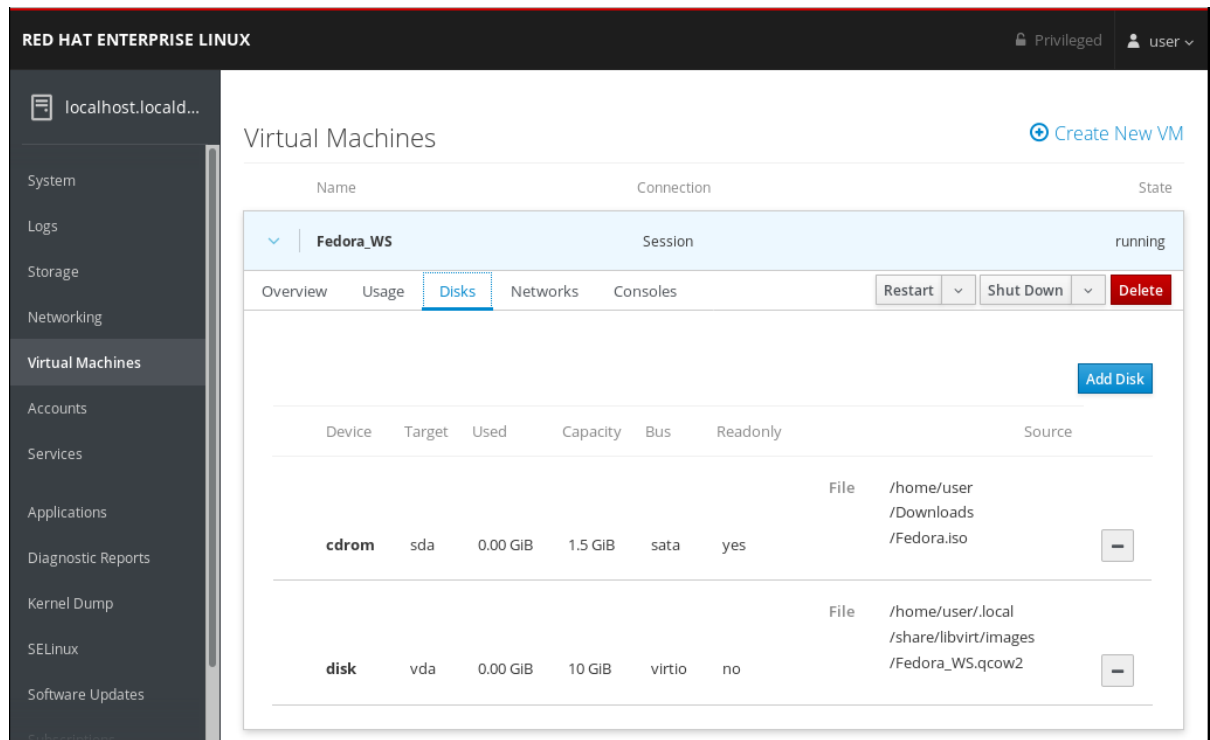


##### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

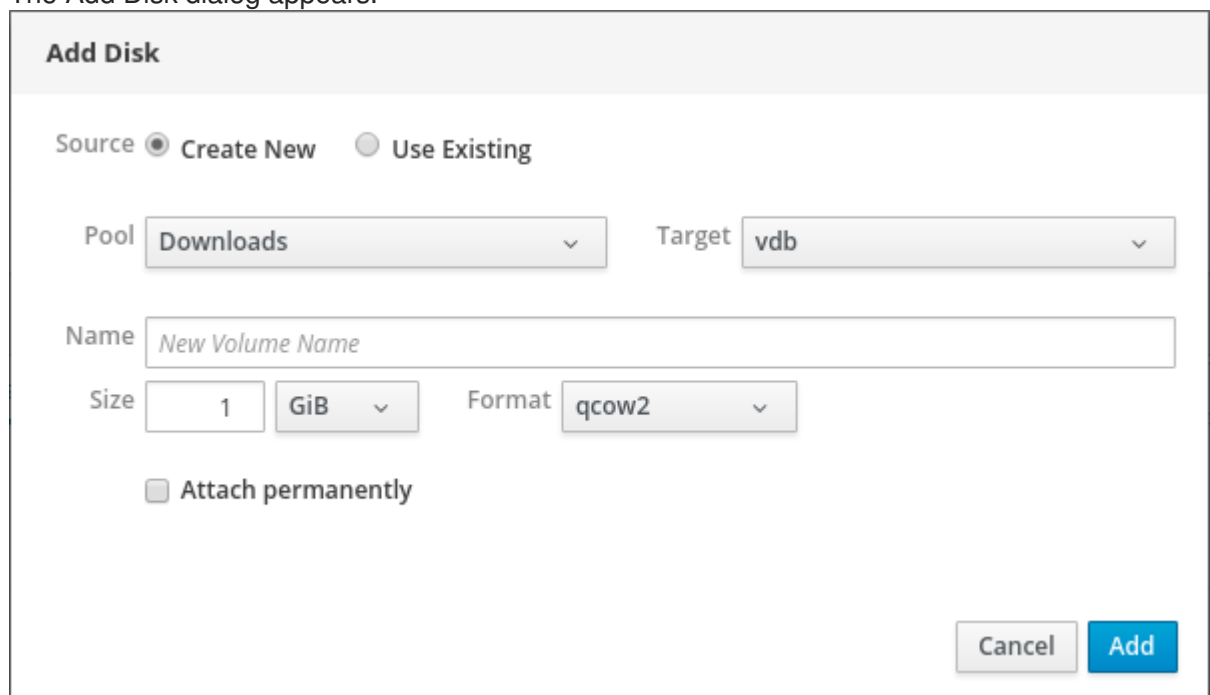
#### Procedure

1. Click a row with the name of the virtual machine for which you want to create and attach a new disk.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Disks**.  
The Disks pane appears with information about the disks configured for the virtual machine.



3. Click **Add Disk**.

The Add Disk dialog appears.



4. Ensure that the *Create New* option button is selected.

5. Configure the new disk.

- **Pool** - Select the storage pool from which the virtual disk will be created.
- **Target** - Select a target for the virtual disk that will be created.
- **Name** - Enter a name for the virtual disk that will be created.
- **Size** - Enter the size and select the unit (MiB or GiB) of the virtual disk that will be created.

- **Format** - Select the format for the virtual disk that will be created. Supported types: qcow2, raw
- **Attach permanently** - Whether or not the virtual disk will be persistent. If checked, the virtual disk is persistent. If not checked, the virtual disk is not persistent.



#### NOTE

When adding a new disk to a virtual machine that is not running, the **Attach permanently** check box does not appear. Only persistent virtual disks can be added to virtual machines that are not running.

1. Click **Add**.

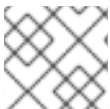
The virtual disk is created and connected to the virtual machine.

#### Additional resources

- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.7.1, “Viewing virtual machine disk information in Cockpit”](#).
- For information on attaching existing disks to virtual machines, see [Section 3.7.3, “Attaching existing disks to virtual machines using Cockpit”](#).
- For information on detaching disks from virtual machines, see [Section 3.7.4, “Detaching disks from virtual machines”](#).

### 3.7.3. Attaching existing disks to virtual machines using Cockpit

The following describes how to attach existing disks to a virtual machine using Cockpit.



#### NOTE

You can only attach directory-type storage pools to virtual machines using Cockpit.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.

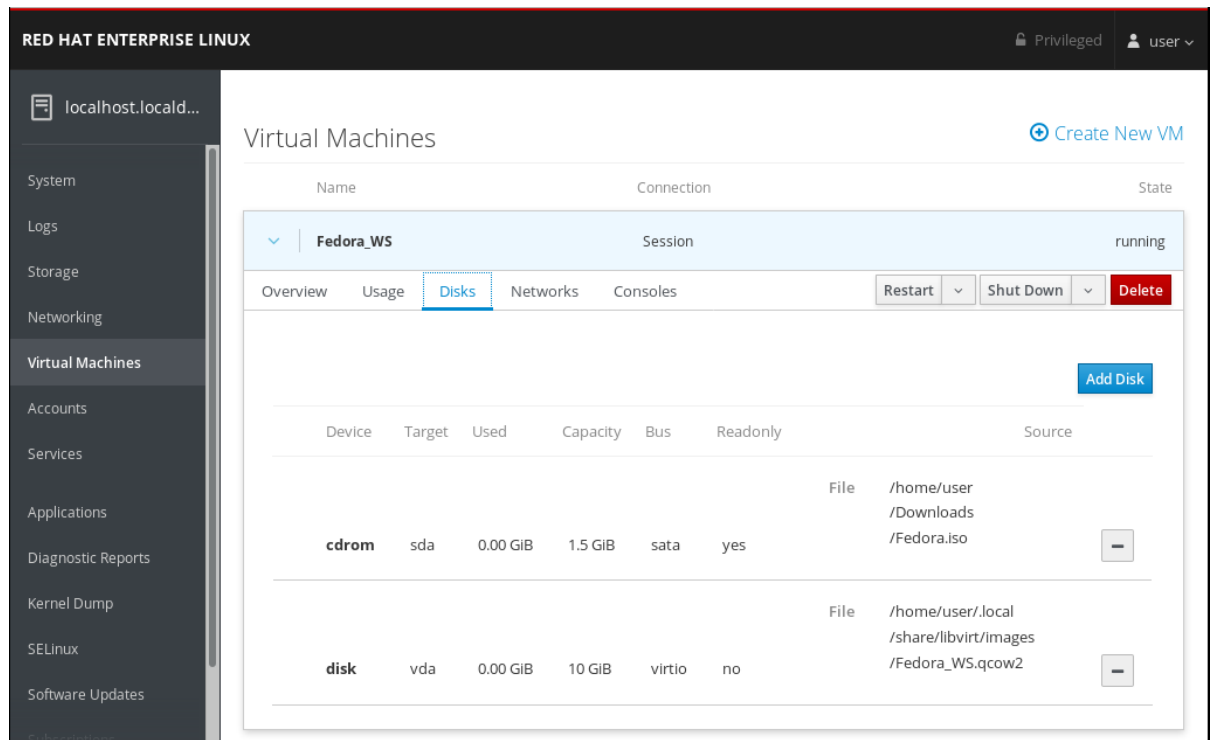


#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

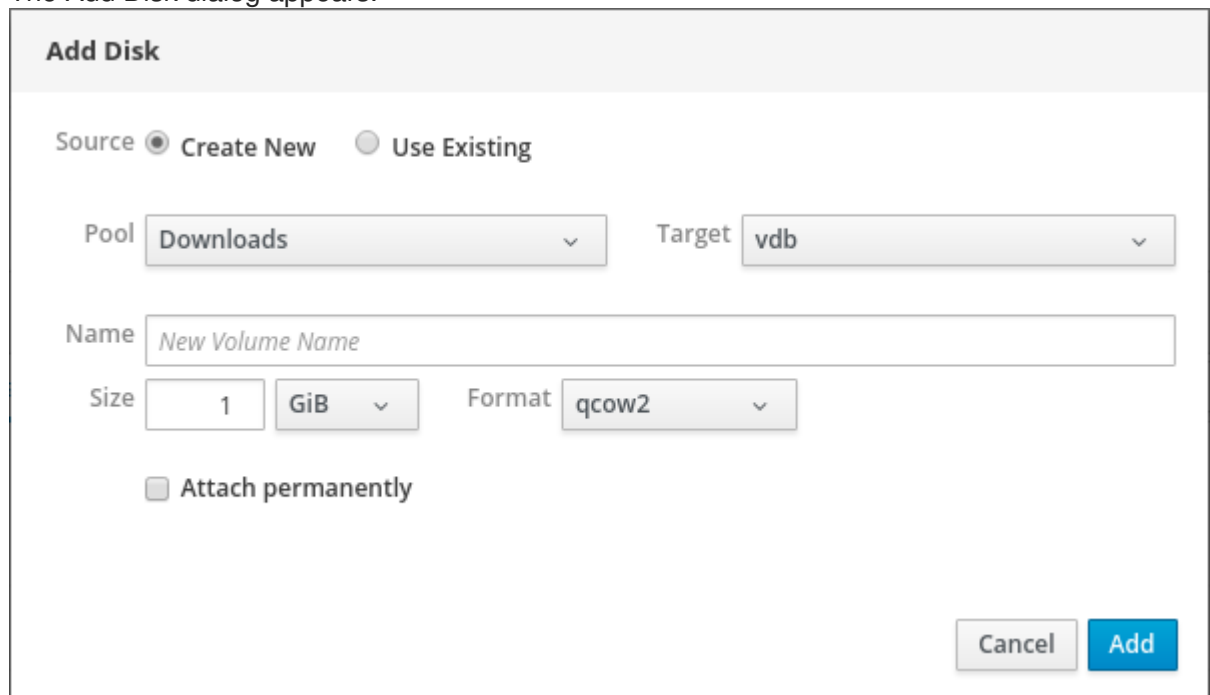
#### Procedure

1. Click a row with the name of the virtual machine to which you want to attach an existing disk. The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Disks**. The Disks pane appears with information about the disks configured for the virtual machine.



### 3. Click **Add Disk**.

The Add Disk dialog appears.



- Click the **Use Existing** option button.

The appropriate configuration fields appear in the Add Disk dialog.

**Add Disk**

Source ☐ Create New ☒ Use Existing

Pool Downloads Target vdb

Volume .directory

☐ Attach permanently

Cancel Add

- Configure the disk for the virtual machine.

- **Pool** - Select the storage pool from which the virtual disk will be attached.
- **Target** - Select a target for the virtual disk that will be attached.
- **Volume** - Select the storage volume that will be attached.
- **Attach Permanently** - Check to make the virtual disk persistent. Clear to make the virtual disk transient.

- Click **Add**

The selected virtual disk is attached to the virtual machine.

### Additional resources

- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.7.1, “Viewing virtual machine disk information in Cockpit”](#).
- For information on creating new disks and attaching them to virtual machines, see [Section 3.7.2, “Adding new disks to virtual machines using Cockpit”](#).
- For information on detaching disks from virtual machines, see [Section 3.7.4, “Detaching disks from virtual machines”](#).

### 3.7.4. Detaching disks from virtual machines

The following describes how to detach disks from virtual machines using Cockpit.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.

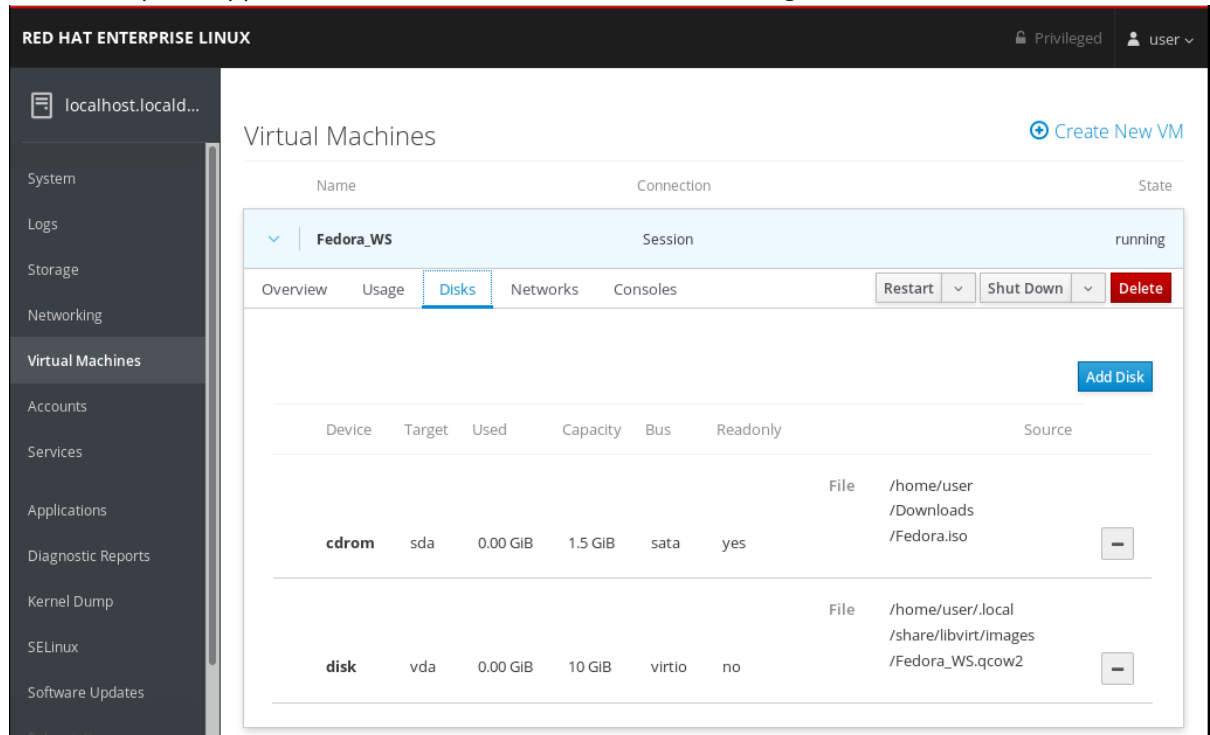



## NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

## Procedure

1. Click a row with the name of the virtual machine from which you want to detach an existing disk. The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Disks**.  
The Disks pane appears with information about the disks configured for the virtual machine.



3. Click  next to the disk you want to detach from the virtual machine. The virtual disk is detached from the virtual machine.

## CAUTION

There is no confirmation before detaching the disk from the virtual machine.

## Additional resources

- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.7.1, “Viewing virtual machine disk information in Cockpit”](#).
- For information on creating new disks and attaching them to virtual machines, see [Section 3.7.2, “Adding new disks to virtual machines using Cockpit”](#).
- For information on attaching existing disks to virtual machines, see [Section 3.7.3, “Attaching existing disks to virtual machines using Cockpit”](#).

## 3.8. USING COCKPIT FOR MANAGING VIRTUAL MACHINE VNICS

Using Cockpit, you can manage the virtual network interface cards (vNICs) configured for the virtual machines to which Cockpit is connected. You can view information about vNICs. You can also connect and disconnect vNICs from virtual machines.

### 3.8.1. Viewing virtual NIC information in Cockpit

The following describes how to view information about the virtual network interface cards (vNICs) on a selected virtual machine:

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

To view information about the virtual network interface cards (NICs) on a selected virtual machine.

1. Click a row with the name of the virtual machine whose information you want to see.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Networks**.  
The Networks pane appears with information about the virtual NICs configured for the virtual machine.

**RED HAT ENTERPRISE LINUX** Privileged user

localhost.locald...

System  
Logs  
Storage  
Networking  
**Virtual Machines**  
Accounts  
Services  
Applications  
Diagnostic Reports  
Kernel Dump  
SELinux  
Software Updates

Virtual Machines [Create New VM](#)

Name	Connection	State
▼ Fedora_WS	Session	running

Overview Usage Disks **Networks** Consoles Restart Shut Down Delete

Type	Model type	MAC Address	Source	State
user	virtio	52:54:00:96:48:5e		down Plug Edit

The information includes the following:



- **Type** - The type of network interface for the virtual machine. Types include direct, network, bridge, ethernet, hostdev, mcast, user, and server.
- **Model type** - The model of the virtual NIC.
- **MAC Address** - The MAC address of the virtual NIC.
- **Source** - The source of the network interface. This is dependent on the network type.
- **State** - The state of the virtual NIC.

### 3.8.1.1. Additional resources

- For information on viewing information about all of the virtual machines to which the Cockpit session is connected, see [Section 3.5.1, “Viewing virtual machine information in Cockpit”](#).
- For information on viewing basic information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.2, “Viewing basic virtual machine information in Cockpit”](#).
- For information on viewing resource usage for a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.3, “Viewing virtual machine resource usage in Cockpit”](#).
- For information on viewing disk information about a selected virtual machine to which the Cockpit session is connected, see [Section 3.5.4, “Viewing virtual machine disk information in Cockpit”](#).

### 3.8.2. Connecting virtual NICs in Cockpit

Using Cockpit, you can connect virtual network interface cards (NICs) configured for a selected virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



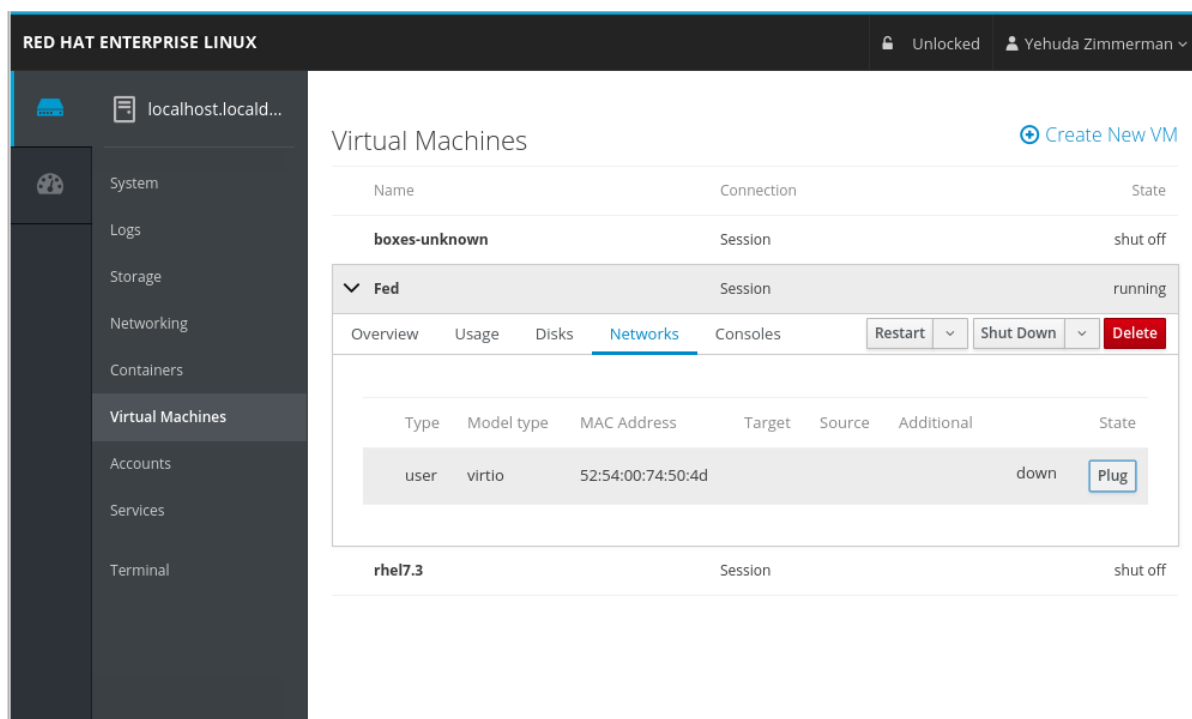
#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

If a virtual NIC is disconnected from the selected virtual machine, you can reconnect it.

1. Click a row with the name of the virtual machine whose virtual NIC you want to connect.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Networks**.  
The Networks pane appears with information about the virtual NICs configured for the virtual machine.



3. Click **Plug** in the row of the virtual NIC you want to connect.  
The selected virtual NIC connects to the virtual machine.

### 3.8.3. Disconnecting virtual NICs in Cockpit

Using Cockpit, you can disconnect the virtual network interface cards (NICs) configured for a selected virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



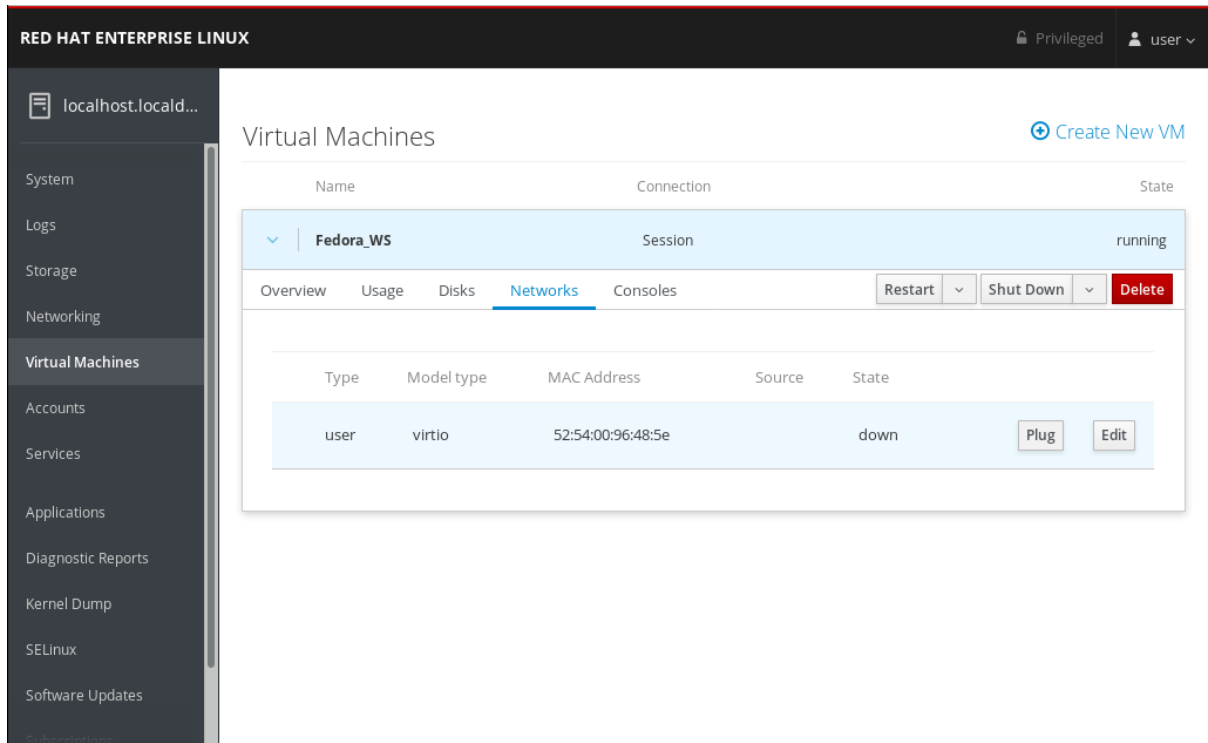
#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

If a virtual NIC is connected to the selected virtual machine, you can disconnect it.

1. Click a row with the name of the virtual machine whose virtual NIC you want to disconnect.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Networks**.  
The Networks pane appears with information about the virtual NICs configured for the virtual machine.



3. Click **Unplug** in the row of the virtual NIC you want to disconnect.  
The selected virtual NIC disconnects from the virtual machine.

## 3.9. VIEWING VIRTUAL MACHINE CONSOLES USING COCKPIT

Using Cockpit, you can view the virtual machine's consoles. These include both graphical and serial consoles.

### 3.9.1. Viewing the virtual machine graphical console in Cockpit

You can view the graphical console of a selected virtual machine in Cockpit. The virtual machine console shows the graphical output of the virtual machine.

#### Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.

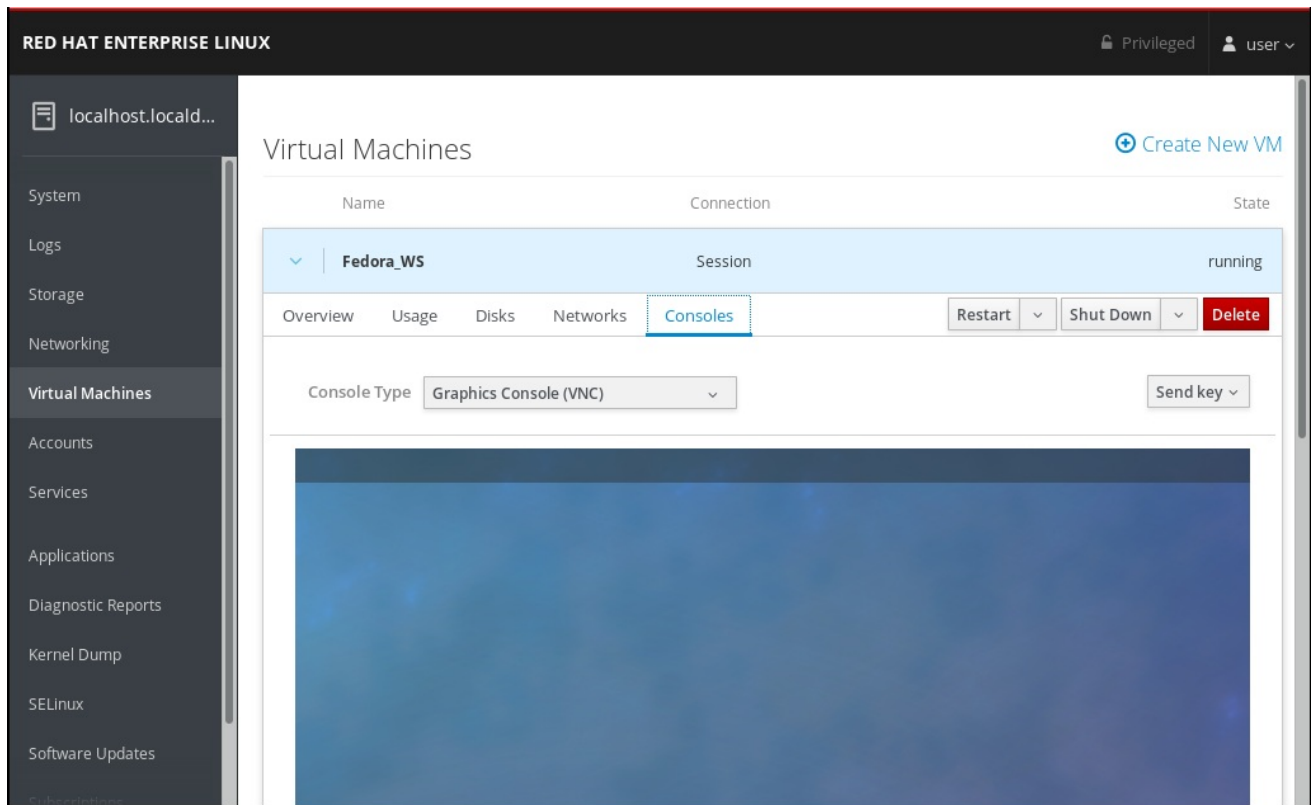


#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

#### Procedure

1. Click a row with the name of the virtual machine whose graphical console you want to view.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Consoles**.  
The graphical console appears in the Cockpit pane.



You can interact with the virtual machine console using the mouse and keyboard in the same manner you interact with a real machine. The display in the virtual machine console reflects the activities being performed on the virtual machine.

The server on which Cockpit is running can intercept specific key combinations preventing them from being sent to the virtual machine. To send the **Ctrl+Alt+Del** combination to the virtual machine, click **Ctrl+Alt+Del**.

From the virtual machine pane, click the Send key menu and select the key sequence to send.

### Additional Resources

- For details on viewing the graphical console in a remote viewer, see [Section 3.9.2, “Viewing virtual machine consoles in remote viewers using Cockpit”](#).
- For details on viewing the serial console in Cockpit, see [Section 3.9.3, “Viewing the virtual machine serial console in Cockpit”](#).

## 3.9.2. Viewing virtual machine consoles in remote viewers using Cockpit

You can view the virtual machine’s consoles in a remote viewer. The connection can be made by Cockpit or manually.

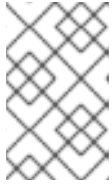
### 3.9.2.1. Viewing the graphical console in a remote viewer

You can view the graphical console of a selected virtual machine in a remote viewer. The virtual machine console shows the graphical output of the virtual machine.

You can launch Virt Viewer from within Cockpit.

### Prerequisites

- Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

- Before you can view the graphical console in Virt Viewer, Virt Viewer must be installed on the machine to which Cockpit is connected. To view information on installing Virt Viewer, select the **Graphics Console in Desktop Viewer** Console Type and click **More Information** in the Consoles window.

[▼ More Information](#)

Clicking "Launch Remote Viewer" will download a .vv file and launch *Remote Viewer*.

*Remote Viewer* is available for most operating systems. To install it, search for it in GNOME Software or run the following:

- **RHEL, CentOS:** `sudo yum install virt-viewer`
- **Fedora:** `sudo dnf install virt-viewer`
- **Ubuntu, Debian:** `sudo apt-get install virt-viewer`
- **Windows:** Download the MSI from [virt-manager.org](http://virt-manager.org)

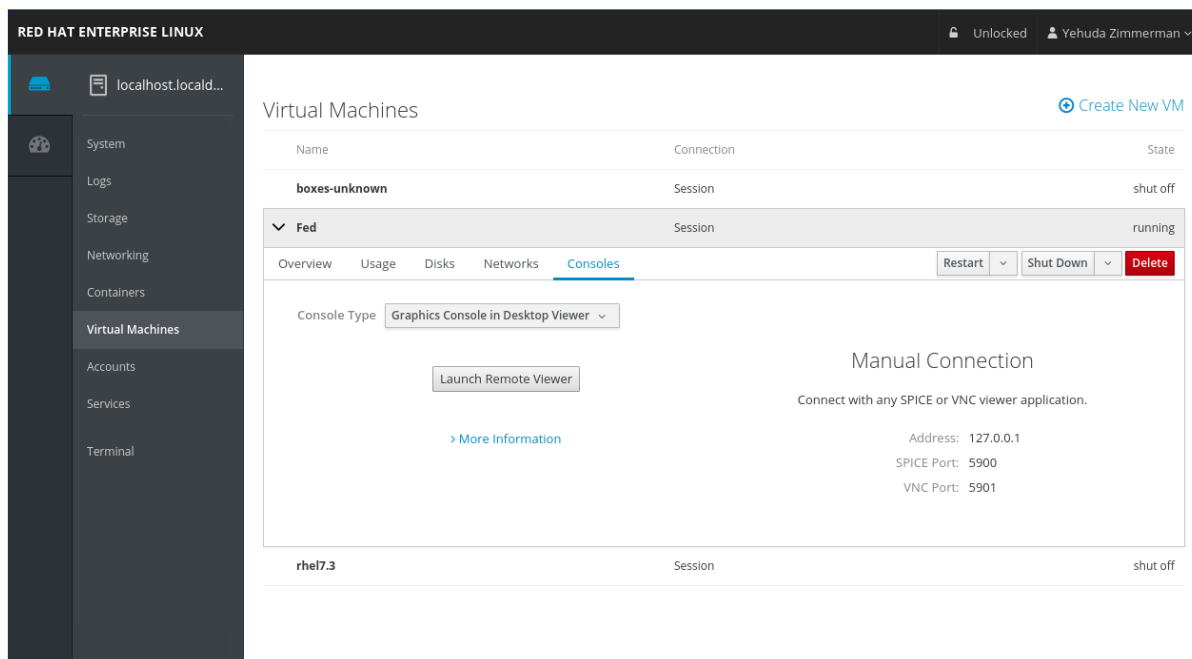


#### NOTE

Some browser extensions and plug-ins do not allow Cockpit to open Virt Viewer.

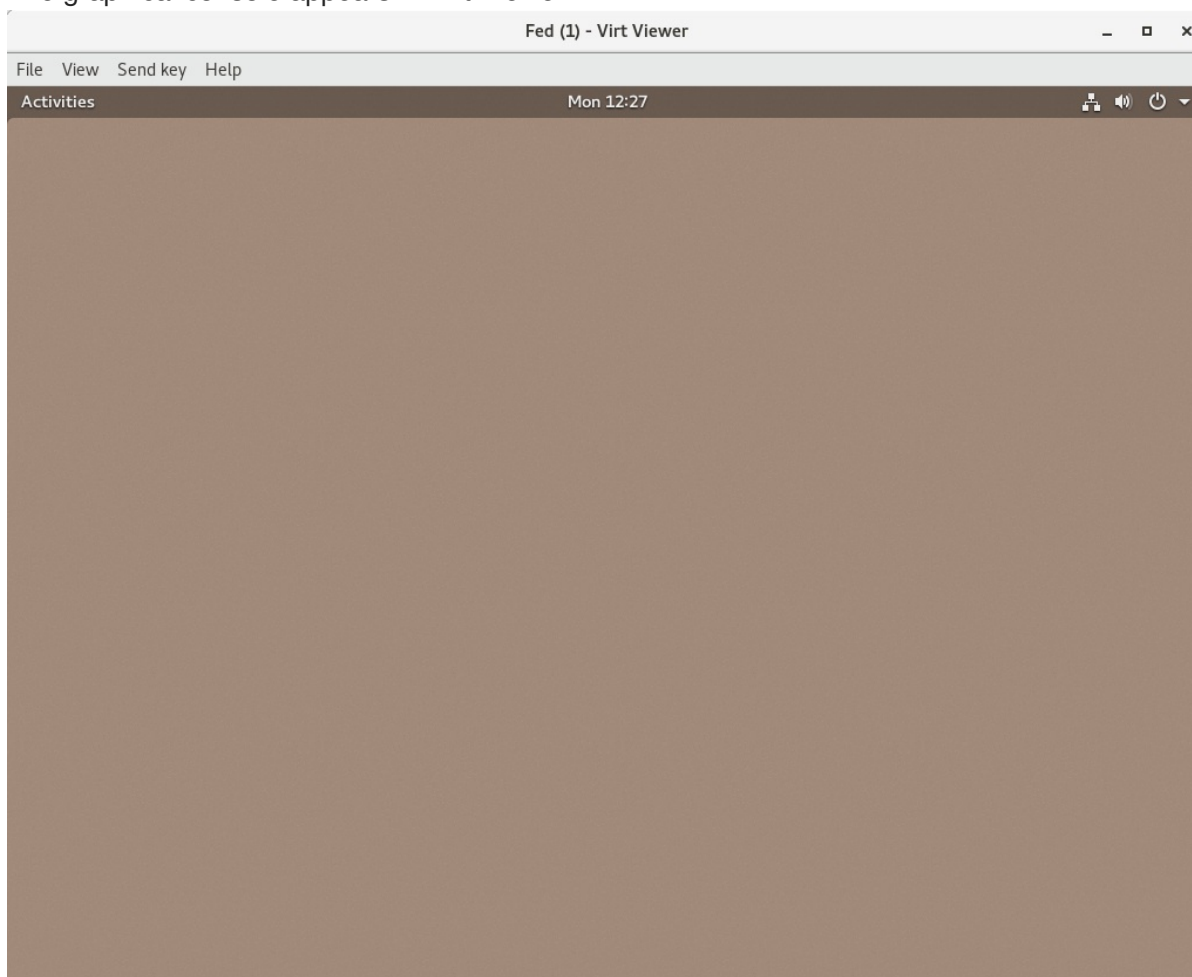
#### Procedure

1. Click a row with the name of the virtual machine whose graphical console you want to view. The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Consoles**.  
The graphical console appears in the Cockpit window.
3. Select the **Graphics Console in Desktop Viewer** Console Type.



#### 4. Click **Launch Remote Viewer**.

The graphical console appears in Virt Viewer.



You can interact with the virtual machine console using the mouse and keyboard in the same manner you interact with a real machine. The display in the virtual machine console reflects the activities being performed on the virtual machine.

The server on which Cockpit is running can intercept specific key combinations (for example, Ctrl+Alt+F1) to prevent them from being sent to the virtual machine. You can use the menu:Send key

menu to send these sequences. From the virtual machine window, click the menu:Send key menu and select the key sequence to send. In addition, from this menu you can also capture the screen output.

### Additional Resources

- For details on viewing the graphical console in a remote viewer using a manual connection, see [Section 3.9.2.2, “Viewing the graphical console in a remote viewer connecting manually”](#).
- For details on viewing the graphical console in Cockpit, see [Section 3.9.1, “Viewing the virtual machine graphical console in Cockpit”](#).
- For details on viewing the serial console in Cockpit, see [Section 3.9.3, “Viewing the virtual machine serial console in Cockpit”](#).

### 3.9.2.2. Viewing the graphical console in a remote viewer connecting manually

You can view the graphical console of a selected virtual machine in a remote viewer. The virtual machine console shows the graphical output of the virtual machine.

Cockpit provides the information necessary to launch any SPICE or VNC viewer to view the virtual machine console.

### Prerequisites

- Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



#### NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

- Before you can view the graphical console in a remote viewer, ensure that a SPICE or VNC viewer application is installed on the machine to which Cockpit is connected. To view information on installing Virt Viewer, select the **Graphics Console in Desktop Viewer** Console Type and click **More Information** in the Consoles window.

[▼ More Information](#)

Clicking "Launch Remote Viewer" will download a .vv file and launch *Remote Viewer*.

*Remote Viewer* is available for most operating systems. To install it, search for it in GNOME Software or run the following:

- **RHEL, CentOS:** `sudo yum install virt-viewer`
- **Fedora:** `sudo dnf install virt-viewer`
- **Ubuntu, Debian:** `sudo apt-get install virt-viewer`
- **Windows:** Download the MSI from [virt-manager.org](http://virt-manager.org)

## Procedure

You can view the virtual machine graphics console in any SPICE or VNC viewer application.

1. Click a row with the name of the virtual machine whose graphical console you want to view.  
The row expands to reveal the Overview pane with basic information about the selected virtual machine and controls for shutting down and deleting the virtual machine.
2. Click **Consoles**.  
The graphical console appears in the Cockpit window.
3. Select the **Graphics Console in Desktop Viewer** Console Type.  
The following Manual Connection information appears on the right side of the pane.

## Manual Connection

Connect with any SPICE or VNC viewer application.

Address: 127.0.0.1

SPICE Port: 5900

VNC Port: 5901

4. Enter the information in the SPICE or VNC viewer.

For more information, see the documentation for the SPICE or VNC viewer.

## Additional Resources

- For details on viewing the graphical console in a remote viewer using Cockpit to make the connection, see [Section 3.9.2.1, “Viewing the graphical console in a remote viewer”](#).
- For details on viewing the graphical console in Cockpit, see [Section 3.9.1, “Viewing the virtual machine graphical console in Cockpit”](#).
- For details on viewing the serial console in Cockpit, see [Section 3.9.3, “Viewing the virtual machine serial console in Cockpit”](#).

### 3.9.3. Viewing the virtual machine serial console in Cockpit

You can view the serial console of a selected virtual machine in Cockpit.

## Prerequisites

Before using Cockpit to manage virtual machines, you must install the Cockpit virtual machine plug-in.



## NOTE

If the Cockpit plug-in is not installed, see [Section 3.2, “Setting up Cockpit to manage virtual machines”](#) for information about installing the Cockpit virtual machine plug-in.

## Procedure

1. Click a row with the name of the virtual machine whose serial console you want to view.  
The row expands to reveal the Overview pane with basic information about the selected virtual



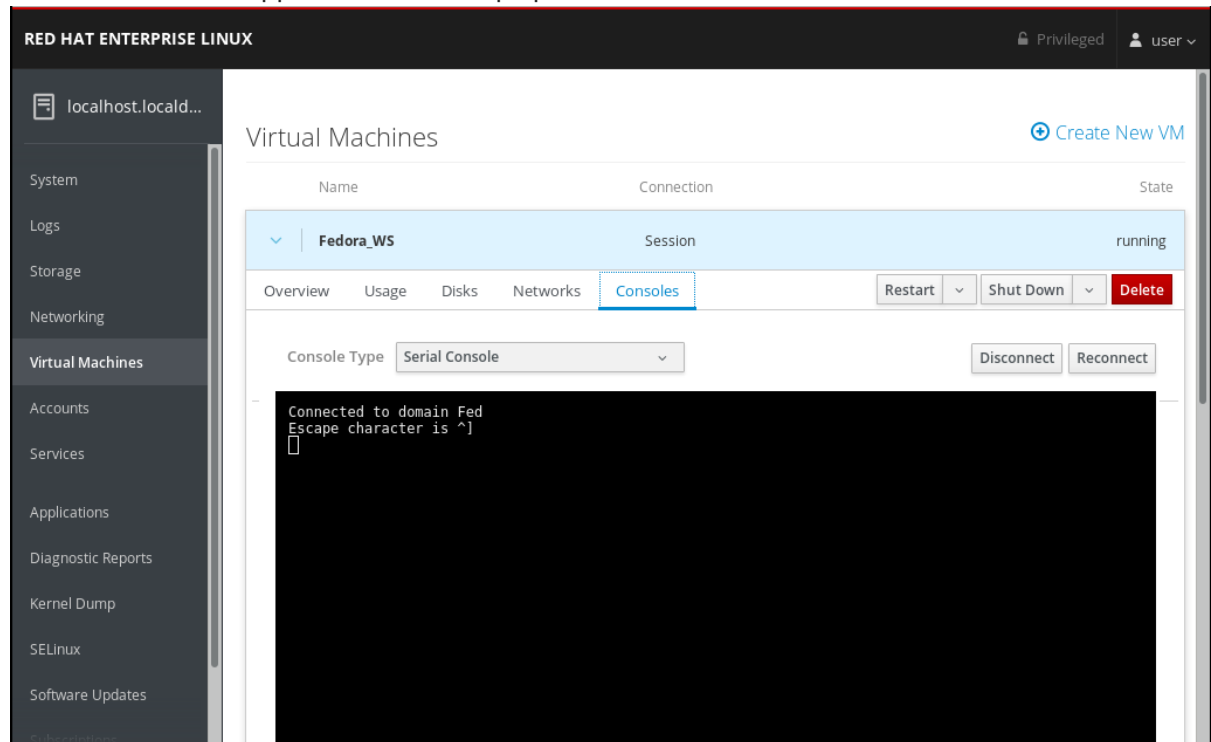
machine and controls for shutting down and deleting the virtual machine.

2. Click **Consoles**.

The graphical console appears in the Cockpit pane.

3. Select the **Serial Console** Console Type.

The serial console appears in the Cockpit pane.



You can disconnect and reconnect the serial console from the virtual machine.

- To disconnect the serial console from the virtual machine, click **Disconnect**.
- To reconnect the serial console to the virtual machine, click **Reconnect**.

### Additional Resources

- For details on viewing the graphical console in Cockpit, see [Section 3.9.1, “Viewing the virtual machine graphical console in Cockpit”](#).
- For details on viewing the graphical console in a remote viewer, see [Section 3.9.2, “Viewing virtual machine consoles in remote viewers using Cockpit”](#).

## CHAPTER 4. CREATING VIRTUAL MACHINE SNAPSHOTS IN RHEL 8

*Snapshots* are files that can contain the disk image and memory state of a virtual machine.

### 4.1. UNDERSTANDING VIRTUAL MACHINE SNAPSHOTS

You can create a snapshot from any virtual machine on the host. The snapshot metadata can be supplied [in an XML file](#) or [on the command line](#). You can also create a snapshot by cloning an existing snapshot.

Snapshots can include:

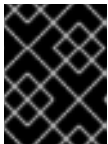
- Only the disk of the virtual machine
- The disk and memory state of the virtual machine

Snapshots can be used to:

- Save a clean copy of an operating system
- Save a virtual machine's state before performing a potentially destructive operation

There are two kinds of snapshots:

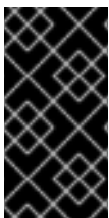
- *Internal snapshots* are contained completely within a qcow2 file and are fully supported by **libvirt**, allowing for creating, deleting, and reverting of snapshots. This is the default setting used by **libvirt** when creating a snapshot, especially when no option is specified. This file type take slightly longer than others for creating the snapshot. In addition, internal snapshots require qcow2 disks.



#### IMPORTANT

Internal snapshots are not being actively developed, and Red Hat discourages their use.

- *External snapshots* work with any type of original disk image, can be taken with no virtual machine downtime, and are more stable and reliable. As such, external snapshots are recommended for use on KVM guest virtual machines. However, external snapshots are currently not fully implemented. A workaround is described on [libvirt upstream pages](#).



#### IMPORTANT

Red Hat Enterprise Linux 8 only supports creating snapshots while the guest virtual machine is paused or powered down. Creating snapshots of running guests (also known as *live snapshots*) is available on Red Hat Virtualization. For details, call your service representative.

### 4.2. CREATING VIRTUAL MACHINE SNAPSHOTS FROM XML FILES

You can create snapshots using XML files that contain the metadata for the snapshots.

For information about the XML format for snapshot metadata, see [the libvirt upstream website](#).

### 4.2.1. Creating virtual machine snapshots using XML files

The **virsh snapshot-create** command creates a snapshot for the specified virtual machine with the snapshot metadata in a named XML file. The command can use the virtual machine *name*, *id*, or *uid* to specify the virtual machine.

The named XML file contains the snapshot metadata. The snapshot *name*, *description*, and *disks* elements must be specified in the XML file.

#### Procedure

1. Create an XML file with the virtual machine snapshot metadata.  
The following is an example of a minimal XML file for creating a snapshot:

```
<domainsnapshot>
  <name>Snapshot1</name>
  <description>First snapshot of the domain</description>
  <disks>
    <disk name='vda' snapshot='external' />
  </disks>
</domainsnapshot>
```

There are several methods for creating the XML file:

- Using a text editor.
  - Using the **virsh snapshot-dumpxml** command to output the metadata of the current snapshot to an XML file. You can edit the file, if necessary.
  - Using the **virsh snapshot-create-as** command with the **print-xml** option to output the metadata of a specified snapshot to an XML file. You can edit the file, if necessary.
2. Run the **virsh snapshot-create** command.

```
# virsh snapshot-create domain xml-file
```

```
Domain snapshot Snapshot1 created
```

The virtual machine snapshot is created from the specified virtual machine using the metadata in the specified XML file.

#### Additional resources

- For information about the available options for the **virsh create-snapshot** command, see the relevant man page.

## 4.3. CREATING VIRTUAL MACHINE SNAPSHOTS ENTERING METADATA ON THE COMMAND LINE

You can create snapshots by entering the metadata for the snapshots as options on the command line.

### 4.3.1. Creating virtual machine snapshots entering metadata on the command line

The **virsh snapshot-create-as** command creates a snapshot for the specified virtual machine with the snapshot metadata specified in the command options. The command can use the virtual machine *name*, *id*, or *uid* to specify the virtual machine.

### Procedure

- Run the **virsh snapshot-create-as** command.

```
# virsh snapshot-create-as domain snapshot-name description
```

The virtual machine snapshot is created from the specified virtual machine using the metadata in the command options.

For example:

```
# virsh snapshot-edit 5 Snapshot1
Domain snapshot Snapshot1 created.
```

The **print-xml** option can be used to display the snapshot XML based on the **virsh snapshot-create-as** command.

For example:

```
# virsh snapshot-create-as 5 Snapshot1 --printxml
<domainsnapshot>
  <name>Snapshot1</name>
</domainsnapshot>
```

### Additional resources

- For information about the available options for the **virsh create-snapshot-as** command, see the relevant man page.

## 4.4. CLONING VIRTUAL MACHINE SNAPSHOTS

You can clone existing snapshots.

### 4.4.1. Cloning virtual machine snapshots

The **virsh snapshot-edit --clone** command clones a specified snapshot for the specified virtual machine. The command can use the virtual machine *name*, *id*, or *uid* to specify the virtual machine.

### Procedure

1. Run the **virsh snapshot-edit --clone** command.

```
# virsh snapshot-edit domain snapshot-name --clone
```

The specified virtual machine snapshot XML file opens in the default text editor. For example, vi.

2. Edit the virtual machine snapshot XML file.  
Ensure that you change the value of the **name** field.
3. Save the edited virtual machine snapshot XML file.  
The snapshot is cloned.

For example:

```
# virsh snapshot-edit 5 Snapshot1 --clone  
Snapshot Snapshot1 cloned to Snapshot2.
```

#### 4.4.2. Related information

- For a full description of all **virsh snapshot-edit** options, refer to the **virsh snapshot-edit** man page.

### 4.5. RELATED INFORMATION

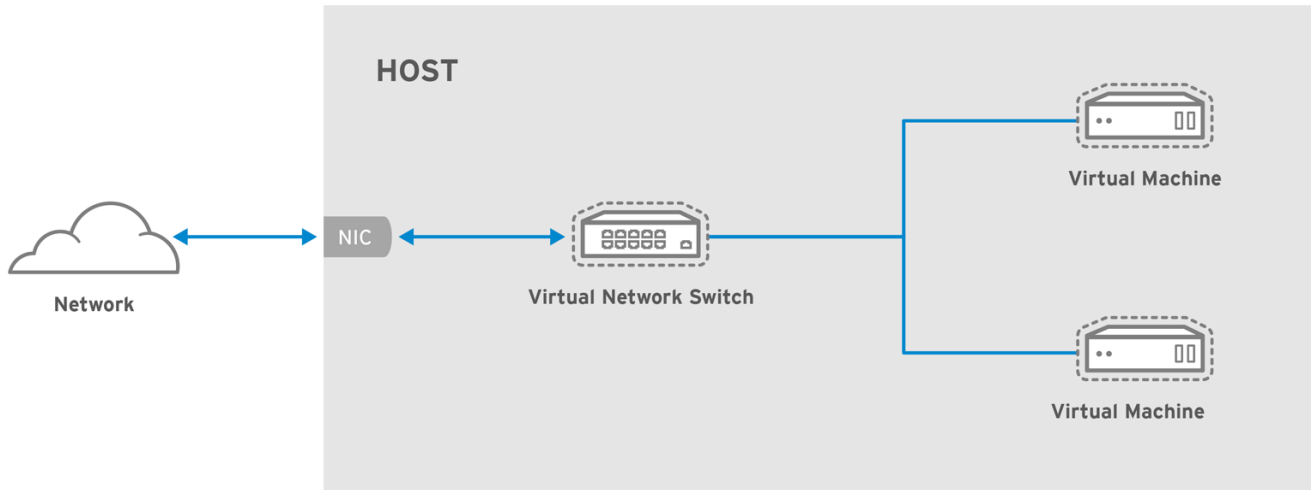
- For information about the XML format for snapshot metadata, see [the libvirt upstream website](#).

## CHAPTER 5. UNDERSTANDING VIRTUAL NETWORKING

The connection of virtual machines to other devices and locations on a network has to be facilitated by the host hardware.

Virtual networking uses the concept of a virtual network switch. A virtual network switch is a software construct that operates on a host physical machine server. Virtual machines (guests) connect to the network through the virtual network switch.

The following figure shows a virtual network switch connecting two virtual machines to the network:



RHEL\_437030\_0217

From the perspective of a guest operating system, a virtual network connection is the same as a physical network connection.

Host physical machine servers view virtual network switches as network interfaces. When the `libvirt` daemon (**libvirtd**) is first installed and started, the default network interface representing the virtual network switch is **virbr0**.

This interface can be viewed with the **ip** command like any other network interface.

```
$ ip addr show virbr0
3: virbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN link/ether 1b:c4:94:cf:fd:17 brd ff:ff:ff:ff:ff:ff
inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
```

By default, all guests on a single host are connected to the same **libvirt** virtual network, named **default**. Guests on this network can make the following connections:

- **With each other and with the virtualization host**  
Both inbound and outbound traffic is possible, but is affected by the firewalls in the guest operating system's network stack and by **libvirt** network filtering rules attached to the guest interface.
- **With other hosts on the network beyond the virtualization host**  
Only outbound traffic is possible and is affected by Network Address Translation (NAT) rules, as well as the host system's firewall.

For more information on the default configuration, see [Section 5.6, “Virtual networking default configuration”](#).

If needed, guest interfaces can instead be set to one of the following modes:

- [Routed mode](#)
- [Bridged mode](#)
- [Isolated mode](#)

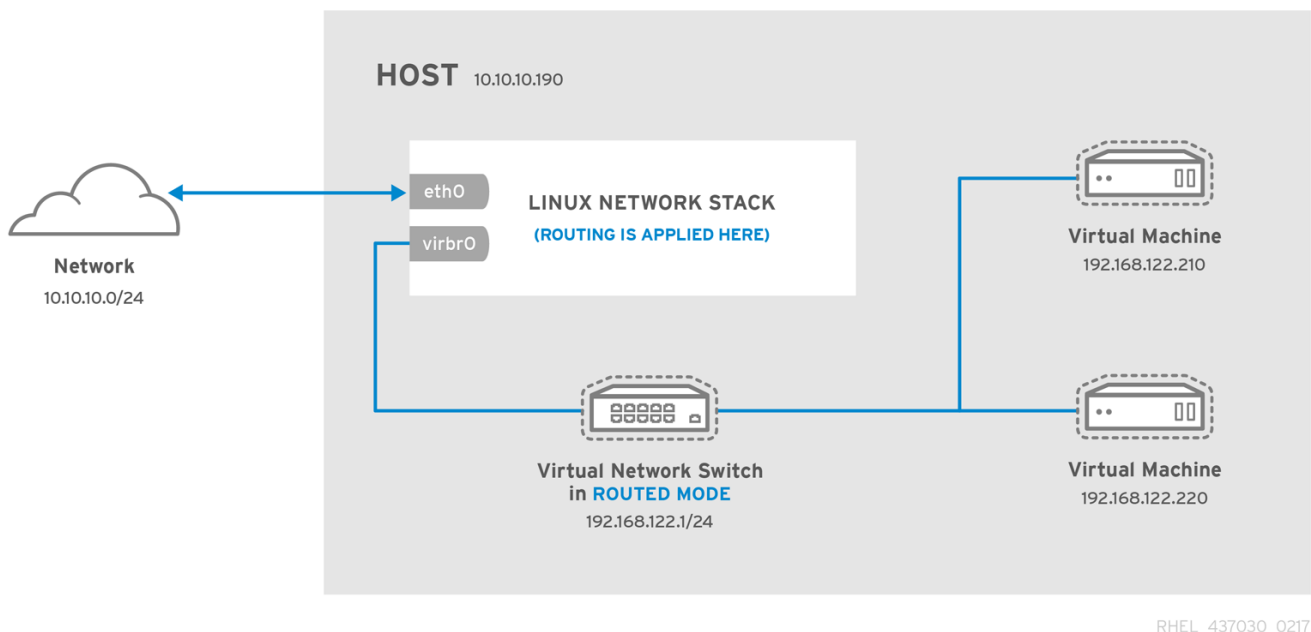
The virtual network uses network address translation (NAT) to assign IP address ranges to virtual networks and **dnsmasq** to automatically assign IP addresses to virtual machine network interface cards (NICs) and connect to a domain name service (DNS).

The following features are available for virtual networking:

- [Network address translation \(NAT\)](#)
- [DNS and DHCP](#)

## 5.1. VIRTUAL NETWORKING IN ROUTED MODE

When using *Routed* mode, the virtual switch connects to the physical LAN connected to the host physical machine, passing traffic back and forth without the use of NAT. The virtual switch can examine all traffic and use the information contained within the network packets to make routing decisions. When using this mode, all of the virtual machines are in their own subnet, routed through a virtual switch. This enables incoming connections, but requires extra routing-table entries for systems on the external network. Routed mode operates at Layer 3 of the OSI networking model.



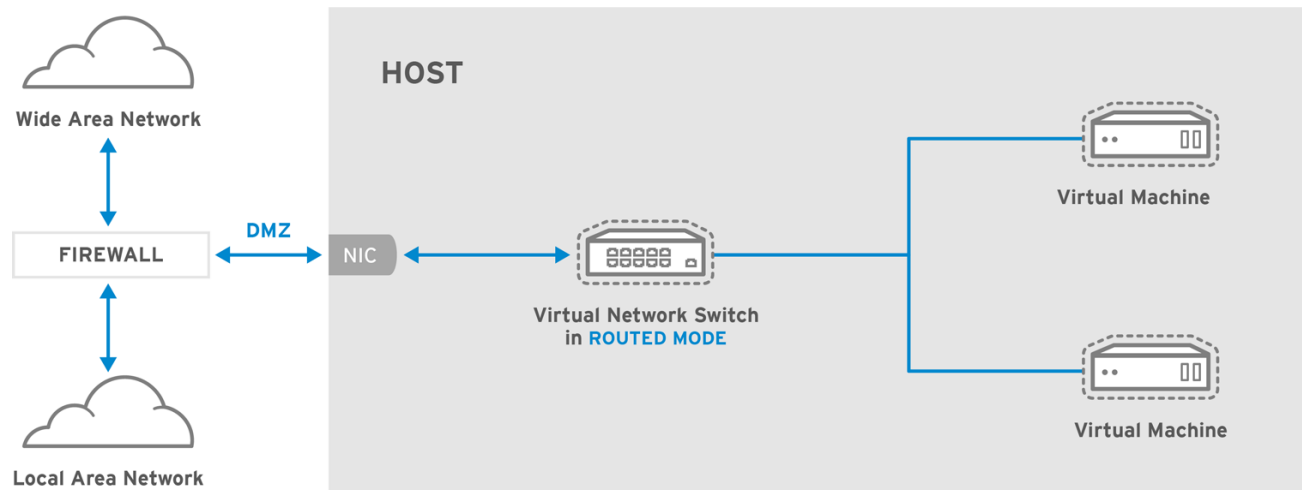
### Common topologies

The following are two common topologies in which routed mode is used:

- DMZ
- Virtual server hosting

### DMZ

You can create a network where one or more nodes are placed in a controlled sub-network for security reasons. Such a sub-network is known as a demilitarized zone (DMZ).

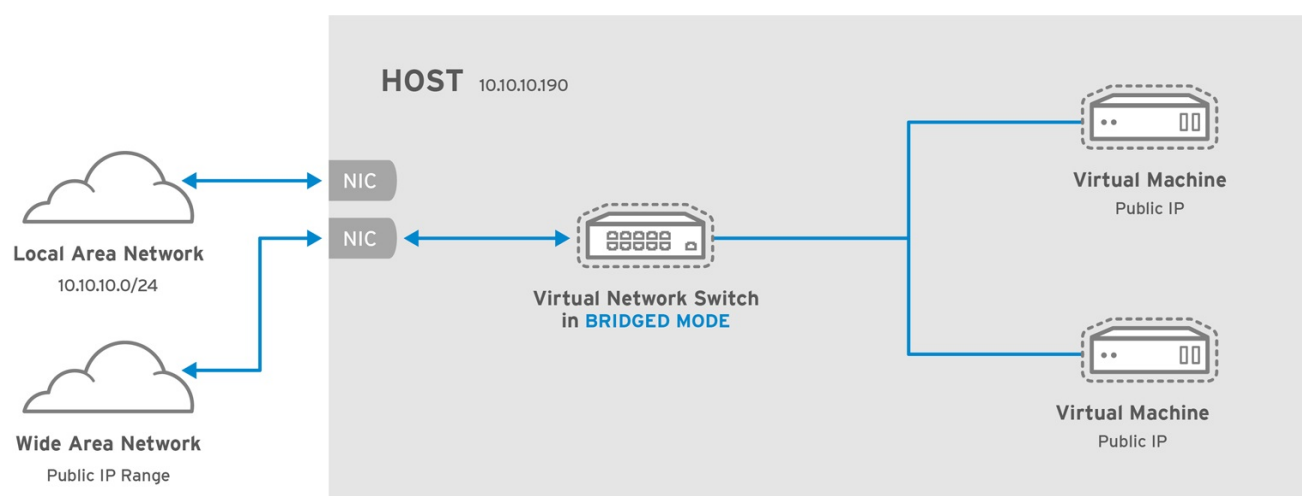


RHEL\_437030\_0217

Host physical machines in a DMZ typically provide services to WAN (external) host physical machines as well as LAN (internal) host physical machines. Since this requires them to be accessible from multiple locations, and considering that these locations are controlled and operated in different ways based on their security and trust level, routed mode is the best configuration for this environment.

### Virtual server hosting

A virtual server hosting provider may have several host physical machines, each with two physical network connections. One interface is used for management and accounting, the other is for the virtual machines to connect through. Each virtual machine has its own public IP address, but the host physical machines use private IP addresses so that management of the virtual machines can only be performed by internal administrators.

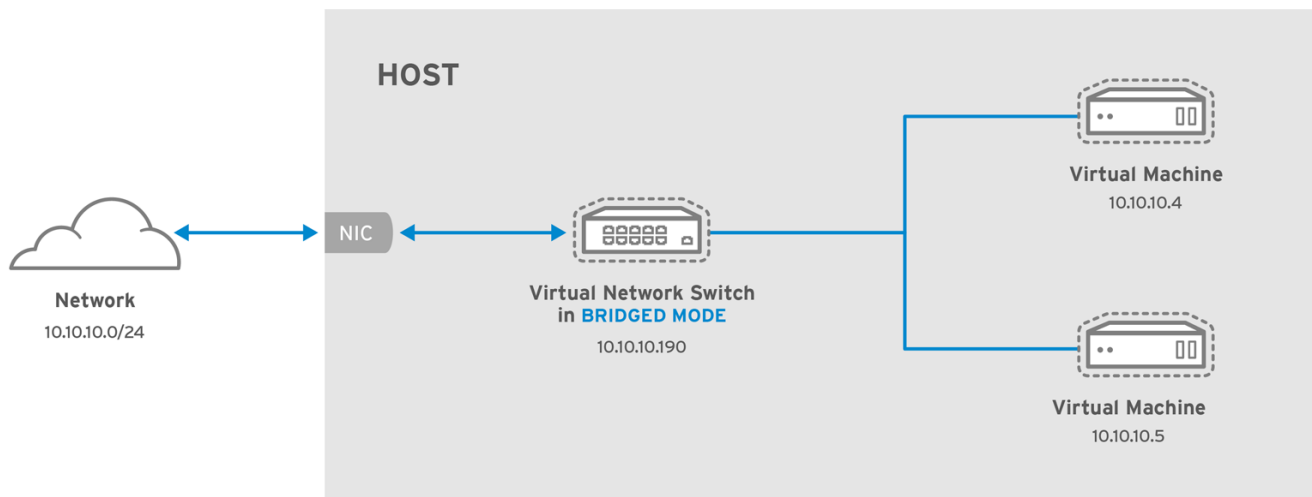


RHEL\_437030\_0217

## 5.2. VIRTUAL NETWORKING IN BRIDGED MODE

When using *Bridged* mode, guests are connected to a bridge device that is also connected directly to a physical ethernet device connected to the local ethernet. As a result, the guest directly visible on the physical network. This enables incoming connections, but does not require any extra routing-table entries.





RHEL\_437030\_0217

For basic outbound-only network access from virtual machines, no additional network setup is usually needed, because the default network is installed along with the **libvirt** package, and automatically started when the **libvirtd** service is started.

If more advanced functionality is needed, additional networks can be created and configured using **virsh**, and the guest XML configuration file can be edited to use one of these new networks.

All of the guest virtual machines appear within the same subnet as the host physical machine. All other physical machines on the same physical network are aware of the virtual machines, and can access them. Bridging operates on Layer 2 of the OSI networking model.

It is possible to use multiple physical interfaces on the hypervisor by joining them together with a bond. The bond is then added to a bridge and then guest virtual machines are added onto the bridge as well. However, the bonding driver has several modes of operation, and only a few of these modes work with a bridge where virtual guest machines are in use.



### WARNING

When using bridged mode, the only bonding modes that should be used with a guest virtual machine are Mode 1, Mode 2, and Mode 4. Using modes 0, 3, 5, or 6 is likely to cause the connection to fail. Also note that Media-Independent Interface (MII) monitoring should be used to monitor bonding modes, as Address Resolution Protocol (ARP) monitoring does not work.

### Common scenarios

The most common use cases for bridged mode include:

- Deploying guest virtual machines in an existing network alongside host physical machines, making the difference between virtual and physical machines transparent to the end user.
- Deploying guest virtual machines without making any changes to existing physical network configuration settings.

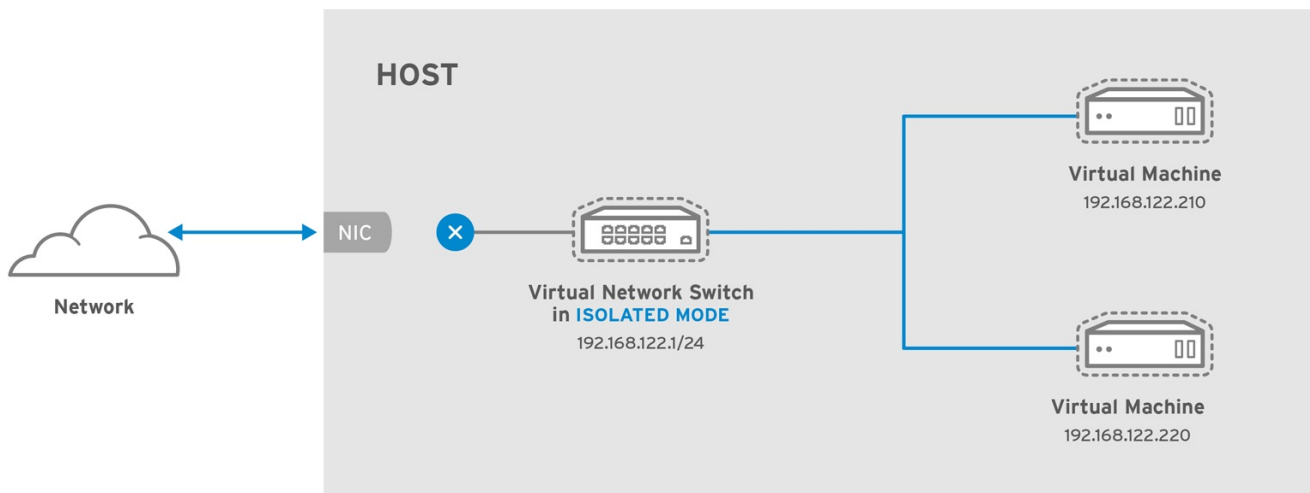
- Deploying guest virtual machines that must be easily accessible to an existing physical network. Placing guest virtual machines on a physical network where they must access services within an existing broadcast domain, such as DHCP.
- Connecting guest virtual machines to an existing network where VLANs are used.

### Additional resources

- For a detailed explanation of `bridge_opts` parameters, used to configure bridged networking mode, see the [Red Hat Virtualization Administration Guide](#).

## 5.3. VIRTUAL NETWORKING ISOLATED MODE

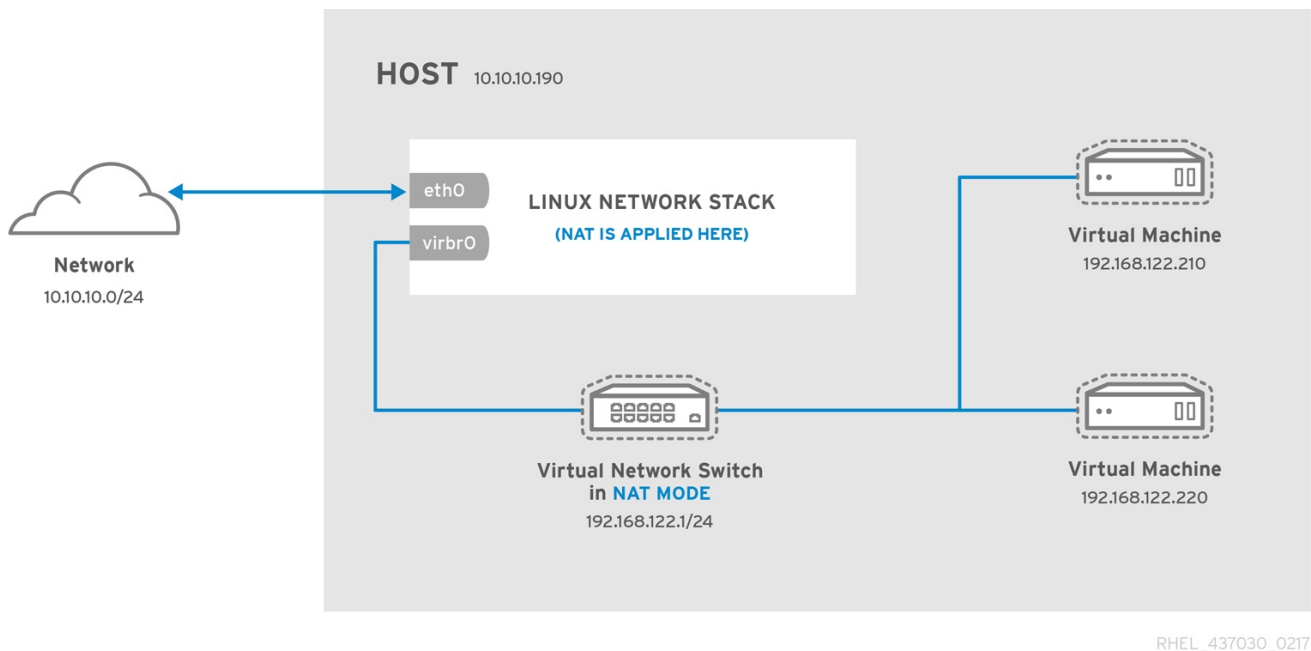
When using *Isolated* mode, guests connected to the virtual switch can communicate with each other, and with the host physical machine, but their traffic will not pass outside of the host physical machine, and they cannot receive traffic from outside the host physical machine. Using `dnsmasq` in this mode is required for basic functionality such as DHCP. However, even if this network is isolated from any physical network, DNS names are still resolved. Therefore, a situation can arise when DNS names resolve but ICMP echo request (ping) commands fail.



RHEL\_437030\_0217

## 5.4. VIRTUAL NETWORKING NETWORK ADDRESS TRANSLATION

By default, virtual network switches operate in NAT mode. They use IP masquerading rather than Source-NAT (SNAT) or Destination-NAT (DNAT). IP masquerading enables connected guests to use the host physical machine IP address for communication to any external network. By default, computers that are placed externally to the host physical machine cannot communicate to the guests inside when the virtual network switch is operating in NAT mode.



### WARNING

Virtual network switches use NAT configured by iptables rules. Editing these rules while the switch is running is not recommended, because incorrect rules may result in the switch being unable to communicate.

If the switch is not running, you can set the public IP range for forward mode NAT in order to create a port masquerading range by running:

```
# iptables -j SNAT --to-source [start]-[end]
```

## 5.5. VIRTUAL NETWORKING DNS AND DHCP

The **libvirt** package includes **dnsmasq** to provide a Dynamic Host Configuration Protocol (DHCP) server and a Domain Name System (DNS) forwarder for virtual networks.

The **dnsmasq** DHCP service can assign a pool of addresses to a virtual network switch. IP information can be assigned to virtual machines via DHCP.

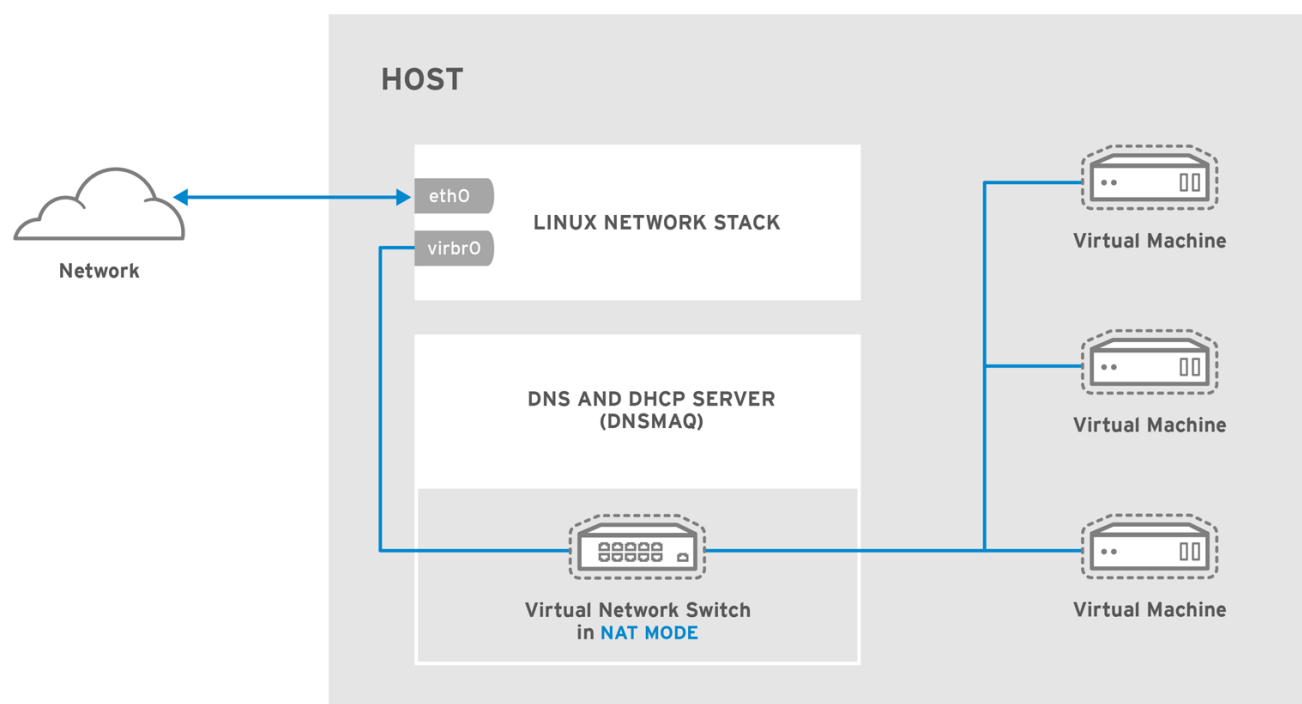
**dnsmasq** accepts DNS queries from virtual machines on the virtual network and forwards them to a real DNS server.

An instance of **dnsmasq** is automatically configured and started by **libvirt** for each virtual network switch that needs it.

## 5.6. VIRTUAL NETWORKING DEFAULT CONFIGURATION

When the **libvirtd** daemon (**libvirtd**) is first installed, it contains an initial virtual network switch configuration in NAT mode. This configuration is used so that installed guests can communicate to the external network through the host physical machine. The following figure shows the default configuration

for **libvirtd**:



RHEL\_437030\_0217



## NOTE

A virtual network can be restricted to a specific physical interface. This may be useful on a physical system that has several interfaces (for example, **eth0**, **eth1**, and **eth2**). This is only useful in routed and NAT modes, and can be defined in the **dev=<interface>** option, or in Cockpit when creating a new virtual network.

## CHAPTER 6. FEATURE SUPPORT AND LIMITATIONS IN RHEL 8 VIRTUALIZATION

This document provides information on feature support and restrictions in Red Hat Enterprise Linux 8 (RHEL 8) virtualization.



### WARNING

No actual support for RHEL 8 features is provided until it becomes globally available. For the Alpha and Beta stages, the following speaks about the planned scope of support for virtualization, and is subject to change.

### 6.1. RHEL 8 VIRTUALIZATION SUPPORT - THE BASICS

A set of support limitations applies to virtualization in Red Hat Enterprise Linux 8 (RHEL 8). This means that when you use certain features or exceed a certain amount of allocated resources when using virtual machines in RHEL 8, Red Hat will not support these guests unless you have a specific subscription plan.

Features listed in [Section 6.2, “Recommended features in RHEL 8 virtualization”](#) have been tested and certified by Red Hat to work with the KVM hypervisor on a RHEL 8 system. Therefore, they are fully supported and recommended for use in virtualization in RHEL 8.

Features listed in [Section 6.3, “Unsupported features in RHEL 8 virtualization”](#) may work, but are not supported and not intended for use in RHEL 8. Therefore, Red Hat strongly recommends not using these features in RHEL 8 with KVM.

[Section 6.4, “Resource allocation limits in RHEL 8 virtualization”](#) lists the maximum amount of specific resources supported on a KVM guest in RHEL 8. Guests that exceed these limits are not supported by Red Hat.

In addition, unless stated otherwise, all features and solutions used by the documentation for RHEL 8 virtualization are supported. However, some of these have not been completely tested and therefore may not be fully optimized.



### NOTE

Many of these limitations do not apply to other virtualization solutions provided by Red Hat, such as Red Hat Virtualization (RHV) or Red Hat OpenStack Platform (RHOSP).

### 6.2. RECOMMENDED FEATURES IN RHEL 8 VIRTUALIZATION

The following features are recommended for use with the KVM hypervisor included with Red Hat Enterprise Linux 8 (RHEL 8):

- **Host system architectures**

Red Hat Enterprise Linux with KVM is supported only on the following host architectures:

- AMD64 and Intel 64

- IBM Z
- IBM POWER8
- IBM POWER9

**NOTE**

Currently, RHEL 8 documentation primarily describes AMD64 and Intel 64 features and usage.

- **Guest operating systems**

Red Hat supports KVM virtual machines that use the following operating systems:

- Red Hat Enterprise Linux 6 and later
- Microsoft Windows 10 and later
- Microsoft Windows Server 2016 and later

- **Q35 guests**

The recommended machine type for KVM guest virtual machines is QEMU Q35, which emulates the ICH9 chipset.

**Additional resources**

- For information about unsupported guest OS types and features in RHEL 8 virtualization, see [Section 6.3, “Unsupported features in RHEL 8 virtualization”](#).
- For information about the maximum supported amounts of resources that can be allocated to a virtual machine, see [Section 6.4, “Resource allocation limits in RHEL 8 virtualization”](#).

## 6.3. UNSUPPORTED FEATURES IN RHEL 8 VIRTUALIZATION

The following features are not supported by the KVM hypervisor included with Red Hat Enterprise Linux 8 (RHEL 8):

**NOTE**

Many of these limitations do not apply to other virtualization solutions provided by Red Hat, such as Red Hat Virtualization (RHV) or Red Hat OpenStack Platform (RHOSP).

- **Guest operating systems**

KVM virtual machines using the following guest operating systems on a RHEL 8 host are not supported:

- Microsoft Windows 8.1 and earlier
- Microsoft Windows Server 2012 and earlier
- macOS
- Solaris for x86 systems
- Any OS released prior to 2009

- **vCPU hot unplug**  
Removing a virtual CPU (vCPU) from a running KVM virtual machine, also referred to as a vCPU hot unplug, is unsupported in RHEL 8 and using it may lead to VM crashes. Therefore, Red Hat strongly advises against its use.
- **Memory hot unplug**  
Decreasing the memory limit allocated to running a KVM virtual machine, also referred to as a memory hot unplug, is unsupported in RHEL 8 and using it may lead to VM crashes. Therefore, Red Hat strongly advises against its use.
- **I/O throttling**  
Configuring maximum input and output levels for operations on virtual disk, also known as I/O throttling, is not supported in RHEL 8.
- **Storage live migration**  
Migrating a disk image of a running virtual machine between hosts is not supported in RHEL 8.
- **Live snapshots**  
Creating or loading a snapshot of a running virtual machine, also referred to as a live snapshot, is not supported in RHEL 8.
- **S3 and S4 system power states**  
Suspending a virtual machine to the **Suspend to RAM** (S3) or **Suspend to disk** (S4) system power states is not supported. Note that these features are disabled by default, and enabling them will make your virtual machine not supportable by Red Hat.

### Additional resources

- For information about supported guest OS types and recommended features in RHEL 8 virtualization, see [Section 6.2, “Recommended features in RHEL 8 virtualization”](#).
- For information about the maximum supported amounts of resources that can be allocated to a virtual machine, see [Section 6.4, “Resource allocation limits in RHEL 8 virtualization”](#).

## 6.4. RESOURCE ALLOCATION LIMITS IN RHEL 8 VIRTUALIZATION

The following limits apply to virtualized resources that can be allocated to a single guest virtual machine on a Red Hat Enterprise Linux 8 (RHEL 8) host.



### NOTE

Many of these limitations do not apply to other virtualization solutions provided by Red Hat, such as Red Hat Virtualization (RHV) or Red Hat OpenStack Platform (RHOSP).

- **Maximum guest virtual machines per host**  
A single RHEL 8 host supports up to **4** guests running at the same time.
- **Maximum vCPUs per guest**  
RHEL 8 supports up to **240** vCPUs allocated to a single KVM guest.
- **PCI devices per guest**  
RHEL 8 supports **32** PCI device slots per virtual machine bus, and **8** PCI functions per device slot. This gives a theoretical maximum of 256 PCI functions per bus when multi-function capabilities are enabled in the virtual machine, and no PCI bridges are used.

Each PCI bridge adds a new bus, potentially enabling another 256 device addresses. However, some buses do not make all 256 device addresses available for the user; for example, the root bus has several built-in devices occupying slots.

- **Virtualized IDE devices**

KVM is limited to a maximum of **4** virtualized IDE devices per virtual machine.