

# Chapter 1. First Steps

## Table of Contents

[1.1. Why is Virtualization Useful?](#)

[1.2. Some Terminology](#)

[1.3. Features Overview](#)

[1.4. Supported Host Operating Systems](#)

[1.4.1. Host CPU Requirements](#)

[1.5. Installing Oracle VM VirtualBox and Extension Packs](#)

[1.6. Starting Oracle VM VirtualBox](#)

[1.7. VirtualBox Manager](#)

[1.7.1. The Machine List](#)

[1.7.2. The Details Pane](#)

[1.7.3. VirtualBox Manager Tools](#)

[1.7.4. Help Viewer](#)

[1.7.5. About VirtualBox Manager Wizards](#)

[1.8. Creating Your First Virtual Machine](#)

[1.8.1. Create Virtual Machine Wizard: Name and Operating System](#)

[1.8.2. \(Optional\) Create Virtual Machine Wizard: Unattended Guest OS Install](#)

[1.8.3. Create Virtual Machine Wizard: Hardware](#)

[1.8.4. Create Virtual Machine Wizard: Virtual Hard Disk](#)

[1.8.5. Create Virtual Machine Wizard: Summary](#)

[1.8.6. Some Examples of Unattended Installation](#)

[1.9. Running Your Virtual Machine](#)

[1.9.1. Starting a New VM for the First Time](#)

[1.9.2. Capturing and Releasing Keyboard and Mouse](#)

[1.9.3. Typing Special Characters](#)

[1.9.4. Changing Removable Media](#)

[1.9.5. Resizing the Machine's Window](#)

[1.9.6. Saving the State of the Machine](#)

[1.10. Using VM Groups](#)

[1.11. Snapshots](#)

[1.11.1. Taking, Restoring, and Deleting Snapshots](#)

[1.11.2. Snapshot Contents](#)

[1.12. Virtual Machine Configuration](#)

[1.13. Removing and Moving Virtual Machines](#)

[1.14. Cloning Virtual Machines](#)

[1.15. Importing and Exporting Virtual Machines](#)

[1.15.1. About the OVF Format](#)

[1.15.2. Importing an Appliance in OVF Format](#)

[1.15.3. Exporting an Appliance in OVF Format](#)

[1.16. Integrating with Oracle Cloud Infrastructure](#)

[1.16.1. Preparing for Oracle Cloud Infrastructure Integration](#)

[1.16.2. Creating an API Signing Key Pair](#)

[1.16.3. Uploading the Public Key to Oracle Cloud Infrastructure](#)

[1.16.4. Creating a Cloud Profile](#)

[1.16.5. Using the Cloud Profile Manager](#)

[1.16.6. Using Oracle VM VirtualBox With Oracle Cloud Infrastructure](#)

[1.16.7. Using Cloud Virtual Machines](#)

[1.16.8. Exporting an Appliance to Oracle Cloud Infrastructure](#)

[1.16.9. Importing an Instance from Oracle Cloud Infrastructure](#)

[1.16.10. Using a Cloud Network](#)[1.16.11. Using VBoxManage Commands With Oracle Cloud Infrastructure](#)[1.17. Preferences](#)[1.18. Alternative Front-Ends](#)[1.19. Soft Keyboard](#)[1.19.1. Using the Soft Keyboard](#)[1.19.2. Creating a Custom Keyboard Layout](#)[1.20. Monitoring of Virtual Machines](#)[1.20.1. VM Activity Overview](#)[1.20.2. Session Information Dialog](#)[1.21. The Log Viewer](#)

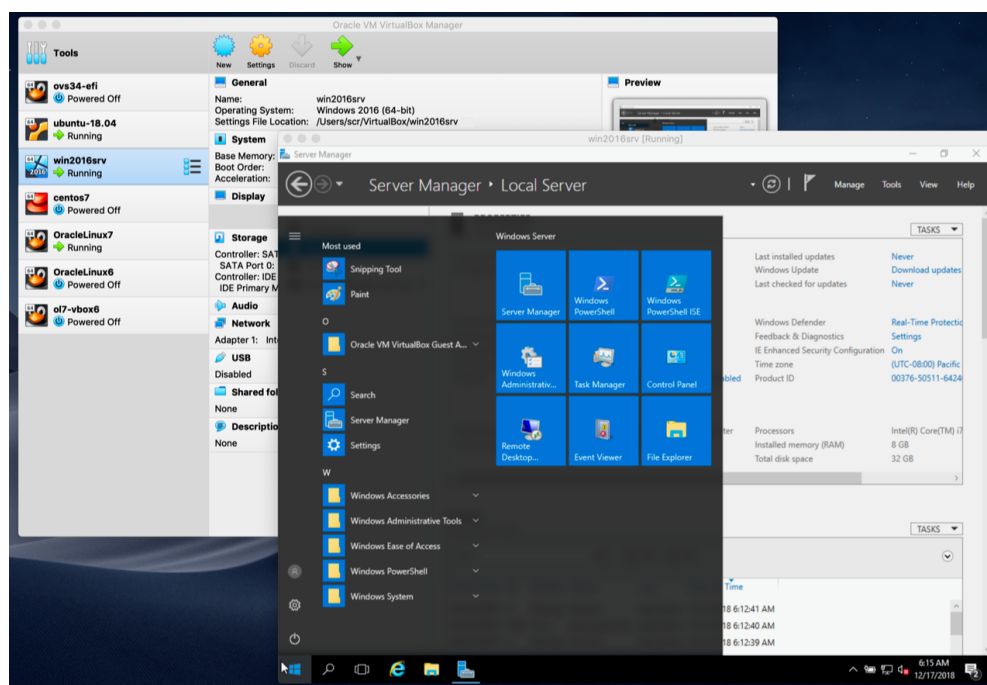
Welcome to Oracle VM VirtualBox.

Oracle VM VirtualBox is a cross-platform virtualization application. What does that mean? For one thing, it installs on your existing Intel or AMD-based computers, whether they are running Windows, macOS, Linux, or Oracle Solaris operating systems (OSes). Secondly, it extends the capabilities of your existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time. As an example, you can run Windows and Linux on your Mac, run Windows Server on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. You can install and run as many virtual machines as you like. The only practical limits are disk space and memory.

Oracle VM VirtualBox is deceptively simple yet also very powerful. It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

The following screenshot shows how Oracle VM VirtualBox, installed on an Apple Mac computer, is running Windows Server 2016 in a virtual machine window.

**Figure 1.1. Windows Server 2016 Virtual Machine, Displayed on a macOS Host**



In this User Manual, we will begin simply with a quick introduction to virtualization and how to get your first virtual machine running with the easy-to-use Oracle VM VirtualBox graphical user interface. Subsequent chapters will go into much more detail covering more powerful tools and features, but fortunately, it is not necessary to read the entire User Manual before you can use Oracle VM VirtualBox.

You can find a summary of Oracle VM VirtualBox's capabilities in [Section 1.3, "Features Overview"](#). For existing Oracle VM VirtualBox users who just want to find out what is new in this release, see the [Chapter 15, Change Log](#).

## 1.1. Why is Virtualization Useful?

The techniques and features that Oracle VM VirtualBox provides are useful in the following scenarios:

- **Running multiple operating systems simultaneously.** Oracle VM VirtualBox enables you to run more than one OS at a time. This way, you can run software written for one OS on another, such as Windows software on Linux or a Mac, without having to reboot to use it. Since you can configure what kinds of *virtual* hardware should be presented to each such OS, you can install an old OS such as DOS or OS/2 even if your real computer's hardware is no longer supported by that OS.
- **Easier software installations.** Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With Oracle VM VirtualBox, such a complex setup, often called an *appliance*, can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into Oracle VM VirtualBox.
- **Testing and disaster recovery.** Once installed, a virtual machine and its virtual hard disks can be considered a *container* that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.

Using virtual machines enables you to build and test a multi-node networked service, for example. Issues with networking, operating system, and software configuration can be investigated easily.

In addition to that, with the use of another Oracle VM VirtualBox feature called *snapshots*, one can save a particular state of a virtual machine and revert back to that state, if necessary. This way, one can freely experiment with a computing environment. If something goes wrong, such as problems after installing software or infecting the guest with a virus, you can easily switch back to a previous snapshot and avoid the need of frequent backups and restores.

Any number of snapshots can be created, allowing you to travel back and forward in virtual machine time. You can delete snapshots while a VM is running to reclaim disk space.

- **Infrastructure consolidation.** Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

## 1.2. Some Terminology

When dealing with virtualization, and also for understanding the following chapters of this documentation, it helps to acquaint oneself with a bit of crucial terminology, especially the following terms:

- **Host operating system (host OS).** This is the OS of the physical computer on which Oracle VM VirtualBox was installed. There are versions of Oracle VM VirtualBox for Windows, macOS, Linux, and Oracle Solaris hosts. See [Section 1.4, "Supported Host Operating Systems"](#).

Most of the time, this manual discusses all Oracle VM VirtualBox versions together. There may be platform-specific differences which we will point out where appropriate.

- **Guest operating system (guest OS).** This is the OS that is running inside the virtual machine. Theoretically, Oracle VM VirtualBox can run any x86 OS such as DOS, Windows, OS/2, FreeBSD, and

OpenBSD. But to achieve near-native performance of the guest code on your machine, we had to go through a lot of optimizations that are specific to certain OSes. So while your favorite OS *may* run as a guest, we officially support and optimize for a select few, which include the most common OSes.

See [Section 3.1, “Supported Guest Operating Systems”](#).

- **Virtual machine (VM).** This is the special environment that Oracle VM VirtualBox creates for your guest OS while it is running. In other words, you run your guest OS *in* a VM. Normally, a VM is shown as a window on your computer's desktop. Depending on which of the various frontends of Oracle VM VirtualBox you use, the VM might be shown in full screen mode or remotely on another computer.

Internally, Oracle VM VirtualBox treats a VM as a set of parameters that specify its behavior. Some parameters describe hardware settings, such as the amount of memory and number of CPUs assigned. Other parameters describe the state information, such as whether the VM is running or saved.

You can view these VM settings in VirtualBox Manager, in the **Settings** window, and by running the **VBoxManage** command. See [Chapter 8, \*VBoxManage\*](#).

- **Guest Additions.** This refers to special software packages which are shipped with Oracle VM VirtualBox but designed to be installed *inside* a VM to improve performance of the guest OS and to add extra features. See [Chapter 4, \*Guest Additions\*](#).

## 1.3. Features Overview

The following is a brief outline of Oracle VM VirtualBox's main features:

- **Portability.** Oracle VM VirtualBox runs on a large number of 64-bit host operating systems. See [Section 1.4, “Supported Host Operating Systems”](#).

Oracle VM VirtualBox is a so-called *hosted* hypervisor, sometimes referred to as a *type 2* hypervisor. Whereas a *bare-metal* or *type 1* hypervisor runs directly on the hardware, Oracle VM VirtualBox requires an existing OS to be installed. It can thus run alongside existing applications on that host.

To a very large degree, Oracle VM VirtualBox is functionally identical on all of the host platforms, and the same file and image formats are used. This enables you to run virtual machines created on one host on another host with a different host OS. For example, you can create a virtual machine on Windows and then run it on Linux.

In addition, virtual machines can easily be imported and exported using the Open Virtualization Format (OVF), an industry standard created for this purpose. You can even import OVFs that were created with a different virtualization software. See [Section 1.15, “Importing and Exporting Virtual Machines”](#).

For users of Oracle Cloud Infrastructure the functionality extends to exporting and importing virtual machines to and from the cloud. This simplifies development of applications and deployment to the production environment. See [Section 1.16.8, “Exporting an Appliance to Oracle Cloud Infrastructure”](#).

- **Guest Additions: shared folders, seamless windows, 3D virtualization.** The Oracle VM VirtualBox Guest Additions are software packages which can be installed *inside* of supported guest systems to improve their performance and to provide additional integration and communication with the host system. After installing the Guest Additions, a virtual machine will support automatic adjustment of video resolutions, seamless windows, accelerated 3D graphics and more. See [Chapter 4, \*Guest Additions\*](#).

In particular, Guest Additions provide for *shared folders*, which let you access files on the host system from within a guest machine. See [Section 4.3, “Shared Folders”](#).

- **Comprehensive hardware support.** Among other features, Oracle VM VirtualBox supports the following:

- **Guest multiprocessing (SMP).** Oracle VM VirtualBox can present up to 32 virtual CPUs to each virtual machine, irrespective of how many CPU cores are physically present on your host.
  - **USB device support.** Oracle VM VirtualBox implements a virtual USB controller and enables you to connect arbitrary USB devices to your virtual machines without having to install device-specific drivers on the host. USB support is not limited to certain device categories. See [Section 3.11.1, “USB Settings”](#).
  - **Hardware compatibility.** Oracle VM VirtualBox virtualizes a vast array of virtual devices, among them many devices that are typically provided by other virtualization platforms. That includes IDE, SCSI, and SATA hard disk controllers, several virtual network cards and sound cards, virtual serial and parallel ports and an Input/Output Advanced Programmable Interrupt Controller (I/O APIC), which is found in many computer systems. This enables easy cloning of disk images from real machines and importing of third-party virtual machines into Oracle VM VirtualBox.
  - **Full ACPI support.** The Advanced Configuration and Power Interface (ACPI) is fully supported by Oracle VM VirtualBox. This enables easy cloning of disk images from real machines or third-party virtual machines into Oracle VM VirtualBox. With its unique *ACPI power status support*, Oracle VM VirtualBox can even report to ACPI-aware guest OSes the power status of the host. For mobile systems running on battery, the guest can thus enable energy saving and notify the user of the remaining power, for example in full screen modes.
  - **Multiscreen resolutions.** Oracle VM VirtualBox virtual machines support screen resolutions many times that of a physical screen, allowing them to be spread over a large number of screens attached to the host system.
  - **Built-in iSCSI support.** This unique feature enables you to connect a virtual machine directly to an iSCSI storage server without going through the host system. The VM accesses the iSCSI target directly without the extra overhead that is required for virtualizing hard disks in container files. See [Section 5.10, “iSCSI Servers”](#).
  - **PXE Network boot.** The integrated virtual network cards of Oracle VM VirtualBox fully support remote booting using the Preboot Execution Environment (PXE).
- **Multigeneration branched snapshots.** Oracle VM VirtualBox can save arbitrary snapshots of the state of the virtual machine. You can go back in time and revert the virtual machine to any such snapshot and start an alternative VM configuration from there, effectively creating a whole snapshot tree. See [Section 1.11, “Snapshots”](#). You can create and delete snapshots while the virtual machine is running.
  - **VM groups.** Oracle VM VirtualBox provides a groups feature that enables the user to organize and control virtual machines collectively, as well as individually. In addition to basic groups, it is also possible for any VM to be in more than one group, and for groups to be nested in a hierarchy. This means you can have groups of groups. In general, the operations that can be performed on groups are the same as those that can be applied to individual VMs: Start, Pause, Reset, Close (Save state, Send Shutdown, Poweroff), Discard Saved State, Show in File System, Sort.
  - **Clean architecture and unprecedented modularity.** Oracle VM VirtualBox has an extremely modular design with well-defined internal programming interfaces and a clean separation of client and server code. This makes it easy to control it from several interfaces at once. For example, you can start a VM simply by clicking on a button in the Oracle VM VirtualBox graphical user interface and then control that machine from the command line, or even remotely. See [Section 1.18, “Alternative Front-Ends”](#).

Due to its modular architecture, Oracle VM VirtualBox can also expose its full functionality and configurability through a comprehensive **software development kit (SDK)**, which enables integration of Oracle VM VirtualBox with other software systems. See [Chapter 11, Oracle VM VirtualBox Programming Interfaces](#).

- **Remote machine display.** The VirtualBox Remote Desktop Extension (VRDE) enables high-performance remote access to any running virtual machine. This extension supports the Remote Desktop Protocol (RDP) originally built into Microsoft Windows, with special additions for full client USB support.

The VRDE does not rely on the RDP server that is built into Microsoft Windows. Instead, the VRDE is plugged directly into the virtualization layer. As a result, it works with guest OSes other than Windows, even in text mode, and does not require application support in the virtual machine either. The VRDE is described in detail in [Section 7.1, “Remote Display \(VRDP Support\)”](#).

On top of this special capacity, Oracle VM VirtualBox offers you more unique features:

- **Extensible RDP authentication.** Oracle VM VirtualBox already supports Winlogon on Windows and PAM on Linux for RDP authentication. In addition, it includes an easy-to-use SDK which enables you to create arbitrary interfaces for other methods of authentication. See [Section 7.1.5, “RDP Authentication”](#).
- **USB over RDP.** Using RDP virtual channel support, Oracle VM VirtualBox also enables you to connect arbitrary USB devices locally to a virtual machine which is running remotely on an Oracle VM VirtualBox RDP server. See [Section 7.1.4, “Remote USB”](#).

## 1.4. Supported Host Operating Systems

Currently, Oracle VM VirtualBox runs on the following host OSes:

- **Windows hosts (64-bit):**
  - Windows 8.1
  - Windows 10
  - Windows 11 21H2
  - Windows Server 2012
  - Windows Server 2012 R2
  - Windows Server 2016
  - Windows Server 2019
  - Windows Server 2022
- **macOS hosts (64-bit):**
  - 10.15 (Catalina)
  - 11 (Big Sur)
  - 12 (Monterey)

Intel hardware is required. See also [Chapter 14, \*Known Limitations\*](#).

An installer package is available for macOS/Arm64, for systems using an Apple silicon CPU. With this package, you can run some guest operating systems for Intel x86/x64 CPUs in an emulation.

The macOS/Arm64 installer package for Apple silicon platform is available as a Developer Preview release. This package represents a work in progress project and the performance is very modest.



**Note**

Developer Preview is a public release for developers, which provides early access to unsupported software release and features.

- **Linux hosts (64-bit).** Includes the following:
  - Ubuntu 18.04 LTS, 20.04 LTS and 22.04
  - Debian GNU/Linux 10 ("Buster") and 11 ("Bullseye")
  - Oracle Linux 7, 8 and 9
  - CentOS/Red Hat Enterprise Linux 7, 8 and 9
  - Fedora 35 and 36
  - Gentoo Linux
  - SUSE Linux Enterprise server 12 and 15
  - openSUSE Leap 15.3

It should be possible to use Oracle VM VirtualBox on most systems based on Linux kernel 2.6, 3.x, 4.x or 5.x using either the Oracle VM VirtualBox installer or by doing a manual installation. See [Section 2.3, "Installing on Linux Hosts"](#). However, the formally tested and supported Linux distributions are those for which we offer a dedicated package.

Note that Linux 2.4-based host OSes are no longer supported.

- **Oracle Solaris hosts (64-bit only).** The following versions are supported with the restrictions listed in [Chapter 14, Known Limitations](#):
  - Oracle Solaris 11.4

Note that any feature which is marked as *experimental* is not supported. Feedback and suggestions about such features are welcome.

### 1.4.1. Host CPU Requirements

SSE2 (Streaming SIMD Extensions 2) support is required for host CPUs.

## 1.5. Installing Oracle VM VirtualBox and Extension Packs

Oracle VM VirtualBox comes in many different packages, and installation depends on your host OS. If you have installed software before, installation should be straightforward. On each host platform, Oracle VM VirtualBox uses the installation method that is most common and easy to use. If you run into trouble or have special requirements, see [Chapter 2, Installation Details](#) for details about the various installation methods.

Oracle VM VirtualBox is split into the following components:

- **Base package.** The base package consists of all open source components and is licensed under the GNU General Public License V2.
- **Extension packs.** Additional extension packs can be downloaded which extend the functionality of the Oracle VM VirtualBox base package. Currently, Oracle provides a single extension pack, available from: <http://www.virtualbox.org>. The extension pack provides the following added functionality:

- VirtualBox Remote Desktop Protocol (VRDP) support. See [Section 7.1, “Remote Display \(VRDP Support\)”](#).
- Host webcam passthrough. See [Section 9.5, “Webcam Passthrough”](#).
- Intel PXE boot ROM.
- Disk image encryption with AES algorithm. See [Section 9.29, “Encryption of Disk Images”](#).
- Cloud integration features. See [Section 1.16, “Integrating with Oracle Cloud Infrastructure”](#).

For details of how to install an extension pack, see [Section 2.5, “Installing an Extension Pack”](#).

## 1.6. Starting Oracle VM VirtualBox

After installation, you can start Oracle VM VirtualBox as follows:

- **Windows hosts.** In the **Programs** menu, click on the item in the **VirtualBox** group. On some Windows platforms, you can also enter **VirtualBox** in the search box of the **Start** menu.
- **macOS hosts.** In the Finder, double-click on the **VirtualBox** item in the Applications folder. You may want to drag this item onto your Dock.
- **Linux or Oracle Solaris hosts.** Depending on your desktop environment, an Oracle VM VirtualBox item may have been placed in either the System or System Tools group of your **Applications** menu. Alternatively, you can enter **VirtualBox** in a terminal window.

When you start Oracle VM VirtualBox, the VirtualBox Manager interface is shown. See [Section 1.7, “VirtualBox Manager”](#).

## 1.7. VirtualBox Manager

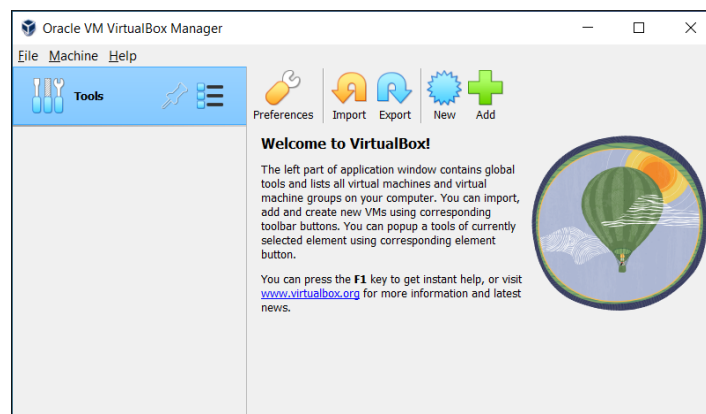
VirtualBox Manager is the user interface for Oracle VM VirtualBox. You can use VirtualBox Manager to create, configure, and manage your virtual machines.

This section describes the main features of the VirtualBox Manager user interface. Subsequent sections and chapters describe how to use VirtualBox Manager to perform tasks in Oracle VM VirtualBox.

When you start Oracle VM VirtualBox, the **VirtualBox Manager** window is displayed.

[Figure 1.2, “VirtualBox Manager, Showing Welcome Screen After Initial Startup”](#) shows VirtualBox Manager the first time you start Oracle VM VirtualBox, before you have created any virtual machines.

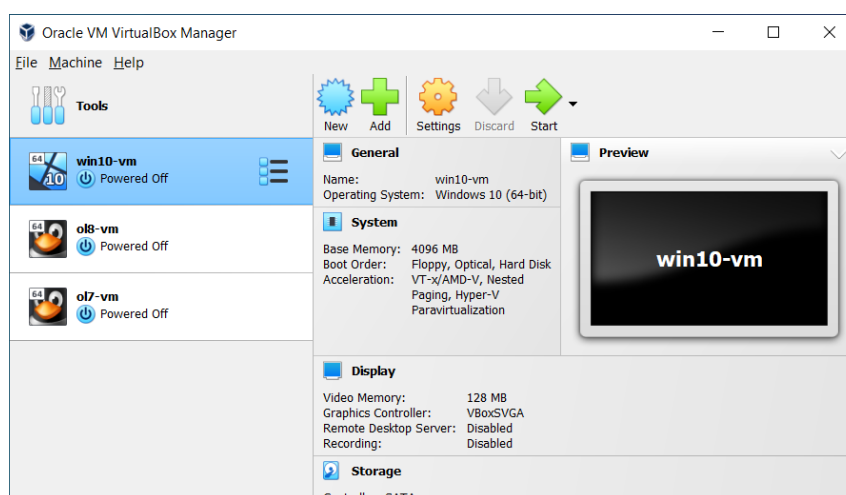
**Figure 1.2. VirtualBox Manager, Showing Welcome Screen After Initial Startup**





[Figure 1.3, “VirtualBox Manager Window, After Creating Virtual Machines”](#) shows how VirtualBox Manager might look after you have created some virtual machines.

**Figure 1.3. VirtualBox Manager Window, After Creating Virtual Machines**



The main components of the VirtualBox Manager window are as follows:

- **The machine list.** The left pane of the **VirtualBox Manager** window lists all your virtual machines. If you have not yet created any virtual machines, this list is empty. See [Section 1.7.1, “The Machine List”](#).
- **The Details pane.** The pane on the right displays the properties of the currently selected virtual machine. If you do not have any machines yet, the pane displays a welcome message.

The toolbar buttons on the Details pane can be used to create and work with virtual machines. See [Section 1.7.2, “The Details Pane”](#).

- **Help Viewer.** A window that displays context-sensitive help topics for VirtualBox Manager tasks. See [Section 1.7.4, “Help Viewer”](#).

### 1.7.1. The Machine List

The list of virtual machines in the left pane is called the *machine list*.

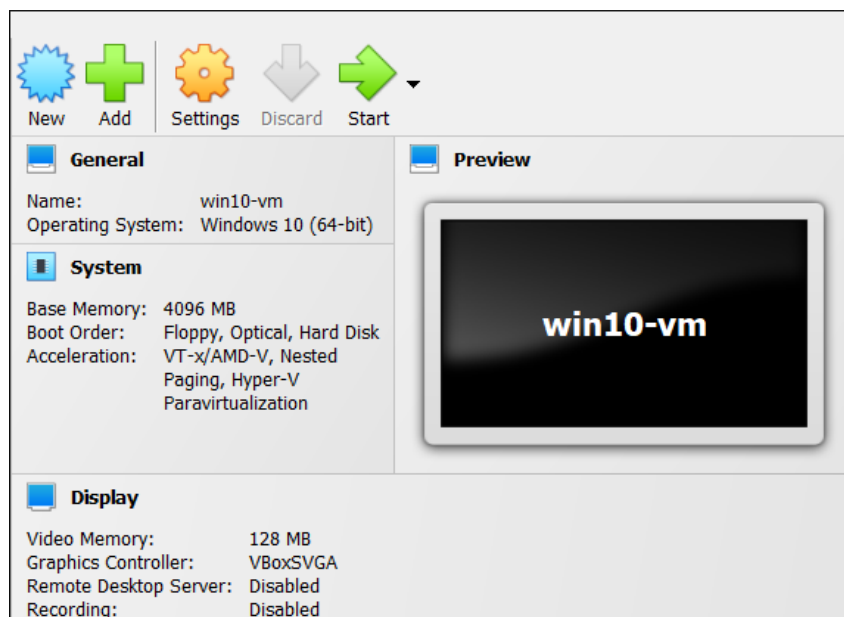
The following methods can be used to control and configure virtual machines in the machine list:

- Right-click on the virtual machine name, to display menu options.
- Click on the Machine Tools menu, to the right of the virtual machine name. See [the section called “Machine Tools”](#).
- Click a button in the toolbar in the Details pane. See [Section 1.7.2, “The Details Pane”](#).

### 1.7.2. The Details Pane

The Details pane shows configuration information for a virtual machine that is selected in the machine list. The pane also includes a toolbar for performing tasks.

**Figure 1.4. VirtualBox Manager Details Pane, Including Toolbar**



The Details pane includes the following:

### VirtualBox Manager Toolbar

A toolbar at the top of the Details pane contains buttons that enable you to configure the selected virtual machine, or to create a new virtual machine.

The toolbar includes the following buttons:

- **New.** Creates a new virtual machine, and adds it to the machine list.
- **Add.** Adds an existing virtual machine to the machine list.
- **Settings.** Displays the **Settings** window for the virtual machine, enabling you to make configuration changes.
- **Discard.** For a running virtual machine, discards the saved state for the virtual machine and closes it down.
- **Show/Start.** For a running virtual machine, **Show** displays the virtual machine window. For a stopped virtual machine, **Start** displays options for powering up the virtual machine.

### Settings

A summary of settings is shown for the virtual machine.

You can change some virtual machine settings, by clicking on the setting in the Details pane.

#### Note

If a virtual machine is running, some settings cannot be altered. You must stop the virtual machine first in order to change the setting.

Virtual machine settings can also be changed using the **Settings** button on the VirtualBox Manager toolbar.

The virtual machine settings on the Details pane are organized in sections that correspond to those used in the **Settings** window. See [Chapter 3, Configuring Virtual Machines](#).

Click the arrow icon to hide or show each section.

## Preview Window

The virtual machine display is shown in a small window.

You can use the Preview window to check if your virtual machine has finished booting up.

Click the arrow icon to hide or show the Preview window.

## Notification Center

Notification messages may be shown in a sliding panel on the right of the Details pane, called the **Notification Center**. Click the warning triangle to show the notification messages.

Most system messages that do not require user interaction are displayed in the Notification Center, including task failure alerts.

The progress of some tasks can be observed and stopped using the Notification Center.

### 1.7.3. VirtualBox Manager Tools

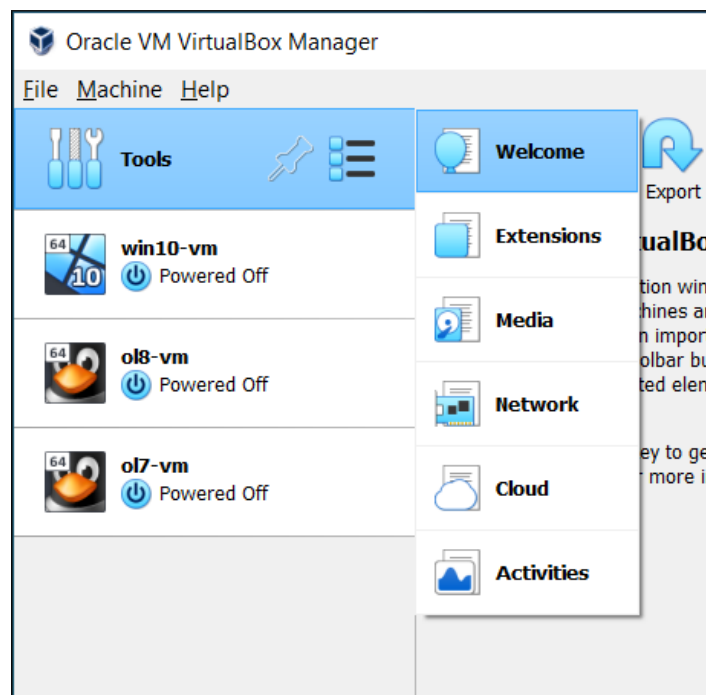
VirtualBox Manager provides two types of user tools, to enable you to perform common tasks easily.

- **Global Tools.** These tools apply to *all* virtual machines. See [the section called “Global Tools”](#).
- **Machine Tools.** These tools apply to a *specific* virtual machine. See [the section called “Machine Tools”](#).

#### Global Tools

In the left pane of the VirtualBox Manager window, click the **Menu** icon in the **Tools** banner located above the machine list. The **Global Tools** menu is displayed.

**Figure 1.5. Global Tools Menu**



A drop-down list enables you to select from the following global tools:

- **Welcome.** Displays the VirtualBox Manager welcome message. The VirtualBox Manager toolbar is also

included, to enable you to get started with using Oracle VM VirtualBox. See [Figure 1.2, “VirtualBox Manager, Showing Welcome Screen After Initial Startup”](#).

- **Extensions.** Displays the **Extension Pack Manager** tool. This tool is used to install and uninstall Oracle VM VirtualBox Extension Packs. See [Section 2.5.1, “The Extension Pack Manager”](#).
- **Media.** Displays the **Virtual Media Manager** tool. This tool is used to manage the disk images used by Oracle VM VirtualBox. See [Section 5.3, “The Virtual Media Manager”](#).
- **Network.** Displays the **Network Manager** tool. This tool is used to create and configure some types of networks used by Oracle VM VirtualBox. See [Section 6.11, “Network Manager”](#).
- **Cloud.** Displays the **Cloud Profile Editor** tool. This tool is used to configure connections to a cloud service, such as Oracle Cloud Infrastructure. See [Section 1.16.5, “Using the Cloud Profile Manager”](#).
- **Activities.** Displays the **VM Activity Overview** tool. This tool is used to monitor performance and resource usage of virtual machines. See [Section 1.20, “Monitoring of Virtual Machines”](#).

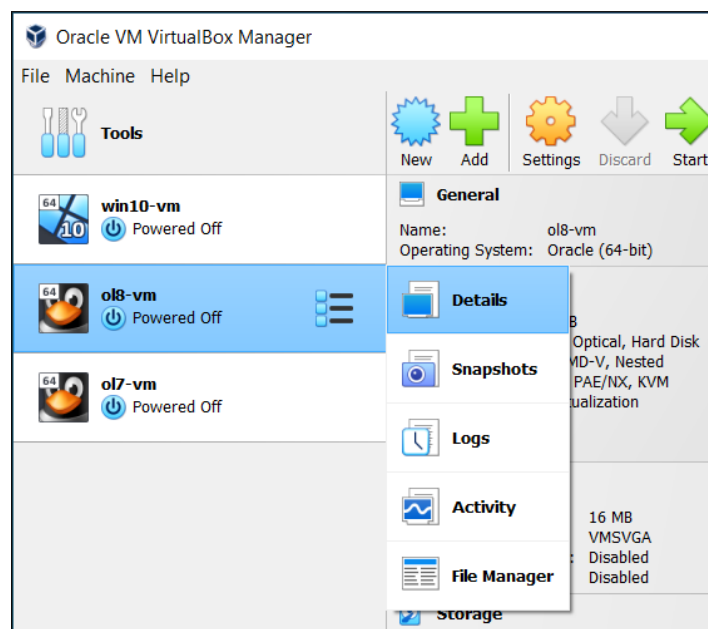
The **Pin** icon is used to keep the **Tools** banner visible as you scroll down the entries in the machine list.

### Machine Tools

In the machine list in the left pane of the VirtualBox Manager window, select a virtual machine.

Click the **Menu** icon to the right of the virtual machine name. The **Machine Tools** menu is displayed.

**Figure 1.6. Machine Tools Menu**



A drop-down list enables you to select from the following machine tools:

- **Details.** Displays the Details pane for the selected virtual machine. See [Section 1.7.2, “The Details Pane”](#).
- **Snapshots.** Displays the **Snapshots** tool. This tool enables you to view and manage snapshots for the virtual machine. See [Section 1.11, “Snapshots”](#).
- **Logs.** Displays the **Log Viewer** tool. This tool enables you to view and search system logs for the virtual machine. See [Section 1.21, “The Log Viewer”](#).

- **Activity.** Displays the **VM Activity** page of the **Session Information** dialog. This dialog enables you to view and analyze performance metrics for the virtual machine. See [Section 1.20, “Monitoring of Virtual Machines”](#).
- **File Manager.** Displays the **Guest Control File Manager** tool. This tool enables you to manage files on the guest system. See [Section 4.8, “Guest Control File Manager”](#).

### 1.7.4. Help Viewer

The Help Viewer is a window that displays context-sensitive help to assist you in completing common VirtualBox Manager tasks. You can display the Help Viewer in the following ways:

- In a VirtualBox Manager wizard or dialog, click **Help** to display the relevant help topic.
- In VirtualBox Manager or from a guest VM, do either of the following:
  - Select the **Help, Contents** menu option.
  - Press the **F1** button.

The keyboard shortcut used to access the Help Viewer can be configured in the **Preferences** window.

The Help Viewer has the following features:

- **Navigation tools.** The left hand pane contains the following navigation tools:
  - **Contents.** Displays the help topic location in the Oracle VM VirtualBox documentation.
  - **Search.** Enables you to search the documentation for help topics.
  - **Bookmarks.** Enables you to bookmark useful help topics.
- **Tabbed browsing.** Help topics that you have visited are displayed in tabs in the main window pane.
- **Zoomable topics.** Zoom controls enable you to enlarge help topic details.
- **Printing.** Help topics can be printed to PDF file or to a local printer.

### 1.7.5. About VirtualBox Manager Wizards

VirtualBox Manager includes wizards that enable you to complete tasks easily. Examples of such tasks are when you create a new virtual machine or use the cloud integration features of Oracle VM VirtualBox.

To display a help topic for the wizard, click the **Help** button.

Some wizards can be displayed in either of the following modes:

- **Guided mode.** This is the default display mode. Wizards are shown in the conventional manner, using a series of pages with descriptions to guide the user through the steps for a task.
- **Expert mode.** This display mode is designed for more advanced users of Oracle VM VirtualBox. All settings are displayed on a single page, enabling quicker completion of tasks.

Click the button at the bottom of the wizard window to switch between Guided mode and Expert mode.

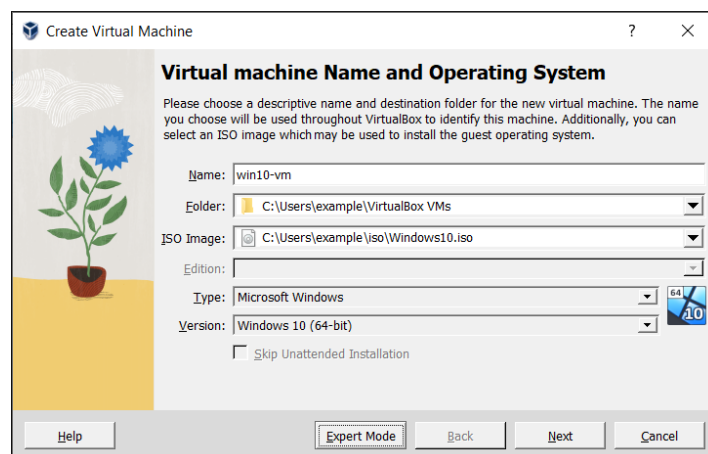
## 1.8. Creating Your First Virtual Machine

Click **New** in the VirtualBox Manager window. The **Create Virtual Machine** wizard is shown, to guide you through the required steps for setting up a new virtual machine (VM).

The **Create Virtual Machine** wizard pages are described in the following sections.

### 1.8.1. Create Virtual Machine Wizard: Name and Operating System

**Figure 1.7. Creating a Virtual Machine: Name and Operating System**



Use this page to specify a name and operating system (OS) for the virtual machine and to change the storage location used for VMs.

You can also choose to disable the unattended guest operating system install feature. See also [Section 1.8.2, “\(Optional\) Create Virtual Machine Wizard: Unattended Guest OS Install”](#).

The following fields are available on this wizard page:

- **Name.** A name for the new VM. The name you enter is shown in the machine list of VirtualBox Manager and is also used for the virtual machine's files on disk.

Be sure to assign each VM an informative name that describes the OS and software running on the VM. For example, a name such as `Windows 10 with Visio`.

- **Folder.** The location where VMs are stored on your computer, called the *machine folder*. The default folder location is shown.

Ensure that the folder location has enough free space, especially if you intend to use the snapshots feature. See also [Section 10.1.1, “The Machine Folder”](#).

- **ISO Image.** Select an ISO image file. The image file can be used to install an OS on the new virtual machine or it can be attached to a DVD drive on the new virtual machine.
- **Type and Version.** These fields are used to select the OS that you want to install on the new virtual machine.

The supported OSes are grouped into types. If you want to install something very unusual that is not listed, select the **Other** type. Depending on your selection, Oracle VM VirtualBox will enable or disable certain VM settings that your guest OS may require. This is particularly important for 64-bit guests. See [Section 3.1.2, “64-bit Guests”](#). It is therefore recommended to always set this field to the correct value.

If an ISO image is selected and Oracle VM VirtualBox detects the operating system for the ISO, the **Type** and **Version** fields are populated automatically and are disabled.



- **Skip Unattended Installation.** Disables unattended guest OS installation, even if an ISO image is selected that supports unattended installation. In that case, the selected ISO image is mounted automatically on the DVD drive of the new virtual machine and user interaction is required to complete the OS installation.

The unattended installation step in the wizard is skipped.

### Note

This option is disabled if you do not select an installation medium in the **ISO Image** field.

Click **Next** to go to the next wizard page.

## 1.8.2. (Optional) Create Virtual Machine Wizard: Unattended Guest OS Install

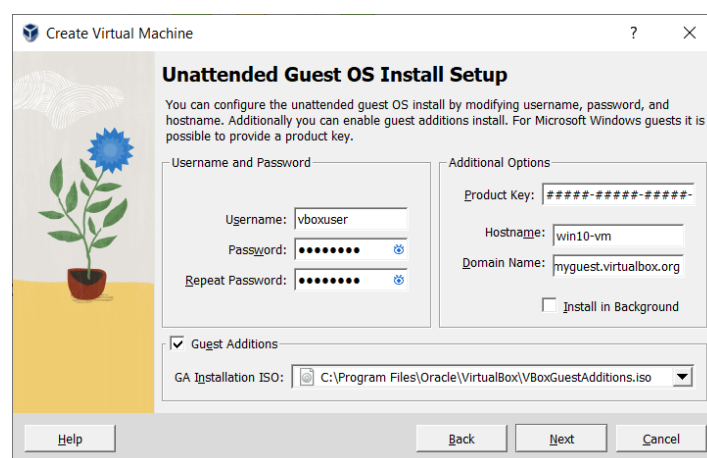
Unattended guest OS installation enables you to install the OS on a virtual machine automatically.

### Note

This page is optional. It is not displayed if you have selected the **Skip Unattended Installation** option on the initial wizard page.

Use this page to set up the required parameters for unattended guest OS installation and to configure automatic installation of the Oracle VM VirtualBox Guest Additions. See also [Section 1.8.6, “Some Examples of Unattended Installation”](#) for some typical scenarios when using automated installation.

**Figure 1.8. Creating a Virtual Machine: Unattended Guest OS Installation**



The following fields are available on this wizard page:

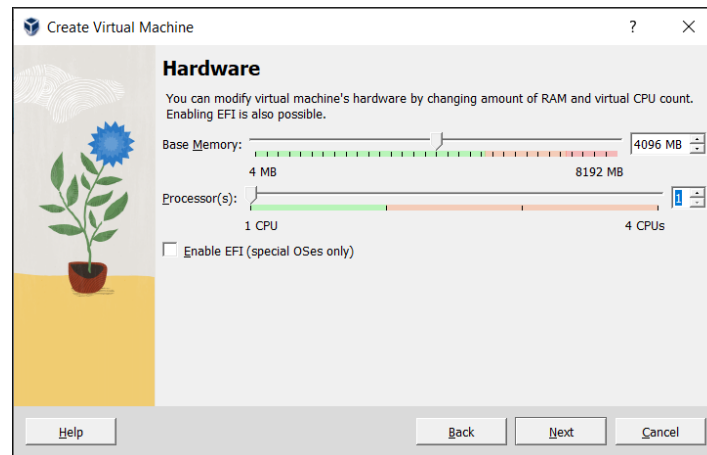
- **Username and Password.** Enter the credentials for a default user on the guest OS.
- **Guest Additions.** Enables automatic installation of the Guest Additions, following installation of the guest OS. Use the drop-down list to select the location of the ISO image file for the Guest Additions.
- **Additional Options.** The following options enable you to perform extra configuration of the guest OS:
  - **Product Key.** For Windows guests only. Enter the product key required for Windows installation.
  - **Hostname.** Host name for the guest. By default, this is the same as the VM name.
  - **Domain Name.** Domain name for the guest.
  - **Install in Background.** Enable headless mode for the VM, where a graphical user interface is not shown.

Click **Next** to go to the next wizard page.

### 1.8.3. Create Virtual Machine Wizard: Hardware

Use this page to configure hardware settings for the virtual machine.

**Figure 1.9. Creating a Virtual Machine: Hardware**



The following fields are available on this wizard page:

- **Base Memory.** Select the amount of RAM that Oracle VM VirtualBox should allocate every time the virtual machine is started. The amount of memory selected here will be taken away from your host machine and presented to the guest OS, which will report this size as the virtual machines installed RAM.

#### Caution

Choose this setting carefully. The memory you give to the VM will not be available to your host OS while the VM is running, so do not specify more than you can spare.

For example, if your host machine has 4 GB of RAM and you enter 2048 MB as the amount of RAM for a particular virtual machine, you will only have 2 GB left for all the other software on your host while the VM is running. If you run two VMs at the same time, even more memory will be allocated for the second VM, which may not even be able to start if that memory is not available.

On the other hand, you should specify as much as your guest OS and your applications will require to run properly. A guest OS may require at least 1 or 2 GB of memory to install and boot up. For best performance, more memory than that may be required.

Always ensure that the host OS has enough RAM remaining. If insufficient RAM remains, the system might excessively swap memory to the hard disk, which effectively brings the host system to a standstill.

As with other **Create Virtual Machine** wizard settings, you can change this setting later, after you have created the VM.

- **Processor(s).** Select the number of virtual processors to assign to the VM.

It is not advised to assign more than half of the total processor threads from the host machine.

- **Enable EFI.** Enables Extensible Firmware Interface (EFI) booting for the guest OS.

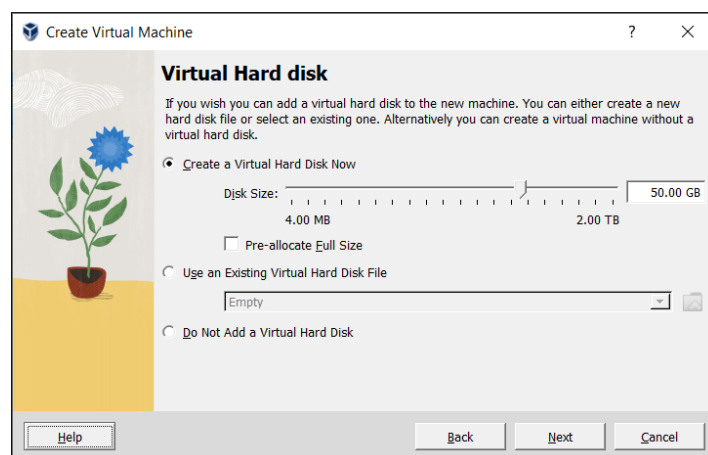
Click **Next** to go to the next wizard page.

### 1.8.4. Create Virtual Machine Wizard: Virtual Hard Disk

Use this page to specify a virtual hard disk for the virtual machine.

There are many ways in which Oracle VM VirtualBox can provide hard disk space to a VM, see [Chapter 5, Virtual Storage](#). The most common way is to use a large image file on your physical hard disk, whose contents Oracle VM VirtualBox presents to your VM as if it were a complete hard disk. This file then represents an entire hard disk, so you can even copy it to another host and use it with another Oracle VM VirtualBox installation.

**Figure 1.10. Creating a New Virtual Machine: Virtual Hard Disk**



The following fields are available on this wizard page:

- **Create a Virtual Hard Disk Now.** Creates a new empty virtual hard disk image, located in the VM's machine folder.

Enter the following settings:

- **Disk Size.** Use the slider to select a maximum size for the hard disk in the new VM.
- **Pre-Allocate Full Size.** This setting determines the type of image file used for the disk image. Select this setting to use a *fixed-size file* for the disk image. Deselect this setting to use a *dynamically allocated file* for the disk image.

The different types of image file behave as follows:

- **Dynamically allocated file.** This type of image file only grows in size when the guest actually stores data on its virtual hard disk. Therefore, this file is small initially. As the drive is filled with data, the file grows to the specified size.
- **Fixed-size file.** This type of image file immediately occupies the file specified, even if only a fraction of that virtual hard disk space is actually in use. While occupying much more space, a fixed-size file incurs less overhead and is therefore slightly faster than a dynamically allocated file.

For more details about the differences, see [Section 5.2, "Disk Image Files \(VDI, VMDK, VHD, HDD\)"](#).

- **Use an Existing Hard Disk File.** Enables you to select an *existing* disk image file to use with the new VM.

The drop-down list presented in the window lists all disk images which are known by Oracle VM VirtualBox. These disk images are currently attached to a virtual machine, or have been attached to a virtual machine.

Alternatively, click on the small folder icon next to the drop-down list. In the **Hard Disk Selector** window

that is displayed, click **Add** to select a disk image file on your host disk.

- **Do Not Add a Virtual Hard Disk.** The new VM is created without a hard disk.

To prevent your physical hard disk on the host OS from filling up, Oracle VM VirtualBox limits the size of the image file. But the image file must be large enough to hold the contents of the guest OS and the applications you want to install. For a Windows or Linux guest, you will probably need several gigabytes for any serious use. The limit of the image file size can be changed later, see [Section 8.31, “VBoxManage modifymedium”](#).

#### Note

You can skip attaching a virtual hard disk file to the new virtual machine you are creating. But you will then need to attach an hard disk later on, in order to install a guest operating system.

After having selected or created your image file, click **Next** to go to the next wizard page.

### 1.8.5. Create Virtual Machine Wizard: Summary

This page displays a summary of the configuration for the virtual machine.

If you are not happy with any of the settings, use the **Back** button to return to the corresponding page and modify the setting.

Click **Finish** to create your new virtual machine. The virtual machine is displayed in the machine list on the left side of the VirtualBox Manager window, with the name that you entered on the first page of the wizard.

### 1.8.6. Some Examples of Unattended Installation

To configure unattended installation, you typically just need to specify an ISO image in the **Create Virtual Machine** wizard. Oracle VM VirtualBox then detects the OS type and the unattended installation process is done automatically when the wizard is completed. However, in some situations the installation may need be completed manually.

The following list describes some common scenarios for unattended installation:

- **OS type is detected automatically.** The following outcomes are possible:
  - If unattended installation is supported for the selected ISO, the guest OS is installed automatically. No user input is required.
  - If unattended installation is not supported for the selected ISO, the ISO image is inserted automatically into the DVD drive of the new VM. The guest OS installation must then be completed manually.

- **OS type is not detected automatically.** You must configure **Type** and **Version** settings in the wizard.

The ISO image is inserted automatically into the DVD drive of the new VM. The guest OS installation must then be completed manually.

- **Unattended Installation is disabled.** Users can disable unattended installation, by selecting the **Skip Unattended Installation** check box on the initial wizard page.

The ISO image is inserted automatically into the DVD drive of the new VM. The guest OS installation must then be completed manually.

See also [Section 3.2, “Unattended Guest Installation”](#) for details of how to perform unattended installation from the command line.

## 1.9. Running Your Virtual Machine

To start a virtual machine, you have the following options:

- Double-click on the VM's entry in the machine list in VirtualBox Manager.
- Select the VM's entry in the machine list in VirtualBox Manager, and click **Start** in the toolbar the top of the window.
- Go to the `VirtualBox VMs` folder in your system user's home directory. Find the subdirectory of the machine you want to start and double-click on the machine settings file. This file has a `.vbox` file extension.

Starting a virtual machine displays a new window, and the virtual machine which you selected will boot up. Everything which would normally be seen on the virtual system's monitor is shown in the window. See [Figure 1.1, "Windows Server 2016 Virtual Machine, Displayed on a macOS Host"](#).

In general, you can use the virtual machine as you would use a real computer. The following topics describe a few points to note when running a VM.

### 1.9.1. Starting a New VM for the First Time

When you start a VM for the first time the OS installation process is started automatically, using the ISO image file specified in the **Create Virtual Machine** wizard.

Follow the onscreen instructions to install your OS.

### 1.9.2. Capturing and Releasing Keyboard and Mouse

Oracle VM VirtualBox provides a virtual USB tablet device to new virtual machines through which mouse events are communicated to the guest OS. If you are running a modern guest OS that can handle such devices, mouse support may work out of the box without the mouse being *captured* as described below. See [Section 3.5.1, "Motherboard Tab"](#).

Otherwise, if the virtual machine detects only standard PS/2 mouse and keyboard devices, since the OS in the virtual machine does not know that it is not running on a real computer, it expects to have exclusive control over your keyboard and mouse. But unless you are running the VM in full screen mode, your VM needs to share keyboard and mouse with other applications and possibly other VMs on your host.

After installing a guest OS and before you install the Guest Additions, described in [Chapter 4, Guest Additions](#), either your VM or the rest of your computer can *own* the keyboard and the mouse. Both cannot own the keyboard and mouse at the same time. You will see a *second* mouse pointer which is always confined to the limits of the VM window. You activate the VM by clicking inside it.

To return ownership of keyboard and mouse to your host OS, Oracle VM VirtualBox reserves a special key on your keyboard: the *Host key*. By default, this is the *right Ctrl* key on your keyboard. On a Mac host, the default Host key is the left Command key. You can change this default using the Preferences window. See [Section 1.17, "Preferences"](#). The current setting for the Host key is always displayed at the bottom right of your VM window.

**Figure 1.11. Host Key Setting on the Virtual Machine Taskbar**



This means the following:

- Your **keyboard** is owned by the VM if the VM window on your host desktop has the keyboard focus. If you

have many windows open in your guest OS, the window that has the focus in your VM is used. This means that if you want to enter text within your VM, click on the title bar of your VM window first.

To release keyboard ownership, press the Host key. As explained above, this is typically the right Ctrl key.

Note that while the VM owns the keyboard, some key sequences, such as Alt+Tab, will no longer be seen by the host, but will go to the guest instead. After you press the Host key to reenable the host keyboard, all key presses will go through the host again, so that sequences such as Alt+Tab will no longer reach the guest. For technical reasons it may not be possible for the VM to get all keyboard input even when it does own the keyboard. Examples of this are the Ctrl+Alt+Del sequence on Windows hosts or single keys grabbed by other applications on X11 hosts such as the GNOME desktop Locate Pointer feature.

- Your **mouse** is owned by the VM only after you have clicked in the VM window. The host mouse pointer will disappear, and your mouse will drive the guest's pointer instead of your normal mouse pointer.

Note that mouse ownership is independent of that of the keyboard. Even after you have clicked on a titlebar to be able to enter text into the VM window, your mouse is not necessarily owned by the VM yet.

To release ownership of your mouse by the VM, press the Host key.

As this behavior is inconvenient, Oracle VM VirtualBox provides a set of tools and device drivers for guest systems called the Oracle VM VirtualBox Guest Additions. These tools make VM keyboard and mouse operations much more seamless. Most importantly, the Guest Additions suppress the second "guest" mouse pointer and make your host mouse pointer work directly in the guest. See [Chapter 4, Guest Additions](#).

### 1.9.3. Typing Special Characters

Some OSes expect certain key combinations to initiate certain procedures. The key combinations that you type into a VM might target the host OS, the Oracle VM VirtualBox software, or the guest OS. The recipient of these keypresses depends on a number of factors, including the key combination itself.

- Host OSes reserve certain key combinations for themselves. For example, you cannot use the **Ctrl+Alt+Delete** combination to reboot the guest OS in your VM, because this key combination is reserved by the host OS. Even though both Windows and Linux OSes can intercept this key combination, the host OS is rebooted automatically.

On Linux and Oracle Solaris hosts, which use the X Window System, the key combination **Ctrl+Alt+Backspace** normally resets the X server and restarts the entire graphical user interface. As the X server intercepts this combination, pressing it will usually restart your *host* graphical user interface and kill all running programs, including Oracle VM VirtualBox, in the process.

On Linux hosts supporting virtual terminals, the key combination **Ctrl+Alt+Fx**, where Fx is one of the function keys from F1 to F12, normally enables you to switch between virtual terminals. As with **Ctrl+Alt+Delete**, these combinations are intercepted by the host OS and therefore always switch terminals on the *host*.

If, instead, you want to send these key combinations to the *guest* OS in the virtual machine, you will need to use one of the following methods:

- Use the items in the **Input, Keyboard** menu of the virtual machine window. This menu includes the settings **Insert Ctrl+Alt+Delete** and **Insert Ctrl+Alt+Backspace**. However, the latter setting affects only Linux guests or Oracle Solaris guests.

This menu also includes an option for inserting the Host key combination.

- Use special key combinations with the Host key, which is normally the right Control key. Oracle VM VirtualBox then translates the following key combinations for the VM:



- **Host key + Del** sends **Ctrl+Alt+Del** to reboot the guest OS.
- **Host key + Backspace** sends **Ctrl+Alt+Backspace** to restart the graphical user interface of a Linux or Oracle Solaris guest.
- **Host key + Function key**. For example, use this key combination to simulate **Ctrl+Alt+Fx** to switch between virtual terminals in a Linux guest.
- For some other keyboard combinations such as **Alt+Tab** to switch between open windows, Oracle VM VirtualBox enables you to configure whether these combinations will affect the host or the guest, if a virtual machine currently has the focus. This is a global setting for all virtual machines and can be found under **File, Preferences, Input**.
- A soft keyboard can be used to input key combinations in the guest. See [Section 1.19, “Soft Keyboard”](#).

### 1.9.4. Changing Removable Media

While a virtual machine is running, you can change removable media in the **Devices** menu of the VM's window. Here you can select in detail what Oracle VM VirtualBox presents to your VM as a CD, DVD, or floppy drive.

The settings are the same as those available for the VM in the **Settings** window of VirtualBox Manager. But as the **Settings** window is disabled while the VM is in the Running or Saved state, the **Devices** menu saves you from having to shut down and restart the VM every time you want to change media.

Using the **Devices** menu, you can attach the host drive to the guest or select a floppy or DVD image, as described in [Section 3.7, “Storage Settings”](#).

The **Devices** menu also includes an option for creating a virtual ISO (VISO) from selected files on the host.

### 1.9.5. Resizing the Machine's Window

You can resize the VM's window while that VM is running. When you do, the window is scaled as follows:

- If you have **scaled mode** enabled, then the virtual machine's screen will be scaled to the size of the window. This can be useful if you have many machines running and want to have a look at one of them while it is running in the background. Alternatively, it might be useful to enlarge a window if the VM's output screen is very small, for example because you are running an old OS in it.

To enable scaled mode, press **Host key + C**, or select **Scaled Mode** from the **View** menu in the VM window. To leave scaled mode, press **Host key + C** again.

The aspect ratio of the guest screen is preserved when resizing the window. To ignore the aspect ratio, press **Shift** during the resize operation.

See [Chapter 14, Known Limitations](#) for additional remarks.

- If you have the Guest Additions installed and they support automatic **resizing**, the Guest Additions will automatically adjust the screen resolution of the guest OS. For example, if you are running a Windows guest with a resolution of 1024x768 pixels and you then resize the VM window to make it 100 pixels wider, the Guest Additions will change the Windows display resolution to 1124x768.

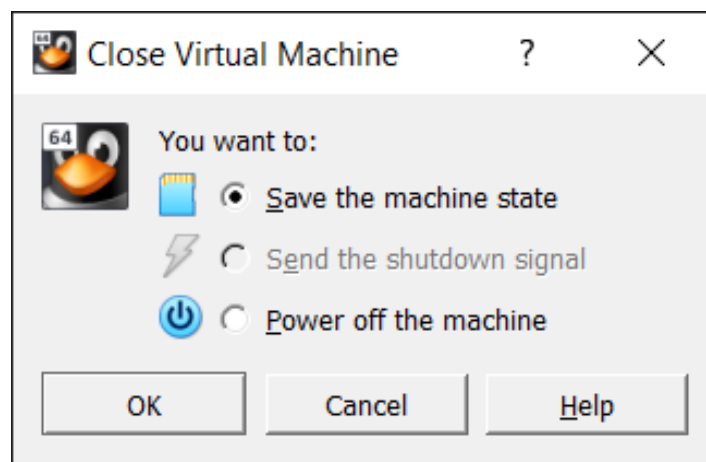
See [Chapter 4, Guest Additions](#).

- Otherwise, if the window is bigger than the VM's screen, the screen will be centered. If it is smaller, then scroll bars will be added to the machine window.

### 1.9.6. Saving the State of the Machine

When you click on the **Close** button of your virtual machine window, at the top right of the window, just like you would close any other window on your system, Oracle VM VirtualBox asks you whether you want to save or power off the VM. As a shortcut, you can also press **Host key + Q**.

**Figure 1.12. Closing Down a Virtual Machine**



The difference between the three options is crucial. They mean the following:

- **Save the machine state:** With this option, Oracle VM VirtualBox *freezes* the virtual machine by completely saving its state to your local disk.

When you start the VM again later, you will find that the VM continues exactly where it was left off. All your programs will still be open, and your computer resumes operation. Saving the state of a virtual machine is thus in some ways similar to suspending a laptop computer by closing its lid.

- **Send the shutdown signal.** This will send an ACPI shutdown signal to the virtual machine, which has the same effect as if you had pressed the power button on a real computer. This should trigger a proper shutdown mechanism from within the VM.
- **Power off the machine:** With this option, Oracle VM VirtualBox also stops running the virtual machine, but *without* saving its state.

### Warning

This is equivalent to pulling the power plug on a real computer without shutting it down properly. If you start the machine again after powering it off, your OS will have to reboot completely and may begin a lengthy check of its virtual system disks. As a result, this should not normally be done, since it can potentially cause data loss or an inconsistent state of the guest system on disk.

As an exception, if your virtual machine has any snapshots, see [Section 1.11, “Snapshots”](#), you can use this option to quickly **restore the current snapshot** of the virtual machine. In that case, powering off the machine will discard the current state and any changes made since the previous snapshot was taken will be lost.

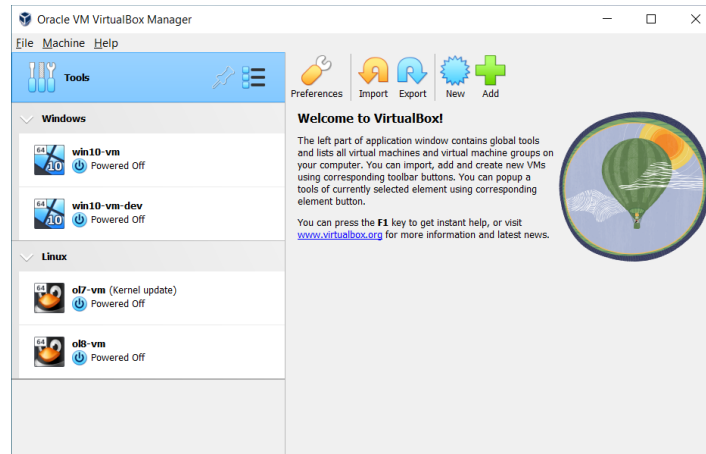
The **Discard** button in the VirtualBox Manager window discards a virtual machine's saved state. This has the same effect as powering it off, and the same warnings apply.

## 1.10. Using VM Groups

VM groups are groups of VMs that you can create as and when required. You can manage and perform functions on them collectively, as well as individually.

The following figure shows VM groups displayed in VirtualBox Manager.

**Figure 1.13. Groups of Virtual Machines**



The following features are available for groups:

- Create a group using VirtualBox Manager. Do one of the following:
  - Drag a VM on top of another VM.
  - Select multiple VMs and select **Group** from the right-click menu.
- Create and manage a group using the command line. Do one of the following:

- Create a group and assign a VM. For example:

```
VBoxManage modifyvm "vm01" --groups "/TestGroup"
```

This command creates a group `TestGroup` and attaches the VM `vm01` to that group.

- Detach a VM from the group, and delete the group if empty. For example:

```
VBoxManage modifyvm "vm01" --groups ""
```

This command detaches all groups from the VM `vm01` and deletes the empty group.

- Create multiple groups. For example:

```
VBoxManage modifyvm "vm01" --groups "/TestGroup,/TestGroup2"
```

This command creates the groups `TestGroup` and `TestGroup2`, if they do not exist, and attaches the VM `vm01` to both of them.

- Create nested groups, having a group hierarchy. For example:

```
VBoxManage modifyvm "vm01" --groups "/TestGroup/TestGroup2"
```

This command attaches the VM `vm01` to the subgroup `TestGroup2` of the `TestGroup` group.

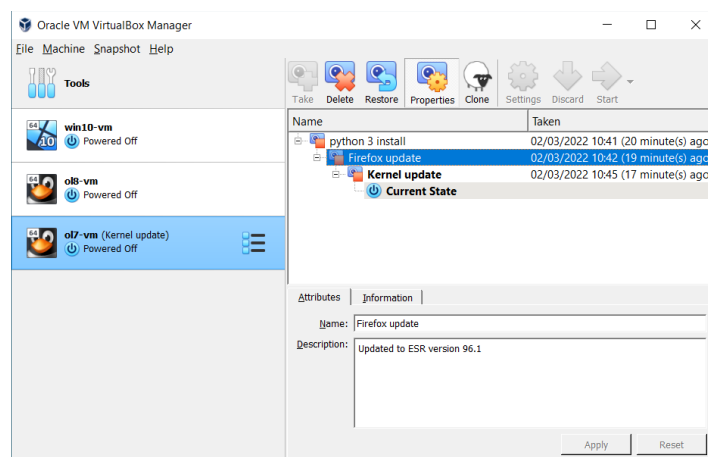
- Use VirtualBox Manager menu options to control and manage all the VMs in a group. For example: **Start**, **Pause**, **Reset**, **Close** (save state, send shutdown signal, poweroff), **Discard Saved State**, **Show in Explorer**, **Sort**.

## 1.11. Snapshots

With snapshots, you can save a particular state of a virtual machine for later use. At any later time, you can revert to that state, even though you may have changed the VM considerably since then. A snapshot of a virtual machine is thus similar to a machine in Saved state, but there can be many of them, and these saved states are preserved.

To see the snapshots of a virtual machine, click on the machine name in VirtualBox Manager. In the machine tools menu for the VM, click **Snapshots**. The Snapshots tool is displayed.

**Figure 1.14. Snapshots Tool, Showing Snapshot Properties**



If you select multiple VMs in the machine list, all snapshots are listed for each VM.

Until you take a snapshot of the virtual machine, the list of snapshots will be empty, except for the **Current State** item. This item represents the current point in the lifetime of the virtual machine.

The Snapshots window includes a toolbar, enabling you to perform the following snapshot operations:

- **Take.** Takes a snapshot of the selected VM. See [Section 1.11.1, “Taking, Restoring, and Deleting Snapshots”](#).
- **Delete.** Removes a snapshot from the list of snapshots. See [Section 1.11.1, “Taking, Restoring, and Deleting Snapshots”](#).
- **Restore.** Restores the VM state to be the same as the selected snapshot. See [Section 1.11.1, “Taking, Restoring, and Deleting Snapshots”](#).
- **Properties.** Displays the properties for the selected snapshot. The **Attributes** tab is used to specify a Name and Description for the snapshot. The **Information** tab shows VM settings for the snapshot.
- **Clone.** Displays the **Clone Virtual Machine** wizard. This enables you to create a clone of the VM, based on the selected snapshot.
- **Settings.** Available for the Current State snapshot only. Displays the **Settings** window for the VM, enabling you to make configuration changes.
- **Discard.** For a running VM, discards the saved state for the VM and closes it down.
- **Start.** Start the VM. This operation is available for the **Current State** item.

### 1.11.1. Taking, Restoring, and Deleting Snapshots

There are three operations related to snapshots, as follows:

1. **Take a snapshot.** This makes a copy of the machine's current state, to which you can go back at any given

time later.

- If your VM is running:

Select **Take Snapshot** from the **Machine** menu in the VM window.

The VM is paused while the snapshot is being created. After snapshot creation, the VM continues to run as normal.

- If your VM is in either the Saved or the Powered Off state, as displayed next to the VM name in the machine list:

Display the Snapshots window and do one of the following:

- Click **Take** in the Snapshots window toolbar.
- Right-click on the **Current State** item in the list and select **Take**.

A dialog is displayed, prompting you for a snapshot name. This name is purely for reference purposes, to help you remember the state of the snapshot. For example, a useful name would be "Fresh installation from scratch, no Guest Additions", or "Service Pack 3 just installed". You can also add a longer text description in the **Snapshot Description** field.

Your new snapshot will then appear in the snapshots list. Underneath your new snapshot, you will see an item called **Current State**, signifying that the current state of your VM is a variation based on the snapshot you took earlier. If you later take another snapshot, you will see that they are displayed in sequence, and that each subsequent snapshot is derived from an earlier one.

**Figure 1.15. Snapshots List For a Virtual Machine**

Name	Taken
python 3 install	02/03/2022 10:41 (20 minute(s) ago)
Firefox update	02/03/2022 10:42 (19 minute(s) ago)
Kernel update	02/03/2022 10:45 (17 minute(s) ago)
Current State	

Oracle VM VirtualBox imposes no limits on the number of snapshots you can take. The only practical limitation is disk space on your host. Each snapshot stores the state of the virtual machine and thus occupies some disk space. See [Section 1.11.2, "Snapshot Contents"](#) for details on what is stored in a snapshot.

2. **Restore a snapshot.** In the Snapshots window, select the snapshot you have taken and click **Restore** in the toolbar. By restoring a snapshot, you go back or forward in time. The current state of the machine is lost, and the machine is restored to the exact state it was in when the snapshot was taken.

### Note

Restoring a snapshot will affect the virtual hard drives that are connected to your VM, as the entire state of the virtual hard drive will be reverted as well. This means also that all files that have been created since the snapshot and all other file changes *will be lost*. In order to prevent such data loss while still making use of the snapshot feature, it is possible to add a second hard drive in *write-through* mode using the **VBoxManage** interface and use it to store your data. As write-through hard drives are *not* included in snapshots, they remain unaltered when a machine is reverted. See [Section 5.4, "Special Image Write Modes"](#).

To avoid losing the current state when restoring a snapshot, you can create a new snapshot before the

restore operation.

By restoring an earlier snapshot and taking more snapshots from there, it is even possible to create a kind of alternate reality and to switch between these different histories of the virtual machine. This can result in a whole tree of virtual machine snapshots.

3. **Delete a snapshot.** This does not affect the state of the virtual machine, but only releases the files on disk that Oracle VM VirtualBox used to store the snapshot data, thus freeing disk space. To delete a snapshot, select the snapshot name in the Snapshots window and click **Delete** in the toolbar. Snapshots can be deleted even while a machine is running.

#### Note

Whereas taking and restoring snapshots are fairly quick operations, deleting a snapshot can take a considerable amount of time since large amounts of data may need to be copied between several disk image files. Temporary disk files may also need large amounts of disk space while the operation is in progress.

There are some situations which cannot be handled while a VM is running, and you will get an appropriate message that you need to perform this snapshot deletion when the VM is shut down.

### 1.11.2. Snapshot Contents

Think of a snapshot as a point in time that you have preserved. More formally, a snapshot consists of the following:

- The snapshot contains a complete copy of the VM settings, including the hardware configuration, so that when you restore a snapshot, the VM settings are restored as well. For example, if you changed the hard disk configuration or the VM's system settings, that change is undone when you restore the snapshot.

The copy of the settings is stored in the machine configuration, an XML text file, and thus occupies very little space.

- The complete state of all the virtual disks attached to the machine is preserved. Going back to a snapshot means that all changes that had been made to the machine's disks, file by file and bit by bit, will be undone as well. Files that were since created will disappear, files that were deleted will be restored, changes to files will be reverted.

Strictly speaking, this is only true for virtual hard disks in "normal" mode. You can configure disks to behave differently with snapshots, see [Section 5.4, "Special Image Write Modes"](#). In technical terms, it is not the virtual disk itself that is restored when a snapshot is restored. Instead, when a snapshot is taken, Oracle VM VirtualBox creates differencing images which contain only the changes since the snapshot were taken. When the snapshot is restored, Oracle VM VirtualBox throws away that differencing image, thus going back to the previous state. This is both faster and uses less disk space. For the details, which can be complex, see [Section 5.5, "Differencing Images"](#).

Creating the differencing image as such does not occupy much space on the host disk initially, since the differencing image will initially be empty and grow dynamically later with each write operation to the disk. The longer you use the machine after having created the snapshot, however, the more the differencing image will grow in size.

- If you took a snapshot while the machine was running, the memory state of the machine is also saved in the snapshot. This is in the same way that memory can be saved when you close a VM window. When you restore such a snapshot, execution resumes at exactly the point when the snapshot was taken.

The memory state file can be as large as the memory size of the VM and will therefore occupy considerable



disk space.

## 1.12. Virtual Machine Configuration

When you select a virtual machine from the list in the VirtualBox Manager window, you will see a summary of that machine's settings on the right.

Clicking on **Settings** displays a window, where you can configure many of the properties of the selected VM. But be careful when changing VM settings. It is possible to change all VM settings after installing a guest OS, but certain changes might prevent a guest OS from functioning correctly if done after installation.

### Note

The **Settings** button is disabled while a VM is either in the Running or Saved state. This is because the **Settings** window enables you to change fundamental characteristics of the virtual machine that is created for your guest OS. For example, the guest OS may not perform well if half of its memory is taken away. As a result, if the **Settings** button is disabled, shut down the current VM first.

Oracle VM VirtualBox provides a wide range of parameters that can be changed for a virtual machine. The various settings that can be changed in the **Settings** window are described in detail in [Chapter 3, Configuring Virtual Machines](#). Even more parameters are available when using the **VBoxManage** command line interface. See [Chapter 8, VBoxManage](#).

## 1.13. Removing and Moving Virtual Machines

You can remove a VM from Oracle VM VirtualBox or move the VM and its associated files, such as disk images, to another location on the host.

- **Removing a VM.** To remove a VM, right-click on the VM in the VirtualBox Manager machine list and select **Remove**.

The confirmation dialog enables you to specify whether to only remove the VM from the list of machines or to remove the files associated with the VM.

Note that the **Remove** menu item is disabled while a VM is running.

- **Moving a VM.** To move a VM to a new location on the host, right-click on the VM in the VirtualBox Manager's machine list and select **Move**.

The file dialog prompts you to specify a new location for the VM.

When you move a VM, Oracle VM VirtualBox configuration files are updated automatically to use the new location on the host.

Note that the **Move** menu item is disabled while a VM is running.

You can also use the **VBoxManage movevm** command to move a VM. See [Section 8.12, "VBoxManage movevm"](#).

For information about removing or moving a disk image file from Oracle VM VirtualBox, see [Section 5.3, "The Virtual Media Manager"](#).

## 1.14. Cloning Virtual Machines

You can create a full copy or a linked copy of an existing VM. This copy is called a *clone*. You might use a cloned

VM to experiment with a VM configuration, to test different guest OS levels, or to back up a VM.

The **Clone Virtual Machine** wizard guides you through the cloning process.

You can start the Clone Virtual Machine wizard in one of the following ways:

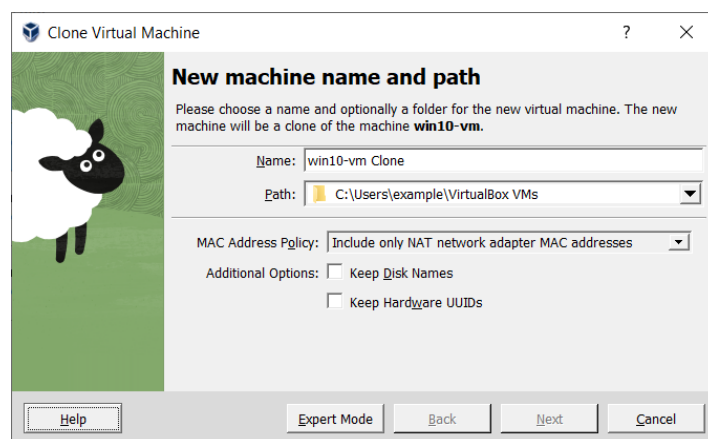
- Click the VM name in the machine list and then select **Clone** from the **Machine** menu.
- Click **Clone** in the **Snapshots** window for the selected VM.

### Note

The **Clone** menu item is disabled while a virtual machine is running.

The **New Machine Name and Path** page is displayed.

**Figure 1.16. Clone Virtual Machine Wizard: New Machine Name and Path**



The following clone options are available:

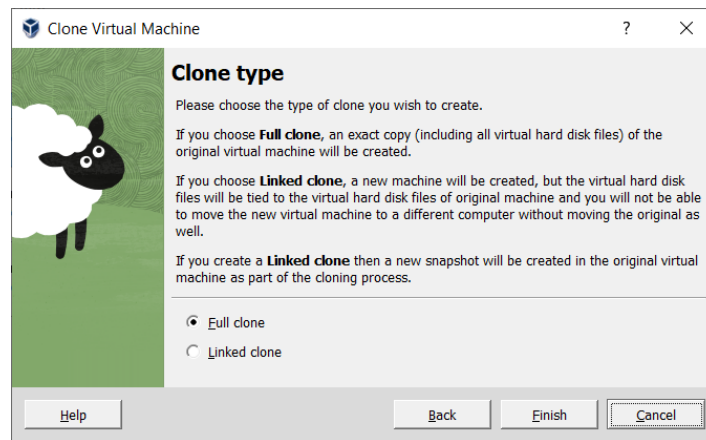
- **Name:** A name for the cloned machine.
- **Path:** Choose a location for the cloned virtual machine, otherwise Oracle VM VirtualBox uses the default machines folder.
- **MAC Address Policy:** Specifies whether to retain network card MAC addresses when cloning the VM.

For example, the **Generate New MAC Addresses For All Network Adapters** value assigns a new MAC address to each network card during cloning. This is the default setting. This is the best option when both the source VM and the cloned VM must operate on the same network. Other values enable you to retain the existing MAC addresses in the cloned VM.

- **Keep Disk Names:** Retains the disk image names when cloning the VM.
- **Keep Hardware UUIDs:** Retains the hardware universally unique identifiers (UUIDs) when cloning the VM.

Click **Next**. The **Clone Type** page is displayed.

**Figure 1.17. Clone Virtual Machine Wizard: Clone Type**



The **Clone Type** option specifies whether to create a clone that is linked to the source VM or to create a fully independent clone:

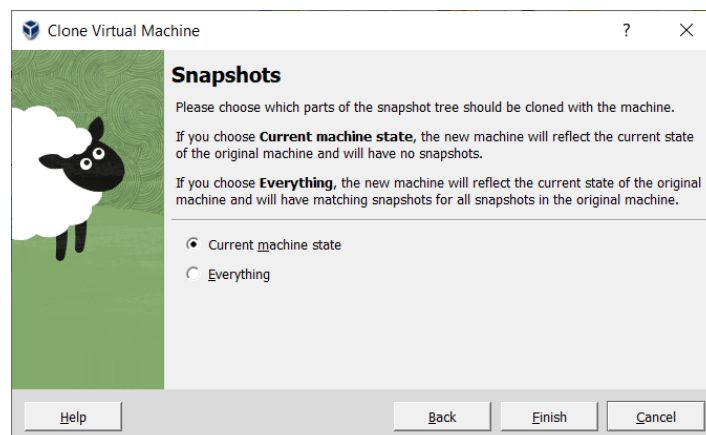
- **Full Clone:** Copies all dependent disk images to the new VM folder. A full clone can operate fully without the source VM.
- **Linked Clone:** Creates new differencing disk images based on the source VM disk images. If you select the current state of the source VM as the clone point, Oracle VM VirtualBox creates a new snapshot.

(Optional) Click **Next**. The **Snapshots** page is displayed.

### Note

The Snapshots page is only displayed for machines that have snapshots and the selected clone type is **Full Clone**.

**Figure 1.18. Clone Virtual Machine Wizard: Snapshots**



You use this page to select which parts of the snapshot tree to include in the clone. The available options are as follows:

- **Current Machine State:** Clones the current state of the VM. Snapshots are not included.
- **Everything:** Clones the current machine state and all its snapshots.

Click **Finish** to start the clone operation.

The duration of the clone operation depends on the size and number of attached disk images. In addition, the clone operation saves all the differencing disk images of a snapshot.

You can also use the **VBoxManage clonevm** command to clone a VM. See [Section 8.11, “VBoxManage clonevm”](#).

## 1.15. Importing and Exporting Virtual Machines

Oracle VM VirtualBox can import and export virtual machines in the following formats:

- **Open Virtualization Format (OVF).** This is the industry-standard format. See [Section 1.15.1, “About the OVF Format”](#).
- **Cloud service formats.** Export to and import from cloud services such as Oracle Cloud Infrastructure is supported. See [Section 1.16, “Integrating with Oracle Cloud Infrastructure”](#).

### 1.15.1. About the OVF Format

OVF is a cross-platform standard supported by many virtualization products which enables the creation of ready-made virtual machines that can then be imported into a hypervisor such as Oracle VM VirtualBox. Oracle VM VirtualBox makes OVF import and export easy to do, using VirtualBox Manager or the command-line interface.

Using OVF enables packaging of *virtual appliances*. These are disk images, together with configuration settings that can be distributed easily. This way one can offer complete ready-to-use software packages, including OSes with applications, that need no configuration or installation except for importing into Oracle VM VirtualBox.

#### Note

The OVF standard is complex, and support in Oracle VM VirtualBox is an ongoing process. In particular, no guarantee is made that Oracle VM VirtualBox supports all appliances created by other virtualization software. For a list of known limitations, see [Chapter 14, \*Known Limitations\*](#).

Appliances in OVF format can appear in the following variants:

- They can come in several files, as one or several disk images, typically in the widely-used VMDK format. See [Section 5.2, “Disk Image Files \(VDI, VMDK, VHD, HDD\)”](#). They also include a textual description file in an XML dialect with an `.ovf` extension. These files must then reside in the same directory for Oracle VM VirtualBox to be able to import them.
- Alternatively, the above files can be packed together into a single archive file, typically with an `.ova` extension. Such archive files use a variant of the TAR archive format and can therefore be unpacked outside of Oracle VM VirtualBox with any utility that can unpack standard TAR files.

#### Note

OVF cannot describe snapshots that were taken for a virtual machine. As a result, when you export a virtual machine that has snapshots, only the current state of the machine will be exported. The disk images in the export will have a *flattened* state identical to the current state of the virtual machine.

### 1.15.2. Importing an Appliance in OVF Format

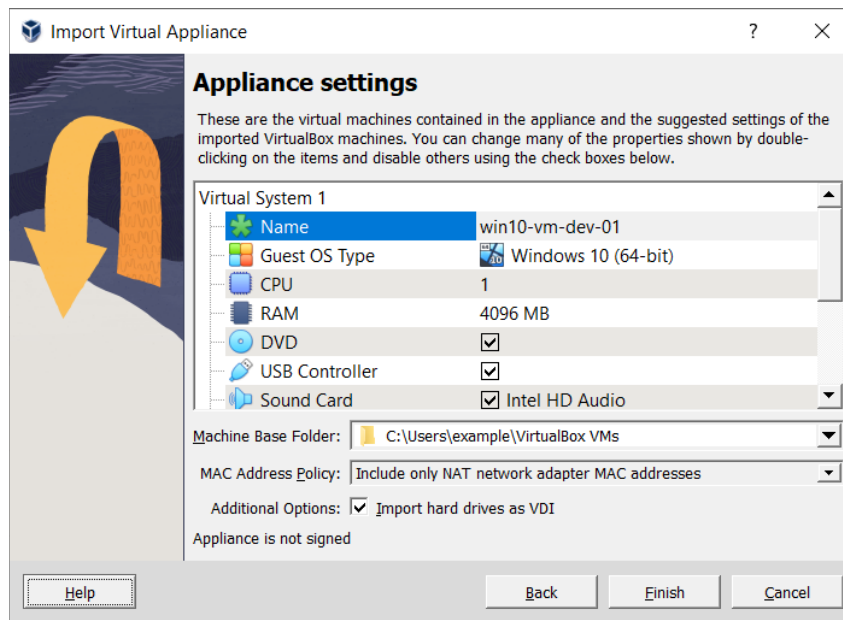
The following steps show how to import an appliance in OVF format.

1. Double-click on the OVF or OVA file.

Oracle VM VirtualBox creates file type associations automatically for any OVF and OVA files on your host OS.

The **Appliance Settings** page of the **Import Virtual Appliance** wizard is shown.

**Figure 1.19. Import Virtual Appliance Wizard: Appliance Settings**



2. The **Appliance Settings** page shows the VMs described in the OVF or OVA file and enables you to change the VM settings.

By default, membership of VM groups is preserved on import for VMs that were initially exported from Oracle VM VirtualBox. You can change this behavior by using the **Primary Group** setting for the VM.

The following global settings apply to all of the VMs that you import:

- **Base Folder:** Specifies the directory on the host in which to store the imported VMs.  
If an appliance has multiple VMs, you can specify a different directory for each VM by editing the **Base Folder** setting for the VM.
- **MAC Address Policy:** Reinitializes the MAC addresses of network cards in your VMs prior to import, by default. You can override the default behavior and preserve the MAC addresses on import.
- **Import Hard Drives as VDI:** Imports hard drives in the VDI format rather than in the default VMDK format.

3. Click **Finish** to import the appliance.

Oracle VM VirtualBox copies the disk images and creates local VMs with the settings described on the **Appliance Settings** page. The imported VMs are shown in the list of VMs in VirtualBox Manager.

Because disk images are large, the VMDK images that are included with virtual appliances are shipped in a compressed format that cannot be used directly by VMs. So, the images are first unpacked and copied, which might take several minutes.

You can use the **VBoxManage import** command to import an appliance. See [Section 8.16, “VBoxManage import”](#).

### 1.15.3. Exporting an Appliance in OVF Format

The following steps show how to export an appliance in OVF format.

1. Select **File, Export Appliance** to display the **Export Virtual Appliance** wizard.

On the initial **Virtual Machines** page, you can combine several VMs into an OVF appliance.

Select one or more VMs to export, and click **Next**.

2. The **Format Settings** page enables you to configure the following settings:

- **Format:** Selects the **Open Virtualization Format** value for the output files.

The **Oracle Cloud Infrastructure** value exports the appliance to Oracle Cloud Infrastructure. See [Section 1.16.8, “Exporting an Appliance to Oracle Cloud Infrastructure”](#).

- **File:** Selects the location in which to store the exported files.
- **MAC Address Policy:** Specifies whether to retain or reassign network card MAC addresses on export.
- **Write Manifest File:** Enables you to include a manifest file in the exported archive file.
- **Include ISO Image Files:** Enables you to include ISO image files in the exported archive file.

3. Click **Next** to show the **Appliance Settings** page.

You can edit settings for the virtual appliance. For example, you can change the name of the virtual appliance or add product information, such as vendor details or license text.

Double-click the appropriate field to change its value.

4. Click **Finish** to begin the export process. Note that this operation might take several minutes.

You can use the **VBoxManage export** command to export an appliance. See [Section 8.17, “VBoxManage export”](#).

## 1.16. Integrating with Oracle Cloud Infrastructure

This section describes how to use the features of Oracle VM VirtualBox to integrate with Oracle Cloud Infrastructure.

Integrating with Oracle Cloud Infrastructure involves the following steps:

- **Prepare for Oracle Cloud Infrastructure Integration.** Before using Oracle VM VirtualBox with Oracle Cloud Infrastructure there are some initial configuration steps you may need to do. See [Section 1.16.1, “Preparing for Oracle Cloud Infrastructure Integration”](#).
- **Use Oracle VM VirtualBox with Oracle Cloud Infrastructure.** [Section 1.16.6, “Using Oracle VM VirtualBox With Oracle Cloud Infrastructure”](#) describes how you can use Oracle VM VirtualBox with Oracle Cloud Infrastructure.

### 1.16.1. Preparing for Oracle Cloud Infrastructure Integration

Perform the following configuration steps before using Oracle VM VirtualBox to integrate with your Oracle Cloud Infrastructure account.

1. **Install the Extension Pack.** Cloud integration features are only available when you install the Oracle VM VirtualBox Extension Pack. See [Section 1.5, “Installing Oracle VM VirtualBox and Extension Packs”](#).
2. **Create a key pair.** Generate an API signing key pair that is used for API requests to Oracle Cloud Infrastructure. See [Section 1.16.2, “Creating an API Signing Key Pair”](#).

Upload the public key of the key pair from your client device to the cloud service. See [Section 1.16.3, “Uploading the Public Key to Oracle Cloud Infrastructure”](#).

3. **Create a cloud profile.** The cloud profile contains resource identifiers for your cloud account, such as your user OCID, and details of your key pair. See [Section 1.16.4, “Creating a Cloud Profile”](#).



### 1.16.2. Creating an API Signing Key Pair

To use the cloud integration features of Oracle VM VirtualBox, you must generate an API signing key pair that is used for API requests to Oracle Cloud Infrastructure.

Your API requests are signed with your private key, and Oracle Cloud Infrastructure uses the public key to verify the authenticity of the request. You must upload the public key to the Oracle Cloud Infrastructure Console.

#### Note

This key pair is not the same SSH key that you use to access compute instances on Oracle Cloud Infrastructure.

1. (Optional) Create a `.oci` directory to store the key pair.

```
$ mkdir ~/.oci
```

The key pair is usually installed in the `.oci` folder in your home directory. For example, `~/.oci` on a Linux system.

2. Generate the private key.

Use the **openssl** command.

- To generate a private key with a passphrase:

```
$ openssl genrsa -out ~/.oci/oci_api_key.pem -aes128 2048
```

- To generate a private key without a passphrase:

```
$ openssl genrsa -out ~/.oci/oci_api_key.pem 2048
```

3. Change permissions for the private key.

```
$ chmod 600 ~/.oci/oci_api_key.pem
```

Generate the public key.

```
$ openssl rsa -pubout -in ~/.oci/oci_api_key.pem -out ~/.oci/oci_api_key_public.pem
```

### 1.16.3. Uploading the Public Key to Oracle Cloud Infrastructure

Use the following steps to upload your public key to Oracle Cloud Infrastructure.

1. Log in to the Oracle Cloud Infrastructure Console.

2. Display the **User Settings** page.

Click **Profile, User Settings**.

3. Display your current API signing keys.

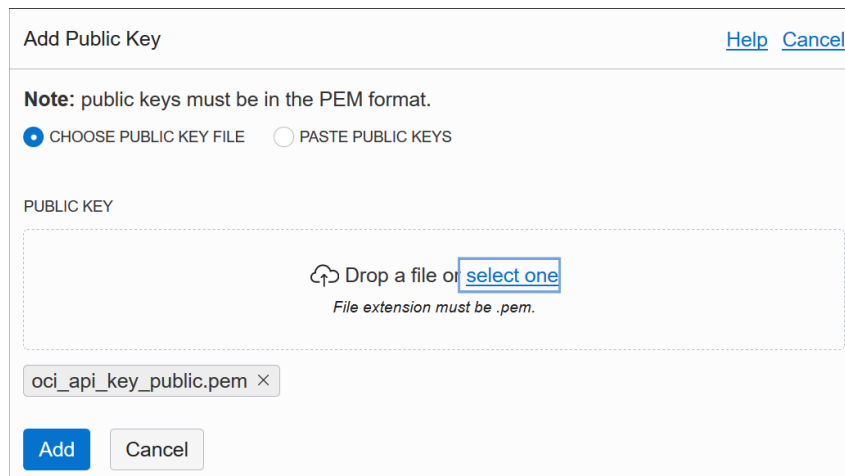
Click **Resources, API Keys**.

4. Upload the public key.

Click **Add Public Key**.

The **Add Public Key** dialog is displayed.

**Figure 1.20. Upload Public Key Dialog in Oracle Cloud Infrastructure Console**



Select one of the following options:

- **Choose Public Key File.** This option enables you to browse to the public key file on your local hard disk.
- **Paste Public Keys.** This option enables you to paste the contents of the public key file into the window in the dialog box.

Click **Add** to upload the public key.

#### 1.16.4. Creating a Cloud Profile

Oracle VM VirtualBox uses a *cloud profile* to connect to Oracle Cloud Infrastructure. A cloud profile is a text file that contains details of your key files and Oracle Cloud Identifier (OCID) resource identifiers for your cloud account, such as the following:

- **Fingerprint of the public key.** To obtain the fingerprint, you can use the **openssl** command:  

```
$ openssl rsa -pubout -outform DER -in ~/.oci/oci_api_key.pem | openssl md5 -c
```
- **Location of the private key on the client device.** Specify the full path to the private key.
- **(Optional) Passphrase for the private key.** This is only required if the key is encrypted.
- **Region.** Shown on the Oracle Cloud Infrastructure Console. Click **Administration, Tenancy Details**.
- **Tenancy OCID.** Shown on the Oracle Cloud Infrastructure Console. Click **Administration, Tenancy Details**.  
A link enables you to copy the Tenancy OCID.
- **Compartment OCID.** Shown on the Oracle Cloud Infrastructure Console. Click **Identity, Compartments**.  
A link enables you to copy the Compartment OCID.
- **User OCID.** Shown on the Oracle Cloud Infrastructure Console. Click **Profile, User Settings**.  
A link enables you to copy the User OCID.

You can create a cloud profile in the following ways:

- Automatically, by using the **Cloud Profile Manager**. See [Section 1.16.5, “Using the Cloud Profile Manager”](#).

The Cloud Profile Manager is a VirtualBox Manager tool that enables you to create, edit, and manage cloud profiles for your cloud service accounts.

- Automatically, by using the **VBoxManage cloudprofile** command. See [Section 8.15, “VBoxManage cloudprofile”](#).
- Manually, by creating an `oci_config` file in your Oracle VM VirtualBox global configuration directory. For example, this is `$HOME/.config/VirtualBox/oci_config` on a Linux host.
- Manually, by creating a `config` file in your Oracle Cloud Infrastructure configuration directory. For example, this is `$HOME/.oci/config` on a Linux host.

This is the same file that is used by the Oracle Cloud Infrastructure command line interface.

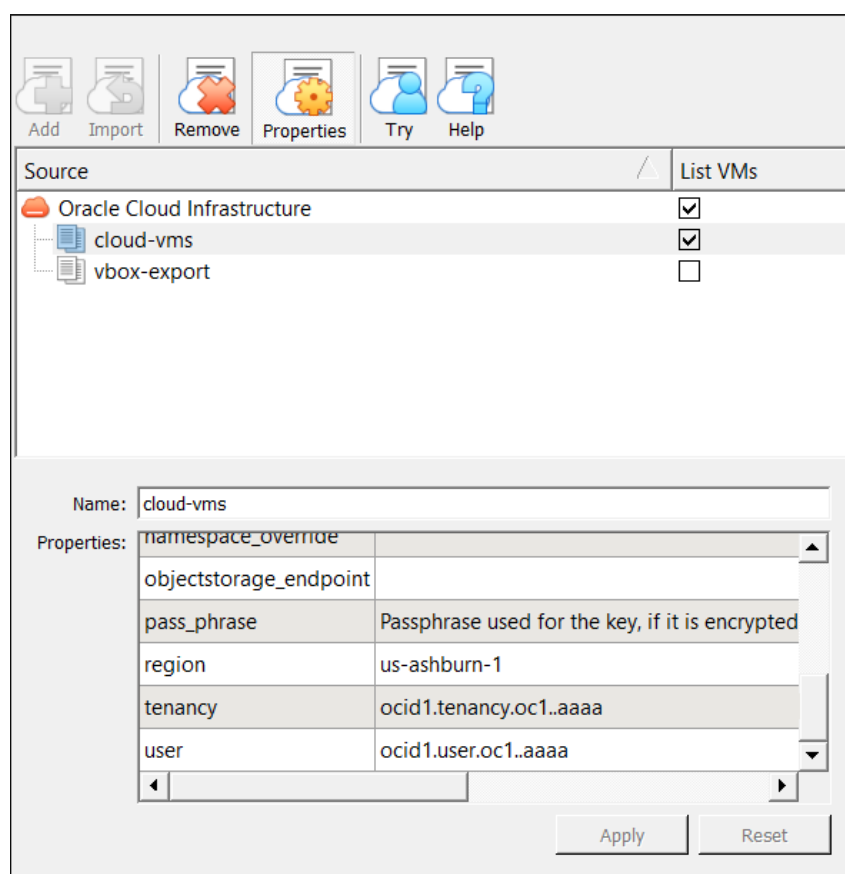
Oracle VM VirtualBox automatically uses the `config` file if no cloud profile file is present in your global configuration directory. Alternatively, you can import this file manually into the Cloud Profile Manager.

### 1.16.5. Using the Cloud Profile Manager

This section describes how to use the Cloud Profile Manager to create a cloud profile.

To open the Cloud Profile Manager click **File, Cloud Profile Manager** in VirtualBox Manager.

**Figure 1.21. The Cloud Profile Manager**



You can use the Cloud Profile Manager in the following ways:

- To create a new cloud profile automatically
- To create a cloud profile by importing settings from your Oracle Cloud Infrastructure configuration file.

Perform the following steps to create a new cloud profile automatically, using the Cloud Profile Manager:

1. Click the **Add** icon and specify a **Name** for the profile.

2. Click **Properties** and specify the following property values for the profile:

- Compartment OCID
- Fingerprint of the public key
- Location of the private key on the client device
- Region OCID
- Tenancy OCID
- User OCID

Some of these are settings for your Oracle Cloud Infrastructure account, which you can view from the Oracle Cloud Infrastructure Console.

3. (Optional) If you are using the cloud profile to connect to cloud virtual machines, select the **Show VMs** check box.

This creates a new subgroup of the **OCI** group in VirtualBox Manager. See [Section 1.16.7.1, “About the OCI VM Group”](#).

4. Click **Apply** to save your changes.

The cloud profile settings are saved to the `oci_config` file in your Oracle VM VirtualBox global settings directory.

Perform the following steps to import an existing Oracle Cloud Infrastructure configuration file into the Cloud Profile Manager:

1. Ensure that a `config` file is present in your Oracle Cloud Infrastructure configuration directory. For example, this is `$HOME/.oci/config` on a Linux host.
2. Click the **Import** icon to open a dialog that prompts you to import cloud profiles from external files.

### Warning

This action overwrites any cloud profiles that are in your Oracle VM VirtualBox global settings directory.

3. Click **Import**.

Your cloud profile settings are saved to the `oci_config` file in your Oracle VM VirtualBox global settings directory.

4. Click **Properties** to show the cloud profile settings.

Double-click on the appropriate field to change the value.

5. Click **Apply** to save your changes.

## 1.16.6. Using Oracle VM VirtualBox With Oracle Cloud Infrastructure

This section describes how you can use Oracle VM VirtualBox with Oracle Cloud Infrastructure to do the following tasks:

- Create, add, and manage Oracle Cloud Infrastructure cloud instances using VirtualBox Manager. See [Section 1.16.7, “Using Cloud Virtual Machines”](#).

- Export an Oracle VM VirtualBox VM to Oracle Cloud Infrastructure. See [Section 1.16.8, “Exporting an Appliance to Oracle Cloud Infrastructure”](#).
- Import a cloud instance into Oracle VM VirtualBox. See [Section 1.16.9, “Importing an Instance from Oracle Cloud Infrastructure”](#).
- Connect from a local VM to an Oracle Cloud Infrastructure cloud subnet. See [Section 1.16.10, “Using a Cloud Network”](#).
- Use the **VBoxManage** commands to integrate with Oracle Cloud Infrastructure and perform cloud operations. See [Section 1.16.11, “Using VBoxManage Commands With Oracle Cloud Infrastructure”](#).

## 1.16.7. Using Cloud Virtual Machines

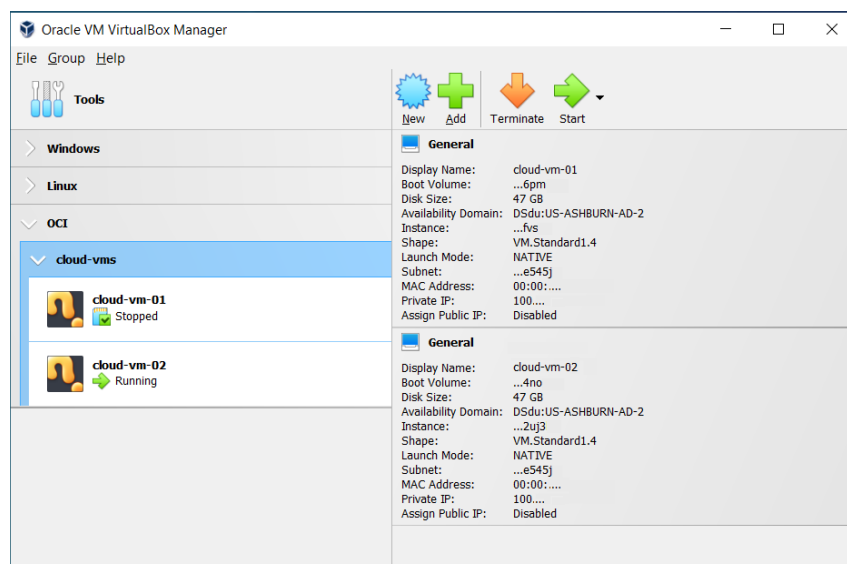
A cloud virtual machine (*cloud VM*) is a type of VM that represents an instance on a cloud service. Cloud VMs are shown in the machine list in VirtualBox Manager, in the same way as local VMs are.

By using cloud VMs you can create, manage, and control your Oracle Cloud Infrastructure instances from VirtualBox Manager.

### Note

Cloud VMs do not install, export, or import instances to the Oracle VM VirtualBox host. All operations are done remotely on the cloud service.

**Figure 1.22. Cloud VMs, Shown in VirtualBox Manager**



Cloud VMs can be used to do the following tasks in Oracle Cloud Infrastructure:

- **Create a new Oracle Cloud Infrastructure instance.** See [Section 1.16.7.2, “Creating a New Cloud VM”](#).
- **Use an existing Oracle Cloud Infrastructure instance.** See [Section 1.16.7.3, “Adding a Cloud VM”](#).
- **Configure an Oracle Cloud Infrastructure instance.** You can change settings for the instance, such as display name and shape. See [Section 1.16.7.4, “Changing Settings for a Cloud VM”](#).
- **Control an Oracle Cloud Infrastructure instance.** Stop, start, and terminate the instance. See [Section 1.16.7.5, “Controlling a Cloud VM”](#).
- **Create a console connection to an Oracle Cloud Infrastructure instance.** See [Section 1.16.7.7, “Creating](#)

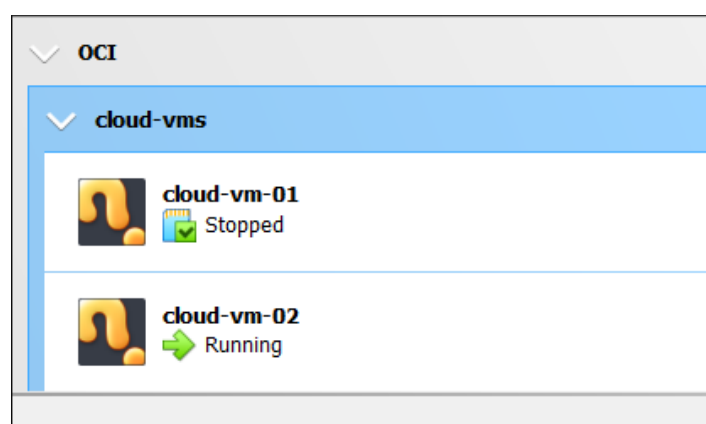
[an Instance Console Connection for a Cloud VM](#)".

### 1.16.7.1. About the OCI VM Group

All cloud VMs are shown in the machine list in VirtualBox Manager, in a special VM group called **OCI**.

Cloud VMs are further grouped according to the cloud profile used to connect to them. The cloud profile identifies the user and compartment for the cloud VM and includes details of the key pair used to connect to cloud instances. See [Section 1.16.4, "Creating a Cloud Profile"](#).

**Figure 1.23. OCI Group, Containing Cloud VMs**



All cloud profiles registered with Oracle VM VirtualBox are listed automatically in the OCI group.

To enable or disable listing of cloud VMs in VirtualBox Manager for a specific cloud profile, do the following:

Display the **Cloud Profile Manager** and select or deselect the **List VMs** check box for each cloud profile.

### 1.16.7.2. Creating a New Cloud VM

When you create a new cloud VM, a *new* Oracle Cloud Infrastructure instance is created and associated with the cloud VM.

Perform the following steps to create a new cloud VM:

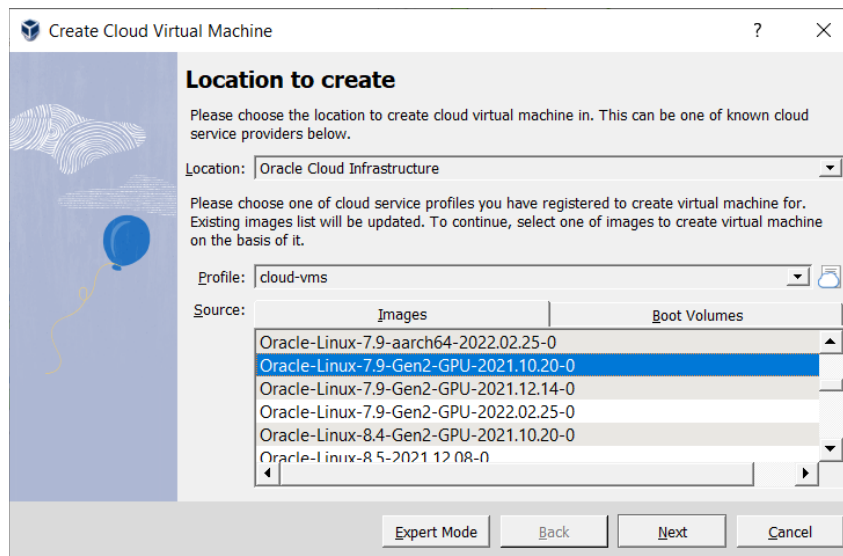
1. Click on a cloud profile in the **OCI** group.

The cloud VMs for the selected cloud profile are displayed.

2. Select **Group, New Machine**.

The **Create Cloud Virtual Machine** wizard is displayed.

**Figure 1.24. Create Cloud Virtual Machine Wizard**



3. On the initial page, configure the following settings for the new cloud VM:

- **Location:** The cloud service provider that will host the new instance. Select **Oracle Cloud Infrastructure**.
- **Profile:** The cloud profile used to connect to the new instance. Select from the available cloud profiles.
- **Source:** The image that the new instance is based on. Choose from the available images and boot volumes.

4. Click **Next** to display the **Cloud Virtual Machine Settings** page.

You can use this page to change the default settings for the new Oracle Cloud Infrastructure instance, such as the display name, shape, and networking configuration.

To add an SSH key to the instance, click the **SSH Authorised Keys** field and paste the public key into the displayed dialog.

5. Click **Finish** to create a new Oracle Cloud Infrastructure instance using the selected image or boot volume. The new instance is started automatically.

The new cloud VM is shown in the **OCI** group in VirtualBox Manager.

### 1.16.7.3. Adding a Cloud VM

When you add a cloud VM, an *existing* Oracle Cloud Infrastructure instance is associated with the cloud VM. You can only add one cloud VM for each instance.

Perform the following steps to add a cloud VM:

1. Click on a cloud profile in the **OCI** group.

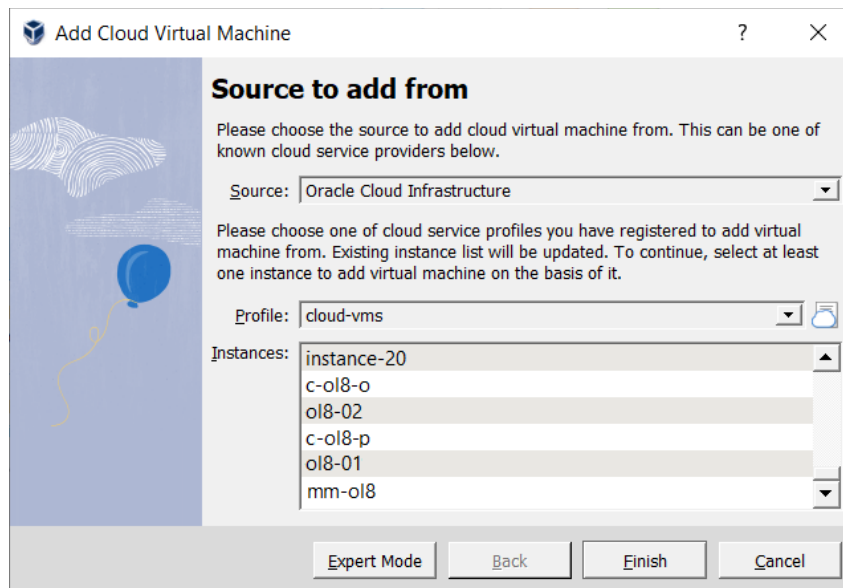
The cloud VMs for the selected cloud profile are displayed.

2. Select **Group, Add Machine**.

The **Add Cloud Virtual Machine** wizard is displayed.

**Figure 1.25. Add Cloud Virtual Machine Wizard**





3. Configure the following settings:

- **Source:** The cloud service provider that hosts the instance used for the cloud VM. Select **Oracle Cloud Infrastructure**.
- **Profile:** The cloud profile used to connect to the running instance. Select from the available cloud profiles.
- **Instances:** The instance to use for the cloud VM. Choose from the available instances on your cloud service.

4. Click **Finish** to add a cloud VM based on the selected instance.

A cloud VM with the same name as the instance is added to the **OCI** group in VirtualBox Manager.

5. (Optional) To change the display name for the instance, click **Settings** and edit the **Display Name** field.

The cloud VM name in VirtualBox Manager is updated automatically.

#### 1.16.7.4. Changing Settings for a Cloud VM

Select the cloud VM in VirtualBox Manager and click **Settings**.

- For a *new* cloud VM, you can change many settings for the Oracle Cloud Infrastructure instance, such as the display name, shape, and disk size.
- When you *add* a cloud VM based on an existing Oracle Cloud Infrastructure instance you can only change the display name.

#### 1.16.7.5. Controlling a Cloud VM

You can use VirtualBox Manager to control a cloud VM as follows:

- **Start.** Use the **Start** button in the VirtualBox Manager toolbar.
- **Stop.** Right-click on the cloud VM name, to display the **Close** menu. Options to shut down and power off the cloud VM are available.
- **Terminate.** Use the **Terminate** button in the VirtualBox Manager toolbar.

## Caution

This action deletes the instance from Oracle Cloud Infrastructure.

When you control a cloud VM in VirtualBox Manager the machine list is updated automatically with the current instance state, such as **Stopped** or **Running**.

When you control an instance using the Oracle Cloud Infrastructure console, VirtualBox Manager updates the status for the corresponding cloud VM automatically.

### 1.16.7.6. Removing a Cloud VM

You can use VirtualBox Manager to remove a cloud VM as follows:

Right-click on the cloud VM name and select **Remove**.

- Click **Remove Only** to remove the cloud VM from the machine list in VirtualBox Manager.
- Click **Delete Everything** to remove the cloud VM from VirtualBox Manager and also to delete the Oracle Cloud Infrastructure instance and any associated boot volumes.

### 1.16.7.7. Creating an Instance Console Connection for a Cloud VM

To create a instance console connection, the cloud VM must be in **Running** state.

1. Right-click on the cloud VM name and select **Console, Create Connection**.
2. The **Public Key** dialog is displayed. Paste the public key used for the instance connection into the dialog and click **OK**.  
  
By default, either the first entry in your SSH keys folder or the public key used for your previous instance console connection is used.
3. Click **Connect** to connect to the instance. An instance console is displayed automatically on the host.
4. (Optional) Click **Show Log** to display log messages for the instance console connection.

See the Oracle Cloud Infrastructure documentation for details about how you can use an instance console connection to troubleshoot instance problems.

### 1.16.8. Exporting an Appliance to Oracle Cloud Infrastructure

Oracle VM VirtualBox supports the export of VMs to an Oracle Cloud Infrastructure service. The exported VM is stored on Oracle Cloud Infrastructure as a custom Linux image. You can configure whether a cloud instance is created and started after the export process has completed.

#### Note

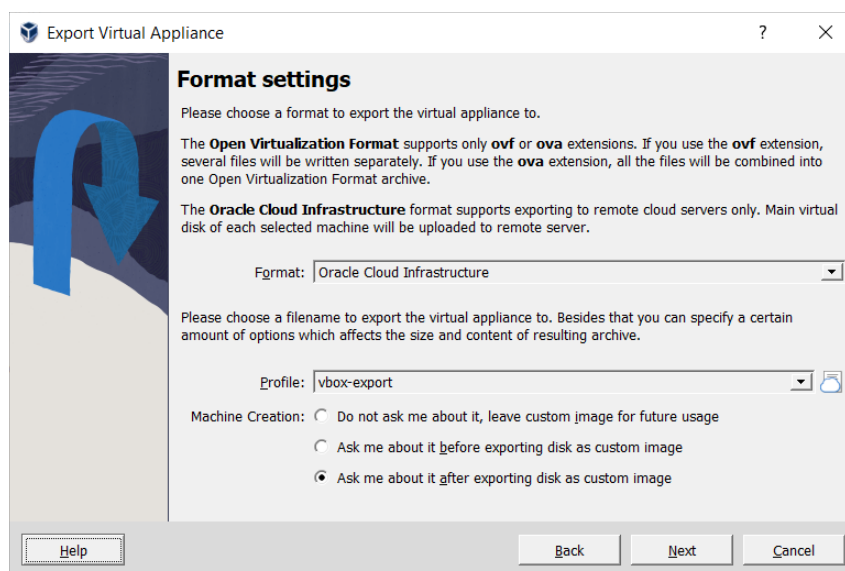
Before you export a VM to Oracle Cloud Infrastructure, you must prepare the VM as described in [Section 1.16.8.1, "Preparing a VM for Export to Oracle Cloud Infrastructure"](#).

Use the following steps to export a VM to Oracle Cloud Infrastructure:

1. Select **File, Export Appliance** to open the **Export Virtual Appliance** wizard.  
  
Select a VM to export and click **Next** to display the **Format Settings** page.
2. From the **Format** drop-down list, select **Oracle Cloud Infrastructure**.

In the **Profile** drop-down list, select the cloud profile used for your Oracle Cloud Infrastructure account.

**Figure 1.26. Export Virtual Appliance Wizard: Format Settings**



In the **Machine Creation** field, select an option to configure settings for the cloud instance created when you export to Oracle Cloud Infrastructure. The options enable you to do one of the following:

- Configure settings for the cloud instance *after* you have finished exporting the VM.
- Configure settings for the cloud instance *before* you start to export the VM.
- Do not create a cloud instance when you export the VM.

Click **Next** to make an API request to the Oracle Cloud Infrastructure service and open the **Appliance Settings** page.

3. (Optional) Edit storage settings used for the exported virtual machine in Oracle Cloud Infrastructure. You can change the following settings:

- The name of the bucket used to store the exported files.
- Whether to store the custom image in Oracle Cloud Infrastructure.
- The display name for the custom image in Oracle Cloud Infrastructure.
- The launch mode for the custom image.

**Paravirtualized** mode gives improved performance and should be suitable for most Oracle VM VirtualBox VMs.

**Emulated** mode is suitable for legacy OS images.

Click **Finish** to continue.

4. (Optional) Depending on the selection in the **Machine Creation** field, the **Appliance Settings** page may be displayed before or after export. This screen enables you to configure settings for the cloud instance, such as Shape and Disk Size.

Click **Finish**. The VM is exported to Oracle Cloud Infrastructure.

Depending on the **Machine Creation** setting, a cloud instance may be started after upload to Oracle Cloud

Infrastructure is completed.

5. Monitor the export process by using the Oracle Cloud Infrastructure Console.

You can also use the **VBoxManage export** command to export a VM to Oracle Cloud Infrastructure. See [Section 8.17.2.2, “Export a Virtual Machine to Oracle Cloud Infrastructure”](#).

### 1.16.8.1. Preparing a VM for Export to Oracle Cloud Infrastructure

Oracle Cloud Infrastructure provides the option to import a custom Linux image. Before an Oracle VM VirtualBox image can be exported to Oracle Cloud Infrastructure, the custom image needs to be prepared to ensure that instances launched from the custom image can boot correctly and that network connections will work. This section provides advice on how to prepare a Linux image for export from Oracle VM VirtualBox.

The following list shows some tasks to consider when preparing an Oracle Linux VM for export:

- **Use DHCP for network addresses.** Configure the VM to use a DHCP server to allocate network addresses, rather than using a static IP address. The Oracle Cloud Infrastructure instance will then be allocated an IP address automatically.
- **Do not specify a MAC address.** The network interface configuration for the VM must not specify the MAC address.

Remove the HWADDR setting from the `/etc/sysconfig/ifcfg-devicename` network script.

- **Disable persistent network device naming rules.** This means that the Oracle Cloud Infrastructure instance will use the same network device names as the VM.

1. Change the GRUB kernel parameters.

Add `net.ifnames=0` and `biosdevname=0` as kernel parameter values to the `GRUB_CMDLINE_LINUX` variable.

2. Update the GRUB configuration.

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Disable any `udev` rules for network device naming.

For example, if an automated `udev` rule exists for `net-persistence`:

```
# cd /etc/udev/rules.d
# rm -f 70-persistent-net.rules
# ln -s /dev/null /etc/udev/rules.d/70-persistent-net.rules
```

- **Enable the serial console.** This enables you to troubleshoot the instance when it is running on Oracle Cloud Infrastructure.

1. Edit the `/etc/default/grub` file, as follows:

- Remove the `resume` setting from the kernel parameters. This setting slows down boot time significantly.
- Replace `GRUB_TERMINAL="gfxterm"` with `GRUB_TERMINAL="console serial"`. This configures use of the serial console instead of a graphical terminal.
- Add `GRUB_SERIAL_COMMAND="serial --unit=0 --speed=115200"`. This configures the serial connection.
- Add `console=tty0 console=ttyS0,115200` to the `GRUB_CMDLINE_LINUX` variable. This adds the serial console to the Linux kernel boot parameters.

## 2. Regenerate the GRUB configuration.

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

## 3. To verify the changes, reboot the machine and run the **dmesg** command to look for the updated kernel parameters.

```
# dmesg |grep console=ttyS0
```

- **Enable paravirtualized device support.** You do this by adding the `virtio` drivers to the `initrd` for the VM.

### 1. This procedure works only on machines with a Linux kernel of version 3.4 or later. Check that the VM is running a supported kernel:

```
# uname -a
```

### 2. Use the `dracut` tool to rebuild `initrd`. Add the `qemu` module, as follows:

```
# dracut --logfile /var/log/Dracut.log --force --add qemu
```

### 3. Verify that the `virtio` drivers are now present in `initrd`.

```
# lsinitrd |grep virtio
```

For more information about importing a custom Linux image into Oracle Cloud Infrastructure, see also:

<https://docs.cloud.oracle.com/iaas/Content/Compute/Tasks/importingcustomimagelinux.htm>

## 1.16.9. Importing an Instance from Oracle Cloud Infrastructure

Perform the following steps to import a cloud instance from Oracle Cloud Infrastructure into Oracle VM VirtualBox:

### 1. Select **File, Import Appliance** to open the **Import Virtual Appliance** wizard.

In the **Source** drop-down list, select **Oracle Cloud Infrastructure**.

In the **Profile** drop-down list, select the cloud profile for your Oracle Cloud Infrastructure account.

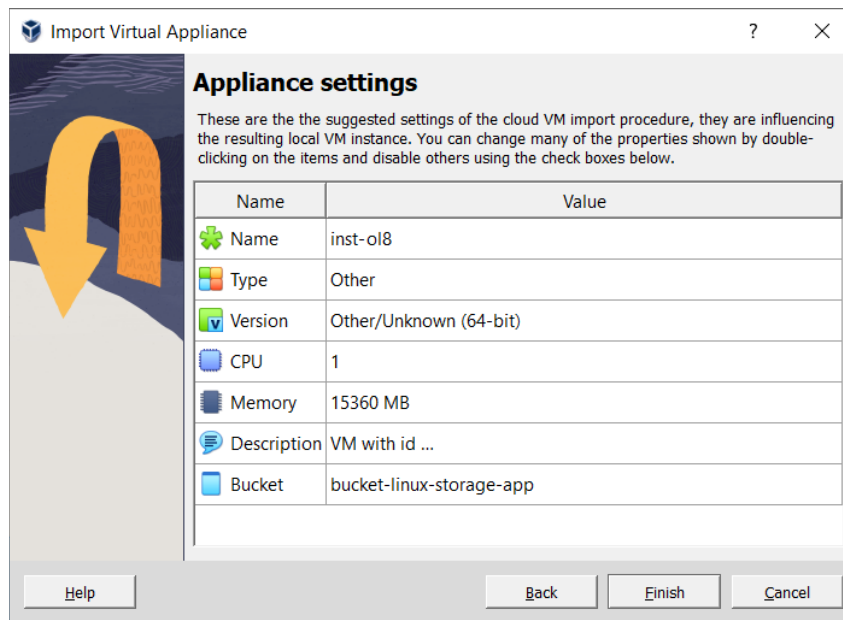
Choose the required cloud instance from the list in the **Machines** field.

Click **Next** to make an API request to the Oracle Cloud Infrastructure service and display the **Appliance Settings** page.

### 2. (Optional) Edit settings for the new local virtual machine.

For example, you can edit the VM name and description.

**Figure 1.27. Import Cloud Instance Wizard: Appliance Settings**



Click **Finish** to import the instance from Oracle Cloud Infrastructure.

3. Monitor the import process by using the Oracle Cloud Infrastructure Console.

You can also use the **VBoxManage import** command to import an instance from Oracle Cloud Infrastructure. See [Section 8.16.2.3, “Cloud Import Options”](#).

### Importing an Instance: Overview of Events

The following describes the sequence of events when you import an instance from Oracle Cloud Infrastructure.

- A custom image is created from the boot volume of the instance.
- The custom image is exported to an Oracle Cloud Infrastructure object and is stored using Object Storage in the bucket specified by the user.
- The Oracle Cloud Infrastructure object is downloaded to the local host. The object is a TAR archive which contains a boot volume of the instance in QCOW2 format and a JSON file containing metadata related to the instance.
- The boot volume of the instance is extracted from the archive and a new VMDK image is created by converting the boot volume into the VMDK format. The VMDK image is registered with Oracle VM VirtualBox.
- A new VM is created using the VMDK image for the cloud instance.

By default, the new VM is not started after import from Oracle Cloud Infrastructure.

- The downloaded TAR archive is deleted after a successful import.

### 1.16.10. Using a Cloud Network

A cloud network is a type of network that can be used for connections from a local VM to a remote Oracle Cloud Infrastructure cloud instance.

To create and use a cloud network, do the following:

1. Set up a virtual cloud network on Oracle Cloud Infrastructure.

The following steps create and configure a virtual cloud network (VCN) on Oracle Cloud Infrastructure. The VCN is used to tunnel network traffic across the cloud.

- a. Ensure that you have a cloud profile for connecting to Oracle Cloud Infrastructure. See [Section 1.16.4, “Creating a Cloud Profile”](#).
- b. Run the following **VBoxManage cloud** command:

```
VBoxManage cloud --provider="OCI" --profile="vbox-oci" network setup
```

where `vbox-oci` is the name of your cloud profile.

Other options are available for the **VBoxManage cloud network setup** command, to enable you to configure details for the VCN. For example, you can configure the operating system used for the cloud gateway instance and the IP address range used by the tunneling network. See [Section 8.14, “VBoxManage cloud”](#).

For best results, use an Oracle Linux 7 instance for the cloud gateway. This is the default option.

2. Register the new cloud network with Oracle VM VirtualBox.

Use the **Cloud Networks** tab in the **Network Manager** tool. See [Section 6.11.3, “Cloud Networks Tab”](#).

3. Add cloud network adaptors to the local VMs that will use the cloud network. See [Section 6.10, “Cloud Networks”](#).

### 1.16.11. Using VBoxManage Commands With Oracle Cloud Infrastructure

This section includes some examples of how **VBoxManage** commands can be used to integrate with Oracle Cloud Infrastructure and perform common cloud operations.

#### Creating a Cloud Profile

To create a cloud profile called `vbox-oci`:

```
VBoxManage cloudprofile --provider "OCI" --profile="vbox-oci" add \
--clouduser="ocid1.user.oc1..." --keyfile="/home/username/.oci/oci_api_key.pem" \
--tenancy="ocid1.tenancy.oc1..." --compartment="ocid1.compartment.oc1..." --region="us-ashburn-1"
```

The new cloud profile is added to the `oci_config` file in your Oracle VM VirtualBox global configuration directory. For example, this is `$HOME/.VirtualBox/oci_config` on a Windows host.

#### Listing Cloud Instances

To list the instances in your Oracle Cloud Infrastructure compartment:

```
VBoxManage cloud --provider="OCI" --profile="vbox-oci" list instances
```

#### Exporting an Oracle VM VirtualBox VM to the Cloud

To export a VM called `myVM` and create a cloud instance called `myVM_Cloud`:

```
VBoxManage export myVM --output OCI:// --cloud 0 --vmname myVM_Cloud \
--cloudprofile "vbox-oci" --cloudbucket myBucket \
--cloudshape VM.Standard2.1 --clouddomain US-ASHBURN-AD-1 --clouddisksize 50 \
--cloudocivcn ocid1.vcn.oc1... --cloudocisubnet ocid1.subnet.oc1... \
--cloudkeepobject true --cloudlaunchinstance true --cloudpublicip true
```

#### Importing a Cloud Instance Into Oracle VM VirtualBox



To import a cloud instance and create an Oracle VM VirtualBox VM called `oci_Import`:

```
VBoxManage import OCI:// --cloud --vmname oci_Import --memory 4000
--cpus 3 --ostype FreeBSD_64 --cloudprofile "vbox-oci"
--cloudinstanceid ocid1.instance.oc1... --cloudbucket myBucket
```

## Creating a New Cloud Instance From a Custom Image

To create a new cloud instance from a custom image on Oracle Cloud Infrastructure:

```
VBoxManage cloud --provider="OCI" --profile="vbox-oci" instance create \
--domain-name="oraclecloud.com" --image-id="ocid1.image.oc1..." --display-name="myInstance" \
--shape="VM.Standard2.1" --subnet="ocid1.subnet.oc1..."
```

## Terminating a Cloud Instance

To terminate an instance in your compartment on Oracle Cloud Infrastructure:

```
VBoxManage cloud --provider="OCI" --profile="vbox-oci" instance terminate \
--id="ocid1.instance.oc1..."
```

For more details about the available commands for cloud operations, see [Section 8.14, “VBoxManage cloud”](#).

## 1.17. Preferences

The Preferences window offers a selection of settings, which apply to all virtual machines of the current user.

To display the Preferences window, do either of the following:

- Select **File, Preferences**.
- Click **Preferences** on the Welcome screen in VirtualBox Manager.

The following settings are available:

- **General.** Enables you to specify the default folder or directory for VM files, and the VRDP Authentication Library.
- **Input.** Enables you to specify keyboard shortcuts, such as the **Host key**. This is the key that toggles whether the cursor is in the focus of the VM or the Host OS windows, see [Section 1.9.2, “Capturing and Releasing Keyboard and Mouse”](#). The Host key is also used to trigger certain VM actions, see [Section 1.9.3, “Typing Special Characters”](#).
- **Update.** Enables you to specify various settings for Automatic Updates.
- **Language.** Enables you to specify the language used for menus, labels, and text in VirtualBox Manager.
- **Display.** Enables you to specify the screen resolution, and its width and height. A default scale factor can be specified for all guest screens.
- **Proxy.** Enables you to configure an HTTP Proxy Server.
- **Interface.** Enables you to select a color theme for the VirtualBox Manager user interface.

### Note

This setting is only available on Windows host platforms.

## 1.18. Alternative Front-Ends

As briefly mentioned in [Section 1.3, “Features Overview”](#), Oracle VM VirtualBox has a very flexible internal design that enables you to use multiple interfaces to control the same virtual machines. For example, you can start a virtual machine with the VirtualBox Manager window and then stop it from the command line. With Oracle VM VirtualBox's support for the Remote Desktop Protocol (RDP), you can even run virtual machines remotely on a headless server and have all the graphical output redirected over the network.

The following front-ends are shipped in the standard Oracle VM VirtualBox package:

- **VirtualBox.** This is the VirtualBox Manager, a graphical user interface that uses the Qt toolkit. This interface is described throughout this manual. While this is the simplest and easiest front-end to use, some of the more advanced Oracle VM VirtualBox features are not included.
- **VBoxManage.** A command-line interface for automated and detailed control of every aspect of Oracle VM VirtualBox. See [Chapter 8, VBoxManage](#).
- **VBoxHeadless.** A front-end that produces no visible output on the host at all, but can act as a RDP server if the VirtualBox Remote Desktop Extension (VRDE) is installed and enabled for the VM. As opposed to the other graphical interfaces, the headless front-end requires no graphics support. This is useful, for example, if you want to host your virtual machines on a headless Linux server that has no X Window system installed. See [Section 7.1.2, “VBoxHeadless, the Remote Desktop Server”](#).

If the above front-ends still do not satisfy your particular needs, it is possible to create yet another front-end to the complex virtualization engine that is the core of Oracle VM VirtualBox, as the Oracle VM VirtualBox core neatly exposes all of its features in a clean API. See [Chapter 11, Oracle VM VirtualBox Programming Interfaces](#).

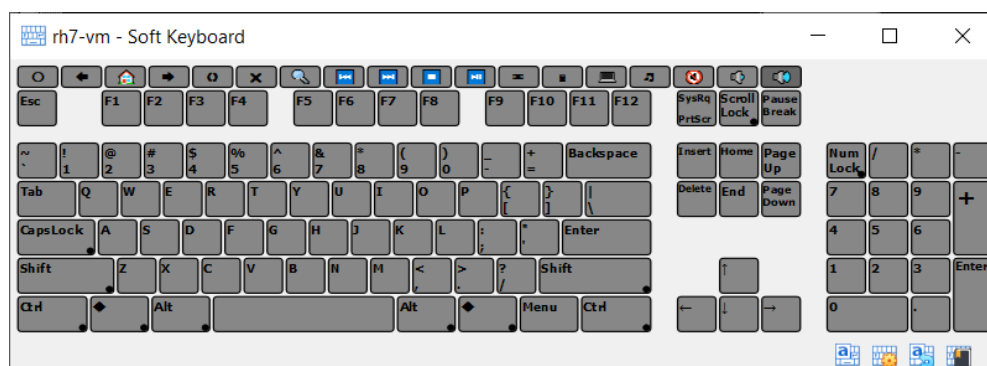
## 1.19. Soft Keyboard

Oracle VM VirtualBox provides a *soft keyboard* that enables you to input keyboard characters on the guest. A soft keyboard is an on-screen keyboard that can be used as an alternative to a physical keyboard. See [Section 1.19.1, “Using the Soft Keyboard”](#) for details of how to use the soft keyboard.

### Caution

For best results, ensure that the keyboard layout configured on the guest OS matches the keyboard layout used by the soft keyboard. Oracle VM VirtualBox does not do this automatically.

**Figure 1.28. Soft Keyboard in a Guest Virtual Machine**



The soft keyboard can be used in the following scenarios:

- When the physical keyboard on the host is not the same as the keyboard layout configured on the guest. For example, if the guest is configured to use an international keyboard, but the host keyboard is US English.
- To send special key combinations to the guest. Note that some common key combinations are also

available in the **Input, Keyboard** menu of the guest VM window. See [Section 1.9.3, “Typing Special Characters”](#).

- For guests in kiosk mode, where a physical keyboard is not present.
- When using nested virtualization, the soft keyboard provides a method of sending key presses to a guest.

By default, the soft keyboard includes some common international keyboard layouts. You can copy and modify these to meet your own requirements. See [Section 1.19.2, “Creating a Custom Keyboard Layout”](#).

### 1.19.1. Using the Soft Keyboard

1. Display the soft keyboard.

In the guest VM window, select **Input, Keyboard, Soft Keyboard**.

2. Select the required keyboard layout.

The name of the current keyboard layout is displayed in the toolbar of the soft keyboard window. This is the previous keyboard layout that was used.

Click the **Layout List** icon in the toolbar of the soft keyboard window. The **Layout List** window is displayed.

Select the required keyboard layout from the entries in the **Layout List** window.

The keyboard display graphic is updated to show the available input keys.

3. Use the soft keyboard to enter keyboard characters on the guest.

- Modifier keys such as Shift, Ctrl, and Alt are available on the soft keyboard. Click once to select the modifier key, click twice to lock the modifier key.

The **Reset the Keyboard and Release All Keys** icon can be used to release all pressed modifier keys, both on the host and the guest.

- To change the look of the soft keyboard, click the **Settings** icon in the toolbar. You can change colors used in the keyboard graphic, and can hide or show sections of the keyboard, such as the NumPad or multimedia keys.

### 1.19.2. Creating a Custom Keyboard Layout

You can use one of the supplied default keyboard layouts as the starting point to create a custom keyboard layout.

#### Note

To permanently save a custom keyboard layout, you must save it to a file. Otherwise, any changes you make are discarded when you close down the **Soft Keyboard** window. Custom keyboard layouts that you save are stored as an XML file on the host, in the `keyboardLayouts` folder in the global configuration data directory. For example, in `$HOME/.config/VirtualBox/keyboardLayouts` on a Linux host.

1. Display the **Layout List**.

Click the **Layout List** icon in the toolbar of the soft keyboard window.

2. Make a copy of an existing keyboard layout.

Highlight the required layout and click the **Copy the Selected Layout** icon.

A new layout entry with a name suffix of `-Copy` is created.

### 3. Edit the new keyboard layout.

Highlight the new layout in the **Layout List** and click the **Edit the Selected Layout** icon.

Enter a new name for the layout.

Edit keys in the new layout. Click on the key that you want to edit and enter new key captions in the **Captions** fields.

The keyboard graphic is updated with the new captions.

### 4. (Optional) Save the layout to a file. This means that your custom keyboard layout will be available for future use.

Highlight the new layout in the **Layout List** and click the **Save the Selected Layout into File** icon.

Any custom layouts that you create can later be removed from the Layout List, by highlighting and clicking the **Delete the Selected Layout** icon.

## 1.20. Monitoring of Virtual Machines

VirtualBox Manager includes the following tools for viewing runtime information and changing the configuration of virtual machines.

- **VM Activity Overview.** Displays an overview of performance metrics for all running VMs.

See [Section 1.20.1, “VM Activity Overview”](#).

- **Session Information Dialog.** Displays configuration and runtime information for the selected guest system.

See [Section 1.20.2, “Session Information Dialog”](#)

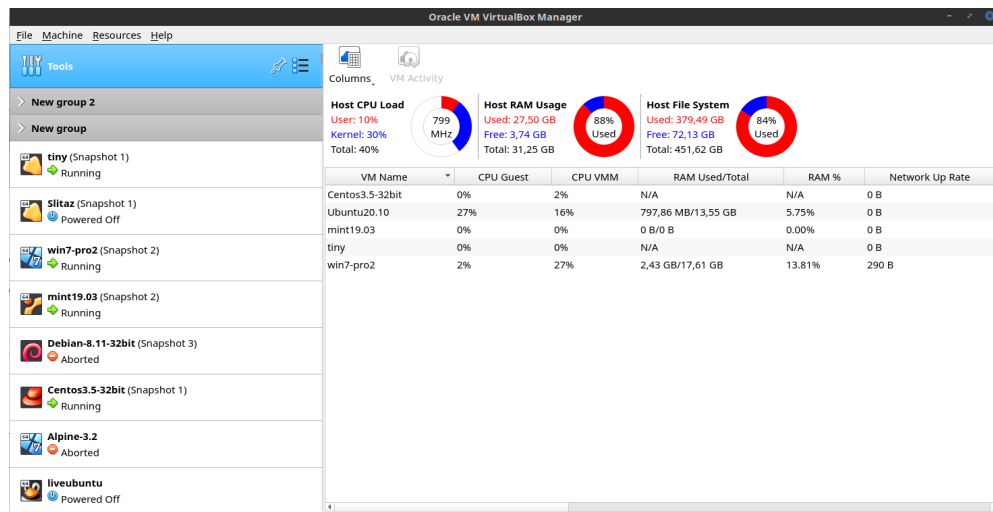
### 1.20.1. VM Activity Overview

The VM Activity Overview tool displays several performance metrics for all running virtual machines and for the host system. This provides an overview of system resources used by individual virtual machines and the host system.

To display the VM Activity Overview tool, do the following:

Go to the global **Tools** menu and click **Activities**. The **VM Activity Overview** window is shown.

**Figure 1.29. VM Activity Overview Tool**



To show metrics for *all* virtual machines, including those that are not running, right-click on the list of virtual machines and select **List All Virtual Machines**.

To configure the set of metrics to be shown, click **Columns** in the toolbar. You can then sort the list of virtual machines by a particular metric.

To see more performance information for a virtual machine, select the VM name and click **VM Activity** in the toolbar. The **VM Activity** tab of the **Session Information** dialog is shown, see [Section 1.20.2, "Session Information Dialog"](#).

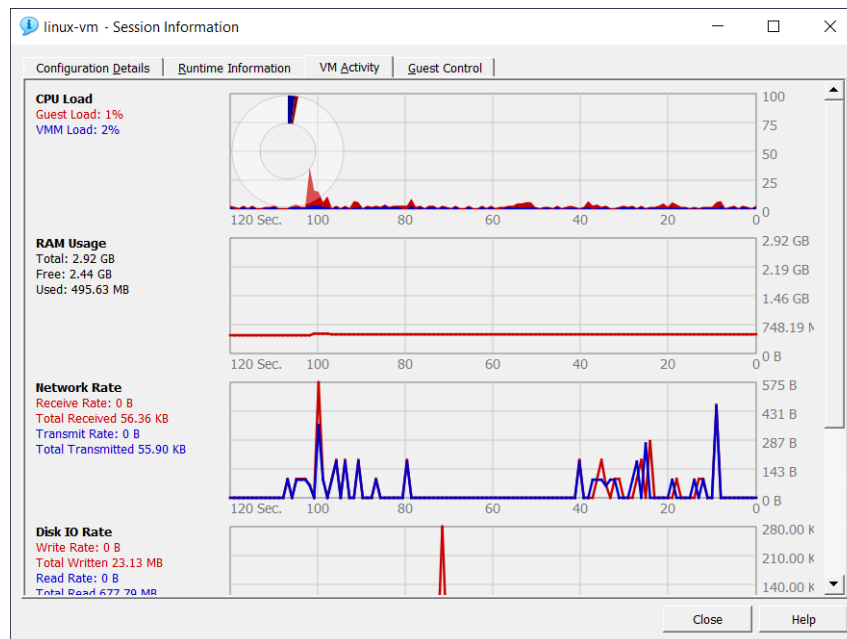
### 1.20.2. Session Information Dialog

The Session Information dialog includes multiple tabs which show important configuration and runtime information for the guest system. The tabs of the dialog are as follows:

- **Configuration Details.** Displays the system configuration of the virtual machine in a tabular format. The displayed information includes details such as storage configuration and audio settings.
- **Runtime Information.** Displays runtime information for the guest session in a tabular format similar to the Configuration Details tab.
- **VM Activity.** Includes several time series charts which monitor guest resource usage including CPU, RAM, Disk I/O, and Network. Note that the RAM chart requires the Guest Additions to be running on the guest system. The VM Activity tab can also be accessed directly from the VM Activity Overview tool. See [Section 1.20.1, "VM Activity Overview"](#).
- **Guest Control.** Details of processes used by the Guest Control File Manager. See [Section 4.8, "Guest Control File Manager"](#).

To display the Session Information dialog, select **Machine, Session Information** in the guest VM.

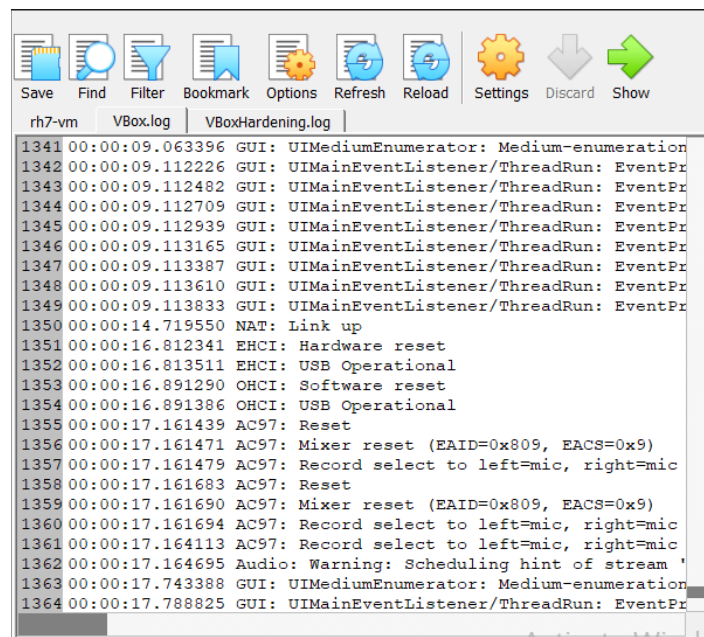
**Figure 1.30. Session Information Dialog, Showing VM Activity Tab**



## 1.21. The Log Viewer

Every time you start up a VM, Oracle VM VirtualBox creates a log file that records system configuration and events. The **Log Viewer** is a VirtualBox Manager tool that enables you to view and analyze system logs.

**Figure 1.31. Log Viewer Tool, Showing System Events**



To display the Log Viewer, do either of the following:

- Click the VM name in the machine list and select **Logs** from the machine tools menu.
- In the guest VM, select **Machine, Show Log**.

Log messages for the VM are displayed in tabs in the Log Viewer window. See [Section 12.1.2, “Collecting Debugging Information”](#) for details of the various log files generated by Oracle VM VirtualBox.

If you select multiple VMs in the machine list, logs are listed for each VM.

The toolbar of the Log Viewer includes the following options:

- **Save:** Exports the contents of the selected log file to a text file. Specify the destination filename and location in the displayed dialog.
- **Find:** Searches for a text string in the log file.
- **Filter:** Uses filter terms to display specific types of log messages. Common log message terms used by Oracle VM VirtualBox, such as Audio and NAT, are included by default. Select one or more terms from the drop-down list. To add your own filter term, enter the text string in the text box field.
- **Bookmark:** Saves the location of a log message, enabling you to find it quickly. To create a bookmark, either click on the line number, or select some text and then click **Bookmark**.
- **Options:** Configures the text display used in the log message window.
- **Refresh:** Refreshes the log file you are currently viewing. Only log messages in the current tab are updated.
- **Reload:** Refreshes all log files. Log messages in every tab are updated.
- **Settings:** Displays the **Settings** window for the VM, enabling you to make configuration changes.
- **Discard:** For a running VM, discards the saved state for the VM and closes it down.
- **Show/Start:** For a running VM, **Show** displays the VM window. For a stopped VM, **Start** displays options for powering up the VM.