

Oracle® VM

Concepts Guide for Release 3.3



E50249-05
July 2015

Oracle Legal Notices

Copyright © 2014, 2015 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Abstract

Document generated on: 2015-07-20 (revision: 4803)

Table of Contents

Preface	vii
1 Audience	vii
2 Related Documents	vii
3 Command Syntax	vii
4 Conventions	viii
1 Introduction to Virtualization	1
1.1 What is the History of Virtualization?	1
1.2 Why use Virtualization?	1
1.3 What are Hypervisors?	3
2 Oracle VM Overview	5
2.1 What is the Oracle VM Architecture?	5
2.2 What is Oracle VM Server?	8
2.2.1 What Hypervisors are used in Oracle VM?	8
2.2.2 What are Domains?	9
2.2.3 What is Oracle VM Agent?	10
2.2.4 What are Server Pools?	11
2.3 What is Oracle VM Manager?	11
2.3.1 Oracle VM Manager Server and API	12
2.3.2 Oracle VM Database Repository	12
2.3.3 Oracle VM Manager Web-Based User Interface	12
2.3.4 Oracle VM Manager Command Line Interface	13
2.3.5 Oracle VM Utilities	13
2.4 What is a Oracle VM Guest?	13
2.5 Authentication and Encryption in Oracle VM	14
2.6 What Else does Oracle VM Require?	16
2.6.1 Storage	16
2.6.2 Networking	17
2.7 What are Virtual Machines?	18
2.7.1 What are Virtualization Modes?	19
2.7.2 What are the Oracle VM Guest Additions?	20
2.8 What Features Does Oracle VM Provide?	20
2.9 What Support is Available for Oracle VM?	21
2.10 Can Oracle Enterprise Manager 12c Manage Oracle VM?	21
3 Understanding Storage	23
3.1 What is Storage Used for in Oracle VM?	23
3.2 What Types of Storage Can be Used?	24
3.2.1 Local Storage	25
3.2.2 Shared Network Attached Storage (NFS)	26
3.2.3 iSCSI Storage Attached Network	26
3.2.4 Fibre Channel Storage Attached Network	27
3.3 How is a Storage Created?	28
3.4 How do Different Storage Types Connect?	28
3.5 What is Multipathing?	30
3.6 What are Uniform and Non-uniform Exports?	30
3.7 What are Access Groups?	31
3.8 How is Storage Used for Server Pool Clustering?	31
3.9 Where are Virtual Machine Resources Located?	32
3.10 Are there Guidelines for Configuring Storage?	32
4 Understanding Repositories	35
4.1 What are the Repository Prerequisites?	36
4.2 How is a Repository Organized?	36

4.3 How is a Repository Created?	37
4.4 How do Oracle VM Servers Access a Repository?	37
4.5 What Virtual Machine Resources are in a Repository?	38
4.6 How are Virtual Machine Templates Managed?	38
4.6.1 Using a Virtual Machine Template	39
4.6.2 Importing a Virtual Machine Template	39
4.6.3 Virtual Machine Template Clone Customizers	40
4.7 How are Assemblies Managed?	40
4.8 How are ISO Files (CD/DVD Images) Managed?	41
4.9 How are Virtual Disks Managed?	41
4.10 How are Virtual Machine Configuration Files Managed?	42
4.11 Can the Repository be Backed Up?	42
5 Understanding Networks	43
5.1 Networking Terminology	44
5.2 How are Networks Used in Oracle VM?	45
5.3 How are IP Addresses Assigned?	48
5.4 How is Network Bonding Used in Oracle VM?	48
5.5 How are VLANs Used in Oracle VM?	50
5.5.1 Configuring VLANs	51
5.5.2 Configuring VLAN Interfaces	51
5.6 How are Network Functions Separated in Oracle VM?	52
5.6.1 Management Channel	54
5.6.2 Cluster Heartbeat Channel	55
5.6.3 Live Migrate Channel	56
5.6.4 Storage Channel	56
5.6.5 Virtual Machine Channel	56
5.7 Network Planning for an Oracle VM Deployment	57
5.7.1 Networks with Ports and/or Bonds	58
5.7.2 Networks with VLANs Only	59
5.7.3 Hybrid Networks (VLANs with Ports and Bonds)	60
5.7.4 Logical Networks on a Single Oracle VM Server (Intra-Server)	61
5.8 Dealing with Failed Network Operations	61
6 Understanding Server Pools and Oracle VM Servers	63
6.1 How are Oracle VM Servers Added?	64
6.2 What are Server Roles?	65
6.3 How is Maintenance Performed on an Oracle VM Server?	65
6.4 Configuring NTP for Oracle VM Servers.	66
6.5 Rebooting and Changing Power State of Oracle VM Servers	67
6.6 What are Server Pools used for in Oracle VM?	67
6.7 How are Server Pools Created?	68
6.8 How do Server Pool Clusters Work?	69
6.8.1 Clustering for x86 Server Pools	69
6.8.2 Clustering for SPARC Server Pools	74
6.9 Unclustered Server Pools	74
6.10 How does High Availability (HA) Work?	76
6.11 What are Server Pool Policies?	78
6.11.1 Distributed Resource Scheduler (DRS)	79
6.11.2 Distributed Power Management (DPM)	79
6.11.3 DRS/DPM Network Policies	80
6.12 What are Anti-Affinity Groups?	80
6.13 What are Server Processor Compatibility Groups?	80
7 Understanding Virtual Machines	83
7.1 What are Virtualization Modes or Domain Types?	84
7.2 How is a Guest OS Installed on a Virtual Machine?	86

7.3 Where are Virtual Machine Resources Stored?	86
7.4 What Networking is Available for Virtual Machines?	87
7.5 How are Virtual Machines Created?	87
7.6 How does Cloning Work?	88
7.7 How Can a Virtual Machine be Moved or Migrated?	89
7.8 Editing Virtual Machine Parameters	90
7.9 Sending Messages to Virtual Machines	90
7.10 Accessing the Virtual Machine Console	91
7.11 How Can I Protect Virtual Machines?	92
7.12 How are Virtual CPUs Allocated to Virtual Machines	92
7.13 How are Huge Pages Enabled for Virtual Machines?	93
7.14 Setting Hard Partitioning for Virtual Machine CPUs	94
7.15 How are Virtual Machines Backed Up?	94
8 Understanding Jobs, Events and Errors	95
8.1 What are Job States?	96
8.2 How are Resources Locked During Jobs?	97
8.3 Are Failed Jobs Rolled Back?	97
8.4 What are Recurring Jobs?	98
8.5 Acknowledging Events/Errors	98
Glossary	99
Index	109

Preface

Table of Contents

1 Audience	vii
2 Related Documents	vii
3 Command Syntax	vii
4 Conventions	viii

The preface contains information on how to use the Oracle VM Concepts Guide.

1 Audience

The Oracle VM Concepts Guide is intended for system administrators and end users who want to learn the fundamentals of virtualization, Oracle VM, and who may require a conceptual overview of the different elements that make up and are used within an Oracle VM environment. This guide outlines the concepts behind the different components of Oracle VM and links to procedures in the [Oracle VM Manager User's Guide](#) that outline how to perform the operations using the Oracle VM Manager Web Interface.

2 Related Documents

For more information, see the following documents in the Oracle VM documentation set:

- [Oracle VM Release Notes](#)
- [Oracle VM Installation and Upgrade Guide](#)
- [Oracle VM Concepts Guide](#)
- [Oracle VM Manager Getting Started Guide](#)
- [Oracle VM Manager User's Guide](#)
- [Oracle VM Manager Command Line Interface User's Guide](#)
- [Oracle VM Administrator's Guide](#)
- [Oracle VM Windows Paravirtual Drivers Installation Guide](#)
- [Oracle VM Web Services API Developer's Guide](#)
- [Oracle VM Security Guide](#)
- [Oracle VM Licensing Information User Manual](#)

You can also get the latest information on Oracle VM by going to the Oracle VM Web site:

<http://www.oracle.com/us/technologies/virtualization/oraclevm>

3 Command Syntax

Oracle Linux command syntax appears in `monospace` font. The dollar character (\$), number sign (#), or percent character (%) are Oracle Linux command prompts. Do not enter them as part of the command. The following command syntax conventions are used in this guide:

Convention	Description
backslash \	A backslash is the Oracle Linux command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash: dd if=/dev/rdsd/c0t1d0s6 of=/dev/rst0 bs=10b \ count=10000
braces { }	Braces indicate required items: .DEFINE {macro1}
brackets []	Brackets indicate optional items: cvtcrt <i>termname</i> [<i>outfile</i>]
ellipses ...	Ellipses indicate an arbitrary number of similar items: CHKVAL fieldname value1 value2 ... valueN
<i>italics</i>	Italic type indicates a variable. Substitute a value for the variable: <i>library_name</i>
vertical line	A vertical line indicates a choice within braces or brackets: FILE <i>filesize</i> [K M]
forward slash /	A forward slash is used as an escape character in the Oracle VM Manager Command Line Interface to escape the special characters ", ', ?, \, /, <, >. Special characters need only be escaped when within single or double quotes: create Tag name=MyTag description="HR/'s VMs"

4 Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<i>monospace</i>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1 Introduction to Virtualization

Table of Contents

1.1 What is the History of Virtualization?	1
1.2 Why use Virtualization?	1
1.3 What are Hypervisors?	3

This chapter provides introductory information on virtualization. It discusses why you would want to use virtualization, the technology provided, and a description of the advantages of using Oracle VM to implement your virtualization solution.

1.1 What is the History of Virtualization?

The concept of virtualization is generally believed to have its origins in the mainframe days in the late 1960s and early 1970s, when IBM invested a lot of time and effort in developing robust time-sharing solutions. Time-sharing refers to the shared usage of computer resources among a large group of users, aiming to increase the efficiency of both the users and the expensive computer resources they share. This model represented a major breakthrough in computer technology: the cost of providing computing capability dropped considerably and it became possible for organizations, and even individuals, to use a computer without actually owning one. Similar reasons are driving virtualization for industry standard computing today: the capacity in a single server is so large that it is almost impossible for most workloads to effectively use it. The best way to improve resource utilization, and at the same time simplify data center management, is through virtualization.

Data centers today use virtualization techniques to provide abstraction from the physical hardware, create large aggregated pools of logical resources consisting of CPUs, memory, disks, file storage, applications, networking, and offer those resources to users or customers in the form of agile, scalable, consolidated virtual machines. Even though the technology and use cases have evolved, the core meaning of virtualization remains the same: to enable a computing environment to run multiple independent systems at the same time.

1.2 Why use Virtualization?

There are many different good reasons for companies and organizations to invest in virtualization today, but it is probably safe to assume that financial motivation is number one on the list: virtualization can save a lot of money. Below is an overview of the key benefits of virtualization.

Table 1.1 Reasons to Use Virtualization

Item	Description
Resource optimization	<p>Today's enterprise level computer resources are so powerful that they often have excess capacity. By virtualizing the hardware and allocating parts of it based on the real needs of users and applications, the available computing power, storage space and network bandwidth can be used much more effectively. Computers no longer need to be idle or performing below their capabilities because there are fewer connected users, or because the hosted application happens to be less demanding than the server can handle.</p> <p>Virtual machines offer software developers isolated, constrained, test environments. Rather than purchasing dedicated physical hardware, virtual machines can be created on the existing hardware. Because each virtual machine is independent and isolated from all the other servers, programmers</p>

Item	Description
	can run software without having to worry about affecting other applications, or external components affecting the execution of their code.
Consolidation	It is common practice to dedicate individual computers to a single application. If several applications only use a small amount of processing power, the administrator can consolidate several computers into one server running multiple virtual environments. For organizations that own hundreds or thousands of servers, consolidation can dramatically reduce the need for floor space, HVAC, A/C power, and co-location resources. This means the cost of ownership is reduced significantly, since less physical servers and floor and rack space are required, which in turn leads to less heat and power consumption, and ultimately a smaller carbon footprint.
Maximizing Uptime	<p>Agility is all about being able to respond to changing requirements as quickly and flexibly as possible. Virtualization brings new opportunities to data center administration, allowing users to enjoy:</p> <ul style="list-style-type: none"> • Guaranteed uptime of servers and applications; speedy disaster recovery if large scale failures do occur. • Instant deployment of new virtual machines or even aggregated pools of virtual machines via template images. • Elasticity, that is, resource provisioning when and where required instead of keeping the entire data center in an <i>always-on</i> state. • Reconfiguration of running computing environments without impacting the users.
Automatically Protect Applications from Server Failure	<p>Server virtualization provides a way to implement redundancy without purchasing additional hardware. Redundancy, in the sense of running the same application on multiple servers, is a safety measure: if for any reason a server fails, another server running the same application takes over, thereby minimizing the interruption in service. This kind of redundancy works in two ways when applied to virtual machines:</p> <ul style="list-style-type: none"> • If one virtual system fails, another virtual system takes over. • By running the redundant virtual machines on separate physical hardware you can also provide better protection against physical hardware failure.
Easily Migrate Workloads as Needs Change	Migration refers to moving a server environment from one place to another. With most virtualization solutions it is possible to move a virtual machine from one physical machine in the environment to another. With physical servers this was originally possible only if both physical machines ran on the same hardware, operating system and processor. In the virtual world, a server can be migrated between physical hosts with entirely different hardware configurations. Migration is typically used to improve reliability and availability: in case of hardware failure the guest system can be moved to a healthy server with limited downtime, if any. It is also useful if a virtual machine needs to scale beyond the physical capabilities of the current host and must be relocated to physical hardware with better performance.
Protect Investment in Existing, Legacy Systems	Server hardware will eventually become obsolete, and switching from one system to another can be difficult. In order to continue offering the services provided by these legacy systems, you can run it as a virtual machine on

Item	Description
	new, modern hardware, while the legacy system itself still behaves as if it were running on the same legacy hardware. From an application perspective, nothing has changed. In fact, its performance may well benefit from the newer underlying hardware. This gives the organization the time to transition to new processes without worrying about hardware issues, particularly in situations where the manufacturer of the legacy hardware no longer exists or cannot fix broken equipment.

1.3 What are Hypervisors?

If virtualization is defined as enabling multiple operating systems to run on a single [host computer](#), then the essential component in the virtualization stack is the [hypervisor](#). This hypervisor, also called Virtual Machine Monitor (VMM), creates a virtual platform on the host computer, on top of which multiple [guest](#) operating systems are executed and monitored. This way, multiple operating systems, which are either multiple instances of the same operating system, or different operating systems, can share the hardware resources offered by the host.

[Hypervisors](#) are commonly classified as one of these two types, as show in [Table 1.2, "Hypervisor Types"](#).

Table 1.2 Hypervisor Types

Classification	Characteristics and Description
Type 1: <i>native or bare metal</i>	Native hypervisors are software systems that run directly on the host's hardware to control the hardware, and to monitor the guest operating systems. Consequently, the guest operating system runs on a separate level above the hypervisor. Examples of this classic implementation of virtual machine architecture are Oracle VM, Microsoft Hyper-V, VMware ESXi and Xen.
Type 2: <i>hosted</i>	Hosted hypervisors are designed to run within a traditional operating system. In other words, a hosted hypervisor adds a distinct software layer on top of the host operating system, and the guest operating system becomes a third software level above the hardware. A well-known example of a hosted hypervisor is Oracle VM VirtualBox. Others include VMware Server and Workstation, Microsoft Virtual PC, KVM, QEMU and Parallels.

Chapter 2 Oracle VM Overview

Table of Contents

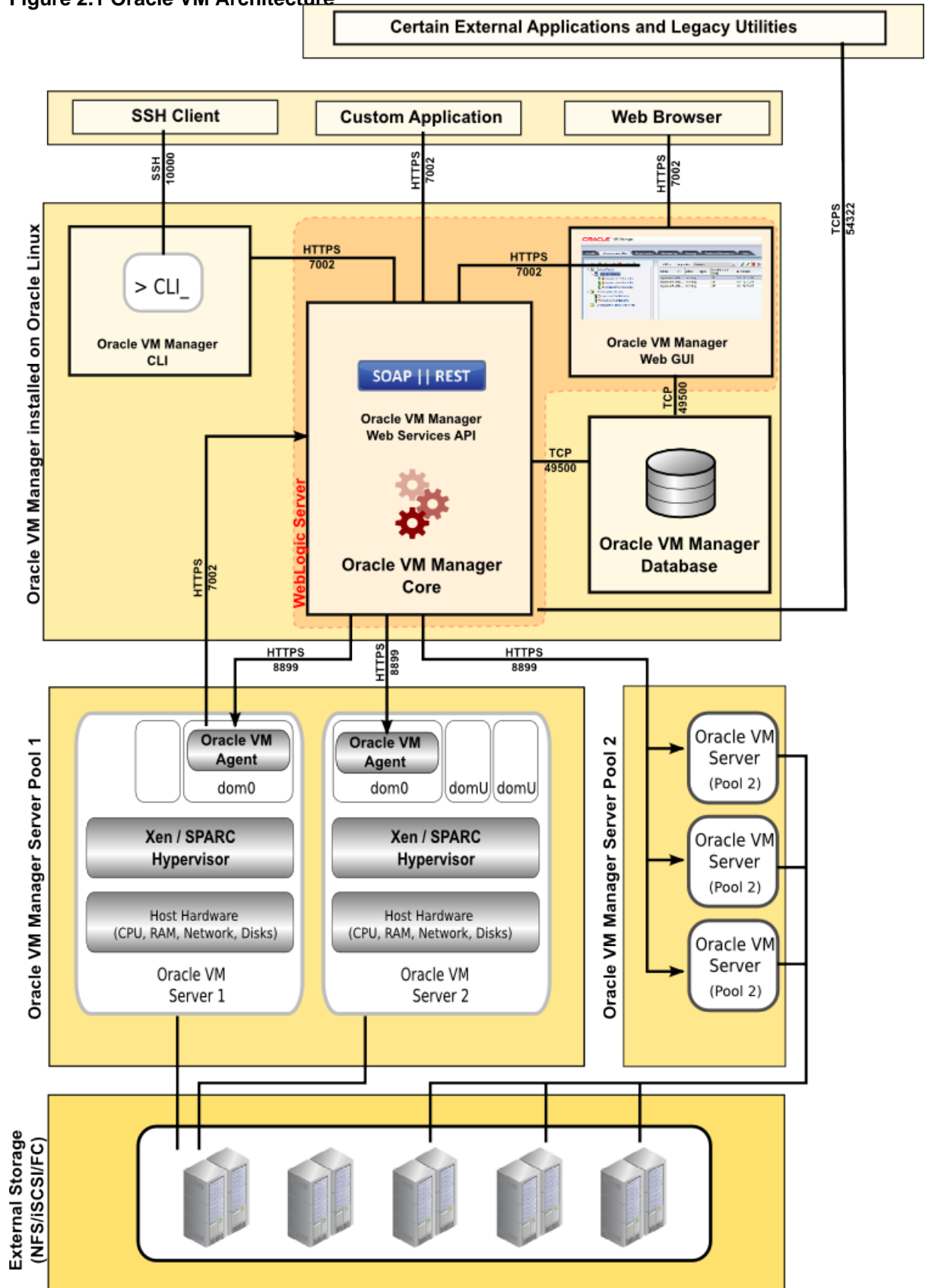
2.1 What is the Oracle VM Architecture?	5
2.2 What is Oracle VM Server?	8
2.2.1 What Hypervisors are used in Oracle VM?	8
2.2.2 What are Domains?	9
2.2.3 What is Oracle VM Agent?	10
2.2.4 What are Server Pools?	11
2.3 What is Oracle VM Manager?	11
2.3.1 Oracle VM Manager Server and API	12
2.3.2 Oracle VM Database Repository	12
2.3.3 Oracle VM Manager Web-Based User Interface	12
2.3.4 Oracle VM Manager Command Line Interface	13
2.3.5 Oracle VM Utilities	13
2.4 What is a Oracle VM Guest?	13
2.5 Authentication and Encryption in Oracle VM	14
2.6 What Else does Oracle VM Require?	16
2.6.1 Storage	16
2.6.2 Networking	17
2.7 What are Virtual Machines?	18
2.7.1 What are Virtualization Modes?	19
2.7.2 What are the Oracle VM Guest Additions?	20
2.8 What Features Does Oracle VM Provide?	20
2.9 What Support is Available for Oracle VM?	21
2.10 Can Oracle Enterprise Manager 12c Manage Oracle VM?	21

In this chapter we deliver a complete overview of the architecture of an Oracle VM deployment. Each component type is discussed to provide an adequate understanding of the platforms involved, the technologies used and the connections between components to provide the entire Oracle VM platform.

2.1 What is the Oracle VM Architecture?

Oracle VM is a platform that provides a fully equipped environment with all the latest benefits of virtualization technology. Oracle VM enables you to deploy operating systems and application software within a supported virtualization environment. Oracle VM insulates users and administrators from the underlying virtualization technology and allows daily operations to be conducted using goal-oriented GUI interfaces. The components of Oracle VM are shown in [Figure 2.1, “Oracle VM Architecture”](#).

Figure 2.1 Oracle VM Architecture



- **Client Applications:** Various user interfaces to Oracle VM Manager are provided, either via the graphical user interface (GUI) accessible using a web-browser; the command line interface (CLI) accessible using an SSH client; custom built applications or scripts that use the Web Services API (WS-API); or external applications, such as Oracle Enterprise Manager, or legacy utility scripts that may still make use of the legacy API over TCPS on port 54322. The legacy API is due to be deprecated in the near future and applications that are using it must be updated to use the new Web Services API instead. All communications with Oracle VM Manager are secured using either a key or certificate based technology.
- **Oracle VM Manager:** Used to manage [Oracle VM Servers](#), virtual machines, and resources. It is comprised of a number of subcomponents, include a web browser-based user interface; and a command line interface (CLI) allowing you to manage your infrastructure directly from the command line either via external scripts or by running manual command sequences. Each of these interfaces runs as a separate application to the Oracle VM Manager core and interfaces with this using the Web Services API.

The Oracle VM Manager core is an Oracle [WebLogic](#) Server application running on Oracle Linux. The user interface uses the Application Development Framework (ADF) application, providing a common look and feel, in line with other Oracle web-based applications. Oracle VM Manager can be on a standalone computer, or part of a virtual machine running on an instance of Oracle VM Server. While Oracle VM Manager core and the Oracle VM Manager GUI are both WebLogic applications, they are separate applications, even though they share the same process space.

While the Oracle VM Manager GUI and CLI both use the Web Services API to interface with the Oracle VM Manager core, the GUI is able to query the Oracle VM Manager Database directly for read-only operations. This design decision allows the GUI to provide a wider range of filtering options and improves performance for particular polling operations.

Oracle VM Manager communicates with each Oracle VM Server via the Oracle VM Agent, using XML-RPC over HTTPS on port 8899. Actions on servers that are initiated within Oracle VM Manager are triggered using this method. The Oracle VM Agent on each Oracle VM Server is equally able to send notifications, statistics and event information back to Oracle VM Manager. Actions within Oracle VM Manager triggered by Oracle VM Agent are achieved using the Web Services API exposed by Oracle VM Manager and are secured using HTTPS.

While Oracle VM Manager is a critical component for configuration actions within the Oracle VM infrastructure, the virtualized environment can continue to function properly even if Oracle VM Manager experiences downtime. This includes the ability to maintain [high availability](#) and to perform [live migration](#) of virtual machines.

- **Oracle VM Manager Database:** Used by Oracle VM Manager core to store and track configuration, status changes and events. Oracle VM Manager uses a MySQL Enterprise database that is bundled in the installer and which runs on the same host where Oracle VM Manager is installed. The database is configured for the exclusive use of Oracle VM Manager and must not be used by any other applications. The database is automatically backed up on a regular schedule, and facilities are provided to perform manual backups as well.
- **Oracle VM Server:** A managed virtualization environment providing a lightweight, secure, server platform which runs virtual machines, also known as domains. At least one Oracle VM Server is required, but several are needed to take advantage of clustering.

Oracle VM Server is installed on a bare metal computer, and contains the [Oracle VM Agent](#) to manage communication with Oracle VM Manager. [dom0](#) is an abbreviation for *domain zero*, the management or [control domain](#) with privileged access to the hardware and device drivers. [DomU](#) is an unprivileged [domain](#) with no direct access to the hardware or device drivers. A user-domain (domU) is started and managed on an Oracle VM Server by dom0.

On x86-based systems, Oracle VM Server is based upon an updated version of the underlying Xen [hypervisor](#) technology, and includes Oracle VM Agent. It also includes a Linux kernel with support for a broad array of devices, file systems, and software RAID volume management. The Linux kernel is run as dom0 to manage one or more domU virtual machines, each of which could be Linux, Oracle Solaris, or Microsoft Windows™.

In contrast, Oracle VM Server for SPARC takes advantage of the hypervisor that is already included within the SPARC firmware, alongside the Oracle VM Agent for SPARC. The default Oracle Solaris operating system is usually promoted to act as the primary domain, which is equivalent to dom0 on x86 systems. Once the primary domain is in place, it can be used to create and manage further domains running different versions of the Oracle Solaris operating system.

Groups of Oracle VM Servers are usually clustered together to create server pools. This allows Oracle VM Manager to handle load balancing and failover for high-availability environments. Virtual machines run within a server pool and can be easily moved between the different servers that make up a server pool. Server pools also provide logical separation of servers and virtual machines. Server pools are required entities within the Oracle VM infrastructure, even if they consist of only one server.

Each Oracle VM Server maintains its own Berkeley Database, used to store local configuration and runtime information. This allows the Oracle VM Server to continue to function normally, even if Oracle VM Manager becomes unavailable for a period. Where Oracle VM Servers are clustered together, a separate cluster database, stored in the server pool file system, is shared between the servers. This allows the server pool to continue to provide clustering features, such as High Availability, even if Oracle VM Manager is unavailable.

- **External Shared Storage:** Provides storage for a variety of purposes and is required to enable high-availability options afforded through clustering. Storage discovery and management is achieved using the Oracle VM Manager, which then interacts with Oracle VM Servers via the storage connect framework to then interact with storage components. This process is discussed in more detail in [Section 3.4, “How do Different Storage Types Connect?”](#). Oracle VM provides support for a variety of external storage types including NFS, iSCSI and Fibre Channel.

2.2 What is Oracle VM Server?

Oracle VM Server can be installed on either x86 or SPARC hardware platforms. Since these platforms are fundamentally dissimilar, the hypervisor used on each platform is different. Oracle VM helps to abstract the hypervisor further by providing a user interface that facilitates the same logical actions across different server types. Much of this abstraction is provided by the Oracle VM Agent that is installed on each Oracle VM Server in the environment.

In this section, the way in which components run on these different hardware platforms and their roles are described in more detail. For more information on the different hardware platforms that are supported and the requirements to run an Oracle VM Server, refer to the [Oracle VM Installation and Upgrade Guide](#).

More detailed coverage of Oracle VM Server is provided in [Chapter 6, Understanding Server Pools and Oracle VM Servers](#).

2.2.1 What Hypervisors are used in Oracle VM?

The [hypervisor](#) present on each [Oracle VM Server](#) is an extremely small-footprint virtual machine manager and scheduler. It is designed so that it is the only fully privileged entity in the system. It controls only the most basic resources of the system, including CPU and memory usage, privilege checks, and hardware interrupts.

2.2.1.1 Oracle VM Server for SPARC

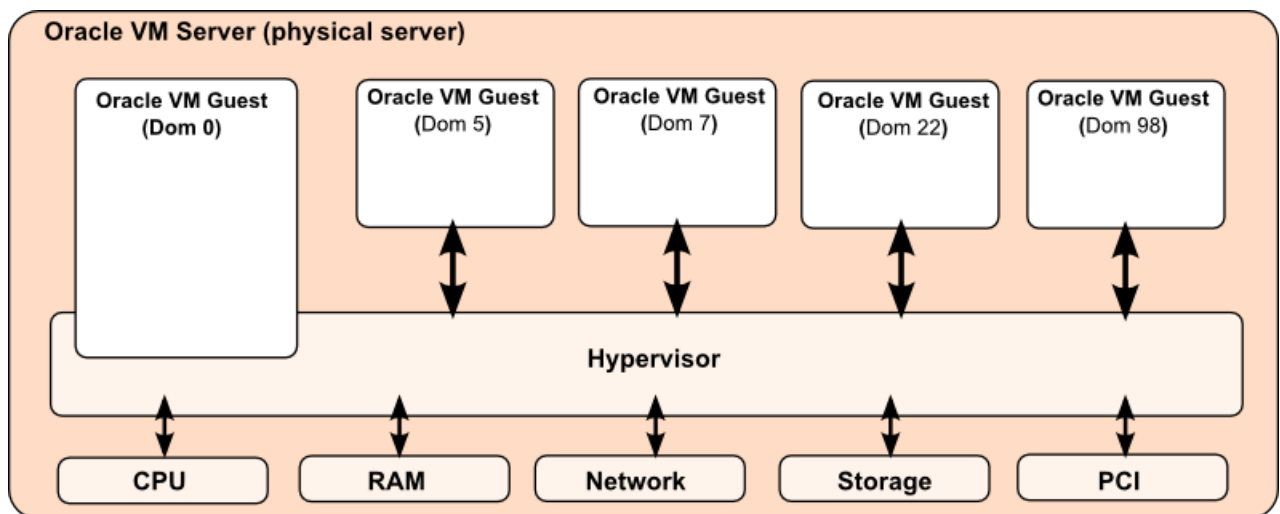
On SPARC systems, the SPARC hypervisor is built into the SPARC firmware and is generally referred to as the [Logical Domains](#) Manager (LDM). If Oracle VM Server for SPARC has not been installed, the default operating system runs on top of the hypervisor transparently. When Oracle VM Server for SPARC is installed, the default operating system becomes the primary [domain](#) and tools are provided for the primary domain to manage how resources and hardware are allocated via the hypervisor to other domains.

As with the Xen hypervisor, each virtual machine is securely executed on a single computer and runs its own [guest](#) Oracle Solaris operating system. The SPARC hypervisor provides a broader range of virtualization features than the Xen hypervisor, due to the nature and design of SPARC hardware.

2.2.1.2 Xen™

Oracle VM makes use of Xen technology, when running on x86 servers, taking advantage of the Xen hypervisor. The Xen hypervisor is a small, lightweight bare metal hypervisor for x86-compatible computers. The Xen hypervisor securely executes multiple virtual machines on one host computer. Each virtual machine runs in its own domain and has its own [guest](#) operating system with almost native performance. A primary management domain, called dom0, also runs as a guest on top of the hypervisor. See [Section 2.2.2, "What are Domains?"](#) for more information on domains. The Xen hypervisor was originally created by researchers at Cambridge University, and derived from work done on the Linux kernel.

Figure 2.2 Oracle VM Server Hypervisor and Guest Domains



2.2.2 What are Domains?

The terms "domain", "guest" and "virtual machine" are often used interchangeably, but they have subtle differences. A *domain* is a configurable set of resources, including memory, virtual CPUs, network devices and disk devices, in which virtual machines run. A *domain* or *virtual machine* is granted virtual resources and can be started, stopped and restarted independently of other domains or the host server itself. A *guest* is a virtualized operating system running within a *domain*. Multiple guests can run on the same Oracle VM Server, each within its own domain.

2.2.2.1 Management Domain (dom0)

Most of the responsibility of hardware detection in an Oracle VM Server environment is passed to the management domain, referred to as [domain zero \(or dom0\)](#). On x86-based servers, the dom0 kernel is

actually a small-footprint Linux kernel with support for a broad array of devices, file systems, and software RAID and volume management. In Oracle VM Server, the dom0 is tasked with providing a view of the system hardware available to the hypervisor. It also allows you to interact directly with the hypervisor to control access to resources as well as performing various tasks such as creating, destroying and controlling *guest* operating systems, and presenting those guests with a set of common virtual hardware. Other domains never interact with dom0 directly. Their requirements are handled by the hypervisor itself. Dom0 only provides a means to administer the hypervisor.

On SPARC-based servers, the management domain, usually referred to as the *primary service domain* or *control domain*, is created when the [logical domains](#) manager is installed. If installed on an existing server that is not already configured for logical domains, the current Operating System automatically gets promoted to primary domain status. The primary domain runs an Oracle Solaris kernel and is responsible for the creation and management of all other domains. It is also responsible for providing access to virtualized hardware resources.

On systems running Oracle VM Server for SPARC, aside from the concepts of a "management domain" and of "guest domains", similar to "user domains", there are a variety of other domain-types that can run alongside the management domain. For instance, it is possible to set up multiple ["service domains"](#) that can act as network switches and virtual disk servers. Not all of these domain types are configurable using the Oracle VM Manager. Most importantly, it is possible to add a secondary service domain to your servers that can take over disk and network I/O in the case where the primary domain must be restarted. This allows running guest virtual machines to continue to access these resources even while the primary domain is unavailable. This topic is discussed in more detail within [Configuring a Secondary Service Domain on SPARC](#) in the [Oracle VM Administrator's Guide](#).

2.2.2.2 Domains (domU)

Guest operating systems each have their own management domain called a "user domain", abbreviated to "domU". These domains are unprivileged domains with no direct access to the hardware or device drivers. Each *domU* is started alongside *dom0* running on Oracle VM Server.



Note

Under Oracle VM Server for SPARC these domains are usually referred to as "guest domains".

2.2.3 What is Oracle VM Agent?

The Oracle VM Agent is a daemon that runs within dom0 on each Oracle VM Server instance. Its primary role is to facilitate communication between Oracle VM Server and Oracle VM Manager. The daemon listens for connections from Oracle VM Manager on the TCP port 8899, and implements a messaging facility that allows Oracle VM Manager to connect to each Oracle VM Server instance and exchange information required for the efficient running of the entire Oracle VM infrastructure.

Oracle VM Agent is responsible for carrying out all of the configuration changes required on an Oracle VM Server instance, in accordance with the messages that are sent to it by Oracle VM Manager. This means that when a networking change is implemented in Oracle VM Manager, it is the Oracle VM Agent that reconfigures the server to cater for the change. Equally, if new storage is discovered and presented to a server in Oracle VM Manager, it is the Oracle VM Agent that handles the actual mount process required on the server.

Oracle VM Agent is also responsible for starting and stopping virtual machines as required by Oracle VM Manager. For this reason, the actual implementation of Oracle VM Agent differs significantly between different hardware platforms, even though actions within Oracle VM Manager are consistent across platforms.

For security reasons, the Oracle VM Agent must authenticate any system attempting to connect to it on port 8899. When an Oracle VM Server is not configured for any particular Oracle VM Manager it is in an *unowned* state. In this state, an Oracle VM Manager must take ownership of the server before it is able to communicate with the Oracle VM Agent. During the process where the Oracle VM Manager instance takes ownership of a server, the Oracle VM Manager authenticates using a password configured for the Oracle VM Agent. This password is exchanged over a connection that is secured using an SSL certificate. Once the Oracle VM Agent has authenticated Oracle VM Manager, an SSL key-certificate pair is set up to authenticate and encrypt all future communications between that Oracle VM Manager instance and the Oracle VM Agent. At this point, no other Oracle VM Manager instance or application can take control of the Oracle VM Server via the Oracle VM Agent. If you wish to allow another Oracle VM Manager instance to take ownership of the server, the original Oracle VM Manager instance must release ownership first.

There are also times when Oracle VM Agent must initiate a connection to Oracle VM Manager to signal an event or to provide statistical information. This is achieved using the Web Services API exposed by Oracle VM Manager via HTTPS on TCP port 7002.

Oracle VM Agent also maintains its own log files on the Oracle VM Server that can be used for debugging issues on a particular server instance or for auditing purposes.

Oracle VM Agent is discussed in more detail in [Chapter 6, Understanding Server Pools and Oracle VM Servers](#).

2.2.4 What are Server Pools?

A [server pool](#) is a required entity in Oracle VM, even if it contains a single Oracle VM Server. In practice, several Oracle VM Servers will form a server pool, and an Oracle VM environment may contain one or several server pools. Server pools are typically clustered, although an unclustered server pool is also possible.

Server pools have shared access to storage repositories and exchange and store vital cluster information in the server pool file system. In a server pool, a [Master server](#) is elected, which is responsible for centralized communication with the Oracle VM Manager. If necessary, any other member of the server pool can take over the Master role. Even in case of server failure, the server pool remains accessible at its virtual IP address, which all servers in the pool share. Since server pools have shared access to storage repositories, live migration of virtual machines is possible for load balancing or for scheduled maintenance, so that a virtual machine can be moved from one Oracle VM Server to another without an interruption of service.

Within a clustered server pool, virtual machines have high availability or HA. The clustering technology used within Oracle VM can take care of monitoring the status of all of the Oracle VM Servers belonging to the server pool. If a pool member disappears for whatever reason, its virtual machines can be recovered and brought back up on another Oracle VM Server because all necessary resources are available on shared storage.

A detailed explanation of the purpose of a server pool and the technology used within one is provided in [Chapter 6, Understanding Server Pools and Oracle VM Servers](#).

2.3 What is Oracle VM Manager?

Management for the Oracle VM environment is provided by [Oracle VM Manager](#), a transaction-based framework that also includes an integrated database, a web-based management user interface and a command line interface. Since Oracle VM can continue to function normally even during periods of

downtime for the Oracle VM Manager, high-availability and redundancy are not generally required for Oracle VM Manager.

Since Oracle VM Manager is a multi-user application with a variety of different interfaces that allow components to be configured and managed, it is possible that two users may attempt to perform a configuration change on the same element at the same time. In this situation, Oracle VM Manager processes the first configuration change that reaches its queue, and temporarily locks the element for any other changes. The second configuration change fails because the information that it has is stale. Usually an error is returned for the configuration change that fails. This error notifies the user that the object was locked while fetching. To work around this problem, the user that receives the error must refresh the view of the object that was being updated and resubmit the changes.

When describing the architecture of Oracle VM Manager, it is best to understand that it is composed of various inter-related components. The fundamental components consist of the Oracle VM Manager Server and the Oracle VM Manager Database. All other Oracle VM Manager components interface with the Oracle VM Manager Server via an API.

2.3.1 Oracle VM Manager Server and API

The Oracle VM Manager Server is a Java-based Oracle [WebLogic](#) 12c application that is usually installed on a dedicated system. On x86 platforms, Oracle VM Manager is installed on Oracle Linux.

Oracle VM Manager Server is sometimes referred to as the Oracle VM Manager *core*. It is used to provide a means to configure, monitor and manage the Oracle VM environment. It does this largely by communicating directly with the Oracle VM Agent on each Oracle VM Server in the environment. It also stores all configuration information and event data within the Oracle VM Manager database.

Oracle VM Manager Server provides an API that allows other components to interface with it. As of Oracle VM Release 3.3.1, this is entirely implemented as a Web Services API, providing both SOAP and REST interfaces to the Oracle VM Manager. The Oracle VM Manager Web Interface and Oracle VM Manager Command Line Interface both make use of this API to interact directly with Oracle VM Manager.

The Web Services API exposed by Oracle VM Manager, makes it possible to programmatically set up and manage an Oracle VM environment. This API is introduced in considerable detail in the [Oracle VM Web Services API Developer's Guide](#).

2.3.2 Oracle VM Database Repository

The Oracle VM Manager database is used by the Oracle VM Manager Server to store configuration information and event data. The actual database management server is a MySQL Enterprise database that is bundled with the Oracle VM Manager installer.

The MySQL database is set up using bundled database software included and licensed for use with Oracle VM Manager. The database management server is installed on the same system as Oracle VM Manager and is intended only for use by this application. Particular configuration parameters specific to the requirements of Oracle VM Manager are applied to this installation.

Details on the installation of the database are covered in the [Oracle VM Installation and Upgrade Guide](#), while database backups are explained in more detail in [Oracle VM Administrator's Guide](#).

2.3.3 Oracle VM Manager Web-Based User Interface

The Oracle VM Manager Web Interface is a web-based application built on top of Oracle WebLogic 12c and Application Development Framework (ADF). It interfaces with Oracle VM Manager Server using the Web Services API. The Oracle VM Manager Web Interface is installed on the same system as Oracle VM Manager Server and shares the same Oracle WebLogic Server space as the Oracle VM Manager core

application, but is a distinct application. By default, the graphical user interface is accessed over HTTPS on the TCP port 7002.

The Oracle VM Manager Web Interface, its configuration and usage are described in detail in the [Oracle VM Manager User's Guide](#).

2.3.4 Oracle VM Manager Command Line Interface

The Oracle VM Manager Command Line Interface is a Java-based daemon that runs on the system where Oracle VM Manager is installed, and interfaces with the Oracle VM Manager Server using the Web Services API. Access to the CLI is provided via an SSH connection available on a customizable TCP port, set to TCP 10000 by default. Any SSH client can be used to access the CLI, which provides its own shell and supports SSH key-based authentication.

The Oracle VM Manager Command Line Interface enables administrators to manage the entire Oracle VM environment from the shell, and facilitates programmatic setup and control of an environment through the use of simple shell or Expect scripts. Every action that is available via the Oracle VM Manager Web Interface is also available via the CLI.

As an independent component of the Oracle VM Manager suite of applications, the Oracle VM Manager Command Line Interface can be disabled if it is not required within an environment.

Full usage of the Oracle VM Manager Command Line Interface, including syntax and examples, is covered in the [Oracle VM Manager Command Line Interface User's Guide](#).

2.3.5 Oracle VM Utilities

The Oracle VM Utilities are a set of largely unsupported Java applications designed to allow administrators to perform actions against Oracle VM Manager from the command line that may not be easily achieved using the officially provided interfaces. In general, most of the tools provided with the Oracle VM Utilities have been superseded by the Oracle VM Manager Command Line Interface, however some tools such as Oracle VM Hostd For Metrics Messaging provide facilities that are not otherwise available.

The Oracle VM Utilities and other assorted tools that assist with the configuration and management of Oracle VM Manager are discussed in the [Oracle VM Administrator's Guide](#).

2.4 What is a Oracle VM Guest?

The terms [domain](#), [guest](#) and virtual machine are often used interchangeably, but they have subtle differences. A domain is a configurable set of resources, including memory, virtual CPUs, network devices and disk devices, in which virtual machines run. A domain is granted virtual resources and can be started, stopped and restarted independently of other domains and of the host server itself. A guest is a virtualized operating system running within a domain.

Oracle VM is designed for the purpose of running multiple guests in a virtualized environment, while providing the tools to easily manage, configure and maintain the virtual machines that guests run on. These virtual machines run on Oracle VM Servers that are grouped together into server pools. This makes it possible to easily migrate the virtual machine where a guest is running to another similar server without interrupting service. In a clustered server pool where a guest is configured to support High Availability, the guest can automatically be restarted on an alternate server when the server where a guest is running is under particular load or where a server becomes unavailable at any point.

Oracle VM guests consume resources that are made available to the domain in which they are running by the Oracle VM Server hypervisor. Additional facilities to allow communication between a guest and the Oracle VM infrastructure can be installed on the guest operating system. These facilities are usually provided in the form of the Oracle VM Guest Additions.

Oracle VM guests and the virtual machines where they run are discussed in more detail in [Chapter 7, Understanding Virtual Machines](#).

2.5 Authentication and Encryption in Oracle VM

Since there are a number of inter-related components that make up the Oracle VM environment, and each of these components must be able to communicate with each other, it is important to understand how each component authenticates itself to another, and how communications are secured. To explain this, we need to first identify the different interactions and communications channels that each component uses. These are presented in the list below:

- Internal component communications within Oracle VM Manager (for example, MySQL, WebLogic, the Oracle VM Manager application, and so on).
- Oracle VM Manager communications with Oracle VM Servers via Oracle VM Agent.
- Web-browser communications with Oracle VM Manager using the web-based user interface.
- Client application communications with Oracle VM Manager via the Web Services API (command line interface, custom scripts and applications, utilities, and so on).

In this section, we discuss the authentication and encryption of these interactions in some detail. Further information is available in the [Oracle VM Security Guide](#).

Internal component interactions

Oracle VM Manager is comprised of a number of different software components running on a single system. These include a MySQL database, an Oracle WebLogic Server that also includes the Application Development Framework (ADF) and the Oracle VM Manager application itself. Since each of these processes runs on the same system, there is no requirement to encrypt communications between them. However, authentication of each component is critical, to ensure that each process performing an interaction is legitimate.

During installation of Oracle VM Manager, a single password is set for all components. This password is used to authenticate communications between each process. The password used by Oracle WebLogic Server and the Oracle VM Manager application to access the MySQL database is stored in a secure keystore.

The same password is used for the administrative user to access the Oracle VM Manager application, via the web-based user interface, the command-line user interface or the web-services API. There are options within the product to change the password for the administrative user or to create additional user and password combinations to access the application. These are discussed in the [Oracle VM Administrator's Guide](#).

Changing the password of the administrative user does not change the password used to authenticate to the Oracle WebLogic Server or MySQL database, and the original password is always used by the Oracle VM Manager application in its interactions with these. Therefore, it is important that this password is always kept safe and that this password properly complies with your security requirements.

Oracle VM Manager and Oracle VM Server Interactions

All communications between Oracle VM Manager and any Oracle VM Server are achieved via the Oracle VM Agent running on the Oracle VM Server. To help keep traffic segregated as far as possible, to improve security and to limit the impact of particular network transactions on each other, Oracle VM introduces the concept of dedicated network channels. By default, a single network channel known as the Management Network is created to facilitate all kinds of traffic between Oracle VM Manager and Oracle VM Servers, and

between Oracle VM Servers themselves. Traffic can be further segregated according to different functions, to improve security and performance. See [Section 5.6, “How are Network Functions Separated in Oracle VM?”](#) for more information. Since these communications take place over a network, they are secured using an SSL certificate. Furthermore, the Oracle VM Manager instance must authenticate itself to the Oracle VM Server as the Oracle VM Manager instance that has ownership of that server.

For this reason, the Oracle VM Manager application creates and manages its own keystores containing various certificates and keys that are used for authentication and encryption purposes. Significantly, Oracle VM Manager generates its own Certificate Authority (CA) certificate and key that it uses to sign and validate certificates used within the infrastructure.

When Oracle VM Manager first takes ownership of an Oracle VM Server, it connects to the server via an SSL certificate that the server accepts to encrypt communications. During the ownership process, the user must provide the password that is set for the Oracle VM Agent. If the Oracle VM Server is already under the ownership of another Oracle VM Manager instance, the request to take ownership fails. Otherwise, the password is used to authenticate the Oracle VM Manager to the server. During this process, Oracle VM Manager uses its CA certificate to sign an SSL certificate and key that it provides to the server to authenticate future communications from the Oracle VM Manager instance.

From this point onwards, all authentication between Oracle VM Manager and the Oracle VM Server is achieved using SSL certificates. If a situation arises where the server is unable to validate or authenticate the certificate used by Oracle VM Manager, Oracle VM Manager must take ownership of the server again so that a new certificate and key pair can be exchanged.

The password for the Oracle VM Agent on a server can be reset when releasing ownership, using the web-based user interface, or can be reset for any server under the current ownership of Oracle VM Manager using the command line interface or Web Services API.

All certificate management for this process occurs internally within the application and does not require any user intervention at all.

Oracle VM Manager Web Interface

When a user connects to the Oracle VM Manager web-based user interface, this is achieved using an HTTPS connection secured with an SSL certificate. The user is still expected to authenticate using a plain text username and password on the Oracle VM Manager user interface login page. The SSL certificate is used to encrypt communications between the user's web-browser application and the Oracle VM Manager web application.

The SSL certificate that is used for this is automatically generated during the installation of Oracle VM Manager. The user may receive a certificate validation warning within the web-browser if the internal Oracle VM Manager CA certificate is not installed as a trusted certificate within the browser.

Oracle VM Manager provides tools to either obtain the CA certificate so that this can be installed within the web-browser, or to substitute the default SSL certificate with an alternate certificate that is signed by a trusted third-party CA. If you opt to do this, the SSL certificate hostname must match the fully qualified domain name of the Oracle VM Manager server and must be the same as the hostname that is used by your users to access the Oracle VM Manager web-based user interface.

More information on configuring this SSL certificate, or on obtaining the internal CA certificate, is provided in the [Oracle VM Administrator's Guide](#).

Oracle VM Manager Web Services API interactions

All client applications, including the Oracle VM Manager command-line interface, are also capable of connecting to Oracle VM Manager across a network. Since all applications interact with Oracle VM

Manager via a web-services API over HTTPS, these connections can be secured using the same SSL certificate as used for the Oracle VM Manager web-based user interface discussed above.

The command-line interface provides access to Oracle VM Manager via an SSH connection that can either be made using an administrator username and password, or which can be secured using standard SSH keys. The command-line interface application, then authenticates with the core Oracle VM Manager application using its own SSL certificate and key, issued and signed by the Oracle VM Manager CA certificate.

If you are developing your own application or script to interface with the Oracle VM Manager Web Services API, it is possible to achieve authentication either using the API login mechanisms that accept a username and password combination, or you can use SSL-based authentication. To do this, you generate your own certificate and key pair. You then sign the certificate and register it with the Oracle VM Manager CA, using one of the provided mechanisms. From this point on, your application can use SSL key-based authentication to obtain secure and authorized access to the API. Further information on this is provided in the [Oracle VM Web Services API Developer's Guide](#).

2.6 What Else does Oracle VM Require?

Two fundamental parts of the Oracle VM infrastructure, storage and networking, are not provided as part of the product itself, but are required in order for the product to function properly. Since the type of storage that a deployment may use can vary dependent on the deployment's requirements, Oracle VM provides support for a range of different storage types and includes mechanisms to enable different support types within the environment. Equally, network design and configuration can vary depending on requirements, hardware and cabling. Oracle VM caters to a range of complex network configurations and provides tools to configure systems within the environment to connect to each other according to a specific network design.

2.6.1 Storage

Storage in Oracle VM can be provided using any of the following technologies:

- Local disks.
- Shared Network Attached Storage - NFS.
- Shared iSCSI SANs: abstracted LUNs or raw disks accessible over existing network infrastructure.
- Fibre Channel SANs connected to one or more host bus adapters (HBAs).



Note

OCFS2 (Oracle Cluster File System) is used in the storage configurations that are not based on NFS.

While it is possible to use a local disk to create a repository for use by virtual machines on a particular Oracle VM Server, to enable [HA](#) or [live migration](#) you must make sure all Oracle VM Servers have access to the same storage resources. Specifically for live migration the Oracle VM Servers also must be part of the same [server pool](#). Also note that clustered server pools require access to a shared file system where server pool information is stored and retrieved, for example in case of failure and subsequent server role changes. The server pool file system can either be on an NFS share or on a LUN of a SAN server.

To configure access to different storage types across Oracle VM Servers within a deployment, Oracle VM makes use of Oracle [Storage Connect](#) plug-ins. [Storage Connect](#) plug-ins are packaged and distributed as RPM packages and deployed on the Oracle VM Servers. They are divided in two major categories:

storage array plug-ins for any block based storage, and file system plug-ins for any network file system based storage.

The main benefits of the plug-in approach are:

- **Flexibility:** Use and integrate with your existing storage infrastructure, choose between file-based and block-based solutions, and use local storage for testing purposes or virtual machines of minor importance. Use generic or vendor-specific plug-ins depending on your available hardware or any new hardware you select.
- **Scalability:** Add more storage providers of your preferred type and present them to your [server pools](#) as your need for storage increases. Reduce the amount of storage again if the higher storage requirements are temporary. Provision your storage with redundancy and [multipathing](#) according to your requirements and preferences.
- **Extensibility:** If you upgrade your storage, consider the added functionality of vendor-specific plug-ins. If you select hardware for which Oracle *Storage Connect* plug-ins are available, ask the manufacturer for the RPM and install the plug-in on the Oracle VM Servers with access to this storage hardware.

Oracle VM always requires a location to store environment resources that are essential to the creation and management of virtual machines. These resources include VM templates and assemblies, ISO files (virtual DVD images), VM configuration files and VM virtual disks. The location of such a group of resources is called a storage repository. You present a storage repository to the Oracle VM Servers that need access to those resources; typically all servers in a server pool.

Storage repositories can be configured on an NFS file system or on a physical disk (LUN) of a storage array. However, for storage repositories on physical disk, the servers with access to it must be members of a *clustered* server pool. For unclustered server pools only file server storage is available. For details about the use of storage repositories, see [Section 3.9, “Where are Virtual Machine Resources Located?”](#).

Clustering adds another storage element to the environment: the server pool file system. During server pool creation, the server pool file system specified for the new server pool is accessed and formatted as an OCFS2 file system, whether the file system is accessed by the Oracle VM Servers as an NFS share, a FC LUN or iSCSI LUN. This formatting creates several management areas on the file system including a region for the global disk heartbeat. The server pool file system plays a key role in clustering and therefore in the high-availability configuration of the Oracle VM environment. For details about server pool clustering, see [Section 6.8, “How do Server Pool Clusters Work?”](#).

The storage element that is most tangible and visible to all users of Oracle VM is the virtual machine disk. A VM disk is either a disk image file in a storage repository or a raw physical disk. If a physical disk (LUN) is used, it is attached directly to the VM in the same way it would be to a physical machine. For details about virtual machine operation, see [Chapter 7, Understanding Virtual Machines](#). Again, the availability of VM disks in a storage location with shared access from all Oracle VM Servers in the server pool is essential for VM high-availability.

Storage is described in more detail in [Chapter 3, Understanding Storage](#).

2.6.2 Networking

The networking infrastructure in the Oracle VM environment comprises connections between:

- Oracle VM Servers within the environment.
- Oracle VM Manager and all Oracle VM Servers in the environment.
- Oracle VM Server and the storage subsystems that it uses.

- Virtual machines running in a server pool.
- Virtual machines and external private or public networks.

These networking connections can leverage features supported by Oracle VM, such as networked file systems, clustering, redundancy and load balancing, bridging, and support for [Virtual LANs](#) (VLANs).

In Oracle VM Manager, network configuration is the mapping of available network interfaces on the Oracle VM Servers to a set of logical Ethernet networks. The physical network is the collection of physical connections in Oracle VM Manager and all Oracle VM Servers, and the switches and routers that allow information to reach its destination. A logical network in Oracle VM is built on top of these physical connections. Before you define the logical networks in Oracle VM Manager, you have to review the physical network configuration that you intend to use, such as [VLAN](#) and subnet usage. You also take into account the number of network interfaces available to your Oracle VM Servers. The minimum recommended number of ports required on a single Oracle VM Server is two, although one would suffice for test or demonstration purposes. If you have more than two ports on your Oracle VM Servers, you can design more redundancy or traffic isolation in your environment.

Oracle VM identifies different network functions: server management, live migration, cluster heartbeat, virtual machine, and storage. All network functions can either be on dedicated or shared physical networks (except for the virtual machine intra-server network). For example, a physical network can be dedicated to Virtual Machine or Storage only, or can be dedicated for all network functions. For details about network functions, see [Section 5.6, “How are Network Functions Separated in Oracle VM?”](#).

After reviewing your physical network environment and deciding on the logical distribution and grouping of these physical objects, you create the logical constructs in Oracle VM Manager to implement your network design. These logical constructs include network bonds, VLAN Interfaces, networks and bridges. If your network design includes interface [bonding](#), or aggregations of two ports, you create these network bonds first. These bonds are often used in conjunction with VLANs, when traffic from several VLANs is allowed to use the same bond. If your network environment comprises VLANs, your next step is to create VLAN Interfaces, determining which port or bond on each Oracle VM Server will accept traffic from which VLANs.

After careful evaluation of the available network building blocks and required network functions, you create the necessary logical networks by choosing one of these types:

- Network with bonds and ports.
- Network with VLANs only.
- Hybrid network connecting bonds and ports, as well as VLAN interfaces.
- Logical network on a single server (intra-server VM network).

For more detailed information about networking within Oracle VM, see [Chapter 5, Understanding Networks](#).

2.7 What are Virtual Machines?

The entire Oracle VM environment is designed for the purpose of running and deploying virtual machines. They are, therefore, a significant component within the Oracle VM architecture. Virtual machines offer different levels of support for virtualization at the level of the operating system, often affecting performance and the availability of particular functionalities afforded by system drivers. The different types of virtualization support are referred to as virtualization modes.

Virtual machine images and configuration data are usually stored within storage repositories on shared storage that is connected to all of the Oracle VM Servers in a server pool. If the server pool has been

configured to take advantage of load balancing, the virtual machine can be started on any Oracle VM Server in the pool, because all of the files required to run it are available to all of the servers. Equally, in the case of failover, if a server in a clustered server pool should become unavailable, the virtual machine can be automatically restarted on another Oracle VM Server in the pool.

Virtual machines can be created from different types of resources: either from a template or [assembly](#) containing preconfigured virtual machines, or from scratch using an ISO file (image) of an installation DVD. Booting a virtual machine using PXE, or network boot for a paravirtualized guest, is also possible.

The creation of a virtual machine from template is based on cloning: the template is imported as an archive, unpacked and stored as a virtual machine configuration file with images of its disks, which are cloned to create a new instance in the form of a virtual machine. In the same way, an existing virtual machine can be cloned to create a new virtual machine, or cloned to create a new template. Cloning is discussed in further detail in [Section 7.6, “How does Cloning Work?”](#).

Assemblies are often described as a template of a group of virtual machines, or a collection of multiple virtual machine templates. In actual fact, an assembly is a more recent format used by Oracle to publish Oracle VM templates. This means that an assembly is equally capable of containing a single virtual machine. In the Oracle VM Manager Web Interface, templates appear on a different tab to virtual machines of the storage repository, but their configuration files and disk images are stored in the same location as those of other virtual machines and templates. Assemblies are stored separately, but when a virtual machine or template is created, the resulting configuration files and disk images are stored in the same location.

Creating a virtual machine from a virtual DVD (image file, ISO) is different depending on the virtualization mode. For [hardware virtualized](#) guests the standard installer provided in the image is used, so no allowance for PV drivers is given. When creating a hardware virtualized guest, you can assign an ISO file located on a storage repository so that the new virtual machine immediately boots from the virtual DVD. Conversely, a PVM guest cannot simply boot from DVD since there is no virtual DVD device loaded at this stage, and uses an ISO file mounted remotely, accessing it via NFS, HTTP or FTP. The distinction here is that a hardware virtualized machine has an emulated BIOS, while a PVM guest does not.

Virtual machine resources are stored in storage repositories. The contents and structure of storage repositories is described in detail in [Chapter 4, Understanding Repositories](#).

When a virtual machine is running, it can be accessed through a console, which allows it to be used as a regular operating system. For more information on the console, see [Section 7.10, “Accessing the Virtual Machine Console”](#).

The next few sections provide an overview of many of the concepts specific to virtual machines. A more detailed look at virtual machines is covered later in [Chapter 7, Understanding Virtual Machines](#).

2.7.1 What are Virtualization Modes?

A virtual machine can be defined as a virtualized operating system with its associated software and applications. It runs in one of three virtualization modes, also named [domain](#) types:

- **Hardware virtualized (HVM):** An unmodified guest operating system executes in complete isolation. Instructions are trapped and emulated at the hardware level (Intel® VT-x/VT-i and AMD-V™). This means that a guest can run without specific modifications to allow for virtualization.
- **Paravirtualized (PVM):** A software interface similar but not identical to the underlying hardware is presented to the guest operating system. Paravirtualization provides hooks for guest instructions so that hardware related tasks such as access to network resources, blocks and underlying files can be handled by the management domain instead of the virtual machine, often significantly improving performance,

particularly for 32-bit environments. Paravirtualization requires that the guest kernel has support for and loads the PVM drivers to be made aware of the virtual environment.

- **Hardware virtualized with paravirtualized drivers (PVHVM):** Similar to [HVM](#) but with additional paravirtualized drivers that are capable of handling I/O related processes directly within the management domain to increase VM performance. This provides the advantages of paravirtualization to an otherwise hardware virtualized guest. This domain type is typically used to run Microsoft Windows™ guests with a limited performance penalty.

2.7.2 What are the Oracle VM Guest Additions?

While not necessary to have a fully functional Oracle VM deployment, Oracle VM Guest Additions can be installed into your virtual machines to allow you to communicate directly with a virtual machine from Oracle VM Manager. The Oracle VM Guest Additions package installs a daemon that runs within the virtual machine and which listens for messages from Oracle VM Manager. By installing this package, Oracle VM Manager is able to provide information about the actual virtual machine's configuration, such as its configured IP address. Furthermore, it is possible to configure virtual machines as they are started by sending configuration information from Oracle VM Manager directly to the virtual machine, triggering configuration scripts that are able to use the data. This facility enables you to control behavior within a virtual machine as different events outside of the virtual machine trigger these scripts.

The Oracle VM Guest Additions, and the way in which configuration scripts can be used in conjunction with Oracle VM's messaging subsystem, are discussed in detail in the [Oracle VM Administrator's Guide](#).

2.8 What Features Does Oracle VM Provide?

This section gives an overview of the Oracle VM Manager features used to manage Oracle VM Servers, virtual machines, storage repositories, networks, and resources. Oracle VM Manager provides the following main capabilities:

- Manages the physical Oracle VM Servers and can, for example, reboot or rediscover the physical hardware.
- Creates and configures server pools.
- Creates and manages Oracle VM Server logical networks, for example, NIC port [bonding](#), and configuring [VLAN](#) networks.
- Manages storage devices such as local disks, SAN storage and Network File Servers.
- Creates and manages storage repositories.
- Manages resources, including ISO files, virtual machine templates, virtual machine images, and virtual machine assemblies.
- Manages the virtual machines. This includes creating virtual machines from either installation media or from templates, starting, logging in, shutting down, and deleting virtual machines.
- Imports, clones and [migrates](#) virtual machines.
- Performs load balancing of virtual machines in server pools.
- Manages [jobs](#) in the Oracle VM environment.
- Manages policies such as High Availability, [Distributed Resource Scheduling](#), and [Distributed Power Management](#).

An Web Services API is available for programmatic access to Oracle VM Manager, and therefore all the objects Oracle VM Manager has ownership of. See the [Oracle VM Web Services API Developer's Guide](#) for a description of an API for programmatic access to Oracle VM Manager functions.

A command line interface is also available to access Oracle VM Manager, see the [Oracle VM Manager Command Line Interface User's Guide](#) for more information.

2.9 What Support is Available for Oracle VM?

Oracle has a unique position in the virtualization market as an Enterprise application, operating system and hardware vendor that delivers technologies across the stack. Owning the entire stack has various advantages:

- Integration and centralized management of all components.
- The ability to pre-package and distribute Oracle technologies via Oracle VM templates.
- Integrated enterprise support across the entire technology stack, from application to hardware.

Oracle VM Support is an add-on component of Oracle's enterprise support package that offers an end-to-end single vendor support solution from the application to the disk. A single support call covers the entire Oracle stack which expedites problem resolution. Using Oracle Support allows an Oracle Support Service Request (SR) to transition between Support teams with issues that require cross stack collaboration. For example, if you open a service request for an application issue, and the root cause is at the virtualization layer, then the service request moves between the application and virtualization Support teams. For more information on Oracle Support, see:

<http://www.oracle.com/us/support/index.html>

2.10 Can Oracle Enterprise Manager 12c Manage Oracle VM?

You can manage your Oracle VM environment using Oracle Enterprise Manager 12c. Oracle VM Manager plugs into Oracle Enterprise Manager 12c and exposes all the functionality of Oracle VM to Oracle Enterprise Manager, so you can manage the entire Oracle VM environment from within Oracle Enterprise Manager. Oracle Enterprise Manager offers extended functionality beyond that of Oracle VM, for example, monitoring and role based access control. Oracle Enterprise Manager is not provided as part of the Oracle VM product. For more information about the Oracle VM versions supported by Oracle Enterprise Manager, see the Oracle Enterprise Manager documentation at:

<http://www.oracle.com/technetwork/oem/grid-control/documentation/oem-091904.html>

Chapter 3 Understanding Storage

Table of Contents

3.1 What is Storage Used for in Oracle VM?	23
3.2 What Types of Storage Can be Used?	24
3.2.1 Local Storage	25
3.2.2 Shared Network Attached Storage (NFS)	26
3.2.3 iSCSI Storage Attached Network	26
3.2.4 Fibre Channel Storage Attached Network	27
3.3 How is a Storage Created?	28
3.4 How do Different Storage Types Connect?	28
3.5 What is Multipathing?	30
3.6 What are Uniform and Non-uniform Exports?	30
3.7 What are Access Groups?	31
3.8 How is Storage Used for Server Pool Clustering?	31
3.9 Where are Virtual Machine Resources Located?	32
3.10 Are there Guidelines for Configuring Storage?	32

Since storage is such a fundamental component within an Oracle VM, and many of the concepts surrounding it are not discussed elsewhere in the Oracle VM documentation, this chapter is dedicated to providing a detailed overview of storage infrastructure and how it is used within Oracle VM. Many of the concepts presented here are designed to assist you during the configuration of your storage using any of the interfaces provided by Oracle VM Manager, including the web-based user interface and the Oracle VM Manager Command Line Interface. A very brief overview of storage components is provided at [Section 2.6.1, “Storage”](#).

To get a more thorough overview of how storage is used within Oracle VM and the supported storage types, it is recommended that you refer to [Section 3.1, “What is Storage Used for in Oracle VM?”](#) and [Section 3.2, “What Types of Storage Can be Used?”](#).

The particular way in which Oracle VM approaches storage is through plug-ins: Oracle has made storage configuration and integration as flexible and modular as possible by creating a [Storage Connect](#) plug-in for each different category and type of storage. These plug-ins are discussed in further detail in [Section 3.4, “How do Different Storage Types Connect?”](#).

For information on managing and working with the contents of a storage repository, such as virtual machine templates, ISO files, assemblies, and so on, see [Section 3.9, “Where are Virtual Machine Resources Located?”](#).

Once you have a good understanding of the Oracle VM storage concepts, please take the time to read [Section 3.10, “Are there Guidelines for Configuring Storage?”](#) before you start planning your storage infrastructure. This can help you to ensure that your storage configuration is well planned and that you avoid common pitfalls.

3.1 What is Storage Used for in Oracle VM?

Storage in Oracle VM is used for a variety of purposes. These can be grouped into two main categories:

- Oracle VM Server clustering resources.
- Oracle VM virtual machine resources (such as templates, ISOs, assemblies, disk images and configuration data).

In the first instance, it is absolutely essential that any Oracle VM Server clustering resources that are used within a deployment are located on shared storage. Shared storage is any storage mechanism that can be made available to all of the Oracle VM Server instances within a server pool. This is a requirement because data relating to the health and availability of components within a pool needs to be shared across all members within a pool. While it is possible to avoid taking advantage of the clustering facilities provided by Oracle VM, thereby avoiding any requirement for shared storage, this is an uncommon configuration and due to the range of support that Oracle VM provides for different shared storage technologies it is highly recommended that some form of shared storage is used for at least this purpose.

In general, it is highly recommended that Oracle VM virtual machine resources also make use of shared storage and this is, in fact, a requirement if clustering is enabled for a server pool. By using shared storage for these resources, it is possible to migrate or restart virtual machines between different servers within a pool. It is possible to use local storage, such as the physical disks attached to any particular Oracle VM Server, within a deployment, for the purpose of provisioning virtual machine resources but doing so severely limits configuration.

The actual provisioning of virtual machine resources is usually achieved by creating a storage *repository*. In Oracle VM terms, a repository is a file system that is logically structured in a particular way so that different resources can be located easily and made available for the purpose of creating, cloning or running virtual machines. If the repository is located on an NFS server share, the directory structure required for the repository is simply created on the existing file system exposed through the share. On shared physical disks exposed through SAN hardware via iSCSI or Fibre Channel, the disk or LUN is formatted with an [OCFS2](#) file system, and the directory structure is created on top of this. OCFS2 is used in this situation because the file system is accessed simultaneously by multiple Oracle VM Servers, something which most file systems cannot handle, but which OCFS2 is explicitly designed to handle. Since OCFS2 is not available on SPARC, iSCSI and Fibre Channel disks cannot be used to host a repository accessible to SPARC-based Oracle VM Servers. For SPARC-based servers you should always use an NFS server share to host a repository.

A repository can also be created on a local storage disk, with the exception that the data within such a repository would only be available to virtual machines running on the Oracle VM Server to which the disk is attached. For SPARC-based servers, ZFS and SAS disks are treated as local storage disks.

3.2 What Types of Storage Can be Used?

Oracle VM is designed to allow you to use a wide variety of storage types so you can adapt your configuration to your needs. Whether you have a limited hardware setup or a full rack of servers, whether you perform an installation for testing and temporary internal use or design a production environment that requires [high availability](#) in every area, Oracle VM offers support for a suitable storage solution.

Making use of both generic and vendor-specific Storage Connect plug-ins, Oracle VM allows you to use the following types of storage:

- [Local Storage](#)
- [Shared Network Attached Storage \(NFS\)](#)
- [iSCSI Storage Attached Network](#)
- [Fibre Channel Storage Attached Network](#)



Important

Oracle VM does not support mixing protocols to access the same storage device. Therefore, if you are using Fibre Channel to access a storage device, you cannot then access the same device using the iSCSI protocol.

Usually the file system of a disk that is used as storage for an Oracle VM Server is formatted using OCFS2 (Oracle Cluster File System). The only exception to this is NFS, since by nature this file system is already designed to be accessed by multiple systems at once. The use of OCFS2 ensures that the file system can properly handle simultaneous access by multiple systems at the same time. OCFS2 is developed by Oracle and is integrated within the mainstream Linux kernel, ensuring excellent performance and full integration within an Oracle VM deployment. OCFS2 is also a fundamental component used to implement [server pool](#) clustering. This is discussed in more detail in [Section 3.8, “How is Storage Used for Server Pool Clustering?”](#). Since OCFS2 is not available on SPARC, only NFS can be used to support SPARC-based server pool clustering or shared repository hosting.

To enable [HA](#) or live migration, you must make sure all Oracle VM Servers have access to the same storage resources. Specifically for live migration and virtual machine HA restarts, the Oracle VM Servers also must be part of the same server pool. Also note that clustered server pools require access to a shared file system where server pool information is stored and retrieved, for example in case of failure and subsequent server role changes. In x86 environments, the server pool file system can either be on an NFS share or on a LUN of a SAN server. In SPARC environments, the server pool file system can only reside on an NFS share.

3.2.1 Local Storage

[Local storage](#) consists of hard disks installed locally in your [Oracle VM Server](#). For SPARC, this includes SAS disks and ZFS volumes. In a default installation, Oracle VM Server detects any unused space on the installation disk and repartitions the disk to make this space available for local storage. As long as no partition and data are present, the device is detected as a raw disk. The choice is yours to use the local disks, either to provision logical storage volumes as disks for virtual machines, or to install a storage repository. See [Creating ZFS Volumes on SPARC](#) in the [Oracle VM Administrator's Guide](#) for more information on setting up ZFS volumes for Oracle VM Server for SPARC. Also note that SAS disk support is limited to the SAS controller using the `mpt_sas` driver. This means that there is internal SAS disk support on T3 and later, but internal SAS disks on T2 and T2+ are unsupported.

If you place a storage repository on a local disk in an x86 environment, an OCFS2 file system is created. If you intend to create a storage repository on a local disk, the disk must not contain any data or meta-data. If it does, it is necessary to clean the disk manually before attempting to create a storage repository on it. This can be done using the `dd` command, for example:

```
# dd if=/dev/zero of=/dev/disk bs=1M
```

Where *disk* is the device name of the disk where you want to create the repository.



Warning

This action is destructive and data on the device where you perform this action may be rendered irretrievable.

In SPARC environments, a storage repository is created using the ZFS file system. If you use a local physical disk for your storage repository, a ZFS pool and file system are automatically created when you create the repository. If you use a ZFS volume to host the repository, the volume is replaced by a ZFS file system and the repository is created within the file system.

Local storage can never be used for a [server pool](#) file system.

Local storage is fairly easy to set up because no special hardware for the disk subsystem is required. Since the virtualization overhead in this set up is limited, and disk access is internal within one physical server, local storage offers reasonably high performance.

While local storage has the advantage that it is fast and easy to use, there is no possibility to share this type of storage and it cannot be used in a high-availability configuration. Consequently, if you need to

store important data and run important virtual machines that require virtually uninterrupted uptime with a high degree of data loss prevention, it is recommended not to use local storage, but to invest in attached storage with proper redundancy instead.

In addition, local storage lacks flexibility in a clustered set up with multiple Oracle VM Servers in a server pool. Resources and virtual disks that live on local storage cannot be shared with other Oracle VM Servers, even if they are within the same server pool. This means that you cannot use a template, ISO, or virtual disk stored on a local repository on another Oracle VM Server, and that [live migration](#) of virtual machines with locally stored disks is impossible. We strongly advise you to take this into account when designing your Oracle VM environment.

The configuration where local storage is most useful is where you have an unclustered server pool that contains only one Oracle VM Server. By creating a storage repository (see [Section 4.3, “How is a Repository Created?”](#)) on local storage you can set up an Oracle VM virtualized environment quickly and easily on a single Oracle VM Server: all your resources, virtual machines and their disks are stored locally. Since there is only one Oracle VM Server, access to resources is guaranteed.

**Note**

In Oracle VM, sharing a local physical disk between virtual machines is possible but not recommended.

3.2.2 Shared Network Attached Storage (NFS)

Network Attached Storage – typically NFS – is a commonly used file-based storage system that is very suitable for the installation of Oracle VM storage repositories. Storage repositories contain various categories of resources such as templates, [virtual disk](#) images, ISO files and virtual machine configuration files, which are all stored as files in the directory structure on the remotely located, attached file system. Since most of these resources are rarely written to but are read frequently, NFS is ideal for storing these types of resources.

With Oracle VM you [discover](#) NFS storage via the server IP or host name and typically present storage to all the servers in a server pool to allow them to share the same resources. This, along with clustering, helps to enable high availability of your environment: virtual machines can be easily [migrated](#) between host servers for the purpose of load balancing or protecting important virtual machines from going off-line due to hardware failure.

NFS storage is exposed to Oracle VM Servers in the form of shares on the NFS server which are mounted onto the Oracle VM Server's file system. Since mounting an NFS share can be done on any server in the network segment to which NFS is exposed, it is possible not only to share NFS storage between servers of the same pool but also across different server pools.

In terms of performance, NFS is slower for [virtual disk](#) I/O compared to a logical volume or a raw disk. This is due mostly to its file-based nature. For better disk performance you should consider using block-based storage, which is supported in Oracle VM in the form of iSCSI or Fibre Channel SANs.

NFS can also be used to store server pool file systems for clustered server pools. This is the only shared storage facility that is supported for this purpose for SPARC-based server pools. In x86 environments, alternate shared storage such as iSCSI or Fibre Channel is generally preferred for this purpose for performance reasons.

Oracle VM Manager does not support accessing the same NFS server via multiple different IP addresses or hostnames. If you attempt to do this, Oracle VM Manager treats each IP address or hostname as a separate NFS server.

3.2.3 iSCSI Storage Attached Network

With Internet SCSI, or iSCSI, you can connect storage entities to client machines, making the disks behave as if they are locally attached disks. iSCSI enables this connectivity by transferring SCSI commands over existing IP networks between what is called an [initiator](#) (the client) and a target (the storage provider).

To establish a link with iSCSI SANs, all Oracle VM Servers can use configured network interfaces as iSCSI initiators. It is the user's responsibility to:

- Configure the disk volumes (iSCSI LUNs) offered by the storage servers.
- Discover the iSCSI storage through Oracle VM Manager. When discovered, unmanaged iSCSI and Fibre Channel storage is not allocated a name in Oracle VM Manager. Use the ID allocated to the storage to reference unmanaged storage devices in the Oracle VM Manager Web Interface or the Oracle VM Manager Command Line Interface.
- Set up access groups, which are groups of iSCSI initiators, through Oracle VM Manager, in order to determine which LUNs are available to which Oracle VM Servers.

Performance-wise, an iSCSI SAN is better than file-based storage like NFS and it is often comparable to direct local disk access. Because iSCSI storage is attached from a remote server, it is perfectly suited for an x86-based clustered [server pool](#) configuration where [high availability](#) of storage and the possibility to live migrate virtual machines are important factors.

In SPARC-based environments, iSCSI LUNs are treated as local disks. Since OCFS2 is not supported under SPARC, these disks cannot be used to host a server pool cluster file system and they cannot be used to host a repository. Typically, if iSCSI is used in a SPARC environment, the LUNs are made available directly to virtual machines for storage.

Provisioning of iSCSI storage can be done through open source target creation software at no additional cost, with dedicated high-end hardware, or with anything in between. The generic iSCSI Oracle VM Storage Connect plug-in allows Oracle VM to use virtually all iSCSI storage providers. In addition, vendor-specific Oracle VM Storage Connect plug-ins exist for certain types of dedicated iSCSI storage hardware, allowing Oracle VM Manager to access additional interactive functionality otherwise only available through the management software of the storage provider. Examples are creating and deleting LUNs, extending existing LUNs and so on. Check with your storage hardware supplier if a Oracle VM Storage Connect plug-in is available. For installation and usage instructions, consult your supplier's plug-in documentation.

Oracle VM is designed to take advantage of Oracle VM Storage Connect plug-ins to perform LUN management tasks. On a storage array do not unmap a LUN and then remap it to a different LUN ID without rebooting the affected Oracle VM Servers. In general, remapping LUNs is risky because it can cause data corruption since the targets have been switched outside of the affected Oracle VM Servers. If you attempt to remap a LUN to a different ID, affected Oracle VM Servers are no longer able to access the disk until they are rebooted and the following error may appear in [/var/log/messages](#):

```
Warning! Received an indication that the LUN assignments on this target have changed.  
The Linux SCSI layer does not automatically remap LUN assignments.
```

3.2.4 Fibre Channel Storage Attached Network

Functionally, a fibre channel SAN is hardly different from an iSCSI SAN. Fibre channel is actually older technology and uses dedicated hardware instead: special controllers on the SAN hardware, host bus adapters or HBAs on the client machines, and special fibre channel cables and switches to interconnect the components.

Like iSCSI, a Fibre Channel SAN transfers SCSI commands between initiators and targets establishing a connectivity that is almost identical to direct disk access. However, whereas iSCSI uses TCP/IP, a Fibre

Channel SAN uses Fibre Channel Protocol (FCP). The same concepts from the iSCSI SAN, as described in [Section 3.2.3, “iSCSI Storage Attached Network”](#), apply equally to the Fibre Channel SAN. Again, generic and vendor-specific Storage Connect plug-ins exist. Your storage hardware supplier can provide proper documentation with the Oracle VM Storage Connect plug-in.


3.3 How is a Storage Created?

To add storage to the Oracle VM environment, you must first discover it. The discovery process connects to the storage and discovers the file systems or disks available, and configures them for use with Oracle VM.

Discovering a File Server


In Oracle VM, the term *file server* is used to indicate file-based storage made available to the environment from another physical server, as opposed to local storage. Describing the technology used to expose file systems, NFS shares and so on, is beyond the scope of this guide. The procedure linked to below explains how you can bring the exposed file-based storage into Oracle VM, prepare it for the installation of a storage repository, and configure the file server and discovered file systems.

Before you discover a file server, make sure that your storage server exposes a writable file system to the storage network of your server pool.

To discover a file server, use **Discover File Servers**  on the **Storage** tab to display the **Discover a File Server** dialog box, which is used to discover the external storage mount points. To discover a file server, see [Discover File Server](#) in the [Oracle VM Manager User's Guide](#).

Discovering a SAN Server

Make sure that your storage array server exposes raw disks (Fibre Channel SAN volumes, iSCSI targets and LUNs) to the storage network of your server pool.

To discover a storage array (SAN) server, use **Discover SAN Server**  on the **Storage** tab to display the **Discover SAN Server** dialog box, which is used to discover the external storage elements. To discover a storage array, see [Discover SAN Server](#) in the [Oracle VM Manager User's Guide](#).

After discovering a file server and its file systems, or a storage array, it is ready to be used either for storage repositories or as server pool file systems. A server pool file system is selected during the creation of the server pool (see [Section 6.7, “How are Server Pools Created?”](#)); to create storage repositories on your file systems, see [Section 4.3, “How is a Repository Created?”](#).

3.4 How do Different Storage Types Connect?

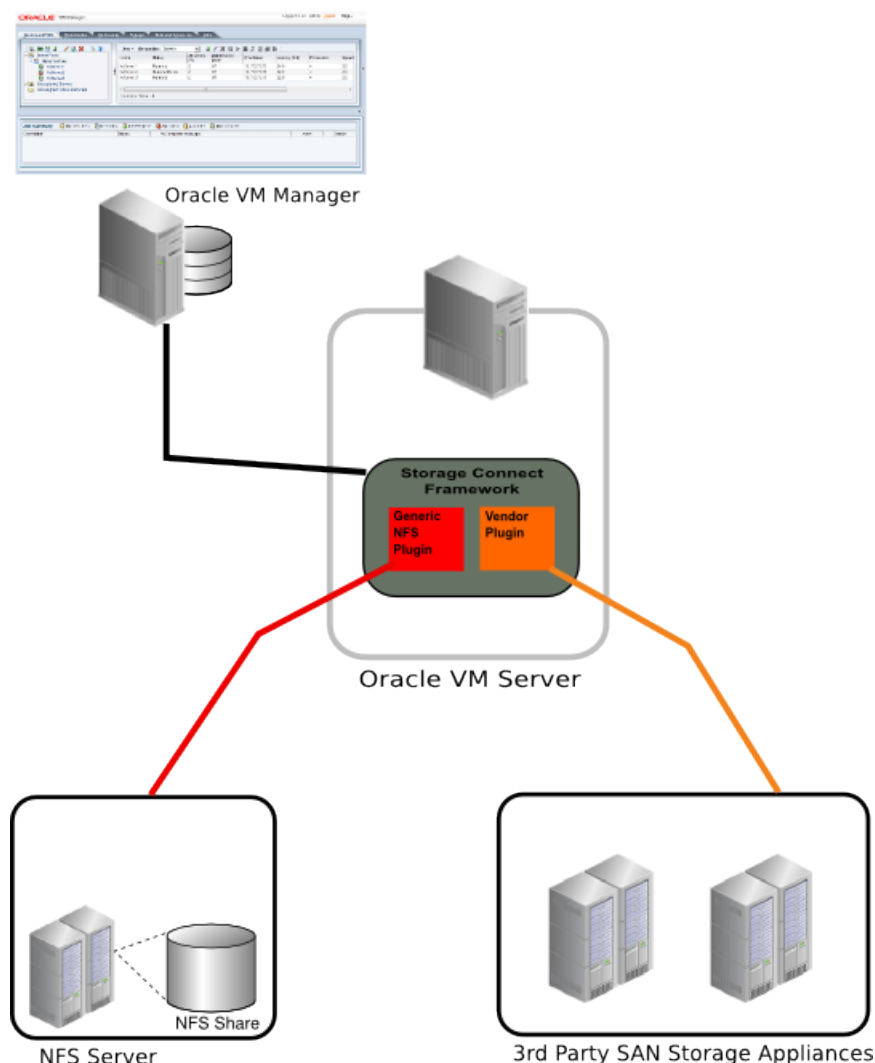
[Oracle VM Manager](#) communicates with all storage through a set of plug-ins, which are part of the Storage Connect framework. These plug-ins are not actually run from the Oracle VM Manager but rather live on some or all of the [Oracle VM Servers](#). You can see these plug-in files in the local file system of an Oracle VM Server in the `/opt/storage-connect/` directory. In the Oracle VM Manager Web Interface you select an available plug-in when creating and configuring storage elements for use in your environment.

The Oracle VM Storage Connect framework provides a storage discovery and provisioning API that enables the provisioning and management of storage platforms directly from within Oracle VM Manager. The abstraction provided by the Storage Connect framework allows administrators to perform provisioning operations without the need to know the specific behavior of any storage platforms that are being used within the Oracle VM infrastructure. The Storage Connect API allows third party storage vendors to

develop their own Oracle VM Storage Connect plug-ins that enable their storage appliances to interface with the Storage Connect framework.

Storage elements are logically divided in File Servers and SAN Servers. This distinction refers to the difference between file-based storage and block-based storage, or raw disks. Both types of storage are handled using Oracle VM Storage Connect plug-ins. The Oracle VM Storage Connect framework is able to present the capabilities of these two different types of storage facilities in a similar way, and a user would need to have only basic knowledge of storage to be able to manage either of them.

Figure 3.1 Storage tab in the Oracle VM Manager Web Interface



Oracle VM Storage Connect plug-ins are split up according to the functionality they offer: there are generic plug-ins and non-generic plug-ins, also referred to as vendor-specific plug-ins. Generic plug-ins offer a limited set of standard storage operations on virtually all storage hardware, such as discovering and operating on existing storage resources. We categorize these operations as 'passive' in the sense that they do not interact with the storage management but simply detect the available storage architecture and allow it to be used in the Oracle VM environment.

Vendor-specific plug-ins include a much larger set of operations, which also includes direct, active interventions on the storage hardware: [clone](#), create LUNs, resize, and so on. To execute generic storage plug-in operations, only an access host or fibre channel connectivity is required (for iSCSI: typically a

host name or IP address with a port number). The non-generic plug-in operations require an additional admin host, with optional administrative user name and password, granting Oracle VM Servers, with the appropriate plug-in installed, direct access to the configuration of the storage hardware.

The following generic plug-ins are included with Oracle VM Server and work with Oracle VM Manager:

- Oracle Generic NFS Plug-in.
- Oracle Generic SCSI Plug-in.

To install vendor-specific plug-ins, see [Installing Oracle VM Storage Connect plug-ins](#) in the [Oracle VM Administrator's Guide](#).

3.5 What is Multipathing?

[Multipathing](#) is the technique of creating more than one physical path between the server and its storage devices. It results in better fault tolerance and performance enhancement. Oracle VM supports multipath I/O out of the box. [Oracle VM Servers](#) are installed with multipathing enabled because it is a requirement for SAN disks to be [discovered](#) by [Oracle VM Manager](#).



Note

Any system disks (disks that contain / or /boot) are blacklisted by Oracle VM Manager and are not available for use in an Oracle VM environment.

Multipath configuration information is stored in `/etc/multipath.conf` on each Oracle VM Server and contains specific settings for Oracle VM along with an extensive set of configuration details for commonly used SAN hardware. In most cases the user should not need to modify this file and is advised not to. Examining the contents of the file may be useful to better understand how it works in Oracle VM and what may need to be configured if your SAN is not using multipathing and your LUNs are not appearing.

More detail on how multipathing is configured is covered in [Enabling Multipath I/O Support](#) in the [Oracle VM Administrator's Guide](#).

3.6 What are Uniform and Non-uniform Exports?

Since Oracle VM Manager never connects directly to any storage, but relies on Oracle VM Servers to handle the administration and the discovery of file systems exported by the NFS file server, Oracle VM Manager requires that you designate an Oracle VM Server as an administrative server and an Oracle VM Server as a refresh server. These servers may be the same Oracle VM Server. If all servers within the deployment have the same access to file systems exposed by the filer, Oracle VM Manager understands these exports to be *uniform*.

In the case of uniform exports, a single Oracle VM Server may be designated for the purpose of performing file system refreshes, although another may be added for redundancy.

In some cases, an NFS file server may be configured to export mount points with various access restrictions. In the case where some Oracle VM Servers in a deployment are not able to access exports that are available to others, Oracle VM Manager understands these exports to be *non-uniform*. If the exports exposed by the NFS file server are non-uniform, a single Oracle VM Server may not have access to all of the exports available on the file server. In this case, multiple Oracle VM Servers may need to be designated for the task of refreshing NFS storage and the file systems that are available on the file server.

Non-uniform exports are not intended for servers within the same server pool. All of the servers that belong to a server pool that requires access to file systems on an NFS file server with non-uniform exports must

be added to an access group. See [Access Groups Perspective](#) in the *Oracle VM Manager User's Guide* for more information on this.

**Note**

If you need to refresh file systems for an NFS file server that has non-uniform exports, it is important to understand that for the operation to complete, each of the refresh servers must perform its own file system refresh, since each server has access to a different set of exports. If any of the refresh servers are unavailable, the job cannot complete, since not all exports can be refreshed. If you are faced with this situation, you may need to edit the NFS file server to remove a refresh server that is out of commission, and add an alternate that has access to the same set of file server exports.

3.7 What are Access Groups?

Access groups provide a means to arrange and restrict access to storage to a limited set of servers. In Oracle VM there are two types of access groups that ultimately provide very similar functionality:

- **File System Access Groups** define which Oracle VM Servers have access to a particular file server export or file system. This is commonly used in environments where a file server has non-uniform exports, since it is important to be able to determine which Oracle VM Servers are able to refresh the file system. This is discussed in more detail in [Section 3.6, “What are Uniform and Non-uniform Exports?”](#).
- **SAN Access Groups** define which storage initiators can be used to access physical disks exposed using some form of SAN storage, such as iSCSI or Fibre Channel. SAN access group functionality is defined by the Oracle VM Storage Connect plug-in that you are using. The generic iSCSI plug-in creates a single access group to which all available storage initiators are added at the time of discovery. Other plug-ins provide more granular control allowing you to create multiple access groups that limit access to particular physical disks according to your requirements.

It is possible that in some environments you may have mixed SAN storage, where both iSCSI and Fibre Channel types are available. This can cause some confusion when configuring access groups, since the storage initiators for an access group must match the type of storage that the access group is being configured for. Oracle VM Manager is unable to filter the storage initiators available on each Oracle VM Server according to type, since access group contents are defined by the Oracle VM Storage Connect plug-in manufacturer. Therefore, it is important to pay attention to the storage initiators that you add to an access group if you are in an environment that supports more than one storage type. In most cases, creating an access group that contains storage initiators of a different type to that of the storage array type causes the Oracle VM Storage Connect plug-in to generate an error.

3.8 How is Storage Used for Server Pool Clustering?

To support server pool clustering, shared storage is required. On x86 hardware, this can be provided either in the form of an NFS share or a LUN on a SAN server. In either case, a disk image is created and formatted using [OCFS2](#) (Oracle Cluster File System). On SPARC hardware, only NFS is supported to host the server pool file system.

For x86 environments, a clustered server pool always uses an OCFS2 file system to store the cluster configuration and to take advantage of OCFS2's heartbeat facility.

For SPARC environments, a clustered server pool is always hosted on an NFS share. Clustering for SPARC makes use of the same mechanisms that are available in OCFS2, but does not use the actual OCFS2 file system. Instead, clustering data is stored directly in the NFS share.

There are two types of heartbeats used within the cluster to ensure [high availability](#):

- The disk heartbeat: all Oracle VM Servers in the cluster write a time stamp to the server pool file system device.
- The network heartbeat: all Oracle VM Servers communicate through the network to signal to each other that every cluster member is alive.

These heartbeat functions exist directly within the kernel and are fundamental to the clustering functionality that Oracle VM offers for server pools. It is very important to understand that the heartbeating functions can be disturbed by I/O-intensive operations on the same physical storage. For example, importing a template or cloning a virtual machine in a storage repository on the same NFS server where the server pool file system resides may cause a time-out in the heartbeat communication, which in turn leads to server fencing and reboot. To avoid unwanted rebooting, it is recommended that you choose a server pool file system location with sufficient and stable I/O bandwidth. Place server pool file systems on a separate NFS server or use a small LUN, if possible. For more information on setting up storage for a server pool file system, see [Section 3.10, “Are there Guidelines for Configuring Storage?”](#).

3.9 Where are Virtual Machine Resources Located?

A storage repository defines where Oracle VM resources may reside. Resources include virtual machine configuration files, templates for virtual machine creation, virtual machine assemblies, ISO files (CD/DVD image files), shared and unshared [virtual disks](#), and so on. For information on the configuration and management of storage repositories and their contents, see [Chapter 4, Understanding Repositories](#).

3.10 Are there Guidelines for Configuring Storage?

It is important to plan your storage configuration in advance of deploying virtual infrastructure. Here are some guidelines to keep in mind:

- Take care when adding, removing, and resizing LUNs as it may require a physical server reboot. Do not resize LUNs that are used as part of Logical disks; instead, create a new LUN and add it to the disk group.
- If you resize a physical disk directly on the SAN server, you must refresh the physical disk within Oracle VM Manager to ensure that it is aware of the change and is capable of correctly determining the available disk space when performing other operations like creating a repository.
- Avoid remapping LUNs as this can cause data corruption since the targets have been switched outside of the server. The Linux SCSI layer does not support dynamic remapping of LUNs from the storage array.
- Test your configuration, especially fail over, in a test environment before rolling into production. You may need to make changes to the [multipath](#) configuration files of [Oracle VM Server](#).
- Plan the size and type of storage that you are using by workload. For example:
 - Boot volumes can typically be on higher capacity drives as most operating systems have minimal I/O activity on the boot disk, but some of that I/O is memory paging, which is sensitive to response times.
 - Applications can be on larger, slower drives (for example, RAID 5) unless they perform a lot of I/O. Write-intensive workloads should use RAID 10 on medium to fast drives. Ensure that log files are on different physical drives than the data they are protecting.
 - Infrastructure servers such as DNS tend to have low I/O needs. These servers can have larger, slower drives.

- If using storage server features such as cloning and snapshots, use raw disks.
- While it may be tempting to create a very large LUN when using logical disks, this can be detrimental to performance as each virtual machine queues I/Os to the same disks. You should size your repositories based on a thorough assessment of your requirements and then keep them to a size that is close to your exact requirements. Sizing of a repository should be based on a sum of the following values:
 - The total amount of disk space required for all running virtual machines. For instance if you have 6 virtual machines sized at 12 GB, you should ensure that the repository has at least 72 GB of available disk space.
 - The total amount of disk space required to provision each virtual machine with a local virtual disk. Consider that for 6 virtual machines, each with a 36 GB local virtual disk, you must have at least 216 GB of disk space available to your repository.
 - The total amount of disk space required to store any templates that you might use to deploy new guest images.
 - The total amount of disk space required to store ISO images for operating systems that may get installed to new guest images.
 - The total amount of disk space that you may require in reserve to deploy an additional number of virtual machines with local virtual disks in the future. As a rule of thumb, assume that you may need to deploy an additional virtual machine for each running virtual machine, so if you intend to initially run 6 virtual machines, you may find it sensible to size your repository for 12 virtual machines
- Be sure to leave some disk space available to create smaller storage entities of, at least, 12 GB each to use as [server pool](#) file systems. The server pool file system is used to hold the server pool and cluster data, and is also used for cluster heartbeating. You create space for server pool file systems the same way as you create storage entities for storage repositories. For more information about the use and management of clusters and server pools, see [Chapter 6, Understanding Server Pools and Oracle VM Servers](#) and [Servers and VMs Tab](#) in the [Oracle VM Manager User's Guide](#).
- Place server pool file systems on a separate NFS server or use a small LUN, if possible. The cluster heartbeating function can be disturbed by I/O-intensive operations on the same physical storage. For example, importing a template or cloning a virtual machine in a storage repository on the same NFS server where the server pool file system resides may cause a time-out in the heartbeat communication, which in turn leads to server fencing and reboot. To avoid unwanted rebooting, it is recommended that you choose a server pool file system location with sufficient and stable I/O bandwidth.
- Disable read and write caching on the underlying disk systems to guarantee I/O synchronization. Caching may result in data loss if the Oracle VM Server or a virtual machine fails abruptly. To disable write caching, change the applicable settings in the RAID controller BIOS. Alternatively, use the `sg_wr_mode` command or use the SCSI disk class directly: `echo "write through" > /sys/class/scsi_disk/scsi-device-id/cache_type`.



Warning

When creating exports on a file server, if you choose to restrict access to a particular set of hosts, then all exports on the file server must have an identical list of permitted hosts in the export list. In [Oracle VM Manager](#) all of the hosts that have been permitted access must be added to the file server's list of [Admin Servers](#). See [Discover File Server](#) in the [Oracle VM Manager User's Guide](#) for more information on adding Admin Servers to your file server in Oracle VM Manager.

Chapter 4 Understanding Repositories

Table of Contents

4.1 What are the Repository Prerequisites?	36
4.2 How is a Repository Organized?	36
4.3 How is a Repository Created?	37
4.4 How do Oracle VM Servers Access a Repository?	37
4.5 What Virtual Machine Resources are in a Repository?	38
4.6 How are Virtual Machine Templates Managed?	38
4.6.1 Using a Virtual Machine Template	39
4.6.2 Importing a Virtual Machine Template	39
4.6.3 Virtual Machine Template Clone Customizers	40
4.7 How are Assemblies Managed?	40
4.8 How are ISO Files (CD/DVD Images) Managed?	41
4.9 How are Virtual Disks Managed?	41
4.10 How are Virtual Machine Configuration Files Managed?	42
4.11 Can the Repository be Backed Up?	42

A storage repository is used to store virtual machine resources, so that these resources can be made available to [Oracle VM Servers](#) in a [server pool](#), without having to copy the resources to each Oracle VM Server. The Oracle VM Servers in a server pool gain access to these virtual machine resources by having the storage repository *presented* to them. If your storage is file-based storage, you can present a storage repository to multiple server pools. Alternatively, if your storage is physical disk-based, you can only present the storage repository to a single server pool.

A storage repository is essentially logical disk space made available through a file system on top of physical storage hardware. If the storage repository is created on a file server, for example an NFS share, then a file system is already present; if the repository is created on a LUN, an [OCFS2](#) file system is first created.

This chapter provides information on the structure of storage repositories, describes how to create and manage them, and describes how you manage virtual machine resources contained within them.

You use the **Repositories** tab in [Oracle VM Manager](#) to create and configure storage repositories, and to present one or more storage repositories to Oracle VM Servers in a server pool. During the creation and configuration of the storage repository the [Oracle VM Agent](#) on the Oracle VM Server acts as the worker component performing the instructions given through Oracle VM Manager. This chapter discusses the flow of the operations you perform after the [discovery](#) of your physical storage hardware, LUNs, file systems and so on, in order to make a storage repository available to the Oracle VM Servers in your server pool. When the storage repository is accessible, you can start adding storage resources and building virtual machines with those resources.

Depending on the configuration of Oracle VM Servers in your environment, some restrictions may apply to the creation of storage repositories due to the nature of the storage. Keep in mind the following guidelines when working with storage repositories:

- When using server pools without clustering functionality, two storage options are available: file servers (NFS) and local physical disks in a local storage array. Remember that local storage imposes severe restrictions, as described in [Section 3.2.1, “Local Storage”](#).

- Local storage, or unused disks in your Oracle VM Servers, are discovered as LUNs in a local storage array. If you want to use a single-server setup with local storage, be sure to deactivate clustering in your server pool. This eliminates the need for a server pool file system, which cannot be on a local disk.
- Only a server pool with multiple servers, active clustering and attached storage (NFS, iSCSI, fibre channel) can offer [high availability](#), load balancing and similar advanced functionality.



Note

Storage repositories are presented to individual Oracle VM Servers, and not necessarily to all Oracle VM Servers within a server pool. This means that it is possible that a repository may not be available for use on a particular server. You should keep this in mind when you deploy a virtual machine on a specific Oracle VM Server. To see which Oracle VM Servers have access to a storage repository, select the repository in the navigation pane and verify the access status in the management pane with the **Perspective** set to **Info**.



Warning

Never manually copy Oracle VM resources such as virtual machine configuration files from one repository to another, as this can cause duplication of UUIDs within the Oracle VM environment and can cause Oracle VM to malfunction. Always use the tools provided by Oracle VM Manager or the Oracle VM Manager Command Line Interface to move resources. For instance, to move a virtual machine configuration from one storage repository to another use the **Move Virtual Machine** wizard described in [Section 7.7, “How Can a Virtual Machine be Moved or Migrated?”](#).

4.1 What are the Repository Prerequisites?

Before you create a storage repository, make sure that these requirements are met:

- **(NFS) File server repository:** At least one (NFS) file server must be discovered. The file server must be accessible to all Oracle VM Servers that are to use it. The file server directory must be writeable.
- **(LUN) Storage array repository:** The storage array must be accessible to all of the Oracle VM Servers that are to use it. The array server directory must be writeable. A [server pool](#) must exist with clustering enabled, and at least one Oracle VM Server must be present in the server pool. By design, a storage repository on a LUN is linked to a clustered server pool because of the nature of the OCFS2 file system it uses.

A repository on a local server storage also belongs in this category, since local disks are always discovered as LUNs. For more information about local storage, see [Section 3.2.1, “Local Storage”](#).



Note

Only file server storage repositories can be shared by multiple server pools.

4.2 How is a Repository Organized?

Each storage repository has a predefined structure, which is visible in Oracle VM Manager, and maps to the directory structure of the underlying physical storage. The directory structure is listed in [Table 4.1, “Storage Repository Directory Structure”](#).

The directories listed in [Table 4.1, “Storage Repository Directory Structure”](#) are subdirectories of a storage repository file system. On disk they are, in fact, located under the path:

/OVS/Repositories/repository_id/...

Table 4.1 Storage Repository Directory Structure

Directory Name	Description
Templates	This is the directory that contains virtual machine templates. See Section 4.6, “How are Virtual Machine Templates Managed?” for information on working with virtual machine templates.
Assemblies	Contains preconfigured sets of virtual machines, typically created with Oracle Assembly Builder. See Section 4.7, “How are Assemblies Managed?” for information on working with assemblies.
ISO files	Contains ISO files that can be mounted as virtual CD/DVD drives on virtual machines. See Section 4.8, “How are ISO Files (CD/DVD Images) Managed?” for information on working with ISO files.
VirtualDisks	Contains virtual disks, which can be either dedicated to a virtual machine or shared by multiple virtual machines. See Section 4.9, “How are Virtual Disks Managed?” for information on working with virtual disk.
VirtualMachines	Contains virtual machine configuration files. See Section 4.10, “How are Virtual Machine Configuration Files Managed?” for information on viewing virtual machine configuration files.

4.3 How is a Repository Created?

After you complete the storage preparation phase described in [Discover File Server](#) and [Discover SAN Server](#) in the *Oracle VM Manager User's Guide*, Oracle VM Manager is fully aware of the underlying physical storage providers available for use as a storage repository. To create a storage repository, see [Create New Repository](#) in the *Oracle VM Manager User's Guide*.



Important

A storage repository should be at least 10 GB in size. In addition to this minimum size requirement, you should include enough storage space for virtual machines, templates, ISO files and other virtual machine resources.

When the storage repository is created, Oracle VM Manager takes ownership of it, and the Oracle VM Servers you selected during the creating process have access to the repository in order to store virtual machines, ISO files, templates and so on. To modify which Oracle VM Servers have access to the storage repository, see [Section 4.4, “How do Oracle VM Servers Access a Repository?”](#).

4.4 How do Oracle VM Servers Access a Repository?

As part of the storage repository creation you can select the Oracle VM Servers to which the repository must be made available. However, you can modify this selection afterwards and present the repository to new servers or unrepresent it from servers previously selected.

Presenting the storage repository to an Oracle VM Server is the equivalent of mounting a file system. Essentially, when you present the repository to the Oracle VM Servers in your server pool, the repository file system is mounted by the *root* user on each Oracle VM Server. This is an essential factor in the [HA](#) configuration of the Oracle VM setup.

To present or unrepresent Oracle VM Servers to a storage repository, see [Present or Unrepresent Repository](#) in the *Oracle VM Manager User's Guide*.



Note

NFS-based storage repositories can be shared by multiple server pools controlled by the same Oracle VM Manager. OCFS2-based storage repositories always belong to a single clustered server pool. Therefore only members of that server pool appear in either pane.

4.5 What Virtual Machine Resources are in a Repository?

The types of virtual machine resources can be categorized as:

- **Virtual machine templates:** Reusable [virtual machine templates](#) used to create multiple virtual machines.
- **Assemblies:** Template containing a configuration of multiple virtual machines with their [virtual disks](#) and the inter connectivity between them.
- **ISO files:** DVD/CD image files used to create virtual machines from scratch using the installation media.
- **Virtual disks:** Virtual disks used by virtual machines to perform boot operations, to run the operating system, and to extend the storage capability of virtual machines.
- **Virtual machine files:** Configuration files of your virtual machines.

Virtual machines can be created from different types of resources: either from a template or [assembly](#) containing preconfigured virtual machines, or from scratch using an ISO file (image) of an installation DVD. Booting a virtual machine via PXE, or network boot for a [paravirtualized guest](#), is also possible.

4.6 How are Virtual Machine Templates Managed?

Oracle VM templates are self-contained and pre-configured virtual machines with key Oracle technologies. Each Oracle VM template is packaged using Oracle best practices, which reduces installation and configuration costs, reduces risk and dramatically shortens deployment time lines.

Oracle VM templates of many key Oracle products are available for download, including Oracle Linux, Oracle Solaris, Oracle Database, Fusion Middleware, and many more.

Oracle VM template licensing includes a free download and free trial use with the option to purchase a product license. Oracle VM templates do not have time limits or feature limitations, that is, Oracle VM templates are full featured and do not have expiration dates. Oracle VM templates can be quickly transitioned from evaluation into production by purchasing Oracle technology licenses.

You can download Oracle VM templates from the Oracle Technology Network:

<http://www.oracle.com/technetwork/server-storage/vm/templates-101937.html>

Please read the documentation provided with any downloaded template, as it informs you of the relevant configuration steps that you must perform, as well as the default username and password for the template.



Important

When using an Oracle VM template, be sure to change the usernames and passwords from the defaults when you clone your virtual machines. Consistently using default usernames and passwords is a security risk.

The creation of a virtual machine from template is based on cloning: the template is imported as an archive, unpacked and stored as a virtual machine configuration file with images of its disks, which are

cloned to create a new instance in the form of a virtual machine. In the same way, an existing virtual machine can be cloned to create a new virtual machine, and to a new template as well.

A [virtual machine template](#) is a fully pre-installed, pre-configured virtual machine that can be repeatedly used to create new virtual machines. A typical virtual machine template contains:

- An operating system.
- A file which contains the basic configuration information, such as the number of virtual CPUs, the amount of memory, the size of disk, and so on.
- Pre installed applications.

Assembly files can contain the one or more virtual machine templates, though in most cases, an assembly contains a single virtual machine template. An assembly file that contains multiple virtual machine templates can be deployed to create a virtual machine template for each virtual machine. Virtual machine templates are shared among users to create new virtual machines. New virtual machines inherit the same contents and configuration from the template.

You can obtain or create a virtual machine template by:

- Downloading an Oracle VM template from the Oracle Technology Network, and importing it into Oracle VM Manager.
- Downloading an Oracle VM template that contains a descriptor file (.ovf file) from the Oracle Technology Network, and importing it into Oracle VM Manager as an [assembly](#), then creating a template from the assembly.
- Cloning an existing virtual machine as a template in Oracle VM Manager. For information on cloning virtual machine templates, see [Clone a Virtual Machine or Template](#) in the *Oracle VM Manager User's Guide*.

4.6.1 Using a Virtual Machine Template

A virtual machine template can be used to create virtual machines, and to create new templates based on the original. To create a virtual machine from a template, see [Clone a Virtual Machine or Template](#) in the *Oracle VM Manager User's Guide*.

4.6.2 Importing a Virtual Machine Template

A virtual machine template contains various components such as the virtual machine configuration information, virtual disks that contain the operating system and any application software. These components are packaged together as an Oracle VM template file. An assembly is similar to a virtual machine template, but in the open standard Open Virtualization Format (OVF) format. Older Oracle VM template files were packaged as Oracle VM template files, and the more recent templates are packaged in OVF format as assemblies. If the template you are importing uses the OVF format, see [Section 4.7, "How are Assemblies Managed?"](#) for information on importing an assembly and creating a template from the assembly.

Before you can use a virtual machine template, you must import it into Oracle VM Manager and make it available to your server pool(s). Virtual machine templates are stored in the server pool's storage repository by importing them from a web server into Oracle VM Manager.

In a storage repository, templates are typically imported as an archive (.tgz, .tar or other). The archive contains a virtual machine configuration file (.cfg) file, and at least one virtual disk image (.img file).

On disk, a template archive is unpacked after import. The .cfg file of the virtual machine is always referenced from the Templates folder, but the virtual disk image files (.img) are placed in the VirtualDisks folder. This makes the creation of virtual machines from template a lot faster, which also applies to cloning of virtual machines.

If you downloaded an Oracle VM template from the Oracle Software Delivery Cloud which contains an *template.tgz* file, you should import it as a template. This format of a template is the older format used by Oracle to publish Oracle VM templates.

For information on importing a virtual machine template, see [Import Template](#) in the *Oracle VM Manager User's Guide*.

4.6.3 Virtual Machine Template Clone Customizers

Cloning a virtual machine or template means making a copy of it, so that you can create multiple virtual machines or templates from the original. You can create a clone customizer to set up the clone parameters, such as networking, and the virtual disk, and ISO resources. A clone customizer is also used when moving a virtual machine or template from one storage repository to another. To create, edit or delete clone customizers, see [Manage Clone Customizers](#) in the *Oracle VM Manager User's Guide*.

4.7 How are Assemblies Managed?

Assemblies can be described as a template of a group of virtual machines, or a collection of multiple virtual machine templates. An [assembly](#) is a kind of infrastructure template containing a configuration of one or more virtual machines with their virtual disks and even the inter connectivity between them. In the Oracle VM Manager Web Interface, templates and assemblies appear in different tabs of the storage repository, but their virtual machine configuration files and disk images are stored in the same location as those of other virtual machines and templates.

Assemblies can be created as a set of .ovf (Open Virtualization Format) and .img (disk image) files, or may all be contained in a single .ova (Open Virtualization Format Archive) file. Disk image files can be different formats such as VMDK, VHD, VDI in addition to raw disk images. After the assemblies are imported, the disk images are converted into raw disk images used by Oracle VM templates and virtual machines.

To use an assembly, you must first import it into a repository, then create one or more templates from the assembly. You create one [virtual machine template](#) for each virtual machine in the assembly. You can then use the template(s) to deploy the virtual machines that originated from the assembly. You cannot deploy all virtual machines in an assembly in one step; you must deploy each virtual machine individually. To perform deployment of all virtual machines, including the associated networking configuration, you should use Oracle Enterprise Manager.

For ease of use, you should use a single .ova file when working with assemblies in Oracle VM. You can also use the .ovf format with the associated disk image files, but you must import all the disk files individually as virtual disks, then import the .ovf file as an assembly before you can use the assembly to create a template.

An assembly can contain a virtual machine from another virtualization software vendor, such as VMware. This enables you to import virtual machines from other virtualization software providers. Just make sure the virtual machine to be imported is correctly saved or created using the OVF standards and all files are saved in an assembly file using the OVF archive file format, then import the .ova file into Oracle VM Manager as you would any other assembly file.



Note

OVF files in the version 0.9 format (as used by VMware ESX 3.5) are not supported. If the assembly that you wish to import uses an unsupported format, you

can try to use VMware's Converter or OVF Tool to convert to a newer version or you can use qemu-img to convert the vmdk files to raw disk files and recreate the VM manually.

To display the virtual machines which use an assembly, select the assembly in the management pane and expand the table row. Note that assembly contents are not automatically refreshed by a refresh operation performed on the repository where the assembly is hosted, if you want to view the contents of the assembly you may need to refresh the assembly first.

To import an assembly, see [Import Assembly](#) in the *Oracle VM Manager User's Guide*. To create a virtual machine template from an assembly, see [Create VM Template](#) in the *Oracle VM Manager User's Guide*.

4.8 How are ISO Files (CD/DVD Images) Managed?

Virtual machines have no access to the physical DVD or CD-ROM drive. You can assign virtual drives to virtual machines by offering ISO files containing the image of a DVD or CDROM. These image files can be found in the ISOs folder in a storage repository.

To be able to use an ISO file with your virtual machine you must first import the file into an appropriate storage repository, namely one that can be accessed by the server pool where the virtual machine is to be created. You import ISO files from a web server into Oracle VM Manager. If your storage repository uses file-based storage, you can make the repository available to multiple server pools, therefore making an ISO file available to multiple server pools. If you are using array-based storage, you can only make a repository available to a single server pool. Virtual machines can only access ISO files on repositories that have been assigned to the server pool to which they belong.

To import an ISO file, see [Import ISO](#) in the *Oracle VM Manager User's Guide*. To create a virtual machine using an ISO file, see [Create Virtual Machine](#) in the *Oracle VM Manager User's Guide*.

4.9 How are Virtual Disks Managed?

A virtual machine needs at least one disk – that is, a disk from which to boot and run the operating system. Virtual disks can be part of a template, assembly, can be created as part of the virtual machine creation process, or can be created independently inside the storage repository. Virtual disks can be shared across virtual machines, or dedicated to one virtual machine. All virtual disks available in the storage repository appear in **Virtual Disks** in the navigation tree when you select a storage repository, regardless of how they are created.

To create a virtual disk, see [Create Virtual Disk](#) in the *Oracle VM Manager User's Guide*. To import a virtual disk, see [Import Virtual Disk](#) in the *Oracle VM Manager User's Guide*. When you create a virtual machine from scratch, you are given the opportunity to either create a new virtual disk, or use an existing one. For more information about creating a virtual machine from scratch, see [Create Virtual Machine](#) in the *Oracle VM Manager User's Guide*.

While virtual disks may prove easy to manage and migrate, there is a performance cost associated, since file systems are nested within each other. It is possible to create a virtual machine that does not use a virtual disk, but uses virtual storage from a raw physical disk in the form of a LUN accessed directly by the virtual machine. Therefore, for virtual machine disks Oracle VM provides two options:

- **Virtual disks::** Disk image files on a file system, usually stored in a storage repository.
- **Raw physical disks:** LUNs accessed directly by the virtual machine.

The design decision of virtual versus physical storage access depends on your server virtualization use cases as well as the existing storage hardware at your disposal. Virtual storage enables quick and easy

on-the-fly configuration, but introduces a higher level of abstraction, since multiple file systems are nested within each other. Therefore, when using virtual storage it is important to balance ease of use against the lower performance compared to physical storage. Physical storage access implies that virtual machine disk storage is mapped directly onto the LUNs configured in the physical storage hardware, which is exactly like the way physical servers access their storage. The advantage here is that existing procedures and storage management practices can be maintained.

Virtual disks can be detected as different device types within a virtual machine. Older templates and virtual machines tend to detect these disks as **hdx** devices, while newer virtual machines and templates detect virtual disks as **xvdx** devices. Oracle VM does not support virtual machines that use mixed device types for virtual disks. If you are working with an older template or virtual machine, it is recommended that you remove all current disk mappings and then recreate them so that they use the new **xvdx** device types. Alternately, if your virtual machine does not support **xvdx** device types at all, you should edit the `vm.cfg` file for the virtual machine by hand to ensure that any storage devices, such as a virtual cdrom device, are changed from **xvdx** devices to **hdx** devices. For instance, you might change the following entry in the `vm.cfg` file:

```
disk = [
file:/OVS/Repositories/0004fb00000300000eaala4a8e297e7c/VirtualDisks\
/0004fb0000120000ce3f55b28a87d068.img,xvda,w,
file:/OVS/Repositories/0004fb00000300000eaala4a8e297e7c/ISOs\
/0004fb000015000040ba0ea64887c0d0.iso,xvdb:cdrom,r]
```

to:

```
disk = [
file:/OVS/Repositories/0004fb00000300000eaala4a8e297e7c/VirtualDisks\
/0004fb0000120000ce3f55b28a87d068.img,hda,w,
file:/OVS/Repositories/0004fb00000300000eaala4a8e297e7c/ISOs\
/0004fb000015000040ba0ea64887c0d0.iso,hdb:cdrom,r]
```

4.10 How are Virtual Machine Configuration Files Managed?

The **VM Files** folder in a storage repository navigation tree lists all the virtual machine configuration files in the storage repository. In the **Create Virtual Machine** wizard you specify which storage repository to use to store the configuration file of your virtual machine. Consider the **VM Files** folder to be the home location of all the virtual machines that have been created in the selected storage repository. You cannot perform any actions on the virtual machine configuration files. If you want to rename, move or delete any of these files, you should perform those operations on the virtual machine, not just the virtual machine configuration file.

4.11 Can the Repository be Backed Up?

An Oracle VM Server can be configured to enable third party applications to perform a back up of the contents of a storage repository. To enable this, an Oracle VM Server is configured to provide an NFS share that a third party back up tool can use to access the contents of the repository. The Oracle VM Server must be in a clustered server pool and have the OCFS2-based storage repository presented to it.

When you have created a repository export, use the **Repository Path** (displayed in the management pane table) and the Oracle VM Server hostname or IP address to connect to the NFS mount point from the third party back up software.

To enable third party backup tools to access the contents of a storage repository, see [Repository Exports Perspective](#) in the *Oracle VM Manager User's Guide*.

Chapter 5 Understanding Networks

Table of Contents

5.1 Networking Terminology	44
5.2 How are Networks Used in Oracle VM?	45
5.3 How are IP Addresses Assigned?	48
5.4 How is Network Bonding Used in Oracle VM?	48
5.5 How are VLANs Used in Oracle VM?	50
5.5.1 Configuring VLANs	51
5.5.2 Configuring VLAN Interfaces	51
5.6 How are Network Functions Separated in Oracle VM?	52
5.6.1 Management Channel	54
5.6.2 Cluster Heartbeat Channel	55
5.6.3 Live Migrate Channel	56
5.6.4 Storage Channel	56
5.6.5 Virtual Machine Channel	56
5.7 Network Planning for an Oracle VM Deployment	57
5.7.1 Networks with Ports and/or Bonds	58
5.7.2 Networks with VLANs Only	59
5.7.3 Hybrid Networks (VLANs with Ports and Bonds)	60
5.7.4 Logical Networks on a Single Oracle VM Server (Intra-Server)	61
5.8 Dealing with Failed Network Operations	61

Networking is a very broad concept with many different interpretations. Data center administrators typically have their own idea about what the best network configuration is in terms of performance, security and cost-effectiveness. In some cases physical network connections are readily available so [bonding](#) or data link aggregation is preferred for fail over or higher bandwidth, while other configurations use [VLANs](#) for network segregation or to compensate for the lack of free NICs. Some will use Ethernet connections for storage while others have dedicated fibre channel hardware at their disposal.

Generally speaking, data center operators tend to think essentially in terms of hardware: switches, routers, firewalls, cables, NICs (Network Interface Cards), and so on. The only widespread network virtualization concept to date is VLAN (Virtual LAN) technology. VLANs are also very frequently used in Oracle VM networking.

The networking infrastructure in the Oracle VM environment comprises connections between various components:

- Between [Oracle VM Servers](#) themselves.
- Between Oracle VM Servers and [Oracle VM Manager](#).
- Between Oracle VM Servers and their storage sub-systems.
- Between virtual machines deployed in the environment.
- Between virtual machines and external private or public networks.

These networking connections can leverage features supported by Oracle VM, such as networked file systems, clustering, redundancy and load balancing, bridging, and support for [Virtual LANs](#) (VLANs).

This chapter discusses common networking concepts used within Oracle VM and provides an overview of how and where different networking technology is used. A brief overview of some of these networking concepts is provided in [Section 2.6.2, "Networking"](#).

For quick reference to some of the networking terminology used, refer to [Section 5.1, “Networking Terminology”](#).

For a more thorough overview of how networks are used in Oracle VM, see [Section 5.2, “How are Networks Used in Oracle VM?”](#).

Networking technologies such as data link aggregation or bonding are discussed in [Section 5.4, “How is Network Bonding Used in Oracle VM?”](#), while VLANs are discussed in [Section 5.5, “How are VLANs Used in Oracle VM?”](#).

The way in which network traffic specific to Oracle VM is assigned to different networks is described in detail in [Section 5.6, “How are Network Functions Separated in Oracle VM?”](#).

This chapter also provides some network preparation guidelines in [Section 5.7, “Network Planning for an Oracle VM Deployment”](#).

You may also want to review additional *Oracle VM 3 Networking* white papers available on OTN at:

<http://www.oracle.com/technetwork/server-storage/vm/overview/index.html>

5.1 Networking Terminology

Since this chapter contains a lot of information about different networking components and their relationships, this section provides a breakdown of some of the networking terminology used here.

- **Port::** the physical network interface on a server. This term is used interchangeably with NIC (Network Interface Card). Network ports can be used to host multiple VLAN interfaces. Multiple network ports can be bonded together for redundancy and performance reasons.
- **Bond:** An aggregation of network ports that act as a single network interface for redundancy and performance reasons. Network bonding is also called Data Link Aggregation. Once a port is part of a bond, it can no longer be used outside of the bond. Oracle VM supports a number of different bonding modes or types. A bond can host multiple VLAN interfaces, or be used as an alternative to a physically cabled port.
- **Bridge:** A method of conjoining different networks together to act as a single network. This technology is only used when creating Virtual Machine networks, and configuration is handled automatically within Oracle VM. Using bridges, virtual machines on one Oracle VM Server are able to communicate with virtual machines on another Oracle VM Server across a network that has been configured for this purpose.
- **VLAN:** A method used to virtualize networking at the switch or router for better control over network separation. VLANs are virtual networks that use identifiers to separate traffic into different networks within the switch. Using VLANs can often reduce network maintenance overhead, as network segregation can be achieved virtually, often from a remote location. Using VLANs can allow servers with a minimal number of physical ports to act as if they were using multiple ports cabled into different networks. Since VLANs can be attached to network bonds, it is possible to achieve the same level of bandwidth that could be achieved using physical cabling by bonding ports together. Although Oracle VM Server can use VLANs, the actual VLAN creation occurs on the switch or router. Network administrators create VLANs and assign VLANs to switch ports on Ethernet switches. The physical cabling from the switch to an Oracle VM Server defines which VLANs are available on the ports or bonds on the Oracle VM Server.
- **VLAN Segments (IDs):** VLANs are divided into segments that are usually tagged with an ID. This allows the switch to determine how to direct traffic. VLAN segments can be thought of as separate physical networks. If a VLAN allows untagged traffic, this traffic is all dealt with as if on the same physical

network. When creating a logical network within Oracle VM Manager, VLAN segments are attached to the network in the same way that you would attach a port or a bond.

- **VLAN Interfaces:** Oracle VM Manager introduces the concept of a VLAN interface. To manage network traffic tagged for different VLAN IDs, a separate virtual interface can be created for each VLAN ID. When creating different logical networks in Oracle VM Manager, these VLAN interfaces can be attached to different networks to specify the type of traffic that belongs to a particular VLAN ID.
- **Logical Network:** The networks referred to in Oracle VM Manager are logical networks, in the sense that they do not necessarily represent a single physical network. Since physical components can be bonded together, bridged, or come in the form of multiple VLAN segments using a single port or bond; networks created in Oracle VM Manager are mapped to all of the individual physical or virtual components that make them up. Therefore, creating a network in Oracle VM Manager is a process of aggregating the information about individual network elements that group together into a single logical network. A logical network may incorporate multiple VLANs, physical ports or bonds.
- **Network Channel:** Each logical network in Oracle VM Manager can be used for a variety of functional purposes. For performance, security and stability reasons it usually makes sense to separate these functions across different logical networks. Each function is referred to as a network channel. Network channels are used to separate types of traffic to reduce the impact that they have on each other. For instance, cluster heartbeat traffic is very sensitive to latency, while virtual machine traffic should be separated from infrastructure traffic for security reasons. Any number of channels may be attached to a logical network.
- **VNIC:** Virtual machines are assigned VNICs or virtual network interface cards, which are allocated faux MAC addresses. This allows each virtual machine to connect to a network. The VNICs are bridged interfaces that are connected to a logical network that has the Virtual Machine channel enabled. A VNIC is only ever assigned to a virtual machine. A virtual machine can have as many VNICs as required within the limitations posed by the virtualization method used. For instance, hardware virtualized virtual machines are able to support a limited number of VNICs, while paravirtualized virtual machines can have an unlimited number of VNICs.
- **Intra-Server Network:** It is possible to create a virtual network that exists on a single server and does not connect to any physical interface on that server. A network like this is called an Intra-Server Network and it is usually only used to connect virtual machines running on the server. Virtual Machines that make use of an Intra-Server Network cannot be migrated or moved to another server unless the Intra-Server Network is removed.
- **Hybrid Network:** A hybrid network is a network that makes use of VLAN segments for some servers, while making use of physical bonds or ports for other servers within the same network. This is typically done when using heterogeneous hardware where VLANs are used to accommodate multiple networks on a number of servers with a lack of physical ports, or for servers with low priority or bandwidth requirements. In general, it is better to choose one networking methodology to use across your infrastructure for maintenance reasons.

5.2 How are Networks Used in Oracle VM?

Oracle VM allows you to define logical Ethernet networks by mapping information about your existing physical network infrastructure within [Oracle VM Manager](#). Therefore, it is very important that you are clear on the physical connections that exist within your environment before you begin configuring networks within Oracle VM Manager itself.

The physical network is the collection of physical connections in Oracle VM Manager and all [Oracle VM Servers](#), and the switches and routers that allow information to reach its destination.

A logical network in Oracle VM is built on top of these physical connections. Each physical connection is called a network port. Other names for this physical connection include network interface card, or NIC, or network interface. Oracle VM supports both 1 Gbit and 10 Gbit NICs. Structuring your logical networks depends on the number of network ports available to your Oracle VM Servers. The minimum recommended number of ports required on a single Oracle VM Server is two, although a single port can be used for test or demonstration purposes. If you have more than two ports on your Oracle VM Servers, you can design more redundancy or traffic isolation in your environment.

Traffic isolation can be achieved by defining different logical networks dedicated to particular network functions and then assigning particular network ports to each logical network. If you have a limited number of NICs available in a system, isolation can also be achieved by configuring [VLANs](#) and then assigning these to a logical network. Redundancy can be achieved by using network bonding, so that multiple NICs essentially act as a single port. Oracle VM Manager provides many of the tools that you need to configure all of this functionality, with the exception of the creation of VLANs which must actually be performed on your switch.

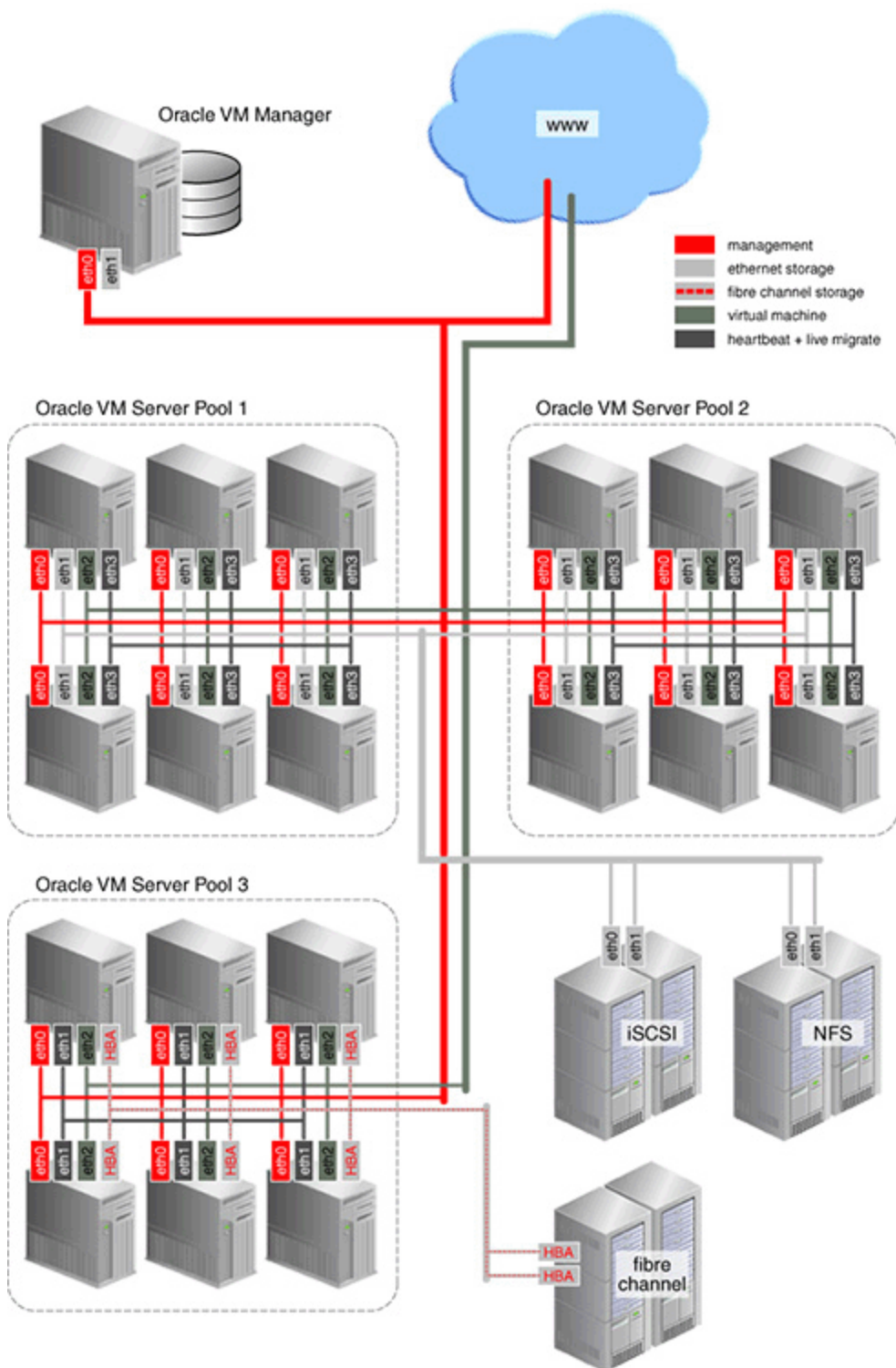
Oracle VM Manager allows you to define a name or alias for each logical network that you create. When you have created your networks, you connect the physical network ports to the logical networks. In the case of VLANs, you define individual VLAN interfaces for each VLAN ID and then attach these VLAN interfaces to the network as if they were physical ports. Logical networks defined in Oracle VM Manager can then be assigned different functions or channels. All network channels can either be on dedicated or shared physical networks, except for the virtual machine intra-server network. For example, a physical network can be dedicated to Virtual Machine or Storage only, or can be used for all network channels. These different network channels are as follows:

- Management Network
- Cluster Heartbeat Network
- Live Migrate Network
- Storage Network
- Virtual Machine Network

Each of these network channels, how they are configured and used, is discussed in detail in [Section 5.6, "How are Network Functions Separated in Oracle VM?"](#).

All of the network elements as well as the networks that you create in Oracle VM Manager are stored as networking objects in the Oracle VM Manager database. Your [Oracle VM Servers](#) are unaware of these Oracle VM Manager network objects. Creating and managing network objects in Oracle VM Manager results in the configuration or deletion of the network devices (for example: ports, VLAN interfaces, bridges) present on Oracle VM Servers. In the case where an Oracle VM Manager database is corrupted or lost, the networking already configured on the Oracle VM Servers remains configured. This means that if preconfigured Oracle VM Servers are discovered in a new instance of Oracle VM Manager, these networking elements are also discovered, although aliases assigned to them in a previous instance of Oracle VM Manager are lost.

Figure 5.1 Oracle VM Networking Example



The table, titled [Figure 5.1, “Oracle VM Networking Example”](#), shows an example of an Oracle VM environment with split network functions. Each Oracle VM Server is connected to the management network, regardless of which *server pool* they belong to.

It is best practice to define separate networks for heartbeat functionality and for [live migration](#). This is because functions like live migration can generate peaks in network traffic. Since heartbeat functionality is sensitive to peak loads, it is better that this function is not affected by a loaded network. Temporarily high network load could cause the heartbeat to fail for a server, resulting in the server being fenced out of a cluster unnecessarily. Since these types of network traffic occur at the level of an individual server pool, the networks do not need a gateway. It is important to understand that when creating different networks to handle separate functions, it is not possible for a server to belong to more than one network that has been assigned the same function.

Virtual machine traffic is often routed over a dedicated network, although it can be combined with the other network functions. In this example the dedicated virtual machine network has a route to the internet (or corporate wide area network). You can create as many virtual machine networks as permitted by your network infrastructure.

The first two server pools are connected to a storage network with Ethernet based storage providers. Ethernet based storage is provided as either NFS file servers or iSCSI LUNs. Server pool 3 has dedicated fibre channel storage, which requires a fibre channel switch and host bus adapters (HBAs) in all connected hardware components. Similar to networks for virtual machines, you create as many storage networks as needed to implement your storage strategy.

To see how you are able to create and manage different networks, see [Networks](#) in the [Oracle VM Manager User's Guide](#).

5.3 How are IP Addresses Assigned?

When setting up a network within [Oracle VM Manager](#) you have the option to configure the IP addressing mechanism that should be used for each [Oracle VM Server](#) and also for each virtual machine. There are three options, [None](#), [DHCP](#), and [Static](#). Setting the IP addressing mechanism to [None](#) leaves the IP addressing for the network unconfigured. If you choose to make use of DHCP to automatically assign IP addresses to servers or virtual machines within a network, you must ensure that a DHCP server is set up and available within your Oracle VM environment. Oracle VM Manager does not function as a DHCP server by itself. In order for DHCP to function properly, the DHCP server must be connected to the physical network ports that you have specified for each Oracle VM Server within the network. Furthermore, it is important that your DHCP is configured to assign static IP addresses to interfaces on your Oracle VM Servers, since there is a requirement that the IP address for an Oracle VM Server does not change. The behavior of the Oracle VM Server host is undefined if used in an environment where your IP address may change due to DHCP lease expiry.

If using static IP addresses for your Oracle VM Servers or virtual machines, Oracle VM Manager automatically configures the network parameters for each Oracle VM Server or virtual machine, via the Oracle VM Agent, according to the IP address, netmask and gateway that you specify for each server or virtual machine within Oracle VM Manager.

When configuring network bridges, or networks that solely function as virtual machine networks, it is possible to not specify an IP addressing mechanism for the bridge. In this case, the bridge functions as a Layer 2 switch for the virtual machines making use of it. See [Section 5.6.5.1, "Network Bridges"](#) for more information on this.

5.4 How is Network Bonding Used in Oracle VM?

Network [bonding](#) refers to the combination of network interfaces on one host for redundancy and/or increased throughput. Redundancy is the key factor, it is desirable to protect the entire virtualized environment from loss of service due to failure of a single physical link. This network bonding is the same

as the Linux network bonding or Oracle Solaris data link aggregation. Using network bonding in Oracle VM may require some switch configuration.



Important

While Oracle VM Manager uses the Linux terminology for network bonds, Oracle Solaris users should understand this to be equivalent to data link aggregation.

In Oracle VM, there are three modes of network bonding:

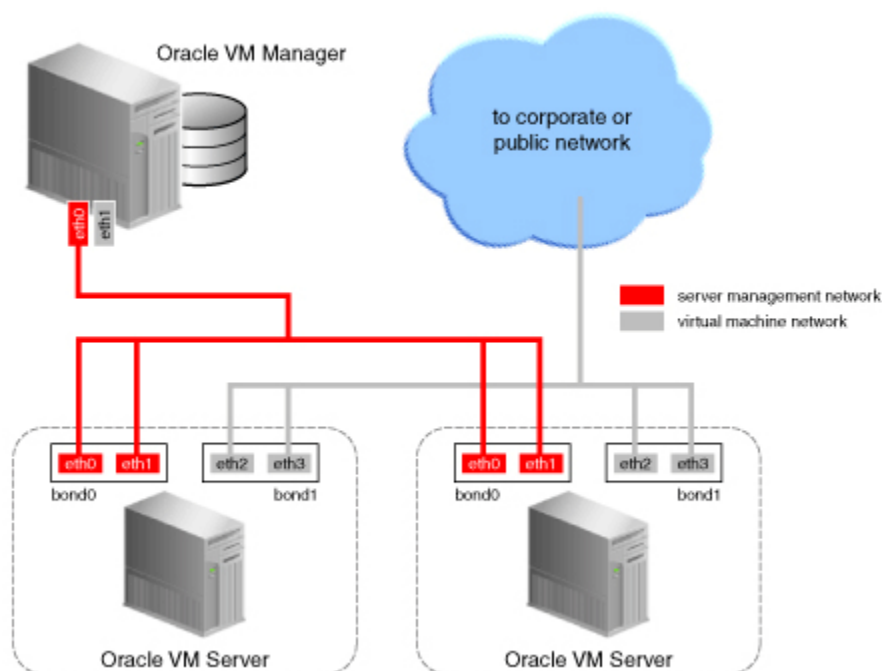
- **Active Backup** (mode 1): There is one NIC active while another NIC is asleep. If the active NIC goes down, another NIC becomes active. While this mode does not increase throughput, it provides redundancy in case of failure. Active Backup is a safe option if you intend to make use of VLANs.
- **Dynamic Link Aggregation** (mode 4): Aggregated NICs act as one NIC which results in a higher throughput, but also provides failover in the case that a NIC fails. Dynamic Link Aggregation requires a switch that supports IEEE 802.3ad. Dynamic Link Aggregation is the preferred mode of network bonding, but requires that the network is configured correctly on the switch. Furthermore, you should be aware that there are significant cabling requirements. An initial guideline is that all of the Ethernet ports aggregated into a single bond using this bonding mode must be connected to the same switch.
- **Adaptive Load Balancing** (mode 6): The network traffic is equally balanced over the NICs of the machine and failover is also supported to provide redundancy. Unlike Dynamic Link Aggregation, Adaptive Load Balancing does not require any particular switch configuration. Adaptive Load Balancing is only supported in x86 environments. Adaptive Load Balancing may not work correctly with VLAN traffic.



Note

Adaptive Load Balancing (mode 6) is currently not supported for SPARC servers.

Figure 5.2 Network bonding



During installation of Oracle VM Server, the network interface (selected when prompted for the management port) is configured as a bonded interface. The bond is created with only one interface. This is done because the reconfiguration of the management interface on the Oracle VM Servers is not supported. You can add a second interface to the already existing bond device without affecting the configuration of the original interface. This is illustrated in [Figure 5.2, “Network bonding”](#), where a second network interface is added to bond0, the network bond created during installation. By default, the bond mode is set to Active Backup for the management network.

[Figure 5.2, “Network bonding”](#) also illustrates the configuration of a second bonded interface, bond1, which can be used for other network usage, such as the virtual machine channel. Separation of network functions into different channels is discussed in more detail in [Section 5.6, “How are Network Functions Separated in Oracle VM?”](#).

It is important to understand that the actual cabling of Ethernet interfaces is important when using network bonds. If you are using Active Backup (mode 1) or Adaptive Load Balancing (mode 6), the Ethernet ports can be connected to alternate switches as shown in [Figure 5.3, “Network bonding for modes 1 and 6”](#). On the other hand, if you are using Dynamic Link Aggregation (mode 4), the Ethernet ports *must* be cabled to the same switch, which is configured for dynamic link aggregation (IEEE 802.3ad). This is illustrated in [Figure 5.4, “Network bonding for mode 4”](#).

Figure 5.3 Network bonding for modes 1 and 6

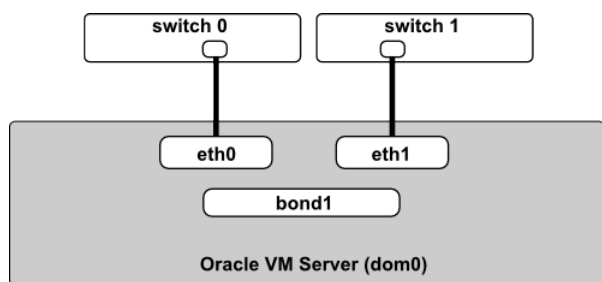
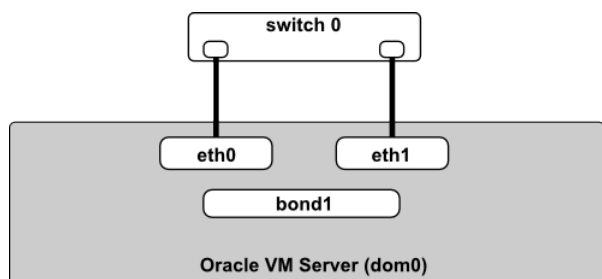


Figure 5.4 Network bonding for mode 4



For more information on configuring bonds in Oracle VM, see [Bond Ports Perspective](#) in the [Oracle VM Manager User's Guide](#).

5.5 How are VLANs Used in Oracle VM?

Oracle VM supports multiple virtual networks, or [VLANs](#), on the same network port or bond. Each VLAN is essentially an independent logical network operating with other VLANs over the same physical connection. This means that virtual machines deployed on different networks, connected through the same Oracle VM Server port (or bond), can have traffic directed to different VLANs. This feature is implemented using VLAN Interfaces.

Configuring VLANs within Oracle VM Manager involves creating VLAN interfaces. A VLAN segment or ID is assigned to each VLAN interface when you create it. The VLAN interface is a virtual interface attached to the bond or port that physically handles the traffic for the entire VLAN. When configuring different logical networks within Oracle VM Manager, the VLAN interfaces that you have defined can be attached to the network in the same way as any ports or bonds. This allows you to treat each VLAN ID as a separate logical network interface. When traffic is routed through the VLAN interface, it is automatically tagged with the VLAN ID that is configured for that interface, and is then routed through the actual port or bond that the VLAN interface is attached to.

The VLAN ID is used by an attached VLAN switch to segregate traffic among the different VLANs operating on the same physical link. When a VLAN is configured, it functions exactly like a separate physical connection. The original physical port that the VLAN interface is attached to, can continue to be used as a separate port to route untagged traffic.

5.5.1 Configuring VLANs

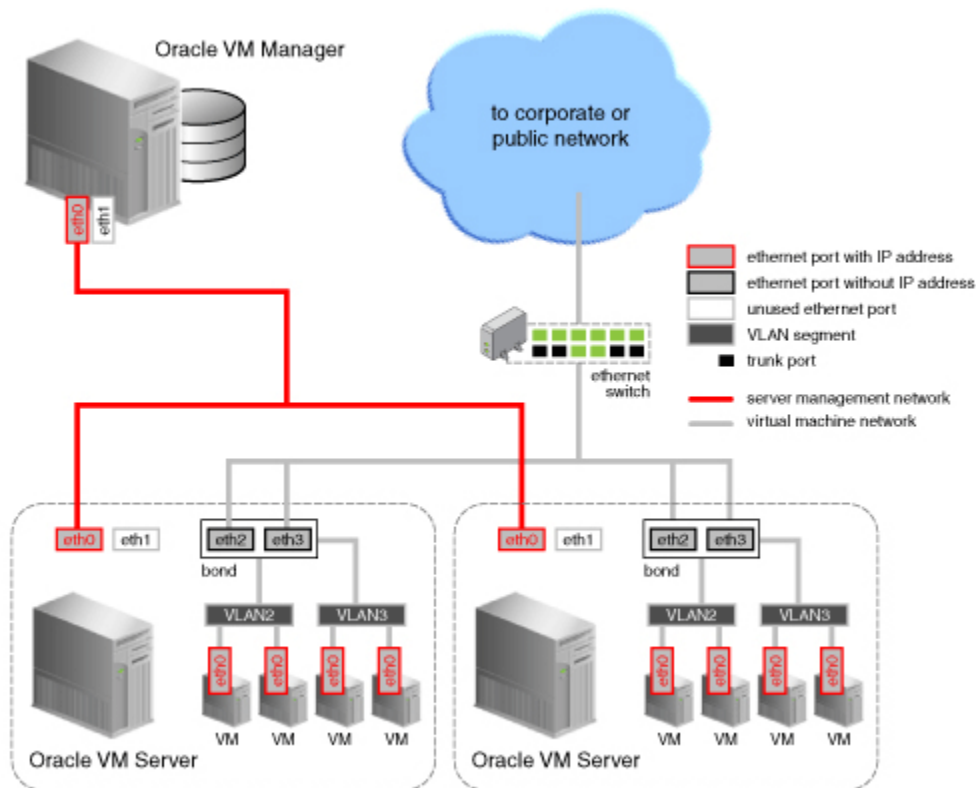
You must configure the VLANs needed to support your network before you can use them. This is usually accomplished using switch trunking. Trunking involves configuring ports on the switch to allow multiple VLAN traffic on these ports, to ensure that packets are correctly transmitted to their final destination. Consult your switch vendor's documentation for information regarding trunking.

5.5.2 Configuring VLAN Interfaces

A VLAN Interface is a virtual interface that is attached to the physical network port or bond that your VLAN is configured on. The VLAN Interface is used to automatically tag traffic that is routed through it with the appropriate VLAN ID. Equally, traffic tagged with a VLAN ID that comes in on a physical port is routed through the appropriate VLAN interface. If no matching VLAN interface is found, the traffic is dropped.

You create VLAN interfaces to direct the traffic from several VLANs onto a single port or bond on each Oracle VM Server in the [server pool](#). For example, if a port or bond is expected to carry traffic for VLAN with ID 2 and for VLAN with ID 3, you create a VLAN Interface for each of these VLAN segments. After creating the VLAN Interfaces, you create a network and specify one of the VLAN Interfaces as belonging to the network. Each packet transmitted from virtual machines on this network is tagged with the VLAN ID for the VLAN interface specified during network creation. If you added the physical network port or bond during network creation, the packets can still flow through the port or bond, but the packets are untagged. The Ethernet switch, to which the Oracle VM Servers are connected, is responsible for the transmission of packets to the appropriate VLAN.

[Figure 5.5, “Networks with VLANs and VLAN Interfaces”](#) illustrates the case of two virtual machine networks, whose network traffic flows through the same bonded interface.

Figure 5.5 Networks with VLANs and VLAN Interfaces

In the diagram, titled [Figure 5.5, “Networks with VLANs and VLAN Interfaces”](#), two VLAN interfaces are defined on the bond for each server. The VLAN2 interface handles traffic tagged with the VLAN ID 2 and the VLAN3 interface handles traffic tagged with the VLAN ID3. Two virtual machine networks have been created in Oracle VM Manager, even though these networks use the same physical network infrastructure. The first virtual machine network has the VLAN2 interface attached for each server; while the second virtual machine network has the VLAN3 interface attached for each server. For each logical network, a bridge is automatically created for the specified VLAN interface. The bridge is configured without an IP address since none is specified during configuration. Network packets from virtual machines deployed on VLAN segment 2 travel through the bridge and acquire a tag which identifies the packets as belonging to VLAN 2. Similarly, the packets issued from the virtual machines deployed on the network for VLAN segment 3 are tagged for VLAN3 with ID 3. The packets from both networks use either path to the switch if the bond is configured for Dynamic Link Aggregation. The receiving ports on the Ethernet switch are configured using trunking or similar configuration to recognize network traffic for the two VLANs in this setup. As such, the trunk ports will direct the packets to the correct VLAN on the switch, or other connected switches.

To see how you are able to create VLAN Interfaces in the Oracle VM Manager Web Interface, see [VLAN Interfaces](#) in the [Oracle VM Manager User's Guide](#) and to see how VLAN Interfaces are attached to logical networks within the Oracle VM Manager Web Interface see [Networks](#) in the [Oracle VM Manager User's Guide](#).

5.6 How are Network Functions Separated in Oracle VM?

Depending on the number of available network ports on your Oracle VM Servers, and whether or not you use [VLANs](#), you can create additional networks and assign network functions to them. The

exception would be the Management function, which is already assigned, and cannot be removed from the management network(s) created when the Oracle VM Servers were [discovered](#). For example, if your Oracle VM Servers have two NICs, you might create a second network with the Virtual Machine channel. Equally, networks can share functions, so you can add the Storage function to your Management network if your storage is connected to the same network as defined by the Management network.

Since it is possible that a single network can be used for multiple functions, the term used for a network function is *channel*. Therefore, you may want to separate different network functions into different channels. Some of these channels may share the same logical network, but ideally each channel should be assigned its own logical network.

After your management networks are in place, you can plan for the creation of other types of network. Note that once a port is selected for a particular network, it cannot be selected again when creating additional networks. You can use a combination of network [bonding](#) and VLAN Interfaces to create all the networks needed for your environment using your existing ports. Network bonding is covered in [Section 5.4, “How is Network Bonding Used in Oracle VM?”](#); VLAN Interfaces are covered in [Section 5.5, “How are VLANs Used in Oracle VM?”](#).

If you have more than two ports on your Oracle VM Servers, or if you are using VLANs, you can create additional networks for Storage channels. These networks might be used to connect your Oracle VM Servers to either iSCSI or NFS-based storage. Generally, all Oracle VM Servers that belong to the same pool access the same storage. For each network created, you select a port, bond or VLAN interface on each Oracle VM Server to participate in this network.

You can also create a separate network for the Live Migrate channel. After the initial server discovery, the Live Migrate channel is assigned to the Management network. Oracle VM encrypts migration traffic using SSL, to protect sensitive data from exploitation and to eliminate the requirement for a dedicated network. Nonetheless, if you have sufficient network resources on your Oracle VM Servers within a server pool, you can choose to create a separate network for [live migration](#) of virtual machines.

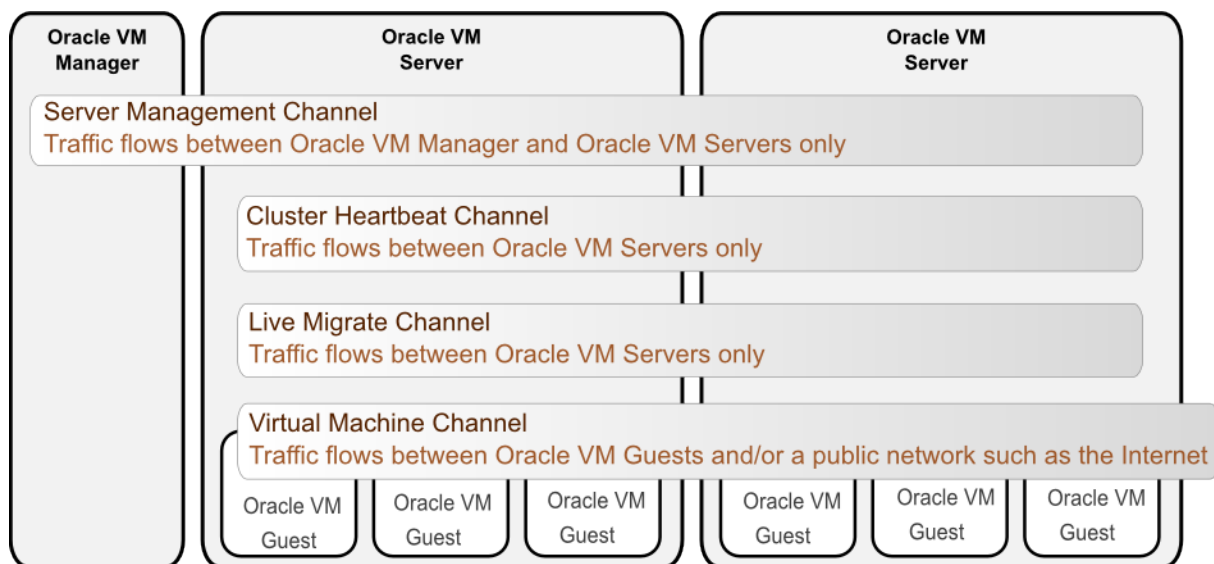
Similarly, the Cluster Heartbeat network channel is assigned to the Management network upon discovering the first Oracle VM Server. The heartbeat communication does not generate a lot of traffic on the network, and therefore does not have much impact on the Management network. It is however susceptible to latency. For this reason, you can choose to create a separate network for the cluster heartbeat function.



Note

Though you can create several networks for the heartbeat and live migration functions, each Oracle VM Server can only participate in one heartbeat and live migration network.

Figure 5.6 Oracle VM Network Channels



Network configuration is independent of your server pool configuration, but both entities must be taken into account when designing your overall networking infrastructure. Oracle VM Manager communicates with all Oracle VM Servers in the environment, using the management port, independent of how Oracle VM Servers are grouped to form server pools. Some network configuration in your environment might be dependent on the storage available to specific server pools. Virtual machines deployed from separate server pools might use the same external network. For this reason, it is best to plan your network design based on current network and storage setup as well as anticipated growth.

For more information on creating a network, refer to [Create New Network](#) in the *Oracle VM Manager User's Guide*.

5.6.1 Management Channel

The Management channel is used to manage the physical [Oracle VM Servers](#) in a [server pool](#), for example, to update the [Oracle VM Agent](#) on the different Oracle VM Servers. This network function is assigned to at least one network by default.



Note

In Oracle VM the management network interface and the public interface (i.e. default route) are expected to be the same on each Oracle VM Server. Other types of network usage are allowed on the same interface, for example through the use of [VLANs](#) and/or network bridges.

The first step in configuring your Oracle VM environment is to [discover](#) your Oracle VM Servers. This step assumes that the Oracle VM Manager host and all of the Oracle VM Servers can communicate over the same network, though the Oracle VM Servers and Oracle VM Manager can reside in different subnets. When you discover the first Oracle VM Server, the management network is created automatically and takes its name from the subnet to which the Oracle VM Server is connected. Each additional Oracle VM Server discovered is either on a previously known subnet, or a new subnet. If the Oracle VM Server is on a new subnet then a new management network is constructed; if the Oracle VM Server is on a known subnet then it is simply added to that subnet. Each server in your Oracle VM environment can only have one interface designated for management, belonging to a single management network object in the Oracle VM Manager's database.

**Warning**

Although the Oracle VM Manager and its discovered and owned Oracle VM Servers may be on different subnets as long as they can reach each other, Network Address Translation (NAT) is not supported in this configuration. NAT would lead to a discrepancy between the actual management IP of the Oracle VM Server and the IP provided during discovery.

A network port on every Oracle VM Server is designated as the management interface during the installation of the Oracle VM Server and is configured as a bonded interface. Ports can be added to this bond or removed from it. Once a management network is created, it can only be deleted if no servers have ports in the management network anymore.

5.6.2 Cluster Heartbeat Channel

The Cluster Heartbeat channel is used to verify if the Oracle VM Servers in a clustered *server pool* are up and running. The heartbeat function has a network component, where a TCP/IP communication channel is created with each Oracle VM Server. Each Oracle VM Server sends regular keep-alive packets and these packets are used to determine if each Oracle VM Server is alive.

**Important**

It is recommended to separate the Cluster Heartbeat function from networks with high load, such as Storage and Live Migrate networks. If bandwidth drops too low, heartbeating connectivity might be interrupted, which could lead to rebooting of virtual machines and Oracle VM Servers.

Oracle VM uses [OCFS2](#) as its underlying clustering file system to manage its storage repositories and provide access to shared storage.

A cluster *heartbeat* is an essential component in any OCFS2 cluster. It is charged with accurately designating *nodes* (in this case, nodes are [Oracle VM Servers](#)) as dead or alive. There are two types of heartbeats used in OCFS2:

- The disk heartbeat where all the Oracle VM Servers in the cluster write a time stamp to the [server pool](#) file system device. See [Section 3.8, “How is Storage Used for Server Pool Clustering?”](#) for more information on this part of the clustering technology.
- The network heartbeat which is where the Oracle VM Servers communicate through the network to signal to each other that every cluster member is alive.

The *quorum* is the group of Oracle VM Servers in a cluster that is allowed to operate on the shared storage. When there is a failure in the cluster, Oracle VM Servers may be split into groups that can communicate within their groups and with the shared storage, but not between groups. In this case, OCFS2 determines which group is allowed to continue and initiates *fencing* of the other group(s). Fencing is the act of forcefully removing an Oracle VM Server from a cluster. An Oracle VM Server with OCFS2 mounted will fence itself when it realizes that it does not have quorum in a degraded cluster. It does this so that other Oracle VM Servers are not stuck trying to access the cluster's resources. When an Oracle VM Server is fenced, it is rebooted and rejoins the cluster. If an Oracle VM Server is fenced, the virtual machines running on the fenced Oracle VM Server are migrated and restarted on other Oracle VM Servers if the virtual machines are HA enabled (virtual machines that are not HA enabled are not migrated).

The cluster heartbeat is sensitive to network interruptions and therefore the Cluster Heartbeat network should be given special attention and be treated separately to make sure that:

- It is not sharing the same links with high traffic networks or networks that may experience sudden traffic spikes like the Storage or Live Migrate networks.

- It has redundancy using a bond which ensures continued operation if one network path fails. See [Section 5.4, “How is Network Bonding Used in Oracle VM?”](#) for more information on configuring bonding.

5.6.3 Live Migrate Channel

The Live Migrate channel is used to migrate virtual machines from one Oracle VM Server to another in a *server pool*, without changing the status of the virtual machine.

While live migration should not be occurring frequently, during a live migration network traffic may spike. This could cause interruption to other services, particularly the cluster heartbeat functionality used for server pool clustering. As a result, configuring a separate network for this purpose can improve the performance and availability of other services within the environment.

5.6.4 Storage Channel

The Storage channel is reserved for future use and currently has no practical function or application. The Storage channel doesn't control anything about which subnet storage traffic is allowed to flow over. The storage channel simply tells the Oracle VM Manager to allow you to add IP addresses to ports on the Dom0. This is only needed if you create an Oracle VM network that you want NFS traffic to flow over on Dom0 and you have no reason to assign any of the other network channels. Basically, you don't need to assign this role unless you create an Oracle VM network that you don't need any of the other four channels, but you need to be able to assign an IP to the Dom0 interface only.

In reality there is nothing in Oracle VM that actually restricts the type of traffic on this network, so the distinction here is really a logical one. In practice, the storage network could be used for other types of network traffic between components that are physically connected to this network, but doing so may cause confusion for other administrators viewing the configuration of your environment.

5.6.5 Virtual Machine Channel

The Virtual Machine channel is used for the network traffic between the different virtual machines in a *server pool*. The Virtual Machine channel can be either standard Inter-Server (routable through standard switches), or Intra-Server (without a route to an external physical network and dedicated to the selected Oracle VM Server). The implications of using an Intra-Server network for your Virtual Machine channel are discussed more in [Section 5.7.4, “Logical Networks on a Single Oracle VM Server \(Intra-Server\)”](#).

Note that it is possible, and very likely, to have multiple networks with the Virtual Machine channel in one [Oracle VM Manager](#).

Virtual Machine networks necessarily make use of network bridging to allow virtual machines running on different Oracle VM Servers to be able to communicate. This network bridging is discussed in more detail in [Section 5.6.5.1, “Network Bridges”](#).

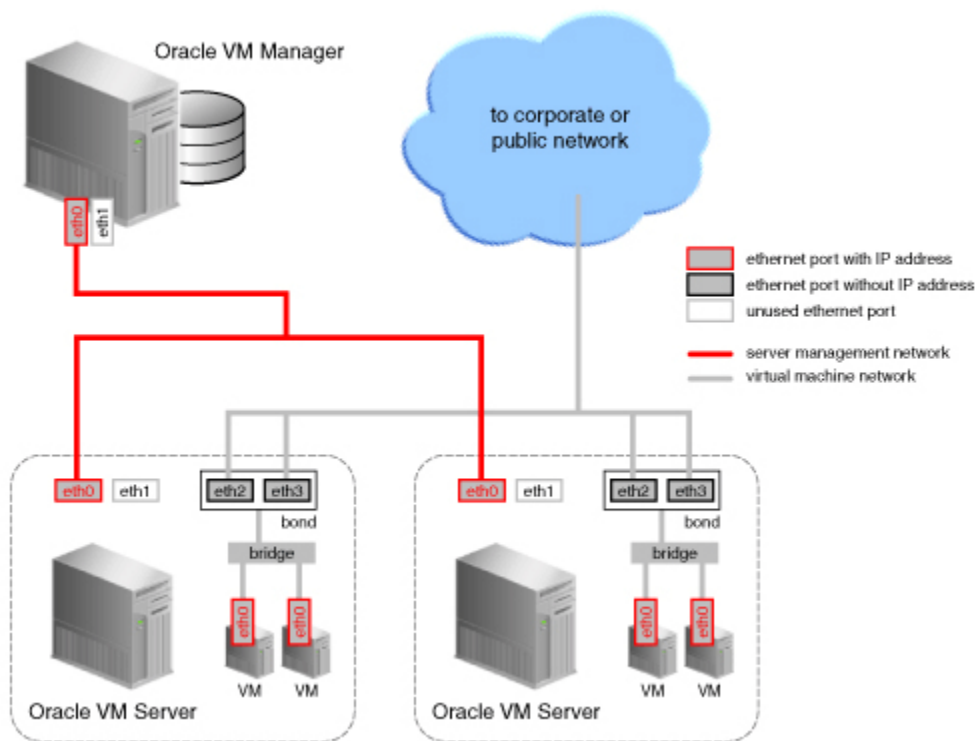
5.6.5.1 Network Bridges

When creating a network with the virtual machine channel, a bridge is created automatically on the port or bond added to the network for each [Oracle VM Server](#) participating in this network. All network packets generated by the virtual machines are sent to the bridge configured for the virtual machines' network. The bridge acts as a Layer 2 switch, and directs packets to other virtual machines running on the Oracle VM Server, or to the port or bond, if the packets' destination is outside of the Oracle VM Server.

Though each virtual machine deployed within a network is usually assigned an IP address, either static or assigned using DHCP, there is no need to configure an IP address for the bridge on the Oracle VM Servers. When configuring your Virtual Machine network, if you specify an IP address for the port or bond you selected for this network, it is assigned to the bridge. You can choose not to assign an IP address to

the selected port or bond. In this case, the bridge does not acquire an address but still functions as a Layer 2 switch.

Figure 5.7 Network bridge



In [Figure 5.7, “Network bridge”](#), two network ports are specified for the network with the virtual machine channel. Therefore, these ports should be configured as a bonded interface. Since this network is configured with the virtual machine channel, a bridge is automatically created on each Oracle VM Server in the network. Neither the bridge nor the ports in the virtual machine network, have IP addresses assigned to them, though you may assign IP addresses if you wish during network creation.

Bridges are only created for networks with the virtual machine channel.

5.7 Network Planning for an Oracle VM Deployment

Sufficient planning and preparation work, with regard to networking, can turn days of configuration for a large deployment into something that you can achieve in a matter of hours. In this section, we consider some of the steps that you might take to ensure that you are ready to set up networking with more ease.

Since the logical network constructs within Oracle VM are comprised by combining a variety of individual components, or building blocks, into whatever network infrastructure you need, it makes sense to take the time to identify each of these components and gather any information that may be required while you attempt to map these components into virtual constructs within Oracle VM Manager. You may find that maintaining a spreadsheet of this information helps you to prepare and properly plan for your deployment.

Before you begin, using the information provided in [Section 5.6, “How are Network Functions Separated in Oracle VM?”](#), you should determine how much network separation you require within your deployment and what different networks you want to make use of. Remember, that in a test or demonstration environment, it is perfectly acceptable to use a single network to cater to all of your networking requirements. In a production level deployment, it is desirable to provide as much network separation as possible to ensure

the best possible performance for the components that are used in an Oracle VM environment. On the other hand, depending on your networking infrastructure, the number of networks that you create may be limited by the hardware that you have at your disposal. With this in mind, prioritize the different networks based on your requirements and determine which networks may share functions.

Now identify the Oracle VM Servers that you intend to use within your deployment, along with the number of NICs available on each server. Ensure that you understand how they are cabled into your switching infrastructure. This helps to ensure that when you configure network ports in Oracle VM Manager, these ports are connected to your network infrastructure in the way that you expect. Plan how you intend to group servers into server pools. Remember that for each server pool you require a dedicated IP address to act as the server pool Virtual IP (VIP).

The next part of your planning involves selecting the type of network elements that you intend to use to construct your logical networks within Oracle VM for each server pool. Logical network types consist of any of the following:

- Networks with ports and/or bonds.
- Networks with VLANs only.
- Hybrid networks consisting of ports and bonds, and VLANs.
- Logical networks on a single server (Intra-Server).

Preparation for each network type can vary both in terms of complexity and in terms of content, since different networking components are involved in each case.

Regardless of the logical network type that you choose to use, you should bear in mind the following general networking rules:

- By default the management network interface after an initial installation is created as a network bond called *bond0*. This interface can have a static or dynamic IP address, but the IP address should remain constant. It is possible to change the management interface, but this requires careful planning as the change may impact other networks used in a deployment.
- DHCP can be used to assign IP addresses to any network interface within the deployment, however if you choose to use DHCP for the allocation of IP address information on any of the network interfaces other than the management network interface, you must ensure that a default route is not set for any interface other than the management network interface.
- You should never configure more than one interface on the same sub-network. If you do this, the first available interface is selected for all network traffic. If you intend to use multiple network interface cards on the same sub-network, you must set up a bond.

5.7.1 Networks with Ports and/or Bonds

If you cannot support VLANs in your switches, the number of logical networks available to you is immediately limited by the number of physical NICs within each server, and whether or not you choose to make use of network bonding for redundancy and performance.

If you choose to set up network bonds on your servers, you do this as part of your server configuration process, prior to defining any additional networks within Oracle VM. Configuring network bonds using the Oracle VM Manager Web Interface is discussed in [Bond Ports Perspective](#) in the *Oracle VM Manager User's Guide*.

If you are only using network ports without bonding, simply keep track of how many ports are available in each server and ensure that you understand how these are connected to your switches as these ports

have a one-to-one correlation with the logical networks that you are able to create. Make notes of the IP addressing and subnetworks that should be used for each port.

If you choose to use network bonding, you should identify which ports belong to which bonds you intend to create. In this case, each bond has a direct correlation with a logical network that you can create within Oracle VM. Once network bonding is enabled, the ports can no longer be used independently and your IP address and subnetwork information is specific to the network bond itself. Make notes of the IP addressing and subnetworks that should be used for each bond.

You should now be able to map the information that you have about servers, ports, bonds and IP addressing directly onto the different logical networks that you wish to create. For each logical network, make a note of the different network channels that you wish to support within each of these subnetworks. Ensure that the components that are used within the channel of the network are all physically connected to the same network.

With all of this information, you are ready to begin configuration. It is important to bear the following points in mind:

- All servers that you wish to include in a network must have already been discovered in Oracle VM Manager
- If you create a network with ports, these ports, located on the Oracle VM Servers that will participate in the network, cannot be part of an already existing network.
- If you are using network bonds, you must create these on each of your servers before you begin adding networks

5.7.2 Networks with VLANs Only

If your switching infrastructure supports VLANs and you have opted to make use of them, you are able to perform traffic separation across each VLAN regardless of the number of actual physical network ports that are available within each of your servers. Nonetheless, different VLAN segments may be available on different ports or bonds that you may have already configured.

Therefore, for each server, you should list the VLAN segment IDs on each port or bond. Make a note of the IP address and subnet information for each VLAN segment. If your network traffic is routed, also include the default gateway IP address here.

Once you have a list of ports or bonds and the VLAN segments that belong to them, you have the information that you need to define your VLAN Interfaces within Oracle VM. List the VLAN Interfaces that you intend to create for each server pool. Within this list, assign each VLAN Interface to the logical networks that you intend to create. For each logical network, list the channels that the network can support. Ensure that the components that are used within each channel of the network are all physically connected to the same network and that the VLAN segments on each NIC are configured correctly on your switches.

With all of this information, you are ready to begin configuration. It is important to bear the following points in mind:

- All servers that you wish to include in a network must have already been discovered in Oracle VM Manager
- If you are using network bonds, you must create these on each of your servers before you begin adding networks or creating VLAN Interfaces
- You must create VLAN Interfaces before you start adding networks
- When creating a VLAN Interface, you provide the following information:

- The port on the server where VLAN traffic for the VLAN ID or segment is routed.
- The VLAN ID that the VLAN Interface should handle.
- And, optionally, the IP address to assign to each port or VLAN interface.

Note that when actually creating VLAN Interfaces for a server, the Oracle VM Manager Web Interface provides the facility to generate multiple VLAN Interfaces for a range of VLAN IDs in a single step, reducing the amount of work involved in configuring each VLAN segment.

- When creating the network do not specify any ports or bonds to add to the network. Instead, add the appropriate VLAN interfaces to the network.

5.7.3 Hybrid Networks (VLANs with Ports and Bonds)

If your servers have multiple NICs and your switching infrastructure supports VLANs, there may be some scenarios where you need to set up a hybrid network that utilizes a combination of VLANs and network bonds or ports. Consider a situation where some of the servers within a pool are connected to a switch that is configured for VLANs, while another group of servers with multiple NICs is connected to a switch that does not offer VLAN support. In this situation, a network can be created that uses the VLAN Interfaces defined for the first set of servers, but uses ports or bonds for the second set of servers.

Since these network types are using a combination of technologies, they are also the most complex to plan for. Therefore, performing adequate preparation work is essential and can help avoid confusion when you are configuring networks within Oracle VM Manager.

Maintain a list of all of the Oracle VM Servers for each server pool. Ensure that the list includes the information for each port on each server. If the server is configured to use network bonding, list the bonds and include the actual physical NICs that are part of the bond.

Group together servers that are going to use VLANs and servers that are only using ports or bonds. For the servers that are using ports or bonds, ensure that the IP address and subnet information is listed. For servers using VLANs, you should list the VLAN segment IDs on each port or bond. Make a note of the IP address and subnet information for each VLAN segment. If your network traffic is routed, also include the default gateway IP address here.

For each logical network that you intend to create, list the channels that the network can support. Ensure that the components that are used within each channel of the network are all physically connected to the same network and that the VLAN segments on each port or bond are configured correctly on your switches. Also ensure that for each server not using VLAN technology, the cabling is correct for each network port and that the components within each channel are all connected to the same network.

With all of this information, you are ready to begin configuration. It is important to bear the following points in mind:

- All servers that you wish to include in a network must have already been discovered in Oracle VM Manager.
- If you are using network bonds, you must create these on each of your servers before you begin adding networks or creating VLAN Interfaces.
- You must create VLAN Interfaces before you start adding networks.
- When creating a VLAN Interface, you provide the following information:
 - The port on the server where VLAN traffic for the VLAN ID or segment is routed.

- The VLAN ID that the VLAN Interface should handle.
- And, optionally, the IP address to assign to each port or VLAN interface.

Note that when actually creating VLAN Interfaces for a server, the Oracle VM Manager Web Interface provides the facility to generate multiple VLAN Interfaces for a range of VLAN IDs in a single step, reducing the amount of work involved in configuring each VLAN segment.

- Ensure that servers that are not using VLAN Interfaces are added to the network and that the correct port or bond is added to the network.
- Finally for the servers using VLAN Interfaces, add the appropriate VLAN interfaces to the network.

5.7.4 Logical Networks on a Single Oracle VM Server (Intra-Server)

You may also create a network which is intended for a single server. This type of network allows communication between the virtual machines running on a single Oracle VM Server, and does not allow external network traffic. A computing environment made up of several virtual machines, where the virtual machines provide services to each other over the network, could benefit from this type of network, without requiring additional network ports on the Oracle VM Server. As the network traffic can never leave the server, it can be a high speed network. Virtual machines using this type of network cannot be moved or migrated to another Oracle VM Server unless the network configuration is removed from the virtual machine. This is known as an Intra-Server network.

Little additional planning or preparation for Intra-Server networks is required, however if you use networks of this type, you should decide which Oracle VM Servers you intend to create them on and understand that the virtual machines running on these servers cannot be moved or migrated easily.

5.8 Dealing with Failed Network Operations

Network configuration is a complex operation involving many different elements in the physical and logical environment. Many instructions are sent to the Oracle VM Servers in the process, and if a single instruction in a whole sequence goes wrong, the resulting state of the network configuration is unpredictable. To avoid badly or partly configured network objects, which become unusable, Oracle VM Manager has a mechanism in place that is triggered when a network operation fails: a network discovery is launched for each Oracle VM Server that participated in the operation, and the commands that completed successfully are reflected in the network model displayed in the Oracle VM Manager Web Interface.

The moment a network operation fails, an [event](#) is created. In the Oracle VM Manager Web Interface this event is displayed in the **Events** tab of each affected Oracle VM Server. When the subsequent automatic network discovery completes, the event is also automatically acknowledged. Note that if you acknowledge the event manually, the discovery operation is not cancelled. Jobs and events are discussed in more detail in [Chapter 8, Understanding Jobs, Events and Errors](#).



Note

The automatic network discovery is not instantaneous. The operations start as soon as the job fails, but could take some time to finish: from a few seconds to a couple of minutes. This will depend on how complicated the network configuration is and how many Oracle VM Servers are involved.

During network discovery, resources may be locked or may change to reflect the new state of the Oracle VM Servers. To avoid resource locking issues and further failing operations, it is recommended that you wait for the discovery operation to

complete before you resume the reconfiguration of the network. Check the events for each Oracle VM Server to know the status of the operation.

In case your network configuration returns a failed job, you have to manually go through all the physical and logical network elements involved and make the necessary changes one by one in Oracle VM Manager. Typical network elements include: network interfaces, Ethernet ports, bond ports, VLAN interfaces and IP addresses. The amount of manual reconfiguration depends on the complexity of your network configuration and the number of Oracle VM Servers involved.

Chapter 6 Understanding Server Pools and Oracle VM Servers

Table of Contents

6.1 How are Oracle VM Servers Added?	64
6.2 What are Server Roles?	65
6.3 How is Maintenance Performed on an Oracle VM Server?	65
6.4 Configuring NTP for Oracle VM Servers.	66
6.5 Rebooting and Changing Power State of Oracle VM Servers	67
6.6 What are Server Pools used for in Oracle VM?	67
6.7 How are Server Pools Created?	68
6.8 How do Server Pool Clusters Work?	69
6.8.1 Clustering for x86 Server Pools	69
6.8.2 Clustering for SPARC Server Pools	74
6.9 Unclustered Server Pools	74
6.10 How does High Availability (HA) Work?	76
6.11 What are Server Pool Policies?	78
6.11.1 Distributed Resource Scheduler (DRS)	79
6.11.2 Distributed Power Management (DPM)	79
6.11.3 DRS/DPM Network Policies	80
6.12 What are Anti-Affinity Groups?	80
6.13 What are Server Processor Compatibility Groups?	80

Oracle VM Servers are very much the work-horses in a Oracle VM environment. It is important to understand their purpose within a deployment, how they are added to a running environment and how they are maintained. A brief overview of Oracle VM Server, its components and its general relationship to other Oracle VM entities is provided in [Section 2.2, “What is Oracle VM Server?”](#).

Some discussion about how servers are added to a deployment is provided in [Section 6.1, “How are Oracle VM Servers Added?”](#). Details on server maintenance are provided in [Section 6.3, “How is Maintenance Performed on an Oracle VM Server?”](#).

Oracle VM Servers are grouped together to form [server pools](#). These server pools are a [domain](#) of physical and virtual resources to host virtual machines, perform virtual machine [migration](#), [high availability](#) (HA), and so on. Server pools are discussed in [Section 6.6, “What are Server Pools used for in Oracle VM?”](#).

Server pools can take advantage of Oracle VM's clustering facility to achieve [HA](#). Implementation of a clustered server pool is very easy using Oracle VM Manager, but it is important to gain a deeper understanding of the technologies involved and how clustering is facilitated. A very thorough description of clustered server pools is provided in [Section 6.8, “How do Server Pool Clusters Work?”](#)

While the most common deployment strategy tends to take advantage of Oracle VM's clustering facility, it is possible to configure a server pool to run in an unclustered arrangement. This approach is discussed in [Figure 6.2, “Unclustered Server Pools Using Only NFS Storage”](#).

While clustering helps to provide HA for the servers within a server pool, Oracle VM allows you to configure whether or not HA should be applied to individual virtual machines. We take a closer look at HA in [Section 6.10, “How does High Availability \(HA\) Work?”](#)

Oracle VM provides additional features specific to how servers and server pools are configured. These include the ability to optimize resource usage or power consumption (see [Section 6.11, “What are Server Pool Policies?”](#)); the ability to define which individual servers virtual machines should be allowed to run on (see [Section 6.12, “What are Anti-Affinity Groups?”](#)); and the ability to create groups of servers that have

compatible processor types to facilitate live migration of virtual machines (see [Section 6.13, “What are Server Processor Compatibility Groups?”](#)).

6.1 How are Oracle VM Servers Added?

After Oracle VM Server has been installed, it is in an *unowned* state and has no relationship to any single Oracle VM deployment. For an Oracle VM Server to be used, it must be added to an Oracle VM Manager instance that can take ownership of the server. This is a process of server [discovery](#).

Server discovery is a straightforward process performed from within Oracle VM Manager, using one of the provided interfaces. The Oracle VM Manager uses a provided IP address or hostname and password to attempt a connection to the Oracle VM Agent running on Oracle VM Server. When authenticated, the server is added to the Oracle VM Manager where various configuration actions such as adding networks or storage, grouping within a server pool, or server updates can be performed. The actions required to discover servers within the Oracle VM Manager Web Interface are discussed in [Discover Servers](#) in the [Oracle VM Manager User's Guide](#).

Each Oracle VM Manager instance is assigned a UUID that identifies it to the Oracle VM Servers that it discovers. The Oracle VM Agent running on an Oracle VM Server stores the UUID of the Oracle VM Manager that discovers and takes ownership of it. The UUID is stored on both Oracle VM Manager and the Oracle VM Server. On Oracle VM Manager you can find the UUID in `/u01/app/oracle/ovm-manager-3/.config`. On Oracle VM Server, the UUID of the owning Oracle VM Manager instance is stored in a Berkeley DB database file that you are able to dump the contents for by doing the following as the root user on the Oracle VM Server system:

```
# cd /etc/ovs-agent/db
# ovs-agent-db dump_db server
```

In the case of an Oracle VM Server that is in an unowned state, Oracle VM Manager automatically takes ownership of the server upon discovery. However, for servers that report an owned state, Oracle VM Manager cannot take ownership of the server until the Oracle VM Manager that already has ownership relinquishes it. This is important as configuration actions performed by two separate Oracle VM Manager instances could result in conflicts. Therefore, it is not possible to perform any server configuration until an Oracle VM Manager instance actually has ownership of the server. Taking ownership of a server using the Oracle VM Manager Web Interface is described in [Edit Server](#) in the [Oracle VM Manager User's Guide](#).

In addition to the UUID of the Oracle VM Manager that owns a server, each Oracle VM Server also stores a large amount of configuration data locally. This means that if the Oracle VM Manager instance experiences downtime, the environment can continue to function normally. It also means that if an Oracle VM Manager instance is entirely replaced, a fair portion of the configuration can be loaded directly from each newly discovered server. Typical examples of this include networking and clustering information as well as Storage Connections. While this may provide some consolation, information such as the aliases for different elements is not stored on each server, so recovery of an Oracle VM Manager instance in this way is not recommended. A proper backup strategy should be followed instead. It is also important to realize that for this to work, the new Oracle VM Manager instance must have the same UUID as the original instance.

This has implications for server discovery. If an Oracle VM Server has been under the ownership of a separate Oracle VM Manager instance, its existing networking, clustering and storage configuration is not automatically loaded into Oracle VM Manager at the same time. Only the fundamental information about the server is loaded into Oracle VM Manager. This is known as a *partial-discovery*. Full discovery can only be performed once ownership of the server is under the control of the Oracle VM Manager instance. In a case where an alternate instance of Oracle VM Manager has ownership of a server, it would need to relinquish ownership first. This can only be done for servers that are not part of a server pool and that do not have a repository presented to them. If a server is partially discovered by an Oracle VM Manager

instance and the original Oracle VM Manager instance releases ownership, the Oracle VM Manager instance that wishes to take ownership must perform a full rediscovery of the server.

6.2 What are Server Roles?

Not all Oracle VM Servers within a server pool may be required for the purpose of running virtual machines. Oracle VM Servers are also required to perform *utility* functions, including performing actions on storage repositories on behalf of Oracle VM Manager. While all Oracle VM Servers can be configured to be capable of performing both functions, Oracle VM provides the option to separate servers within a server pool based on their functional roles. These roles are defined as follows:

- **VM Server role:** The [VM Server](#) role enables an Oracle VM Server to run virtual machines.
- **Utility Server role:** A [Utility Server](#) will be favored to do operations other than hosting virtual machines, for example, importing virtual machine templates and assemblies, creating virtual machine templates from assemblies, and creating repositories. If no Utility Servers are available, a non-utility server is chosen to do the work.

The role of a server can be configured within Oracle VM Manager once the server is part of a server pool. At least one server within the server pool must be assigned the VM Server role, or no virtual machines are able to run in the server pool. When you add an Oracle VM Server to a server pool it has both the Utility Server role and the VM Server role automatically selected by default. The VM Server role is required to run virtual machines. The Utility Server role is not required, it is advisory only.

Since Oracle VM Servers are capable of being configured for either or both roles, role separation is used for performance purposes. For instance, in the case of the import of a virtual machine template or assembly, it is preferable that there is a server dedicated to this task, to reduce the impact of such a task on the running of virtual machines within the server pool.

You cannot update the VM Server role for an Oracle VM Server if the server is not in a server pool or if the Oracle VM Server has any virtual machines, running or not running. You must own the Oracle VM Server to change the VM Server role. You can change the Utility Server role anytime as long as you own the Oracle VM Server.

More information about the configuration of server roles can be found at [Edit Server](#) in the [Oracle VM Manager User's Guide](#).

6.3 How is Maintenance Performed on an Oracle VM Server?

Once an Oracle VM Server is installed and discovered, an administrator has no need to directly access the system unless directed to do so by Oracle Support. All maintenance and configuration is handled directly from the Oracle VM Manager instance that has ownership of the server. This makes system administration such as performing server updates, configuring networks, attaching storage, configuring clustering and the management of virtual machines much easier for a systems administrator and reduces the security implications of provisioning access to multiple systems within a deployment.

Oracle VM Manager is able to communicate directly with each Oracle VM Server via the Oracle VM Agent installed and running on the server. Communications are secured using TLS and authentication against the Oracle VM Agent is required. The password used for the Oracle VM Agent is configured at install time, and is required by Oracle VM Manager to perform server discovery. Once Oracle VM Manager has taken ownership of the Oracle VM Server, authentication of the Oracle VM Manager against the Oracle VM Agent is achieved using certificates to improve the security of the Oracle VM Agent. This ensures that only the Oracle VM Manager that has ownership of an Oracle VM Server is authorized to perform any action on the server. The Oracle VM Agent is responsible for performing all configuration changes and maintenance initiated from the Oracle VM Manager.

Oracle VM Server software updates and upgrades are handled by configuring one or more server update repositories for the Oracle VM Servers from within Oracle VM Manager. One or more update repositories can be configured for x86-based Oracle VM Servers, or for SPARC-based servers. These repositories are available to all Oracle VM Servers with that hypervisor type. You can also override these repositories at the server pool level. When a repository has been configured, Oracle VM Manager reports whether or not updates are available for each Oracle VM Server in the environment. Updating a server can be triggered from Oracle VM Manager. See [Update Server](#) in the *Oracle VM Manager User's Guide* for more information about Oracle VM Server updates.

During an update, the Oracle VM Server is put into *maintenance mode*. If the server is part of a clustered server pool, any virtual machines running on the server are automatically migrated to an alternate server within the server pool. The update is performed without affecting existing services. If the server requires a reboot after an upgrade, this is performed automatically. Once the server has rebooted, it rejoins the cluster and resumes its normal functions. See [Edit Server](#) in the *Oracle VM Manager User's Guide* for more information about maintenance mode.

Additional networking configuration is also handled from within Oracle VM Manager. During installation, a primary network is defined to allow the server to connect to the Server Management network. From this point onward, any further network configuration is performed from within Oracle VM Manager. Networking for Oracle VM is described in more detail in [Chapter 5, Understanding Networks](#).

The process of defining Storage Connections and mount points is also handled from within Oracle VM Manager. These connections and how they are created are discussed in more detail in [Chapter 3, Understanding Storage](#).

It should be clear that other than the initial installation required for an Oracle VM Server, all administrative tasks related to server maintenance and configuration are handled by Oracle VM Manager. This centralized administration not only eases the management of servers, but also improves security and reduces the likelihood of configuration errors that could result in server downtime.

6.4 Configuring NTP for Oracle VM Servers.

In environments where data is constantly shared between systems, it is important that time is properly synchronized. This is particularly important within Oracle VM where clustering is frequently used and where virtual machines can move between Oracle VM Servers as required. The most common approach to this requirement is to configure an NTP server to run on the same host where you install Oracle VM Manager. Each Oracle VM Server can then use the Oracle VM Manager host as an NTP server. You can find out more about the installation and configuration of NTP in [Configure the NTP Service on the Oracle VM Manager Host](#) in the *Oracle VM Installation and Upgrade Guide*.

By default, Oracle VM Servers that are not under the ownership of any Oracle VM Manager instance, are configured to point to localhost (127.127.1.0) for NTP. Once an Oracle VM Server has been discovered and ownership is taken inside of Oracle VM Manager, it becomes possible to configure the NTP servers that should be used for that Oracle VM Server. The Oracle VM Manager allows you to configure the NTP servers that should be used for any Oracle VM Server directly within the server configuration in the Oracle VM Manager Web Interface. See [Edit Server](#) in the *Oracle VM Manager User's Guide* for more information on how to configure NTP for each server. If you need to batch edit NTP for a large number of servers, the preferred method is to use the Oracle VM Manager Command Line Interface or the Oracle VM Web Services API. See [edit Server](#) for an example of how to edit an Oracle VM Server to configure NTP using the Oracle VM Manager Command Line Interface.

Removing all NTP servers from an Oracle VM Server configuration results in the configuration reverting to its default state, where the Oracle VM Server is configured to point to localhost as its NTP server. This is the same behavior that is applied when an Oracle VM Server is released from ownership of an Oracle VM Manager instance. Therefore, whenever an Oracle VM Server is newly discovered and the Oracle VM

Manager instance takes ownership, the NTP configuration on the Oracle VM Server is always configured to point to the localhost.

6.5 Rebooting and Changing Power State of Oracle VM Servers

Oracle VM Servers can be rebooted from Oracle VM Manager, using the web user interface or the command line interface. See [Restart Server](#) in the [Oracle VM Manager User's Guide](#). A reboot has different consequences depending on the hardware architecture that you the Oracle VM Server is running on.

On x86 hardware, a reboot causes the entire system to restart. This means that depending on the configuration of your server pool and whether the server has been put into maintenance mode first, all virtual machines that are running on the server are either migrated to an alternative server or stopped, before the server is rebooted.

On SPARC hardware, a reboot is only applied to the control domain. This mean that any virtual machines running on the SPARC-based Oracle VM Server continue to run while the control domain is rebooted. However, network access and disk I/O are blocked while the control domain is rebooting, unless a second service domain or shadow domain is available to facilitate this activity. See [Configuring a Secondary Service Domain on SPARC](#) in the [Oracle VM Administrator's Guide](#). Once the control domain has finished rebooting, network access and disk I/O are restored for the running virtual machines and activity resumes as normal. If you are using server pool clustering, virtual machines may be migrated to an alternative server if the reboot of the control domain takes too long or does not complete for some reason.

It is worthwhile noting that a power cycle has the same effect on both x86 and SPARC servers. The entire system is restarted. All virtual machines are either stopped or migrated, depending on whether or not clustering is enabled for the server pool that the server belongs to. Killing an Oracle VM Server from within Oracle VM Manager has an equivalent effect to a power cycle. See [Kill Server](#) in the [Oracle VM Manager User's Guide](#) for more information on this.



Warning

Standby mode is not supported for Oracle VM Server. Do not attempt to put an Oracle VM Server into standby mode. Setting a server's power state to standby may cause the server to freeze.

6.6 What are Server Pools used for in Oracle VM?

A [server pool](#) consists of one or more [Oracle VM Servers](#), and represents a logical grouping of the servers where a particular set of virtual machines can run. It is a requirement of all server pools that the Oracle VM Servers within them share the same CPU architecture. All servers within a server pool must be in the same physical location. Stretching server pools across geographical locations is not supported.

Oracle VM deployments can vary in design. It is equally plausible that one deployment may only use a single server pool for all of its virtualization requirements, while another deployment may consist of several server pools either catering for different hardware platforms or for complete virtual machine separation.

Server pools help to provide horizontal scalability. If you find that a server pool does not have sufficient resources, such as CPU or memory, to run the virtual machines, you can expand the server pool by adding more Oracle VM Servers. This process is discussed in [Edit Server Pool](#) in the [Oracle VM Manager User's Guide](#). In this way, a server pool can be described as the set of resources available to a group of virtual machines.

Therefore, before creating a server pool, it is useful to consider how many Oracle VM Servers are to be included in the server pool, and which function(s) or role(s) each Oracle VM Server is to perform. See [Section 6.2, "What are Server Roles?"](#) for more information of server functions and roles. Each virtual

machine running in a server pool requires resources to be available to it, such as CPU, network, and memory. You should size your server pool accordingly.

Oracle VM's usual deployment architecture utilizes server pools, with shared access to storage across Oracle VM Servers in the server pool. Virtual machines are stored on the shared storage and loaded onto any one of the Oracle VM Servers to balance the workloads of the server pool.

In a deployment that uses shared storage that is accessible to all of the Oracle VM Servers in the server pool, many other facilities are available to ensure [high availability](#) and excellent failover. A server pool can be configured as a cluster, so that virtual machines are automatically live-migrated between servers in the event of server downtime.

Since the virtual machines are not bound to any specific Oracle VM Server in the server pool, virtual machines are not prevented from starting up simply because an individual Oracle VM Server happens to be down for maintenance or otherwise unavailable at the time. Further, options are provided to specify the start policy for the virtual machines in the server pool. The start policy can implement a load-balancing algorithm that assures that a virtual machine is only started on the Oracle VM Server with the most resources available. Load balancing is achieved using the same algorithms used for [Distributed Power Management \(DPM\)](#) and for the [Distributed Resource Scheduler \(DRS\)](#). Load-balancing further helps assure the maximum aggregate performance from the server pool.

When you create a server pool in Oracle VM, you specify:

- Server pool name and description.
- A virtual IP address.
- Whether or not to activate the cluster.
- A server pool file system for the global heartbeat and other cluster information.

The server pool name and description are used as friendly identifiers within Oracle VM Manager to reference different server pools and to understand their purpose.

The virtual IP address is used by Oracle VM Manager to communicate with the server that is designated as the Master in the server pool. If the master changes, the virtual IP address is transferred to the new Master, ensuring that Oracle VM Manager continues to communicate with the Master server. It is possible to manually configure which Oracle VM Server is designated as the Master server in the server pool using the Oracle VM Manager Web Interface or Oracle VM Manager Command Line Interface. See [Edit Server Pool](#) in the [Oracle VM Manager User's Guide](#) for more information on setting the Master server in the server pool.

If you opt to enable clustering, the servers in the server pool are clustered and all of the configuration steps required are performed automatically through Oracle VM Manager. In this case, it is necessary for you to configure the server pool file system that is to be used to store cluster information. See [Section 6.8, "How do Server Pool Clusters Work?"](#) for more information. If you opt for a server pool without clustering, see [Section 6.9, "Unclustered Server Pools"](#).

6.7 How are Server Pools Created?

A [server pool](#) consists of at least one, but usually multiple Oracle VM Servers. All Oracle VM Servers in a server pool should have CPUs in the same CPU family and of the same type. If they are not in the same CPU family and type, some operations such as [live migration](#) may fail. Though the CPUs should be in the same CPU family, they may have differing configurations, such as different number of cores. Other hardware components on the [host computer](#) may also differ, such as the amount of RAM, the number and size of disk drives, and so on.

Although the host computers may have differing configurations, Oracle recommends that all Oracle VM Servers in a server pool are identical. Oracle VM Manager contains rules for [processor compatibility groups](#). If live migration is attempted between incompatible processors, an error message is displayed.


Before creating a server pool, you must have:

- IP addresses or hostnames for the Oracle VM Servers.
- IP address to use as the virtual IP address that is used to facilitate communication between Oracle VM Manager and the server pool via the designated Master server.
- Password to access the Oracle VM Agent installed on the Oracle VM Server(s).

**Note**

The Oracle VM Agent password should be the same on each Oracle VM Server in a server pool. Once an Oracle VM Server is under the ownership of an Oracle VM Manager instance, password authentication is replaced with certificate-based authentication to improve the security of the Oracle VM Agent.

A clustered server pool must have a dedicated file system (either a NAS export, or a LUN) to use as the server pool's file system. Oracle recommends that you create this storage with a size of at least 10 GB. If you are creating a SPARC-based server pool, only an NFS file system is supported for the server pool file system.

The **Create Server Pool**  icon is available on the **Servers and VMs** tab within the Oracle VM Manager Web Interface and is used to open the **Create Server Pool** wizard that guides you through the server pool creation process.

For information on creating a server pool, see [Create Server Pool](#) in the [Oracle VM Manager User's Guide](#).

6.8 How do Server Pool Clusters Work?

The implementation of [server pool](#) clustering within Oracle VM differs depending on whether the [Oracle VM Servers](#) used in the server pool are based on an x86 or SPARC architecture, however the behavior of a server pool cluster and the way in which it is configured within Oracle VM Manager is largely seamless regardless of the platform used. This means that server pool clustering is handled automatically by Oracle VM Manager as soon as you enable it for a server pool, as long as the necessary requirements are met to allow clustering to take place.

In this section we discuss how server pool clusters work for each of the different architectures, and describe the requirements for clustering to be enabled.

6.8.1 Clustering for x86 Server Pools

Oracle VM works in concert with Oracle [OCFS2](#) to provide shared access to server pool resources residing in an OCFS2 file system. This shared access feature is crucial in the implementation of [high availability](#) (HA) for virtual machines running on x86 Oracle VM Servers that belong to a server pool with clustering enabled.

OCFS2 is a cluster file system developed by Oracle for Linux, which allows multiple nodes (Oracle VM Servers) to access the same disk at the same time. OCFS2, which provides both performance and [HA](#), is used in many applications that are cluster-aware or that have a need for shared file system facilities. With Oracle VM, OCFS2 ensures that Oracle VM Servers belonging to the same server pool access and modify resources in the shared repositories in a controlled manner.

The OCFS2 software includes the core file system, which offers the standard file system interfaces and behavioral semantics and also includes a component which supports the shared disk cluster feature.

The shared disk component resides mostly in the kernel and is referred to as the O2CB cluster stack. It includes:

- A disk heartbeat to detect live servers.
- A network heartbeat for communication between the nodes.
- A Distributed Lock Manager (DLM) which allows shared disk resources to be locked and released by the servers in the cluster.

OCFS2 also offers several tools to examine and troubleshoot the OCFS2 components. For detailed information on OCFS2, see the OCFS2 documentation at:

<http://oss.oracle.com/projects/ocfs2/documentation/>

Oracle VM decouples storage repositories and clusters so that if a storage repository is taken off-line, the cluster is still available. A loss of one heartbeat device does not force an Oracle VM Server to self fence.

When you create a server pool, you have a choice to activate the cluster function which offers these benefits:

- Shared access to the resources in the repositories accessible by all Oracle VM Servers in the cluster.
- Protection of virtual machines in the event of a failure of any Oracle VM Server in the server pool.

You can choose to configure the server pool cluster and enable HA in a server pool, when you create or edit a server pool within Oracle VM Manager. See [Create Server Pool](#) and [Edit Server Pool](#) in the [Oracle VM Manager User's Guide](#) for more information on creating and editing a server pool.

During server pool creation, the server pool file system specified for the new server pool is accessed and formatted as an OCFS2 file system. This formatting creates several management areas on the file system including a region for the global disk heartbeat. Oracle VM formats the server pool file system as an OCFS2 file system whether the file system is accessed by the Oracle VM Servers as an NFS share, a FC LUN or iSCSI LUN. See [Section 3.8, "How is Storage Used for Server Pool Clustering?"](#) for more information on how storage is used for the cluster file system and the requirements for a stable cluster heartbeat function.

As Oracle VM Servers are added to a newly created server pool, Oracle VM:

1. Selects a Master Oracle VM Server.
2. Configures the Virtual IP address selected during pool creation as a virtual network interface on top of the management interface for the Master Oracle VM Server.
3. Creates the cluster configuration file and the cluster time-out file.
4. Pushes the configuration files to all Oracle VM Servers in the server pool.
5. Starts the cluster.

On each Oracle VM Server in the cluster, the cluster configuration file is located at `/etc/ocfs2/cluster.conf`, and the cluster time-out file is located at `/etc/sysconfig/o2cb`. Cluster timeout can be configured during server pool creation. The recommended approach to setting timeout values is to use the functionality provided within the Oracle VM Manager when creating or editing a server pool. See [Create Server Pool](#) and [Edit Server Pool](#) in the [Oracle VM Manager User's Guide](#) for more information on setting the timeout value.

Starting the cluster activates several services and processes on each of the Oracle VM Servers in the cluster. The most important processes and services are discussed in [Table 6.1, "Cluster services"](#).

Table 6.1 Cluster services

Service	Description
o2net	The o2net process creates TCP/IP intra-cluster node communication channels on port 7777 and sends regular keep-alive packages to each node in the cluster to validate if the nodes are alive. The intra-cluster node communication uses the network with the Cluster Heartbeat role. By default, this is the Server Management network. You can however create a separate network for this function. See Section 5.6, “How are Network Functions Separated in Oracle VM?” for information about the Cluster Heartbeat role. Make sure the firewall on each Oracle VM Server in the cluster allows network traffic on the heartbeat network. By default, the firewall is disabled on Oracle VM Servers after installation.
o2hb-diskid	The server pool cluster also employs a disk heartbeat check. The o2hb process is responsible for the global disk heartbeat component of cluster. The heartbeat feature uses a file in the hidden region of the server pool file system. Each pool member writes to its own block of this region every two seconds, indicating it is alive. It also reads the region to maintain a map of live nodes. If a server pool member's block is no longer updated, the Oracle VM Server is considered dead. If an Oracle VM Server dies, the Oracle VM Server is fenced. Fencing forcefully removes dead members from the server pool to make sure active pool members are not obstructed from accessing the fenced Oracle VM Server's resources.
o2cb	The o2cb service is central to cluster operations. When an Oracle VM Server boots, the o2cb service starts automatically. This service must be up for the mount of shared repositories to succeed.
ocfs2	The OCFS2 service is responsible for the file system operations. This service also starts automatically.
ocfs2_dlm and ocfs2_dlmfs	<p>The DLM modules (ocfs2_dlm, ocfs2_dlmfs) and processes (user_dlm, dlm_thread, dlm_wq, dlm_reco_thread, and so on) are part of the Distributed Lock Manager.</p> <p>OCFS2 uses a DLM to track and manage locks on resources across the cluster. It is called distributed because each Oracle VM Server in the cluster only maintains lock information for the resources it is interested in. If an Oracle VM Server dies while holding locks for resources in the cluster, for example, a lock on a virtual machine, the remaining Oracle VM Servers in the server pool gather information to reconstruct the lock state maintained by the dead Oracle VM Server.</p>

**Warning**

Do not manually modify the cluster configuration files, or start and stop the cluster services. Oracle VM Manager automatically starts the cluster on Oracle VM Servers that belong to a server pool. Manually configuring or operating the cluster may lead to cluster failure.

When you create a repository on a physical disk, an OCFS2 file system is created on the physical disk. This occurs for local repositories as well. The resources in the repositories, for example, virtual machine configuration files, virtual disks, ISO files, templates and assemblies, can then be shared safely across the server pool. When a server pool member stops or dies, the resources owned by the departing server are recovered, and the change in status of the server pool members is propagated to all the remaining Oracle VM Servers in the server pool.

Figure 6.1, “Server Pool clustering with OCFS2 features” illustrates server pool clustering, the disk and network heartbeats, and the use of the DLM feature to lock resources across the cluster.

Figure 6.1 Server Pool clustering with OCFS2 features

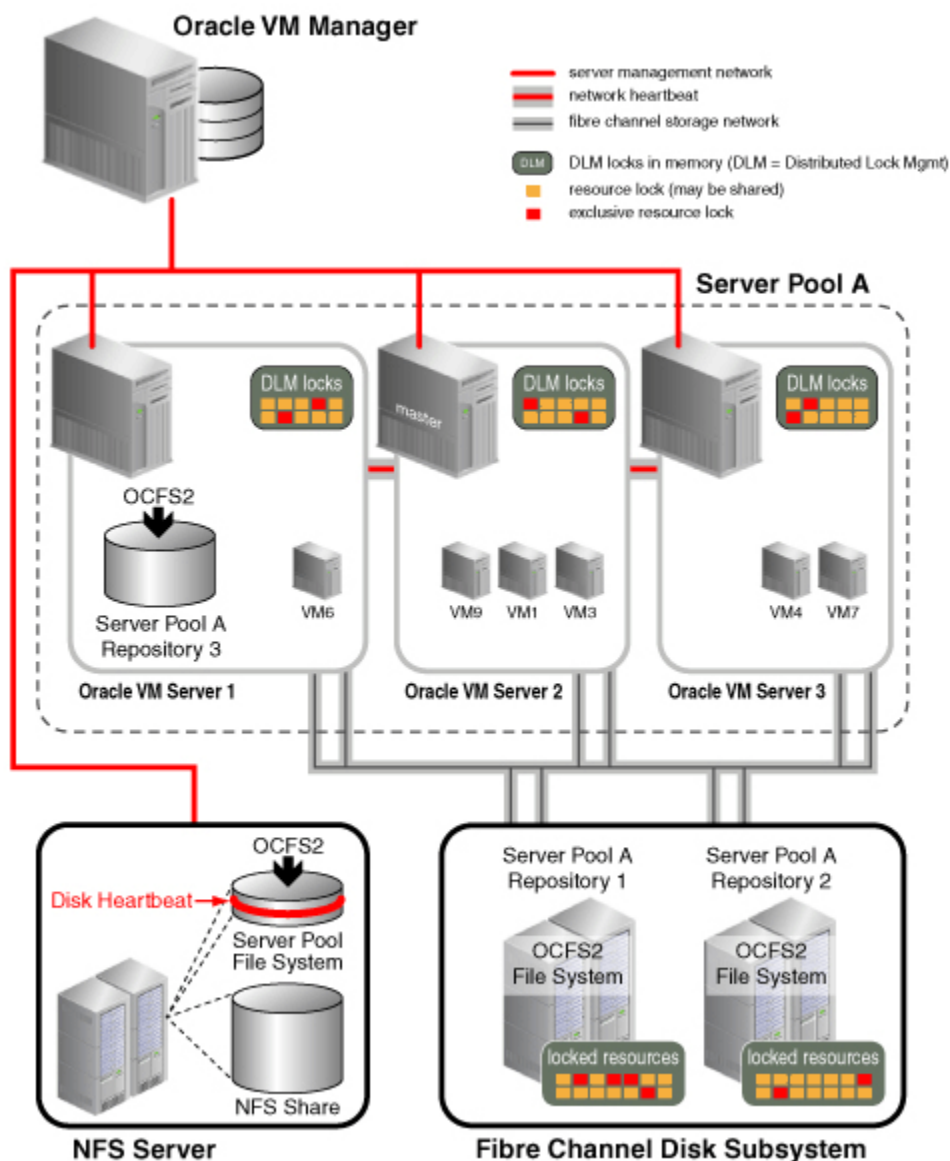


Figure 6.1, “Server Pool clustering with OCFS2 features” represents a server pool with three Oracle VM Servers. The server pool file system associated with this server pool resides on an NFS share. During server pool creation, the NFS share is accessed, a disk image is created on the NFS share and the disk image is formatted as an OCFS2 file system. This technique allows all Oracle VM Server pool file systems to be accessed in the same manner, using OCFS2, whether the underlying storage element is an NFS share, an iSCSI LUN or a Fibre Channel LUN.

The network heartbeat, which is illustrated as a private network connection between the Oracle VM Servers, is configured before creating the first server pool in your Oracle VM environment. After the server pool is created, the Oracle VM Servers are added to the server pool. At that time, the cluster configuration is created, and the cluster state changes from *off-line* to *heartbeating*. Finally, the server pool file system is mounted on all Oracle VM Servers in the cluster and the cluster state changes from *heartbeating* to

DLM ready. As seen in [Figure 6.1, “Server Pool clustering with OCFS2 features”](#), the heartbeat region is global to all Oracle VM Servers in the cluster, and resides on the server pool file system. Using the network heartbeat, the Oracle VM Servers establish communication channels with other Oracle VM Servers in the cluster, and send keep-alive packets to detect any interruption on the channels.

For each newly added repository on a physical storage element, an OCFS2 file system is created on the repository, and the repository is usually presented to all Oracle VM Servers in the pool. [Figure 6.1, “Server Pool clustering with OCFS2 features”](#) shows that Repository 1 and Repository 2 are accessible by all of the Oracle VM Servers in the pool. While this is the usual configuration, it is also feasible that a repository is accessible by only one Oracle VM Server in the pool. This is indicated in the figure by Repository 3, which is accessible by Oracle VM Server 1 only. Any virtual machine whose resources reside on this repository cannot take advantage of the high availability feature afforded by the server pool.

Note that repositories built on NFS shares are not formatted as OCFS2 file systems. See [Section 3.9, “Where are Virtual Machine Resources Located?”](#) for more information on repositories.

[Figure 6.1, “Server Pool clustering with OCFS2 features”](#) shows several virtual machines with resources in shared Repositories 1 and 2. As virtual machines are created, started, stopped, or [migrated](#), the resources for these virtual machines are locked by the Oracle VM Servers needing these resources. Each Oracle VM Server ends up managing a subset of all the locked resources in the server pool. A resource may have several locks against it. An exclusive lock is requested when anticipating a write to the resource while several read-only locks can exist at the same time on the same resource. Lock state is kept in memory on each Oracle VM Server as shown in the diagram. The distributed lock manager (DLM) information kept in memory is exposed to user space in the synthetic file system called dlmfs, mounted under /dlm. If an Oracle VM Server fails, its locks are recovered by the other Oracle VM Servers in the cluster and virtual machines running on the failed Oracle VM Server are restarted on another Oracle VM Server in the cluster. If an Oracle VM Server is no longer communicating with the cluster via the heartbeat, it can be forcibly removed from the cluster. This is called fencing. An Oracle VM Server can also fence itself if it realizes that it is no longer part of the cluster. The Oracle VM Server uses a machine reset to fence. This is the quickest way for the Oracle VM Server to rejoin the cluster.

6.8.1.1 Troubleshooting Cluster-related Problems for x86 Server Pools

There are some situations where removing an Oracle VM Server from a server pool may generate an error. Typical examples include the situation where an OCFS2-based repository is still presented to the Oracle VM Server at the time that you attempt to remove it from the server pool, or if the Oracle VM Server has lost access to the server pool file system or the heartbeat function is failing for that Oracle VM Server. The following list describes steps that can be taken to handle these situations.

- Make sure that there are no repositories presented to the server when you attempt to remove it from the server pool. If this is the cause of the problem, the error that is displayed usually indicates that there are still OCFS2 file systems present. See [Present or Unpresent Repository](#) in the [Oracle VM Manager User's Guide](#) for more information.
- If a pool file system is causing the remove operation to fail, other processes might be working on the pool file system during the unmount. Try removing the Oracle VM Server at a later time.
- In a case where you try to remove a server from a clustered server pool on a newly installed instance of Oracle VM Manager, it is possible that the file server has not been refreshed since the server pool was discovered in your environment. Try refreshing all storage and all file systems on your storage before attempting to remove the Oracle VM Server.
- In the situation where the Oracle VM Server cannot be removed from the server pool because the server has lost network connectivity with the rest of the server pool, or the storage where the server pool file system is located, a critical event is usually generated for the server in question. Try acknowledging

any critical events that have been generated for the Oracle VM Server in question. See [Events Perspective](#) in the [Oracle VM Manager User's Guide](#) for more information. Once these events have been acknowledged you can try to remove the server from the server pool again. In most cases, the removal of the server from the server pool succeeds after critical events have been acknowledged, although some warnings may be generated during the removal process. Once the server has been removed from the server pool, you should resolve any networking or storage access issues that the server may be experiencing.

- If the server is still experiencing trouble accessing storage and all critical events have been acknowledged and you are still unable to remove it from the server pool, try to reboot the server to allow it to rejoin the cluster properly before attempting to remove it again.
- If the server pool file system has become corrupt for some reason, or a server still contains remnants of an old stale cluster, it may be necessary to completely erase the server pool and reconstruct it from scratch. This usually involves performing a series of manual steps on each Oracle VM Server in the cluster and should be attempted with the assistance of Oracle Support. More information on this procedure is available on My Oracle Support in [KM Note 1615376.1](#) at:

<https://support.us.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=1615376.1>

6.8.2 Clustering for SPARC Server Pools

Since the Oracle [OCFS2](#) file system caters for Linux and not for Solaris, SPARC server pools are unable to use this file system to implement clustering functionality. Therefore, clustering for SPARC server pools cannot be implemented on physical disks and are limited to using NFS storage to host the cluster file system. Clustering for SPARC server pools relies on an additional package, which must be installed in the control domain for each [Oracle VM Server](#) in the server pool. This package contains a distributed lock manager (DLM) that is used to facilitate the cluster. Installation of this package is described in more detail in [Installing DLM for SPARC Server Pool Clusters](#) in the [Oracle VM Installation and Upgrade Guide](#).

The DLM package used to achieve clustering for SPARC server pools is a port of the tools that are used within OCFS2 on Linux, but exclude the actual OCFS2 file system itself. The DLM package includes:

- A disk heartbeat to detect live servers.
- A network heartbeat for communication between the nodes.
- A Distributed Lock Manager (DLM) which allows shared disk resources to be locked and released by the servers in the cluster.

The only major difference between clustering on SPARC and on x86 is that there are limitations to the types of shared disks that SPARC infrastructure can use to host the cluster file system. Without OCFS2, clustering depends on a file system that is already built to facilitate shared access and this is why only NFS is supported for this purpose. Unlike in x86 environments, when NFS is used to host a SPARC server pool file system, an OCFS2 disk image is not created on the NFS share. Instead, the cluster data is simply stored directly on the NFS file system itself.

With this information in mind, the description provided in [Section 6.8.1, "Clustering for x86 Server Pools"](#) largely applies equally to clustering on SPARC, although the implementation does not use OCFS2.

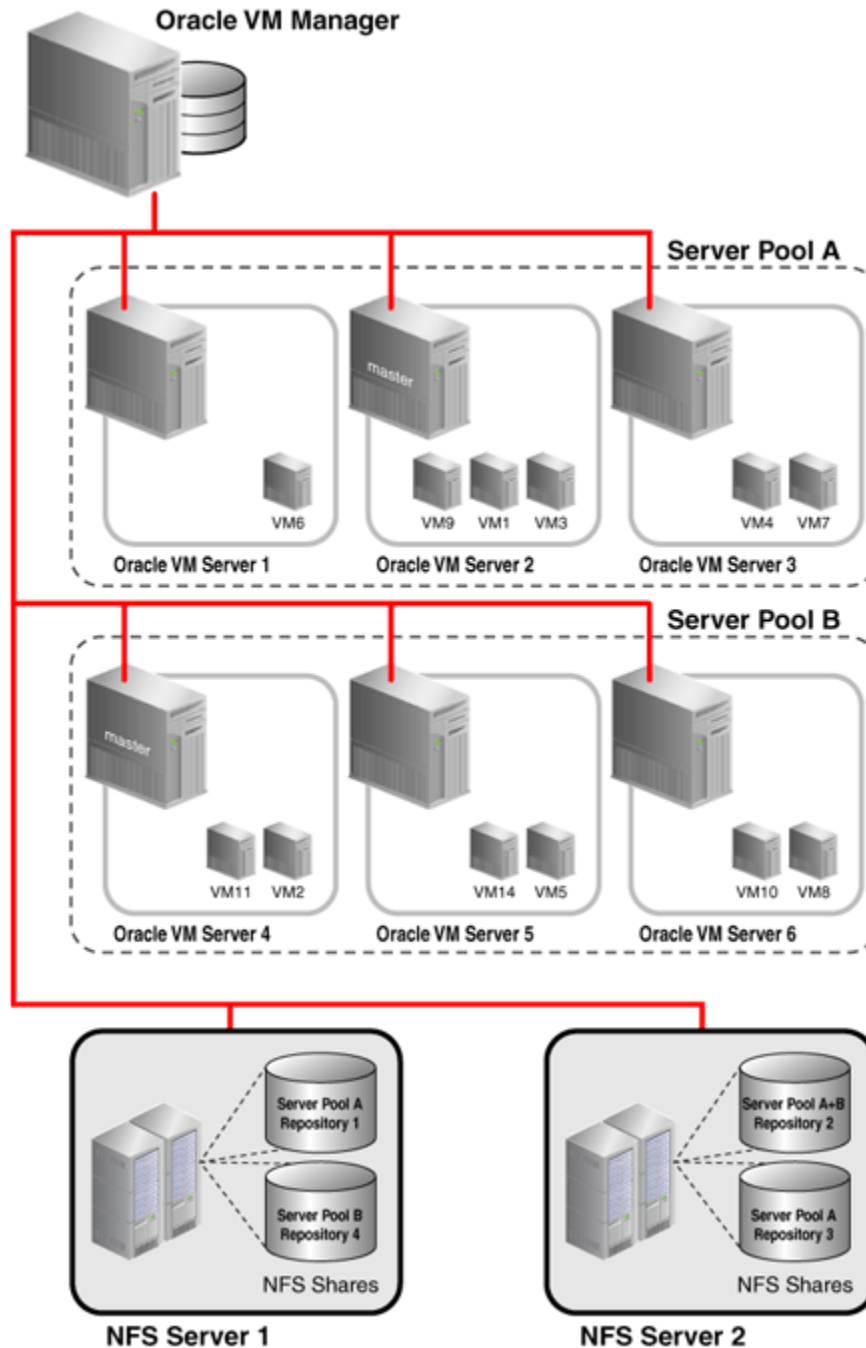
A final point to bear in mind is that clustering for SPARC server pools is only supported where a single control domain has been configured on all of the Oracle VM Servers in the server pool. If you have decided to make use of multiple service domains, you must configure an unclustered server pool. See [Section 6.9, "Unclustered Server Pools"](#) for more information.

6.9 Unclustered Server Pools

When creating a [server pool](#), you specify whether the servers in the pool will be part of a cluster or not. In most cases, you create a clustered server pool. You can create a non-clustered pool when all servers in the pool are expected to use only NFS shares as repositories. If your [Oracle VM Servers](#) are also expected to access repositories on shared physical disks, then these servers must be part of a clustered server pool. Unclustered server pools are more common in SPARC environments, where the use of multiple service domains prevents clustering but offers a more fault-tolerant and robust platform.

Figure 6.2, “Unclustered Server Pools Using Only NFS Storage” illustrates server pools in an unclustered configuration, with shared access to resources on NFS storage but no [HA](#) features for the servers.

Figure 6.2 Unclustered Server Pools Using Only NFS Storage



Non-clustered server pools do not require a server pool file system, though a Virtual IP is still required and the Master function is also assigned to one of the server pool members.

A non-clustered server pool does not support HA for virtual machines deployed on its servers. If a server fails, the virtual machines on this server have to be restarted manually on a server in this server pool, or possibly on a server in another server pool, if that server pool also has access to the repositories needed for deploying the virtual machines on the failed server. [Live Migration](#) is supported between servers in a non-clustered pool if the servers have the same CPU [affinity](#) (same family and type of CPU).



Note

Converting non-clustered server pools to clustered server pools is not supported in Release 3.3 of Oracle VM.

6.10 How does High Availability (HA) Work?

Oracle VM has [high availability](#) (HA) functionality built in. Even though there is only one Oracle VM Manager in the environment, it distributes vital information over the servers it manages, so that in case of failure the Oracle VM Manager and its infrastructure database can be rebuilt. For virtual machine HA, [Oracle VM Servers](#) can be clustered so that if one server fails, the [virtual machines](#) can be automatically [migrated](#) to another server as all virtual machine data is on shared storage and not directly on the Oracle VM Server. In case of predictable failures or scheduled maintenance, virtual machines can be [moved](#) to other members of the [server pool](#) using [live migration](#).

In addition, Oracle VM supports HA networking and storage, but these are configurations the system administrator must implement outside Oracle VM Manager (RAID, [multipathing](#), etc.).

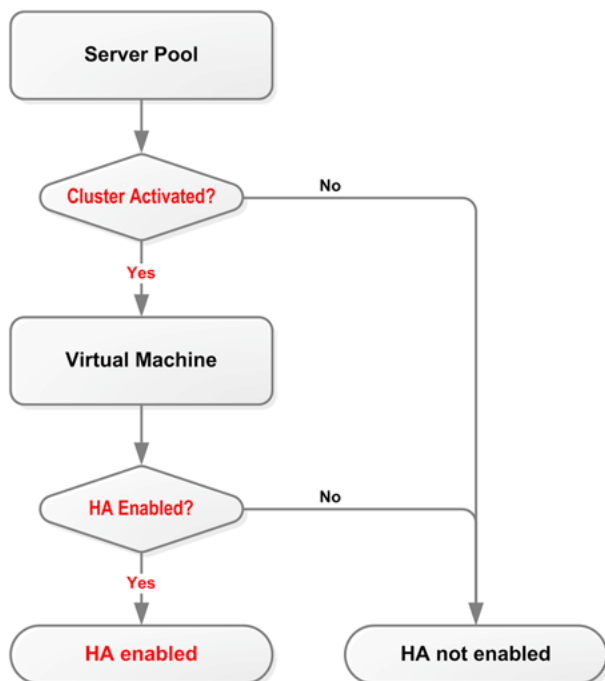
You can set up HA to help ensure the uninterrupted availability of a virtual machine. If HA is configured and a [Oracle VM Server](#) is restarted or shut down, the virtual machines running on it are either restarted on, or [migrated](#) to, another Oracle VM Server.

If you have set the [inbound migration lock](#) feature to disallow new virtual machines on an Oracle VM Server, then HA cannot be used to create or migrate new virtual machines on that server, but virtual machines already running on the server may be migrated to other Oracle VM Servers in a server pool. See [Section 7.11, “How Can I Protect Virtual Machines?”](#) for more information on using the inbound migration lock feature.

The following prerequisites are requirement to implement HA:

- The [server pool](#) must contain multiple Oracle VM Servers. HA cannot be implemented with a stand-alone Oracle VM Server.
- The server pool must be clustered.
- All Oracle VM Servers must be Oracle VM Server Release 3.0 or above.

To use [HA](#), you must first enable HA on the server pool, then on all virtual machines, as shown in [Figure 6.3, “Enabling HA”](#). If you enable HA on the server pool and then for virtual machines, when an Oracle VM Server is shut down or fails, the virtual machines are migrated or restarted on another available Oracle VM Server. HA must be enabled for **both** the server pool **and** for virtual machines.

Figure 6.3 Enabling HA

To automatically configure the server pool cluster and enable HA in a server pool, the server pool must be created with clustering enabled. See [Create Server Pool](#) in the *Oracle VM Manager User's Guide* for more information on creating a server pool.

To enable HA on a virtual machine, high availability must be enabled when you create or edit a virtual machine. See [Create Virtual Machine](#) and [Edit Virtual Machine](#) in the *Oracle VM Manager User's Guide* for more information on creating and editing a virtual machine.

If HA is enabled and you want to restart, shut down, or delete an Oracle VM Server, you must first *migrate* the running HA-enabled virtual machines to another available Oracle VM Server. For information on migrating virtual machines, see [Migrate Virtual Machines](#) in the *Oracle VM Manager User's Guide*.

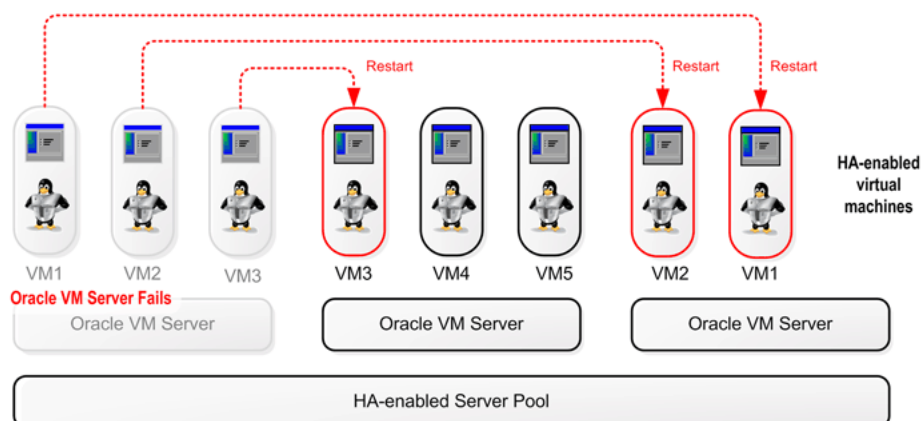
If there are no Oracle VM Servers available, the virtual machines are shut down (Powered Off) and are restarted when an Oracle VM Server becomes available.

The possible HA scenarios are:

- If you want to shut down or restart an Oracle VM Server, you must first migrate the HA-enabled virtual machines to another Oracle VM Server.
- If an Oracle VM Server fails, all running virtual machines are restarted automatically on another available Oracle VM Server.
- If an Oracle VM Server fails and no other Oracle VM Servers are available, all running virtual machines are restarted when an Oracle VM Server becomes available.

Figure 6.4, “HA in effect for an Oracle VM Server failure” shows an Oracle VM Server failing and the virtual machines restarting on other Oracle VM Servers in the server pool.

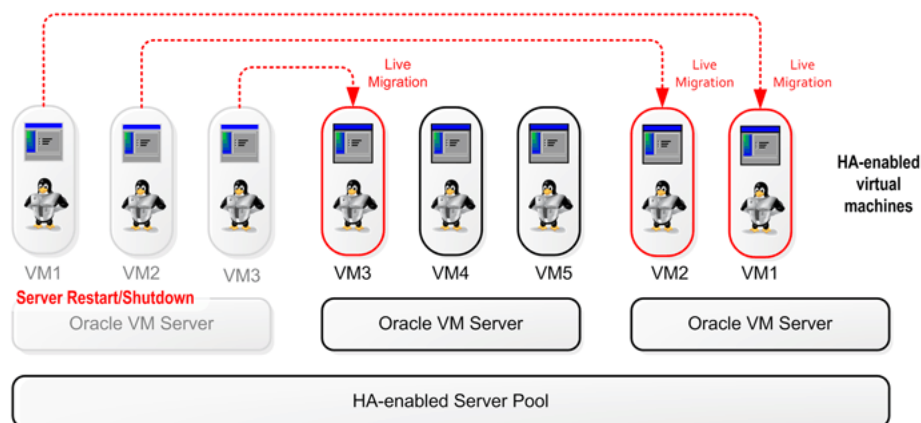
Figure 6.4 HA in effect for an Oracle VM Server failure



You should test your HA configuration to ensure it is properly configured in the event of a real failure.

Figure 6.5, “HA in effect for an Oracle VM Server restart or shut down” shows an Oracle VM Server restarting or shutting down and the virtual machines migrating to other Oracle VM Servers in the server pool. In this example, the virtual machines are running and so [live migration](#) can be performed and the virtual machines continue to run, uninterrupted. Live migration is not a feature of HA, but can be used in conjunction with, or independently of, HA. For more information on live migration, see [Migrate Virtual Machines](#) in the [Oracle VM Manager User's Guide](#).

Figure 6.5 HA in effect for an Oracle VM Server restart or shut down



If you do not have HA enabled, before you shut down an Oracle VM Server, you should migrate all virtual machines to another Oracle VM Server (either using standard virtual machine migration or live migration), or have them automatically migrated by placing the server into maintenance mode.

6.11 What are Server Pool Policies?

Managing load and reducing power consumption are two of the major benefits of virtualization. When a server is already under significant load, it is preferable to distribute the virtual machines that it is running across less utilized servers within the server pool. Equally, during periods of low utilization it is preferable to consolidate virtual machines across as few servers as possible so that unused servers can be powered off to reduce energy consumption.

Oracle VM provides a facility to handle this kind of behavior automatically. This facility is handled by creating a server pool policy. Server pool policies allow you to define the different options that you wish to support. These are defined as follows:

- **Distributed Resource Scheduling:** Used to optimize resource utilization through a form of load balancing. See [Section 6.11.1, “Distributed Resource Scheduler \(DRS\)”](#) for more information.
- **Distributed Power Management:** Used to reduce energy consumption by consolidating virtual machines across fewer servers. See [Section 6.11.2, “Distributed Power Management \(DPM\)”](#) for more information.

It is also possible to apply these policies to the networks that are available within a server pool, by setting network utilization thresholds that trigger these behaviors within the server pool. This is discussed in more detail in [Section 6.11.3, “DRS/DPM Network Policies”](#).

If you have set the inbound migration lock feature to disallow new virtual machines on an Oracle VM Server, then any server pool policies you set are restricted from migrating virtual machines, or creating new ones on the server. See [Section 7.11, “How Can I Protect Virtual Machines?”](#) for more information on using the inbound migration lock feature.

6.11.1 Distributed Resource Scheduler (DRS)

The [Distributed Resource Scheduler \(DRS\)](#) optimizes virtual machine resource utilization in a server pool. DRS automatically moves running virtual machines to another Oracle VM Server in a server pool if any of the Oracle VM Servers exceed a specified CPU threshold for a specified period of time. DRS continuously samples performance data from every Oracle VM Server and every virtual machine.

The movement of virtual machines is policy-driven. When a threshold is reached, [Oracle VM Manager](#) live migrates the running virtual machine from one Oracle VM Server to another, without down time. Oracle VM Manager allows you to specify a DRS threshold for each server pool, and to choose which Oracle VM Servers participate in the policy.

See [Define or Edit Server Pool Policies](#) in the [Oracle VM Manager User's Guide](#) for information on enabling and configuring the DRS in a server pool.

6.11.2 Distributed Power Management (DPM)

[Distributed Power Management \(DPM\)](#) is used when there are periods of relative low resource utilization to increase the consolidation ratio on fewer Oracle VM Servers. DPM dynamically *migrates* virtual machines from under-utilized Oracle VM Servers. When there are Oracle VM Servers without virtual machines running the Oracle VM Server can be powered off, conserving power until the Oracle VM Server is needed again.

DPM aims to keep only the minimum necessary number of Oracle VM Servers running. If a periodic check reveals that a Oracle VM Server's CPU utilization is operating at below a user-set level, virtual machines are live migrated to other Oracle VM Servers in the same server pool.

When all virtual machines are migrated, the Oracle VM Server is shut down.

If an Oracle VM Server exceeds the DPM policy CPU threshold, Oracle VM Manager looks for other Oracle VM Servers to migrate virtual machines to from the busy Oracle VM Server. If no powered Oracle VM Servers are available, Oracle VM Manager finds and starts an Oracle VM Server using its Wake-On-LAN capability. When that Oracle VM Server is running, Oracle VM Manager off-loads the virtual machines from the busy Oracle VM Server to the newly started Oracle VM Server to balance the overall load. It is a prerequisite that all the servers that participate in DPM have Wake-On-LAN enabled in the BIOS for the physical network interface that connects to the dedicated management network.

Oracle VM Manager allows you to specify a DPM threshold for each server pool, and to choose which Oracle VM Servers participate in the policy.

See [Define or Edit Server Pool Policies](#) in the *Oracle VM Manager User's Guide* for information on enabling and configuring DPM in a server pool.

6.11.3 DRS/DPM Network Policies

Both the [DRS](#) and [DPM](#) policies can also be set for the networks used by Oracle VM Servers in a [server pool](#). When a network used by an Oracle VM Server exceeds its threshold, virtual machines are [migrated](#) to other Oracle VM Servers to either balance the resources used (DRS), or reduce the power used (DPM). Each network on an Oracle VM Server can have a threshold set. The threshold applies to either the received data or the transmitted data. If the threshold is set to say 50%, when an Oracle VM Server's receive or transmit traffic on that network exceeds 50% of the theoretical capacity of the network, the Oracle VM Server is deemed to be over the threshold. The theoretical capacity of a network on an Oracle VM Server is equal to the port speed of the physical Ethernet adapter on the Oracle VM Server. If the network is bonded in a fail-over configuration, then the port capacity is equal to the port speed of one of the Ethernet adapters. If the network is bonded on a Oracle VM Server with link aggregation, then the network capacity is equal to the sum of the speed of the bonded Ethernet adapters.

You set the network policies for [DRS](#) and DPM when you set up the [server pool](#) policy. See [Define or Edit Server Pool Policies](#) in the *Oracle VM Manager User's Guide* for information on enabling and configuring network DRS and DPM policies in a server pool.

It is important to understand that a network policy can be defined for a server pool, even if that network is not used by any servers in the server pool. In this case, the policy is simply ignored, however if a server with the network attached is added to the server pool at a later date, the policy is automatically enabled for the network attached to that server. If you define a network policy on a server pool and later remove all of the servers that had that network attached, the policy still remains enforced on the server pool. Therefore, it is always good practice to regularly check the server pool policy when adding servers to a server pool, since an old policy may be in place that affects the behavior of the network.

6.12 What are Anti-Affinity Groups?

Anti-affinity groups specify that specific virtual machines should never run on the same [Oracle VM Server](#). An anti-affinity group applies to all the Oracle VM Servers in a [server pool](#). You may want to set up anti-affinity groups when you want to build-in redundancy or load balancing of specific applications in your environment.

If you add a virtual machine to an anti-affinity group that already has a virtual machine in the group running on the same Oracle VM Server, the job is aborted and the virtual machine is not added to the group. To add the virtual machine to the anti-affinity group, [migrate](#) it to another Oracle VM Server, then add it to the group.

If you have set the inbound migration lock feature to disallow new virtual machines on an Oracle VM Server, then any anti-affinity groups you set are restricted from migrating virtual machines, or creating new ones on the server. See [Section 7.11, "How Can I Protect Virtual Machines?"](#) for more information on using the inbound migration lock feature.

Details on creating, editing and deleting anti-affinity groups is available in [Anti-Affinity Groups Perspective](#) in the *Oracle VM Manager User's Guide*.

6.13 What are Server Processor Compatibility Groups?

To ensure successful virtual machine live migration, [Oracle VM Manager](#) requires the processor family and model number of the source and destination computer to be the same. A [server processor compatibility group](#) is a group of [Oracle VM Servers](#) with compatible processors, where a running virtual machine on one Oracle VM Server can safely be [migrated](#) and continue to run on another Oracle VM Server. Although Oracle VM Manager contains rules for server processor compatibility, you can create custom compatibility groups to ensure the ability to do smooth migrations is improved. If live migration is attempted between incompatible processors, an error message is displayed.

All Oracle VM Servers are added to a default server processor compatibility group as they are [discovered](#). A default server processor compatibility group is created when an Oracle VM Server is discovered if that Oracle VM Server has a processor that is new and unique to Oracle VM Manager.

Each server processor compatibility group may include Oracle VM Servers that are members of one or more [server pools](#). An Oracle VM Server may be included in multiple server processor compatibility groups. You can create server processor compatibility groups and select which Oracle VM Servers to include according to your needs. There is no limit to the number of server processor compatibility groups you may have.

More information on configuring server processor compatibility groups is available in [Server Processor Compatibility Perspective](#) in the [Oracle VM Manager User's Guide](#).

Chapter 7 Understanding Virtual Machines

Table of Contents

7.1 What are Virtualization Modes or Domain Types?	84
7.2 How is a Guest OS Installed on a Virtual Machine?	86
7.3 Where are Virtual Machine Resources Stored?	86
7.4 What Networking is Available for Virtual Machines?	87
7.5 How are Virtual Machines Created?	87
7.6 How does Cloning Work?	88
7.7 How Can a Virtual Machine be Moved or Migrated?	89
7.8 Editing Virtual Machine Parameters	90
7.9 Sending Messages to Virtual Machines	90
7.10 Accessing the Virtual Machine Console	91
7.11 How Can I Protect Virtual Machines?	92
7.12 How are Virtual CPUs Allocated to Virtual Machines	92
7.13 How are Huge Pages Enabled for Virtual Machines?	93
7.14 Setting Hard Partitioning for Virtual Machine CPUs	94
7.15 How are Virtual Machines Backed Up?	94

The purpose of Oracle VM is to provide an infrastructure that virtual machines are able to run on, and which is easy to manage, configure and maintain. In this chapter we explore the different types of virtual machines that are supported; how virtual machines work within an Oracle VM environment; how virtual machines can be created, moved and migrated; and how to provision resources for a virtual machine.

The terms [domain](#), [guest](#) and virtual machine are often used interchangeably, but they have subtle differences. A domain is a configurable set of resources, including memory, virtual CPUs, network devices and disk devices, in which virtual machines run. A domain is granted virtual resources and can be started, stopped and restarted independently of other domains and of the host server itself. A guest is a virtualized operating system running within a domain. A guest operating system may be [paravirtualized](#), [hardware virtualized](#), or hardware virtualized with paravirtualized drivers. A description of these different virtualization modes is provided in [Section 7.1, “What are Virtualization Modes or Domain Types?”](#).

Multiple guests can run on the same [Oracle VM Server](#). A virtual machine is a guest operating system and its associated application software. For the sake of simplicity, we use the term virtual machine to encompass domain, guest and virtual machine. They are synonymous with each other and may be used interchangeably.

An operating system installed in a virtual machine is known as a guest operating system. Oracle VM supports a variety of guest operating systems including Linux, Oracle Solaris and Microsoft Windows™. For a list of the supported guest operating systems, see the [Oracle VM Release Notes](#). Guest operating system installation is described in [Section 7.2, “How is a Guest OS Installed on a Virtual Machine?”](#).

You can use Oracle VM Manager to create virtual machines using:

- ISO files in a repository (hardware virtualized only).
- Mounted ISO files on an NFS, HTTP or FTP server (paravirtualized only).
- Virtual machine templates (by cloning a template).
- Existing virtual machine (by cloning the virtual machine).

- Virtual machine assemblies.

Virtual machines require most of these installation resources to be in a storage repository, managed by Oracle VM Manager, with the exception of mounted ISO files for paravirtualized guests. See [Section 7.3, “Where are Virtual Machine Resources Stored?”](#) for information on how these resources are stored within a repository.

When you create a virtual machine that requires network connectivity, or a paravirtualized machine which requires network connectivity to perform the operating system install, you generate some virtual network interfaces that can be bridged to a network defined on the Oracle VM Server. More information on provisioning networking for your virtual machines is provided in [Section 7.4, “What Networking is Available for Virtual Machines?”](#).

In cases where you require multiple virtual machines that are identical in make up, you might consider cloning a virtual machine or template. This process is described in more detail in [Section 7.6, “How does Cloning Work?”](#). Equally, you may need to move a virtual machine from one server or server pool to another, this process is known as migration and is discussed in [Section 7.7, “How Can a Virtual Machine be Moved or Migrated?”](#)

Oracle VM provides a messaging facility that allows messages to be sent backwards and forwards between Oracle VM Manager and a guest operating system. This makes it possible to automate configuration actions when a virtual machine boots, to obtain information about the internal status of a virtual machine, and to trigger events within a virtual machine based on changes within the Oracle VM environment. This messaging facility is described in some detail in [Section 7.9, “Sending Messages to Virtual Machines”](#).

In some cases, usually due to software licensing requirements, you may require that virtual machines are [hard partitioned](#) using CPU pinning. This feature allows you to pin a virtual machine to a specific physical CPU to prevent it from running on alternate systems. This is an advanced feature and is described in [Section 7.14, “Setting Hard Partitioning for Virtual Machine CPUs”](#).

7.1 What are Virtualization Modes or Domain Types?

Virtual machines may run in one of two main modes, [paravirtualized \(PVM\)](#) or [hardware virtualized machine \(HVM\)](#). In paravirtualized mode, the kernel of the guest operating system is recompiled to be made aware of the virtual environment. This allows the paravirtualized guest to run at near native speed, since most memory, disk and network accesses are optimized for maximum performance.

If support for hardware virtualization is available (either Intel® VT or AMD-V™), the guest operating system may run completely unmodified. This hardware virtualized machine is carefully monitored and trapped by [Oracle VM Server](#) when any instruction is executed which would violate the isolation with other guests or dom0. In the current implementation, there may be performance penalty for certain types of guest and access types, but hardware virtualization also allows many Microsoft Windows operating systems and legacy operating systems to run unmodified.

The third virtualization mode is a hardware virtualized machine with paravirtualized drivers (PVHVM). This mode is identical to a hardware virtualized machine, but with additional paravirtualized drivers installed in the guest's operating system to improve virtual machine performance.

Oracle recommends you create paravirtualized virtual machines if possible, as the performance of a paravirtualized virtual machine is often superior to that of a hardware virtualized machine.

There are a number of virtual machine virtualization modes, or [domain](#) types, as shown in [Table 7.1, “Domain Types”](#). When you create a virtual machine you must decide which mode to use.

Table 7.1 Domain Types

Domain Type	Description
<i>Hardware virtualized machine</i> (HVM)	Hardware virtualization, or fully virtualized. When you create an HVM guest, you must supply an ISO file in a repository from which to create the virtual machine. See Section 7.3, “Where are Virtual Machine Resources Stored?” for information on storing an ISO file in a repository. To create HVM guests, you may need to activate the hardware virtualization in the BIOS of the server on which you install the Oracle VM Server.
Hardware virtualized, with paravirtualized drivers (PVHVM)	Identical to HVM, but with additional paravirtualized drivers for improved performance of the virtual machine. See Converting to Paravirtualized Guests or Installing Paravirtualized Drivers for more information about using paravirtualized drivers. This Domain Type is used to run Microsoft Windows guest operating systems with an acceptable performance level.
Paravirtualized (PVM)	When you create a PVM guest, you must supply a location for the mounted ISO file from which to create the virtual machine. Before you create the virtual machine using the paravirtualized method, mount the ISO file on an NFS share, or HTTP or FTP server. You supply the location of the mounted ISO file during the creation process. For information on creating a mounted ISO file, see Section 7.2, “How is a Guest OS Installed on a Virtual Machine?” .
Oracle VM Server for SPARC (OVM/SPARC)	Since virtualization on SPARC platforms is significantly different to that achieved within x86 environments, Oracle VM distinguishes a separate virtualization mode specifically to indicate that the SPARC hypervisor is to be used. When creating a virtual machine that is to run in a server pool consisting of Oracle VM Servers for SPARC, this mode is used during configuration.



Note

If you want to create a PVHVM or PVM, make sure all disks the virtual machine is configured to use are paravirtual devices. An explanation and description of the steps to properly configure paravirtual disks is discussed in [Paravirtualized Guest Disk Devices are Not Recognized](#) in the [Oracle VM Administrator's Guide](#).

Converting to Paravirtualized Guests or Installing Paravirtualized Drivers

For optimized performance, you can install [paravirtualized](#) drivers on *hardware virtualized machines*. Paravirtual drivers are optimized and improve the performance of the operating system in a virtual machine. These drivers enable high performance throughput of I/O operations in *guest* operating systems on top of the [Oracle VM Server](#) hosts.

Creating hardware virtualized machine machines may require that you install paravirtual drivers for your hardware on the guest operating system.

The Oracle Solaris 10 and Oracle Solaris 11 operating system runs as a hardware virtual machine (HVM), which requires HVM support (Intel® VT or AMD-V™) on the underlying hardware platform. By default, Oracle Solaris 10 or Oracle Solaris 11 operating system already has the required paravirtualized drivers installed as part of the operating system.

You can continue using HVM guest, but leverage the paravirtualized I/O drivers. For more information, see the Oracle Support document 757719.1 titled *Comparison of Guest Virtualization Modes; HVM, PVM and PVHVM. Configuration, Mode Validation and Conversion to PVHVM* on the My Oracle Support web site at:

<https://support.oracle.com/epmos/faces/DocumentDisplay?id=757719.1>.

To install the paravirtual drivers for Microsoft Windows operating systems (Oracle VM Paravirtual Drivers for Microsoft Windows), see the [Oracle VM Paravirtual Drivers Installation Guide](#).

7.2 How is a Guest OS Installed on a Virtual Machine?

Virtual machines require some form of installation media, whether it be a template, [assembly](#), ISO file, or mounted ISO file. Different domain types may require slightly different installation source files. [Table 7.2, “Virtual machine installation sources”](#) lists the installation sources available for [HVM](#) and [PVM](#) guests.

Table 7.2 Virtual machine installation sources

Installation Source	HVM	PVM
Template (clone)	Yes	Yes
ISO file in repository	Yes	No
Mounted ISO file on NFS, HTTP or FTP server	No	Yes
Assembly	Yes	Yes

When you create an HVM guest from an ISO file, you must supply an ISO file which has been preloaded into a storage repository that is presented to the Oracle VM Server on which the virtual machine is to be deployed. The ISO is configured as a virtual CDROM device within the virtual machine's BIOS. At boot, the virtual machine is able to boot from the virtual CDROM like a regular physical system. See [Section 7.3, “Where are Virtual Machine Resources Stored?”](#) for information on how these files are stored in a repository.

Unlike a hardware virtualized machine, a paravirtualized machine does not have a BIOS and the kernel must be loaded directly. Therefore, the virtual machine requires direct access to a kernel that can be loaded at boot. Oracle VM Manager cannot be used to manage this process, since it is not possible to predict how networking has been configured for every deployment. The virtual machine network is frequently separated from the management network using VLANs or multiple networks, so that an Oracle VM Server is unable to present a loopback mounted ISO directly to the virtual machine. This means that to install a PVM guest from an ISO file, you must first mount the ISO on an NFS, HTTP or FTP server that is accessible to the virtual machine on the virtual machine network. During the creation of the guest, you provide a network boot path that provides a URI to the mounted installation media. Examples showing how to set up your environment to cater for PVM installation media are provided in [ISO Provisioning for PVM Guest Installations](#) in the [Oracle VM Administrator's Guide](#)

More commonly, virtual machines are deployed using either templates or assemblies. Templates are essentially clones of a single existing virtual machine. As such, they can be reused and distributed to quickly deploy a preinstalled and preconfigured virtual machine. Assemblies are similar to templates, although they can contain multiple virtual machines and are provided in the more universal Open Virtualization Format.

See [Section 4.6, “How are Virtual Machine Templates Managed?”](#) for information on working with virtual machine templates. See [Section 4.7, “How are Assemblies Managed?”](#) for information on working with assemblies. See [Section 4.8, “How are ISO Files \(CD/DVD Images\) Managed?”](#) for information on working with ISO files.

7.3 Where are Virtual Machine Resources Stored?

A storage repository is used to store virtual machine resources, so that these resources can be made available to [Oracle VM Servers](#) in a [server pool](#), without having to copy the resources to each Oracle VM Server. The types of underlying storage that can be used for a repository are discussed in more detail in

[Section 3.9, “Where are Virtual Machine Resources Located?”](#). A more detailed description of storage repositories and how to manage their contents is described in [Chapter 4, Understanding Repositories](#).

7.4 What Networking is Available for Virtual Machines?

A VNIC is a virtualized Network Interface Card, used by a Virtual Machine as its network interface. A VNIC is assigned a MAC address. Each MAC address corresponds with a single virtual NIC, which is used by a virtual machine. You can create a VNIC when you create or edit a virtual machine. There is no real limit to the number of VNICs available to any single virtual machine. Each VNIC must belong to a virtual machine and cannot exist independently of it. This requirement was implemented in version 3.3.1 of Oracle VM. Previous versions allowed you to create VNICs independently of virtual machines and then to assign them as required. This approach was frequently confusing and more difficult to manage.

VNICs are automatically related to a network bridge that is created on the hosting Oracle VM Server. It is possible to configure the network in such a way that virtual machines are able to only interact with each other on the same Oracle VM Server, with all of the virtual machines running across the server pool, or with all of the virtual machines and a public network such as the Internet.

Since MAC addresses need to be unique on a network, Oracle VM Manager dynamically creates MAC addresses to be allocated to each VNIC from a predefined range. Oracle VM Manager attempts to ensure that no two VNICs are able to have the same MAC address. It is possible to modify the MAC address range that Oracle VM Manager uses to allocate for VNICs. This is described in more detail in [Virtual NICs](#) in the [Oracle VM Manager User's Guide](#).

Network configuration on a virtual machine can be achieved using the tools available in the guest operating system. If you require DHCP configuration, you equally require a DHCP server to be configured on a network that is attached to the virtual machine network. For this reason, along with the benefit of added security, it is often useful to implement full network separation particularly for the function of the virtual machine network. This can be achieved using VLANs, or separate physical network ports or bonds.

Different network configurations and network separation are discussed in more detail in [Chapter 5, Understanding Networks](#).

7.5 How are Virtual Machines Created?

Before you create a new [virtual machine](#), make sure that the following resources are available:

- A server pool. See [Section 6.7, “How are Server Pools Created?”](#) for information on creating server pool.
- An [Oracle VM Server](#) as part of the server pool.
- Source file(s) in a repository from which to create the virtual machine. The source files can be any of the following:
 - Imported ISO file. See [Section 4.8, “How are ISO Files \(CD/DVD Images\) Managed?”](#) for more information on how to import ISO files.
 - Virtual machine template. See [Section 4.6, “How are Virtual Machine Templates Managed?”](#) for more information on how to import a virtual machine template.
 - Virtual machine [assembly](#). See [Section 4.7, “How are Assemblies Managed?”](#) for more information on how to import an assembly.

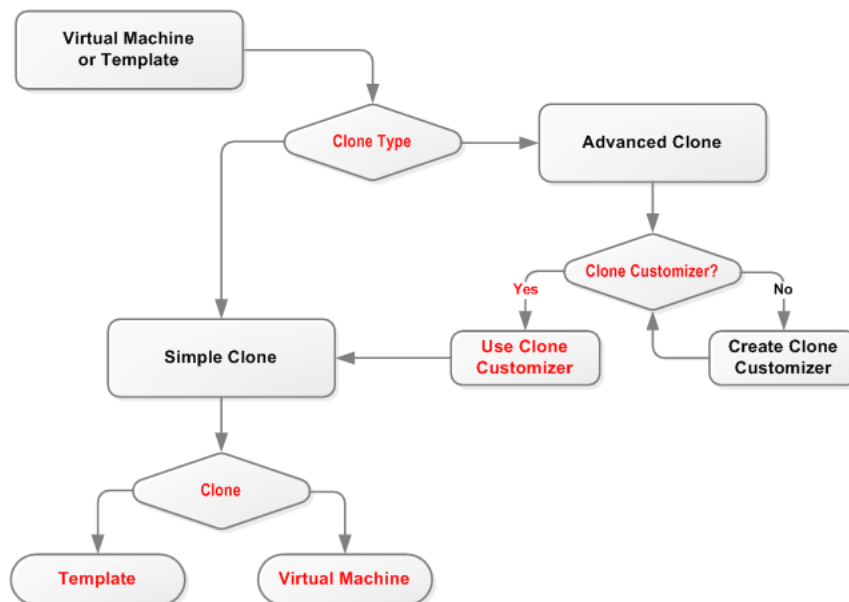
For information on how to create a virtual machine using a template, and creating a virtual machine from an ISO file, or from physical or [virtual disks](#) see [Create Virtual Machine](#) in the [Oracle VM Manager User's Guide](#).

To access the virtual machine, select the **Servers and VMs** tab, select the server pool on which the virtual machine was created in the navigation tree, and select **Virtual Machines** in the **Perspective** drop-down list in the management pane. Select the virtual machine in the table to perform operations on it. Expand the virtual machine in the table to see more detailed configuration information.

7.6 How does Cloning Work?

[Cloning](#) a virtual machine or template enables you to create multiple virtual machines or templates based on the original. There are two methods of cloning; a simple clone, and an advanced clone. A simple clone sets up the clone with the same configuration information as the original. An advanced clone enables you to create and use a clone customizer with differing configuration from the original. For example, you can use a clone customizer to have the clone deploy to a different [server pool](#) or repository, with changed memory, virtual CPU number, network settings, and so on. Figure [Figure 7.1](#), “Cloning a virtual machine or template” shows the process of creating a clone of a virtual machine or template.

Figure 7.1 Cloning a virtual machine or template



To modify the clone parameters, such as [virtual disks](#), network, memory, and so on, you should create a clone customizer, and use that clone customizer to perform cloning. You can create a clone customizer to set up the clone parameters, such as networking, and the virtual disk, and ISO resources. A clone customizer is also used when moving a virtual machine or template. See [Manage Clone Customizers](#) in the [Oracle VM Manager User's Guide](#) to see an example of how you are able to create a clone customizer to use during cloning.

A *cold* clone is a clone created from a *stopped* virtual machine. A cold clone performs a clone of the virtual machine, with safe and consistent virtual disk status. This is useful for creating a virtual machine or template from the original virtual machine.

A *hot* clone is created from a *running* virtual machine. A hot clone is only available on [OCFS2-based](#) file systems, so you must use either iSCSI- or fibre channel-based storage for the source and target repositories. Additionally, Oracle VM Manager allows you to create hot clones for virtual machines only if the clone type for the virtual disk is a [thin clone](#).



Note

A hot clone creates a clone with inconsistent disk status, and should only be used as a *snapshot* or back up of a virtual machine, perhaps on a virtual machine that requires 100% uptime and cannot be shut down. If you want to use the hot-cloned virtual machine, you should first repair any virtual disks, using a disk repair utility such as `fsck`. Do not use hot cloning for virtual machines running an Oracle Database. Instead, you should use an Oracle Database backup strategy, such as the `rman` utility.

A *thin clone* only copies the required virtual machine files and ignores empty disk space, so the cloning process is able to complete much more quickly. This process makes use of the *reflinking* capability within the OCFS2 file system, allowing for a rapid cloning method. As a result, thin cloning can only be used when cloning from and to the same *OCFS2-based* repository, and when the storage used for the storage repository is non-generic (for example, a Sun 7000 or NetApp *Storage Connect* plug in). Thin cloning is the fastest and most efficient cloning method.



Note

Thin provisioning of physical disks on generic storage is not supported.

A clone can also be performed using two other file copy methods: *sparse copy*, and *non-sparse copy*. These two cloning methods can be used when cloning from and to different repositories, and when the storage used for the storage repository uses a generic *Storage Connect* plug in. These cloning methods are slower than thin cloning, but more versatile.

Sparse copy only takes up the amount of disk space actually used, not the full specified disk space. This process only copies used blocks and ignores unused blocks on the disk. This makes it quicker than a non-sparse copy and is efficient in its initial use of disk space, however the process is not as fast as thin cloning. Since sparse copy does not allocate all required disk space immediately, it is important to plan for the possibility that disk sizes may ultimately exceed the repository size where the virtual machine is running and cause I/O errors on all of the running virtual machines in the same repositories. Therefore, when using sparse copy, you should ensure that the repository size is adequate to handle the potential disk space requirements of all of the virtual machines hosted on it.

Non-sparse copy performs a block-level copy of a disk, so that even unused blocks are cloned. This means that the virtual machine disk is fully allocated at the time that the virtual machine is cloned, whether the disk blocks are used or not. The advantage of this type of clone is that you can always be certain of the amount of disk space required for a virtual machine and cannot exceed the disk space available in a repository by hosting it there.

7.7 How Can a Virtual Machine be Moved or Migrated?

Moving a Virtual Machine to Another Repository

You can *move* a non-running virtual machine from one repository to another. During the move you can specify where the disks should be moved to using a clone customizer. If you keep the clone customizers to the default settings (the same as the original virtual machine), a *move* is essentially the same as migrating the virtual machine to another Oracle VM Server. You can change the location of disks and the virtual machine configuration file to another storage repository when you *move* a virtual machine using a clone customizer.

**Note**

The network information is not changed when moving a virtual machine, so you cannot move VNICs between networks. Any network changes you make in a clone customizer are ignored when moving a virtual machine. This allows you to preserve the virtual machine in its original state, while moving the configuration file and storage to a different repository.

Migrating Virtual Machines Between Oracle VM Servers

[Migrating](#) a virtual machine is a process to move a virtual machine from one Oracle VM Server, or [server pool](#), to another. If the virtual machine is running during the migration, the applications continue to run, uninterrupted and this is called [live migration](#). This feature is important, and useful, when the existing Oracle VM Server may be out of commission (if HA is enabled), or on a planned shutdown for maintenance purposes. Live migration can only be performed within the same server pool, so a running virtual machine cannot be moved out of its server pool. Live migration does not require HA to be enabled; it can occur in a server pool simply by selecting a running virtual machine and migrating it, independently of whether or not the server pool is clustered and/or whether the virtual machine has its HA flag set.

You can migrate one or more virtual machines at a time. When migrating multiple virtual machines, the migrations are performed serially and not concurrently, so one migration is performed, then the next, until all the migrations are completed. Cross-server pool migration is allowed, though the virtual machines must first be stopped. A virtual machine must also be in a stopped state before migrating it off of any server (an **Unassigned Virtual Machine**). When you migrate a virtual machine to another server pool, the virtual machine is not deployed to a particular Oracle VM Server until you start the virtual machine, then the placement strategies for the virtual machine depends on the Oracle VM Server roles, and destination server pool policies such as [Distributed Resource Scheduler \(DRS\)](#) and [Distributed Power Management \(DPM\)](#). See [Section 6.2, “What are Server Roles?”](#) for information on Oracle VM Server roles, and [Section 6.11, “What are Server Pool Policies?”](#) for information on server pool policies.

If you have set the inbound migration lock feature to disallow new virtual machines on an Oracle VM Server, then any automatic migration policies you set are restricted from migrating virtual machines, or creating new ones on that server. See [Section 7.11, “How Can I Protect Virtual Machines?”](#) for more information on using the inbound migration lock feature.

The CPU family and model number of the source and destination computers must be compatible in order to perform live migration. This means, for instance, that you cannot migrate a virtual machine from an x86-based [server pool](#) to a SPARC-based server pool, or vice versa. Equally, you cannot perform a live-migration within the same x86-server pool, if the servers have different CPU families or model numbers. For more information on CPU compatibility, please see [Section 6.13, “What are Server Processor Compatibility Groups?”](#).

7.8 Editing Virtual Machine Parameters

It is possible to edit the parameters that are used to start any virtual machine within Oracle VM using Oracle VM Manager. Typical changes may include updating the keymap, mouse device type, memory or CPU allocation or to enable huge page support. It is important to understand that these parameters are runtime parameters used by the hypervisor to initialize a virtual machine. This means that making changes to these parameters do not have an immediate effect on a running virtual machine. For changes to take effect, the virtual machine must be stopped and then started again, so that the hypervisor is forced to reload the virtual machine configuration. A simple restart does not have the same effect, since the hypervisor does not reload the virtual machine configuration for this event.

7.9 Sending Messages to Virtual Machines

Sending a [message](#) to a virtual machine may be useful during certain situations, like developing or starting a [virtual machine template](#). Messages are passed to the virtual machine operating system, and can be of any key/value pair that the operating system or template understands. For example, to set up a virtual machine template to use DHCP, you may want to send the following key/pairs to a virtual machine:

```
com.oracle.linux.network.device.0    eth0
com.oracle.linux.network.onboot.0    yes
com.oracle.linux.network.bootproto.0 dhcp
com.oracle.linux.root-password       password
```



Note

The `com.oracle.linux.root-password` is always required as a parameter and should be sent as the final parameter for any message.

Any messages that are not understood are discarded and ignored by the operating system. You can hide the message for sensitive information such as passwords, so a series of asterisks are displayed in the user interface instead of the sensitive information.

You can optionally keep a log of the message. This feature is especially useful to template or application developers that want to send messages to virtual machines. Message logs are stored by Oracle VM Manager and are not available through any log file or database query. To gain access to these messages, you must use the Oracle VM Web Services API. Messages are limited to 1024 bytes when message logging is selected, or 8192 bytes when logging is not selected. The key is limited to 256 bytes.

Each message sent to a virtual machine is contained within its own job. If you send multiple messages to multiple virtual machines, each one has its own job, so 10 messages to 100 virtual machines produces 1,000 jobs.

To send a virtual machine a message you must have first installed the Oracle VM Guest Additions in the virtual machine. For information on installing the Oracle VM Guest Additions, see [Oracle VM Guest Additions](#) in the *Oracle VM Administrator's Guide*. For information on sending messages to a virtual machine, see [Send VM Messages](#) in the *Oracle VM Manager User's Guide*.

7.10 Accessing the Virtual Machine Console

You can connect to a virtual machine using its *console*. The console is the remote control system of Oracle VM, and enables you to work and interact directly with your virtual machines.

There are two types of virtual machine consoles in [Oracle VM Manager](#): a VNC console used to connect to virtual machines in x86-based [server pools](#), but which is not supported for SPARC-based server pools; and a *serial* console used to connect to the terminal console of a virtual machines and most commonly used for SPARC-based server pools.



Note

You can use the serial console to connect to a Linux guest virtual machine in an x86-based [server pool](#), if the guest supports a serial console, but the console is in read-only mode, and you cannot interact with the virtual machine using the serial console. Furthermore, additional configuration is required within the guest virtual machine to enable this facility. Therefore, use of the serial console should be limited to virtual machines that run within a SPARC-based server pool, and the VNC console should be used to access virtual machines running in an x86-based server pool instead.

It is necessary to install additional packages on the Oracle VM Manager host to take advantage of the VNC or serial console that is included in the Oracle VM Manager web-based user interface. These packages

and their requirements are described in detail in [Installing and Upgrading Packages Required For The VNC and Serial Console](#) in the [Oracle VM Installation and Upgrade Guide](#).

There is no requirement to install or run any additional software on the client system accessing the virtual machine console. The required packages are installed on the Oracle VM Manager host and the console is served to the client system using a combination of JavaScript and HTML5 elements. Make sure the web-browser used on the client system meets the minimum requirements as described in [Installing and Upgrading Packages Required For The VNC and Serial Console](#) in the [Oracle VM Installation and Upgrade Guide](#).

Access to the serial and VNC consoles for each virtual machine is provided via the Oracle VM Manager web-based user interface. This is described in more detail in [Launch Console](#) and [Launch Serial Console](#) in the [Oracle VM Manager User's Guide](#).

7.11 How Can I Protect Virtual Machines?

You can protect the resources on an Oracle VM Server by restricting whether new virtual machines can be created on it or migrated to it. By doing this, you can make sure the resources of an Oracle VM Server are available to the virtual machine(s) running on the server. You can set this up for an Oracle VM Server by using the [inbound migration lock](#) feature. This feature is in addition to the other protection mechanisms such as HA, server pool policies and anti-affinity groups, and overrides any of these other features or policies that might try to create or migrate a virtual machine onto the Oracle VM Server. This feature allows you to set whether to allow additional virtual machines to run on the Oracle VM Server, but does not prevent virtual machines running on the Oracle VM Server from being migrated to another Oracle VM Server.

For more information on using the inbound migration lock feature, see [Edit Server](#) in the [Oracle VM Manager User's Guide](#).

7.12 How are Virtual CPUs Allocated to Virtual Machines

When you create or edit a virtual machine, you have the option to specify two different values for the number of processors (also known as virtual CPUs) that are allocated to a virtual machine. These represent:

- the maximum number of virtual CPUs that can be made available to a running virtual machine (maxvcpu)
- the actual number of virtual CPUs that a virtual machine should be using at runtime (vcpu)

The number of virtual CPUs that the virtual machine actually has available at any one time is dynamically allocated by the hypervisor as required, by changing the number of processors (vcpu) for the running virtual machine within Oracle VM Manager. This value can never exceed the maximum number of processors (maxvcpu) that you have also specified within Oracle VM Manager.

If you want to change the maximum number of virtual CPUs (maxvcpu) that a virtual machine can have, the virtual machine must be rebooted after this value is changed. The maximum number of virtual CPUs (maxvcpu) cannot exceed the values described in [Configuration Limits](#) in the [Oracle VM Release Notes](#).

Guest virtual machine boot behavior, in relation to virtual CPU allocation, differs depending on the mode of virtualization that the guest is configured for. Modes of virtualization are discussed further in [Section 7.1, "What are Virtualization Modes or Domain Types?"](#). For instance, for a paravirtualized virtual machine (PVM), the guest virtual machine boots with as many virtual CPUs as are specified for the maximum number of processors (max_vcpu) in Oracle VM Manager and then works its way down to the configured number of processors (vcpu) that should be available to the virtual machine at runtime. If the difference between these two values is large, then it may take some time before the guest virtual machine is actually

running with the specified number of virtual CPUs (vcpu). On the other hand, a hardware-virtualized virtual machine (HVM) boots with exactly the number of virtual CPUs (vcpu) that are specified in the configuration.

Once the virtual machine is running with the specified number of virtual CPUs, you can increase or decrease this value within Oracle VM Manager as required, and the number of processor cores available to the running guest operating system is automatically adjusted.

7.13 How are Huge Pages Enabled for Virtual Machines?

Paging is a process whereby the CPU, for a system, allocates contiguous blocks of memory for use by a running process. These pages are tracked by the operating system, so that processes access the correct blocks of assigned memory. Typically, these blocks are sized at 4 KB. This means that when a process uses 1 GB of memory, 262144 page (1 GB/4 KB) entries are created and are referenced continually by the process.

Most current CPU architectures support bigger pages to reduce the number of page lookups required by the CPU or Operating System. On Linux systems, these are called Huge Pages, while on Windows systems they are called Large Pages. These terminologies are equivalent.

By default, Huge Pages are available for hardware virtualized virtual machines (HVM and PVHVM) when the new virtual machine is created. For paravirtualized virtual machines (PVM), the Xen hypervisor must start the virtual machine with a specific parameter set to enable Huge Page support. Oracle VM Manager provides an option to enable huge page support for a paravirtualized virtual machine when you create or edit a virtual machine. Do not attempt to enable Huge Page support for an HVM or PVHVM, as this functionality is not handled by the Xen hypervisor, but rather by the virtual machine itself.

For huge pages to be supported in the virtual machine, depending on the operating system of the virtual machine, huge pages may need to be enabled inside the virtual machine. For example, in Linux, this is done by specifying the Linux boot command line flag `balloon_hugepages` inside the virtual machine.

Huge pages are not supported on virtual machines running on SPARC architecture. Attempting to enable Huge Page support for a virtual machine running on a SPARC server, causes an exception to be returned.

The following points are worth considering if enabling Huge Page support on your paravirtualized virtual machines.

- Huge pages can only be changed when the virtual machine is stopped.
- It is always safe to set huge pages on a virtual machine. It just means the entire guest will be allocated with huge pages, regardless of whether it uses them or not.
- It is not always safe to turn off huge pages on a virtual machine. If the virtual machine has been enabled to use huge pages (from inside the virtual machine's operating system) and if Huge Page support is disabled in Oracle VM Manager, then the underlying memory is unlikely to be aligned correctly. The Huge Page code in the operating system may check for this and fail with an error.
- If there are virtual machines configured Huge Page support disabled, the server where they are running may report that it has enough memory to run a given virtual machine. However, if the virtual machine that is scheduled to run on that server has Huge Page support enabled, the start (or migrate) may fail as the memory on that server may be fragmented such that there is not enough huge page memory available to run the virtual machine. To avoid these issues it is not possible to run virtual machines that have Huge Page support enabled on the same server as those with Huge Page support disabled.
- If you have configured PVMs with Huge Page support enabled within Oracle VM Manager and you have PVHVMs or HVMs co-existing within your environment, you are not able to run these virtual machines

on the same server. This is because it is not possible for Oracle VM Manager to determine whether or not Huge Pages are actually enabled within a PVHVM or HVM and then manage memory allocation appropriately.

See [Create Virtual Machine](#) in the *Oracle VM Manager User's Guide* for more information on this option.

7.14 Setting Hard Partitioning for Virtual Machine CPUs

Oracle VM offers an advanced feature for [hard partitioning](#), also known as [CPU pinning](#). Hard partitioning means binding a virtual machine CPU to a physical CPU (on x86) or core (on SPARC), and preventing it from running on other physical cores than the ones specified. This is done for Oracle CPU licensing purposes, since Oracle VM is licensed on a per-CPU basis.

CPU pinning cannot be configured using Oracle VM Manager; it can only be set up using the Oracle VM Utilities.

Hard partitioning is described in its own white paper available on the Oracle Technology Network. This white paper describes setting up hard partitioning using the Oracle VM Utilities. The hard partitioning white paper is titled [Hard Partitioning With Oracle VM Server for x86](#) and located at:

<http://www.oracle.com/technetwork/server-storage/vm/ovm-hardpart-168217.pdf>



Warning

Live-migration of CPU pinned virtual machines to another Oracle VM Server is not permitted under the terms of the license. Consequently, [DRS](#) and [DPM](#) policies should not be enabled for [server pools](#) containing CPU pinned [guests](#).



Note

If your Oracle VM Servers support NUMA (non-uniform memory access), make sure that the systems are running correctly in NUMA mode. In a clustered setup, a CPU can access its local memory faster than non-local and shared memory. To make full use of the performance advantages of NUMA, be sure to pin the virtual VCPUs to the same physical CPU on an Oracle VM Server. For more information about NUMA, consult your server hardware documentation.

For information about special Oracle hard partitioning licensing policies, see the [Oracle Technology Network](#) and open the PDF document on the subject of *Partitioning* located at:

<http://www.oracle.com/us/corporate/pricing/specialty-topics/index.html>

7.15 How are Virtual Machines Backed Up?

Virtual machine backup is not handled directly by Oracle VM. However, there are a number of approaches to virtual machine backup that can take advantage of facilities already provided within Oracle VM. Most commonly virtual machine backup is achieved by performing cloning of a virtual machine. For the best possible result, this should be performed as a cold clone (i.e. where the virtual machine is stopped) as this ensures better data integrity. Cloning is discussed in more detail in [Section 7.6, "How does Cloning Work?"](#).

It is equally possible to stop a virtual machine and then create a backup of the virtual machine files stored in the repository that is used for the server pool.

A more comprehensive discussion of virtual machine backup strategies is provided in [Backing up Virtual Machines](#) in the *Oracle VM Administrator's Guide*.

Chapter 8 Understanding Jobs, Events and Errors

Table of Contents

8.1 What are Job States?	96
8.2 How are Resources Locked During Jobs?	97
8.3 Are Failed Jobs Rolled Back?	97
8.4 What are Recurring Jobs?	98
8.5 Acknowledging Events/Errors	98

Oracle VM makes strong use of a [job](#) operation framework that supports a flexible approach to the configuration of physical and virtual objects. Oracle VM Manager maintains an accurate and consistent view of the virtualization environment while users perform separate and simultaneous [jobs](#). A configuration change on any individual object type within Oracle VM Manager is considered to be a job. Since Oracle VM allows for incredibly granular configuration of each object type within the environment, some configuration changes may affect more than one object type within Oracle VM Manager. This means that some configuration actions, such as the creation of a clustered server pool, may consist of many consecutive jobs, as each different object and the relationships between them are created, updated or removed within the Oracle VM Manager database.

Therefore an operation may consist of multiple jobs. This behavior is distinct from the behavior of jobs within Oracle VM prior to version 3.3.x, where whole operations were often treated as a single job. Examples of operations that translate to a number of different jobs are presented in [Table 8.1, “Job Example”](#).

Table 8.1 Job Example

Operation	Jobs
Create clustered server pool	Create server pool
	Create Cluster Heartbeat Device
	Create Cluster
Create virtual machine ^a	Create Virtual Machine
	Add Virtual Machine to Server Pool
	Create Virtual Disk
	Add Virtual Disk to Virtual Machine
	Create VNIC on Virtual Machine

^a The number of jobs for this action may vary depending on the configuration steps that you perform. The example provided here is merely indicative of how a number of jobs can be triggered by performing a single operation.

A job begins when you make any change in Oracle VM Manager. Each change you make appears in the Job Summary pane on the **Jobs** tab as a discrete operation. Job operations can be comparatively minor actions, such as renaming a [virtual machine](#). Operations may also have a wider scope, such as the creation of a new network or storage device. Performing any of these actions changes the configuration of Oracle VM Manager. When a new job is started, information about the job is displayed in the Job Summary pane at the bottom of the management pane to show the job's progress.

Furthermore, some jobs may be triggered internally within Oracle VM Manager by particular events, as a result of internally set recurring job operations, or as child jobs that are triggered as a subset of larger job

operations initiated by a user. There is no distinction between a job resulting from a user action or from an internal event within Oracle VM Manager.

A single job can contain one or many individual operations. When a job is in progress, a yellow lock appears to the left of the resources included in the job in the Oracle VM Manager Web Interface.

Unlike previous releases, Oracle VM Manager 3.3 handles jobs asynchronously. This means that when an operation using the Oracle VM Manager Web Interface or Oracle VM Manager Command Line Interface triggers one or more jobs, an error may not be returned immediately if a job fails. It is important to monitor job progress when you perform an operation to check that all jobs complete successfully.

The following sections describe jobs, and how resources are locked and released at the start and conclusion of each job, and how to manage jobs.

8.1 What are Job States?

A job listed in Oracle VM Manager can have any of the states defined in [Table 8.2, “Job States”](#).

Table 8.2 Job States

Job State	Definition
Completed	The job has completed.
In Progress	The job is in progress.
Aborted	The job has been aborted. Oracle VM Manager has rolled-back to its previous state and all locks have been released.
Failed	The job has failed. Oracle VM Manager has rolled-back to its previous state and all locks have been released.
Child Queued	The job has spawned a child job. This child job is in queue but is not running yet. The parent job is waiting for all child jobs to complete.
Child Running	The job has spawned a child job. This child job is currently running. The parent job is waiting for all child jobs to complete.

Some job operations, such as renaming an object, complete quickly. Others, such as adjusting the memory used by a virtual machine, take longer. Monitoring Job States allows you to understand how a job is progressing.

If a job is running or fails to complete, you can abort the job to cancel it. For example, a virtual machine or [Oracle VM Server](#) may be in an unresponsive state and fail to respond to a start or stop request. The appropriate action is to abort the job. When a job is cancelled in this way, its state changes to *Aborted*. See [Abort Jobs](#) in the [Oracle VM Manager User's Guide](#) for information on aborting jobs.

Jobs may *hang* or remain *In Progress* every time a virtual machine is started or stopped. A paravirtual machine may be in an unresponsive state for a variety of reasons and consequently fail to respond to a start or stop request. The appropriate action in this case is to abort the job. For example, when starting a PVM virtual machine using PXE type boot with an invalid network URL, this causes the virtual machine status to be *In Progress* indefinitely. To resolve this, abort the virtual machine start job. Edit the virtual machine and provide the correct URL.

If a job has a *Failed* state, more information is available for the job, usually indicating the reason for failure. This information is accessible using any of the interfaces provided for Oracle VM Manager.

Some jobs may spawn child jobs. In this case, the parent job will have the status *Child Queued* or *Child Running*. The parent job must wait until all spawned child jobs are completed before it is able to report its

status. This is typical in situations where multiple actions must be performed on different objects within Oracle VM Manager. For instance, when presenting a repository to multiple servers, a parent job is created to handle the action and report on its overall status, but subsequent child jobs are spawned to actually carry out the task on each server.

If a job seems to hang in a *Child Queued* state for an extended period of time, check the reachability of your server. Oracle VM Manager continues to try to execute a job even if the server is unreachable. It does not stop attempting to run the job until the job has been completed or the job is aborted. This can cause the job queue to lock so that subsequent actions within Oracle VM Manager are not added to the job queue. If this happens, you may need to abort the job that is hanging in this state to free up the job queue.



Note

If you log out of the Oracle VM Manager Web Interface and there are pending or in progress jobs, a warning is displayed confirming you want to log out. If you do log out, any pending or in progress jobs are cancelled. If there is a UI timeout, pending and in progress jobs continue.

8.2 How are Resources Locked During Jobs?

A single task can contain one or many individual [jobs](#). When a job is in progress, the resources affected by the job are locked to all other [Oracle VM Manager](#) users until the job is completed or aborted.

The state of locked objects cannot be known until the locks are cleared. The state of Oracle VM Manager is always accurately reflected by the state of objects that are not locked.

A number of different users may perform jobs simultaneously, provided they are performed on different objects. For example, suppose User A has created a [server pool](#), named 'Finance', and begins a job by moving [Oracle VM Servers](#) into this server pool. At the same time, User B modifies the resources of a server pool named 'Commodities'. Each user is able to see the different jobs in progress and would also see each other's objects as locked. The objects remain locked until the jobs are completed.

Prior to completing a job, a lock can be cleared in two ways:

- By logging out the user who initiated the lock.
- By direct action of an Oracle VM Manager user.

All locks are cleared when a job completes.

8.3 Are Failed Jobs Rolled Back?

Job operations are validated by [Oracle VM Manager](#) as they are added to the **Jobs** tab. The failure of any job operation causes the following to happen:

- The failed job is cancelled.
- Remaining jobs are attempted.
- All locks in the operation are released.

In previous Oracle VM releases (prior to Release 3.3.x), any remaining jobs in a transaction were cancelled, and all jobs rolled back to a consistent state prior to performing the operation. From Release 3.3.x onwards, this does not occur. The current architecture (using the Web Services API) requires that you look through the job list to see which jobs failed, and which succeeded. When you find a failed job, attempt to remedy the error or fault, then retry the job. If the Oracle VM environment is left in an

inconsistent state due to a failed job, then you must clean it up using the Oracle VM Manager Web Interface, or, if this cannot be done, using the Oracle VM Manager Command Line Interface.

8.4 What are Recurring Jobs?

[Oracle VM Manager](#) periodically performs recurring [jobs](#), such as repositories and file systems refresh, and checking server update repositories for any available software updates for [Oracle VM Servers](#). These jobs and the settings available for them can be configured within the Oracle VM Manager Web Interface. The recurring jobs are listed in [Recurring](#) in the [Oracle VM Manager User's Guide](#).

8.5 Acknowledging Events/Errors

When a job operation fails, one or more [events](#) may be generated. These can be viewed through any of the interfaces to [Oracle VM Manager](#) and are displayed in the user interface as flagged with yellow or red icons in the navigation tree. If an object has an error event associated with it you must acknowledge the event to clear the error and return the object to normal operations. For example, this can occur if an [Oracle VM Server](#) or virtual machine is stopped as a result of an error. Oracle VM Servers, virtual machines, repositories and storage objects can have error events associated with them. To acknowledge events, see the object's **Events** perspective in [Events Perspective](#) in the [Oracle VM Manager User's Guide](#).

Glossary

A

admin server

An Oracle VM Server dedicated to performing administrative functions on storage servers such as creating a new LUN or extending a file system. The server must be capable of logging into a storage array or file server as an admin user. The administrative functions available to the server are defined by the Oracle VM Storage Connect plug-in.

affinity

Specify that specific virtual machines should always run on the same host.

assembly

An infrastructure template containing a configuration of multiple virtual machines with their virtual disks, and the inter-connectivity between them. Assemblies can be created as a set of .ovf (Open Virtualization Format) and .img (disk image) files, or may all be contained in a single .ova (Open Virtualization Format Archive) file.

B

bonding

Bonding is a Linux OS feature that provides a method for aggregating several ports into a single bonded interface, to provide load balancing or redundancy. When you discover an Oracle VM Server, the bonded interface is shown as containing a single port.

Network bonding refers to the combination of network interfaces on one host for redundancy and/or increased throughput. Redundancy is the key factor: You want to protect your virtualized environment from loss of service due to failure of a single physical link.

In Oracle VM, there are three modes of network bonding:

- Active - Passive: One Network Interface Card (NIC) is active while the other NIC is standby. If the active NIC goes down, the other NIC becomes active.
- Dynamic Link Aggregation: All NICs act as one NIC and the network traffic flows through all interfaces concurrently, which results in a higher throughput. With this mode, your network administrator must create LACP (Link Aggregation Control Protocol) bonding on the network switch(es).
- Adaptive Load Balancing: The network traffic is equally balanced over the NICs of the bond. This mode does not require any special configuration on connected network switch(es). However, this mode does not support using VLAN with bridges. If using this mode for your bonded interfaces in any network, you cannot use VLANs if this network is configured with the Virtual Machine network channel.

C

clone

The action or result of making an exact copy of an object. The object may be a virtual machine, virtual machine template, ISO file, or virtual disk. Cloning is analogous to copying and maintains the integrity of the original object, while creating a new object based on the original. A clone customizer may be used to define cloning options to specify details of where the object components may reside when cloned, such as in a different storage repository.

control domain

A privileged domain that creates and manages other logical domains and services.

D

discover

The process of adding systems as objects within Oracle VM Manager is known as *discovery*. When you add Oracle VM Servers and storage to your Oracle VM environment, Oracle VM Manager uses the information provided to connect to the resource and perform verification. During this process, information is usually exchanged between the server and the manager. In the case of an Oracle VM Server, Oracle VM Manager obtains information about the server, its network connectivity and any storage that is already attached to the server. Depending on your hardware and networking configuration, external storage may be automatically detected during discovery of Oracle VM Servers. This is always the case with local OCFS2 storage on an Oracle VM Server.

While storage can be automatically discovered during the process of discovering Oracle VM Servers, you may need to perform storage discovery for resources that are not already attached to any of your Oracle VM Servers. It is important that storage is configured outside of the Oracle VM environment prior to discovery. Depending on the storage type, you can perform different storage discovery operations from within Oracle VM Manager.

distributed power management

DPM complements DRS to reduce the Oracle VM Servers in a server pool when there are periods of low resource utilization. It can automatically add capacity as needed when resource utilization ramps up.

See Also: [distributed resource scheduling](#)

distributed resource scheduling

DRS provides real-time monitoring of Oracle VM Server utilization with the goal to rebalance a server pool to provide consistent resources to the running virtual machines. DRS migrates load away from heavily loaded Oracle VM Servers to less heavily loaded Oracle VM Servers.

See Also: [distributed power management](#)

dom0

An abbreviation for *domain zero*. The management domain with privileged access to the hardware and device drivers. Dom0 is the first domain started at boot time. Dom0 has more privileges than domU. It can access the hardware directly and can manage the device drivers for other domains. It can also start new domains.

domain

A configurable set of resources, including memory, virtual CPUs, network devices and disk devices, in which virtual machines run. A domain is granted virtual resources and can be started, stopped and rebooted independently.

See Also: [dom0](#)

See Also: [domU](#)

domU

An unprivileged domain with no direct access to the hardware or device drivers. Each domU is started by dom0.

E

events

Events are used to register status information of "objects" within Oracle VM Manager for future reference or to make problems easier to trace back. Events are often, though not always, related to [jobs](#) that are initiated within Oracle VM Manager. For instance, when a job fails, an event is generated. Events can also be triggered through

changes in the environment such as server crashes or storage disconnects. Therefore, events are used to alert you to potential problems that may need your attention.

Events are categorized by severity. Most events will be informational, but they can also be warnings or errors. If an event has an error level severity, you need to acknowledge the error event to clear the error and to perform further operations on the object that generated the error.

See Also: [jobs](#)

G

guest

A guest operating system that runs within a domain in Oracle VM Server. A guest may be paravirtualized or hardware virtualized. Multiple guests can run on the same Oracle VM Server.

H

hard partitioning

Hard partition, or CPU pinning, is the act of binding a virtual machine to one or more physical CPUs or cores. This prevents software within the virtual machine from running on any cores other than those specified for the virtual machine. By default, Oracle VM takes advantage of distributed resource scheduling, which allows a virtual machine to use all cores on an Oracle VM Server as required. In some situations, such as the requirement to conform with [Oracle licensing policies for partitioned environments](#), it may be desirable to implement hard partitioning.

Hard partitioning can result in restrictions on live migration, DRS and DPM.

hardware virtualized machine (HVM)

A hardware virtualized guest runs on the virtualization platform as it would on a physical host. Because the device drivers of the hardware virtualized guest are emulated, dom0 must contain device emulation code to support the guest's device drivers. The other types of privileged instructions issued by the hardware virtualized guest, for example, for CPU or memory access, are not emulated, but are trapped, which also requires support from CPU vendors.

The guest's OS does not require any modification to run as a hardware virtualized guest.

A virtual machine with an unmodified guest operating system. It is not recompiled for the virtual environment. There may be substantial performance penalties running as a hardware virtualized guest. Enables Microsoft Windows™™ operating system to be run, as well as legacy operating systems. Hardware virtualization is only available on Intel® VT or AMD SVM CPUs.

high availability

High availability (HA) help ensure the uninterrupted availability of a virtual machine. If HA is configured for your virtual machine, and if the Oracle VM Server on which it is running fails or shuts down, the virtual machine is restarted on another available Oracle VM Server in the server pool. The server pool must be clustered. You must enable high availability for both the server pool and the virtual machine.

host computer

The physical computer on which the software is installed. Typically used to refer to either the computer on which Oracle VM Server or Oracle VM Manager is running.

hypervisor

A hypervisor, also called a monitor or Virtual Machine Manager (VMM), is a layer which abstracts the virtual hardware from the real hardware. As such it is the only privileged entity in the system which has full access to real

hardware resources. It controls only the most basic resources of the system, including CPU and memory usage, privilege checks, and hardware interrupts.

Hosted hypervisors are designed to run within a traditional operating system. In other words, a hosted hypervisor adds a distinct software layer to the host operating system, and the guest operating system becomes a third software level above the hardware and the host-based hypervisor. A well-known example of a hosted hypervisor is Oracle VM VirtualBox. Others include VMware Server and Workstation, Microsoft Virtual PC, KVM, QEMU, and Parallels.

Native hypervisors are software systems that run directly on the host's hardware to control the hardware, and to monitor the guest operating systems. The guest operating system runs on a separate level above the hypervisor. Examples of this type of virtualization architecture are Oracle VM, Microsoft Hyper-V, VMware ESX, and Xen.

I

inbound migration lock

A method of protecting virtual machines on an Oracle VM Server by restricting whether new virtual machines can be created on it or migrated to it. By doing this, you can make sure the resources of an Oracle VM Server are available to the virtual machine(s) running on the server. This feature is in addition to the other protection mechanisms such as HA, server pool policies and anti-affinity groups, and overrides any of these other features or policies that might try to create or migrate a virtual machine on the Oracle VM Server.

J

jobs

Jobs are a sequential operations that take place through Oracle VM Manager, such as server discovery, presenting a repository and creating a VM. Jobs are assigned a status that is refreshed according to their progress. A history of all jobs in the environment is stored within Oracle VM Manager.

Since jobs are sequential and sometimes take time to complete, tracking the status of a job allows you to understand what actions the system is currently performing, and which actions are queued to run in sequence after the current job has completed. Jobs also allow you to access system messages that may be useful to debug the failure of an operation.

Most jobs tend to generate events that each have a different severity level.

See Also: [events](#)

L

live migration

Live migration is a process to move a running virtual machine from one Oracle VM Server to another, while applications on the existing virtual machine continue to run. You can only live migrate virtual machines from one Oracle VM Server to another within the same server pool.

local storage

Local storage consists of hard disks installed locally in an Oracle VM Server. Local storage is often not appropriate for production environments, because it prevents or sharply constrains the ability of a virtual machine to run anywhere in the server pool in the event of the failure of the Oracle VM server, which owns the local storage.

logical domain

A virtual machine comprised of a discrete logical grouping of resources, which has its own operating system and identity within a single computer system. Also called a domain.

See Also: [domain](#)

M

master Oracle VM Server

A component of Oracle VM Agent. An application that acts as the contact point to Oracle VM Manager, and to other Oracle VM Agents. Provides virtual machine host load-balancing, and local persistence for Oracle VM Server.

There is only one master Oracle VM Server in a server pool. A physical server can perform as the master Oracle VM Server, Utility Server and Virtual Machine Server simultaneously.

messaging

Oracle VM supports a messaging system that enables communication between the Oracle VM Manager and a guest running within a virtual machine on any Oracle VM Server, as long as the guest has the Oracle VM Guest Additions installed. This messaging system works by sending key/value pairs between the guest and Oracle VM Manager via a secured connection. Messaging allows greater administrative control over virtual machines and facilitates remote and automated configuration of virtual machines as they are started.

migrate

The act of moving a virtual machine from one Oracle VM Server to another, or to the Unassigned Virtual Machines folder. Migration can be performed on either a running or a stopped virtual machine.

move

The act of moving an object from one location to another. This may be moving a stopped virtual machine from one Oracle VM Server to another, moving a virtual machine template from one storage repository to another, or moving an ISO file or virtual disk to another storage location.

multipath

The technique of creating more than one physical path between the server CPU and its storage devices. It results in better fault tolerance and performance enhancement. Oracle VM supports multipath I/O out of the box. Oracle VM Servers are installed with multipathing enabled because it is a requirement for SAN disks to be discovered by Oracle VM Manager

N

non-sparse copy

A clone of the type "non-sparse copy" is a disk image file of a physical disk, taking up the space equivalent to the full specified disk size, including empty blocks.

See Also: [sparse copy](#)

O

OCFS2

OCFS2 is a general-purpose shared-disk cluster file system for Linux capable of providing both high performance and high availability. OCFS2 is developed by Oracle and is integrated within the mainstream Linux kernel. OCFS2 is used within Oracle VM to facilitate clustered server pools, storage of virtual machine images and for the purpose of allowing guests to share the same file system.

A clustered server pool always uses an OCFS2 file system to store the cluster configuration and to take advantage of OCFS2's heartbeat facility. There are two types of heartbeats used in OCFS2 to ensure high availability:

-
- The disk heartbeat: all Oracle VM Servers in the cluster write a time stamp to the server pool file system device.
 - The network heartbeat: all Oracle VM Servers communicate through the network to signal to each other that every cluster member is alive.

These heartbeat functions exist directly within the kernel and are fundamental to the clustering functionality that Oracle VM offers for server pools. The server pool file system should be stored on a separate NFS server or on a small LUN if possible, as OCFS2's heartbeat facility can be disturbed by intensive I/O operations taking place on the same physical storage.

A storage repository configured on a LUN-based repository must be linked to a clustered server pool due to the nature of the OCFS2 file system. As a result, LUN-based repositories cannot be shared between multiple server pools, although it is possible to move an OCFS2 repository from one server pool to another.

For more information on OCFS2, please refer to <https://oss.oracle.com/projects/ocfs2/>.

Oracle VM Agent

An application installed with Oracle VM Server. The Oracle VM Agent receives and processes management requests, and provides event notifications and configuration data to the Oracle VM Manager. Oracle VM Manager manages the virtual machines running on Oracle VM Server by communicating with Oracle VM Agent. It contains three components: master Oracle VM Server, Utility Server, and Virtual Machine Server.

Oracle VM Manager

The Oracle VM Manager is the management platform, which offers an easy-to-use, web-browser interface as well as a command-line interface (CLI). The Oracle VM Manager tracks and manages the resources available in your virtual environment. The user interface, which is an Application Development Framework (ADF) web application, allow you to easily manage Oracle VM Server pools. Manages virtual machine life cycle, including creating virtual machines from templates or from installation media, deleting, powering off, uploading, deployment and live migration of virtual machines. Manages resources including ISO files, templates and shared virtual disks.

Oracle VM Server

A self-contained virtualization environment designed to provide a lightweight, secure, server-based platform for running virtual machines. The Oracle VM Server comprises a hypervisor and a privileged domain (called dom0) that allow multiple domains or virtual machines (that is, Linux, Solaris, Windows, and so on) to run on one physical machine. Includes Oracle VM Agent to enable communication with Oracle VM Manager.

The Oracle VM Server for x86 incorporates an open source Xen hypervisor component, which has been customized and optimized to integrate into the larger, Oracle - developed virtualization server. The Oracle VM Server for x86 is also responsible for access and security management and generally acts as the server administrative entity, because the hypervisor's role is limited.

On Oracle VM Server for SPARC systems, the SPARC hypervisor is built into the SPARC firmware and is generally referred to as the Logical Domains Manager (LDM). As with the Xen hypervisor, each virtual machine is securely executed on a single computer and runs its own guest Oracle Solaris operating system

P

paravirtualized machine (PVM)

A virtual machine with a kernel that is recompiled to be made aware of the virtual environment. Runs at near native speed, with memory, disk and network access optimized for maximum performance.

Paravirtualized guests use generic, idealized device drivers, which are part of the guest's OS. The I/O operations using these generic device drivers are mapped to the real device drivers in dom0. The generic, abstracted drivers in the guest seldom change and provide excellent guest stability. The dom0 domain, alternatively, can use the native hardware vendor drivers, and the guests can safely migrate to another dom0 with slightly different drivers.

For other resources such as CPU and memory, paravirtualized kernels make special “hypercalls” to the Xen hypervisor. These hypercalls provide better performance by reducing the number of instructions and context switches required to handle an incoming request. By contrast, on an emulated (hardware virtualized) guest, driver requests engage the guest’s interrupt handler, increasing the I/O operation overhead.

Q

QEMU

Also referred to as `qemu-dm`, which is the process name. The virtualization process which allows full virtualization of a PC system within another PC system.

S

server pool

Logically an autonomous region that contains one or more physical Oracle VM Servers. Presents a unified view of the storage where the virtual machines reside, and groups the users of these virtual machines into a single community called a *group*, in which each user is a server pool member.

Each server pool can have up to 32 physical servers. Each Oracle VM Server can be a member of only one server pool. The server pool is the operational unit of Oracle VM. Policies are configured and enforced at the server pool level.

server processor compatibility group

A server processor compatibility group is a group of Oracle VM Servers with compatible processors, or CPUs sharing the same processor family and model number. These groups are created to ensure that a virtual machine running on one Oracle VM Server can safely be migrated and continue to run on another Oracle VM Server. Oracle VM Manager automatically creates processor compatibility groups as it discovers servers that have different processor types.

Using Oracle VM Manager you can create custom compatibility groups to improve your ability to do smooth migrations and to group servers according to your own requirements.

service domain

Logical domain that provides devices, such as virtual switches, virtual console connectors, and virtual disk servers, to other logical domains.

sparse copy

A clone of the type “sparse copy” is a disk image file of a physical disk, taking up only the amount of space actually in use; not the full specified disk size.

See Also: [non-sparse copy](#)

Storage Connect

Oracle VM integrates with all types of storage, referred to as *generic storage*, but also provides advanced storage functionality for storage vendors that provide a plug-in to access their storage. This plug-in is part of Oracle VM’s Storage Connect framework.

Oracle VM provides its own Oracle VM Storage Connect plug-in for the Sun ZFS Storage Appliance, and for the Oracle Axiom systems.

storage initiator

To initiate a connection to an iSCSI target, hosts use a client known as a storage initiator. The storage initiator is used to encapsulate SCSI requests in the iSCSI protocol and to handle requests and responses from the iSCSI target. Storage initiators are configured on each Oracle VM Server when an iSCSI Oracle VM Storage Connect plug-in is used.

T

thin clone

A thin clone is a clone of a virtual disk that takes up only the amount of disk space actually in use; not the full specified disk size.

Thin cloning of virtual disks on OCFS2-based repositories is supported. Thin provisioning of physical disks on generic storage is not supported.

U

Utility Server

A component of Oracle VM Agent. An application that handles I/O intensive operations for virtual machines, server pools and servers, for example, copying, moving and renaming files.

There can be more than one Utility Server in a server pool. A physical server can perform as the master Oracle VM Server, Utility Server and Virtual Machine Server simultaneously.

V

virtual disk

A file or set of files, usually on the host file system although it may also be a remote file system, that appears as a physical disk drive to the guest operating system.

virtual machine (VM)

A guest operating system and the associated application software that runs within Oracle VM Server. May be paravirtualized or hardware virtualized machines. Multiple virtual machines can run on the same Oracle VM Server.

virtual machine template

A template of a virtual machine. Contains basic configuration information such as the number of CPUs, memory size, hard disk size, and network interface card (NIC). Create virtual machines based on a virtual machine template using Oracle VM Manager.

VLAN

A layer-2 network may be segregated into partitions, at the switch or router, so that network traffic is isolated to a distinct broadcast domain. VLANs can be tagged so that a trunk can transport data for all of the different VLANs in a network.

VLANs are commonly used in large networks to help simplify network design, provide mechanisms to achieve better scalability, and to improve security.

VM Server

A component of Oracle VM Agent. An application which runs Oracle VM Server virtual machines. It can start and stop virtual machines, and collect performance data for the host and guest operating systems. Enables communication between the master Oracle VM Server, Utility Server and Virtual Machine Servers.

There can be more than one Virtual Machine Server in a server pool. A physical server can perform as the master Oracle VM Server, Utility Server and Virtual Machine Server simultaneously.

W

WebLogic

Oracle WebLogic Server is a platform that includes an application server that can run java applications within a web-based framework. Oracle VM Manager runs as an application within Oracle WebLogic Server, taking

advantage of many of Oracle WebLogic Server's many features to deliver a robust web UI through which Oracle VM can be fully managed.

The installation process behind Oracle VM Manager automatically installs and configures Oracle WebLogic Server on the system where Oracle VM Manager is installed. During this process, a *weblogic* user is set up within Oracle WebLogic Server to manage Oracle WebLogic Server configuration and to administer the underlying system. An *admin* user is also set up within Oracle WebLogic Server and is given permission to access the Oracle VM Manager application. A typical setup uses the same password for both of these users, although this is not always the case and it is possible to configure a different password for each user depending on your requirements.

In general, users of the Oracle VM Manager application should avoid attempting to access the underlying Oracle WebLogic Server, or to change any configuration parameters here without guidance from Oracle Support.

Index

Symbols

/OVS directory, 36

A

- abort
 - jobs, 96
- acknowledge
 - error, 98
 - event, 98
- active backup
 - bonding mode, 49
- adaptive load balancing
 - bonding mode, 49
- agent authentication, 14
- anti-affinity, 80
- API
 - Oracle VM Manager server, 12
- architecture
 - Oracle VM, 5
- assembly, 40
 - virtual machine, 35, 86

B

- back up
 - jobs, 98
 - repository, 42
 - storage repository, 42
 - virtual machine, 88
- bonding mode, 49
 - active backup, 49
 - adaptive load balancing, 49
 - dynamic link aggregation, 49

C

- certificates
 - authentication, 14
- CLI, 13
- clone, 88
 - cold, 88
 - hot, 88
 - ISO file, 41
 - thin, 89
 - virtual machine, 88
 - virtual machine template, 88
- clone customizer, 88
 - virtual machine template, 40
- cluster, 11, 16, 69
 - DLM, 71
 - fencing, 73
 - heartbeat, 70, 71

- services, 71
- cold clone, 88
- components
 - dom0, 7
 - domain zero, 7
 - Oracle VM, 7
 - Oracle VM Agent, 7
 - Oracle VM Server for SPARC, 7
 - shared storage, 7
- configuration file
 - virtual machine, 42
- console, 18, 91
- copy
 - ISO file, 41
 - non-sparse, 89
 - sparse, 89
- CPU pinning, 94
- create
 - server pool, 68
 - storage repository, 37
 - virtual disk, 41
 - virtual machine, 87
 - VNICs, 87

D

- database
 - MySQL, 12
- database repository, 12
- delete
 - VNICs, 87
- DHCP, 48
- Distributed Power Management, 78
- Distributed Resource Scheduler, 78
- DLM, 71
- dom0, 9
- domain, 9
- domU, 10
- DPM, 78
- DRS, 78
- dynamic link aggregation
 - bonding mode, 49

E

- edit
 - virtual machine, 90
- error
 - acknowledge, 98
 - event, 98
- event
 - acknowledge, 98
 - error, 98

F

- features
 - Oracle VM Manager, 20
- fencing, 73
- fibre channel, 27
- file server
 - discover, 28
- file system
 - refresh, 28
 - refresh server, 28
 - select, 28
 - server pool, 32

G

- guest, 9

H

- HA, 76
 - enabling, 68
- hard partitioning, 94
- hardware virtualization
 - overview, 84
- heartbeat, 70
- high availability, 76
- hosted
 - hypervisor, 3
- hot clone, 88
- HTTPS certificates, 14
- hypervisor, 3, 8
 - hosted, 3
 - native, 3
 - SPARC, 9
 - Xen, 9

I

- import
 - ISO file, 41
 - template, 39
- inbound migration lock
 - Oracle VM Server, 76, 78, 80, 92
- installing paravirtual drivers, 85
- IP address, 48
- IPS
 - Oracle VM Server updates, 65
- iSCSI, 27
- ISO file
 - clone, 41
 - copy, 41
 - import, 41
 - virtual machine, 35, 86
- iso_pool, 37

J

- jobs
 - abort, 96
 - events, 98
 - failure, 97
 - operations, 97
 - recurring, 98
 - resource locks, 97
 - states, 96

L

- live migration, 78
- local storage, 25

M

- management domain, 9
- master, 68
- master server
 - server pool, 11
- message
 - send to virtual machine, 90
- migrate
 - virtual machine, 90
- move
 - virtual machine, 89, 90
- multipath, 30
- multiple-users, and jobs, 97
- MySQL
 - database, 12

N

- native
 - hypervisor, 3
- network
 - bond, 44, 48
 - bonding modes, 49
 - bridge, 44, 56
 - channels, 46, 52
 - components, 44
 - data link aggregation, 48
 - design, 17
 - DHCP, 48
 - functions, 17, 46, 52
 - intra-server, 61
 - IP address, 48
 - logical, 44, 45
 - networking, 17
 - overview, 43
 - physical, 45
 - planning, 57
 - port, 44
 - preparation, 57
 - usage, 46, 52

- virtual LAN, 17
- VLAN, 44
- NFS, 26, 28
- non-sparse copy, 89

O

- OCFS2, 16, 69
- Open Virtualization Format, 40
- Open Virtualization Format Archive, 40
- Oracle Enterprise Manager 12c, 21
- Oracle VM
 - architecture, 5
 - authentication, 14
 - components, 7
 - encryption, 14
 - Oracle Enterprise Manager 12, 21
 - Oracle VM Manager, 11
 - support, 21
- Oracle VM Agent
 - overview, 10
- Oracle VM Guest Additions, 20
- Oracle VM Manager
 - agent authentication, 14
 - features, 20
 - Oracle VM, 11
 - performance manager, 98
- Oracle VM Manager Command Line Interface, 13
- Oracle VM Manager database, 7
- Oracle VM Manager server
 - API, 12
- Oracle VM Server
 - inbound migration lock, 76, 78, 80, 92
 - maintenance, 65
 - overview, 8
 - software updates, 65
 - update, 98
- Oracle VM Storage Connect plug-in, 28
- Oracle VM Utilities, 13
- OVA files, 40
- OVF files, 40

P

- paravirtual drivers, 85
- paravirtualization
 - overview, 84
- plug-ins, 28
- policies
 - DRS/DPM, 78
 - Network, 78
 - server pool, 78
- present repository, 37
- processor compatibility groups, 80

R

- recurring
 - jobs, 98
- refresh
 - NFS, 28
 - storage, 28
- refresh server, 28
- repository, 32, 35
 - back up, 42
 - create, 37
 - present, 37
 - refresh, 98
 - requirements, 36
 - structure, 36
 - virtual machine resources, 35, 38, 86
 - virtual machine template, 38
- resource locking, 97
- running_pool, 37

S

- SAN, 27
- SAN server
 - discover, 28
- seed_pool, 37
- serial console, 91
- server
 - shutdown, 78
- server pool, 11, 67
 - cluster, 16, 69
 - create, 68
 - file system, 32
 - master server, 11
 - pool file system, 16
 - unclustered, 11, 74
 - virtual IP, 68
- sharedDisk, 37
- snapshot, 88
- SPARC
 - hypervisor, 9
- sparse copy, 89
- SSL
 - certificates, 14
- storage, 23
 - configuring, 32
 - fibre channel, 27
 - file server, 28
 - iSCSI, 26
 - local, 25
 - NFS, 26
 - OCFS2, 24
 - overview, 23
 - plug-in benefits, 16
 - present repository, 37

- repository, 23, 32, 35
- resources, 23
- server pool clustering, 31
- types, 24
- VM resources, 16
- storage array
 - discover, 28
 - required information, 28
- Storage Connect
 - benefits, 16
- storage repository
 - back up, 42
 - create, 37
 - present, 37
 - requirements, 36
 - structure, 36
 - virtual machine resources, 35, 86
- support, 21

T

- template, 39
 - clone, 88
 - create, 39
 - import, 39
 - virtual machine, 35
- templates
 - virtual machine, 86
- thin clone, 89

U

- user domain, 10

V

- virtual cpu, 92
- virtual disk
 - create, 41
- virtual IP, 68
- virtual machine
 - assembly, 18
 - back up, 88
 - clone, 18, 88
 - clone customizer, 88
 - configuration file, 42
 - console, 18, 91
 - cpus, 92
 - create, 39, 87
 - create using template, 39
 - create VNICs, 87
 - edit, 90
 - hard partitioning, 94
 - huge pages, 93
 - importing as assemblies, 40
 - large pages, 93

- migrate, 90
- move, 89, 90
- paravirtual drivers, 85
- resources, 35, 86
- send message, 90
- serial console, 91
- snapshot, 88
- template, 18, 86
- templates, 35
- virtualization modes, 19
- virtual machine template
 - clone, 88
 - clone customizer, 40, 88
 - import, 39
 - manage, 38
 - using, 39
- virtualization
 - hypervisor, 3
- virtualization modes
 - hardware virtualized, 19
 - paravirtualized virtualized, 19
 - software virtualized, 19
- VLAN, 50
 - interfaces, 51
- VLAN interfaces, 50
- VLAN segment, 50
- VM
 - migrate, 90
 - migration, 78
- VMware
 - importing virtual machines, 40
- VNICs
 - create, 87
 - delete, 87

X

- Xen
 - hypervisor, 9

Y

- YUM, 98
 - Oracle VM Server updates, 65