



An Oracle Technical White Paper
April 2013

Oracle VM: Designing, Creating and Testing an Oracle VM 3.2 Environment

Introduction	1
Stage 1: Designing your Oracle VM environment	3
Oracle VM Management Design	3
System Patching.....	7
Oracle VM Server Pool Design Considerations	7
Oracle VM Server Synchronization	8
Hard Partitioning for Oracle Licenses	9
Oracle VM Server Types within Server Pool	9
Oracle VM Shared Storage Design.....	9
Storage Connect Framework	9
Storage repository	10
Advanced Strategies for Shared Storage Repositories	12
Presenting Virtual and Physical Disks to Virtual Machines.....	12
Basic Sizing Considerations for a Storage Repository	13
Using DM- Multipath with the storage repository	14
Using DM- Multipath with Virtual Machines.....	14
Best Practices for Storage	15
Oracle VM Networking Design.....	18
Stage 2: Creating your Oracle VM environment.....	20
Hardware Preparation	20
Software Preparation.....	20
Deploying Oracle VM Manager as a VM on Oracle VM	21
Create an Oracle VM Server Pool	22
Stage 3: Testing your Oracle VM Environment	25
Appendix A – Essential Reference Information.....	29
Configuration Maxima.....	29
Supported Guest Operating Systems	29
Supported Hardware	29
Important Reference Sites	29
Appendix B – Deployment Planning Checklist	30

Introduction

Oracle VM is an enterprise-class server virtualization solution comprised of [Oracle VM Server for x86](#), [Oracle VM Server for SPARC](#) and [Oracle VM Manager](#). Oracle VM 3 reflects Oracle's strategic commitment to deliver Application Driven Virtualization—Virtualization that makes the entire enterprise software and hardware stack easier to deploy, manage, and support so IT and business can be more agile.

The paper here will focus on the Oracle VM Server for x86 and Oracle VM Manager. This document describes a plan you could follow to ensure that your Oracle VM deployment is organized, repeatable and tested before you mark it ready for production. The plan is broken down into three key stages, each of which may touch many aspects of the overall deployment.

Contained within this document are a number of tips, hints, instructions and even a test plan check-sheet for you to use when planning, building and testing your Oracle VM environment. At the end of the document are several Appendices containing useful reference material related to the deployment of Oracle VM, including some reference configuration files.

Any plan has a number of stages, phases and steps that lead to the successful completion of that plan. We believe that there are three key stages to the successful deployment of Oracle VM:

- Stage 1: Designing your Oracle VM environment
- Stage 2: Creating your Oracle VM environment
- Stage 3: Testing your Oracle VM environment

Each stage is self-contained but requires the total completion of all previous steps; these stages cannot be run in parallel at any point. Attempting to do this has resulted in numerous

failed or buggy deployments, especially where assumptions have been made about the ability to test certain aspects in isolation of the whole.

This document follows from a previous edition written for [Oracle VM 2.2](#) and has been completely updated to reflect the significant differences between Oracle VM 2.2 and Oracle VM 3.2 so even if you are familiar with the concepts of the previous edition you should review this new one.

Stage 1: Designing your Oracle VM environment

Obviously, you have a reason for using Oracle VM and this will have influenced your hardware, storage, networking and software choices up to this point so we are not going to go into the physical installation of the hardware and network here; rather, we are going to assume that you know how to do this or have this done to your specification. Having said that, there are three key design considerations affecting your hardware infrastructure that need to be addressed at this point:

- Oracle VM Management design
- Oracle VM Shared Storage design
- Oracle VM Networking design

These designs can be addressed individually although they will influence the outcome of each of the other designs. It is entirely possible that different teams will have overall responsibility for one or more aspects of these designs, which is why it is essential that someone take overall responsibility for the Oracle VM environment design as a whole. We can tell you from experience that design decisions made outside the overall picture can have serious negative consequences later.

One classic example is for the storage team to just provide the Oracle VM team with a large collection of small LUNs rather than one large one. They have no reason for this other than they thought it best even though they had no idea about Oracle VM or what the team was trying to achieve. This can lead to much wasted space, excessive network overhead and management complexity without any actual benefit.

Oracle VM Management Design

Oracle VM 3.2 comprises of a server component that takes over the whole of a physical computer in order to run virtual machines (VMs) and a manager component that runs on a usual operating system (OS) host as an application, which is used to create, configure and manage one or more of the servers. Oracle VM 3.2 Manager not only provides a browser based UI but actually embodies a complete set of components and APIs designed to manage the complete lifecycle of a virtualized environment. This engenders perhaps the biggest difference between Oracle VM 3 and its earlier incarnations, the fact that other management frameworks can manage Oracle VM 3 environments in combination, or even simultaneously, as well as the standard Oracle VM Manager 3.2 UI. Oracle Enterprise Manager 12c (EM12c), in both its Cloud Control and Ops Center incarnations, can be used to build, configure and manage Oracle VM 3.2 environments by simply configuring their respective Oracle VM plug-ins to point to one or more Oracle VM Managers. Each of these EM12c options has its own values and which one you use, if either, will depend on your particular situation and we will not go into these in this document suffice to say that either or both can perform the same functions and Oracle VM Manager 3.2 is always required to be installed and operating to manage any Oracle VM 3.2 infrastructure. All the Oracle VM design considerations and choices reviewed in this document will be the same irrespective of whether you use Oracle VM 3.2 Manager on its own or with EM12c Cloud Control or Ops Center. Readers of the previous edition of this document should note that EM12c is incompatible with any version of Oracle VM prior to Oracle VM 3 and all versions of Oracle VM 3 and newer are incompatible with versions of Oracle Enterprise Manager older than 12c.

Oracle VM Management Topology

Every aspect of Oracle VM 3.2 environments is managed and controlled using Oracle VM Manager; Oracle VM Servers should be considered stateless and should no longer be logged into directly. Oracle VM Manager discovers Oracle VM Servers and storage devices, creates Server Pools, configures every aspect of those servers, storage devices and pools, creates virtual machines, starts, stops, clones and migrates those VMs. Oracle VM Servers and VMs will remain running if Oracle VM Manager stops running temporarily, although operation of the environment using external management systems such as EM12c will be blocked until the Oracle VM Manager is operational again. A paper discussing how to deploy Oracle VM in a highly available DR type environment is available: [Oracle VM 3: Integrating Oracle VM into a Disaster Recovery Solution using SAN \(PDF\)](#).

Oracle VM Manager stores its management data in a database, which can be either installed locally with the Oracle VM Manager software or remotely. This is entirely up to you as an implementer and your choice has negligible impact on the operation or ongoing maintenance of Oracle VM Manager. Our personal choice tends to be to install and use a local database, as it requires fewer components. Choice of which database is between various editions of MySQL and Oracle Database and your selection will be based on how large you anticipate your environment becoming. For most environments the default MySQL local database install or Oracle Database 11gR2 Standard Edition with an initial allocation of 100GB will be perfectly adequate.

Given the relationships between the Oracle VM Manager, Oracle VM Servers and the storage devices, there are many approaches to designing a management topology; each Oracle VM manager can control multiple server pools, even across a WAN connection (and each EM12c can manage multiple Oracle VM Managers). Usually, the Oracle VM manager is installed on a separate physical server, although it can be deployed in a VM, the process of which is documented in the manuals but also briefly discussed in the section, Deploying Oracle VM Manager as a VM on Oracle VM on page 21. The advantage of using dedicated servers is that no aspect of management of the Oracle VM environment is beholden to the location of the running Oracle VM Manager, including operations such as Oracle VM Server maintenance or upgrade. The options are:

- Centralized management deployed as an Oracle VM guest on a server pool (Figure 1)
- Centralized management deployed on an independent physical server (Figure 2)
- Distributed management deployed as an Oracle VM guest on each server pool (Figure 3)
- Distributed management deployed on independent physical servers (Figure 4)

Centralized management deployed as an Oracle VM guest allows Oracle VM manager to be part of a highly available cluster taking advantage of Oracle VM high availability feature while providing a single point of maintenance.

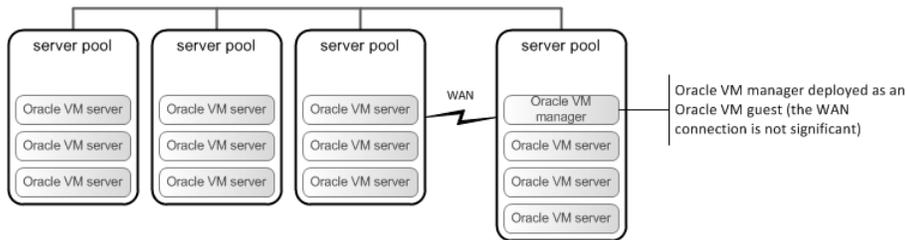


Figure 1: Centralized management deployed as an Oracle VM guest

Although Oracle VM manager would not be highly available, centralized management using a physical server that is independent of any Oracle VM environment is also an acceptable solution. Figure 2 illustrates the concept. Note that it does not matter which network the independent server hosting Oracle VM manager as long as it can reach all Oracle VM servers and guests being managed.

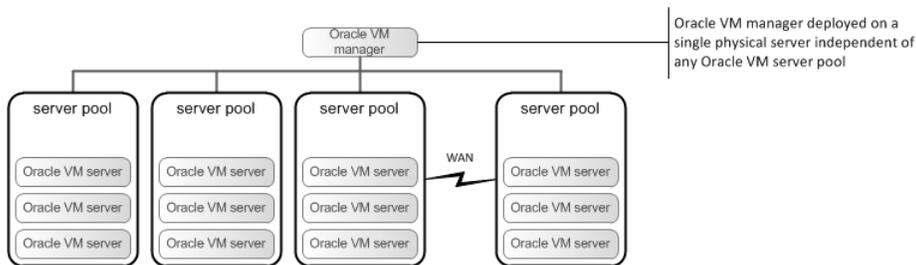


Figure 2: Centralized management deployed on an independent server

The next two examples of Oracle VM management topology are also fine, but are less desirable since either of the schemes is harder to maintain as far as patching and consistency are concerned. Oracle does not recommend either of these management topologies, but they are supported. If you decided to use a distributed management scheme, then Oracle would recommend the following since it takes advantage of Oracle VM's high availability feature.

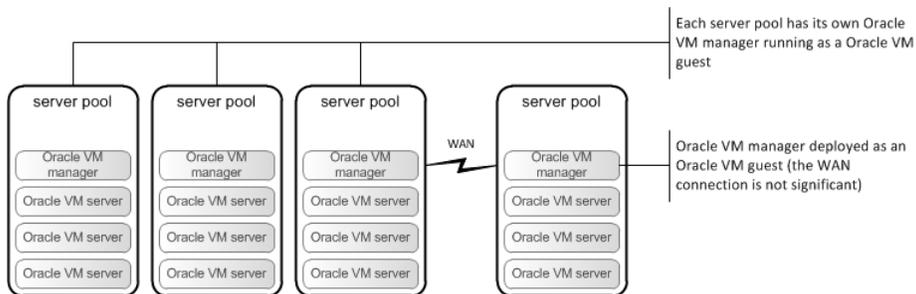


Figure 3: Distributed management topology deployed as Oracle VM guests

The other less desirable management topology is one in which each server pool is managed by its own Oracle VM manager running on a physical server that is independent from any Oracle VM environment as shown in Figure 4

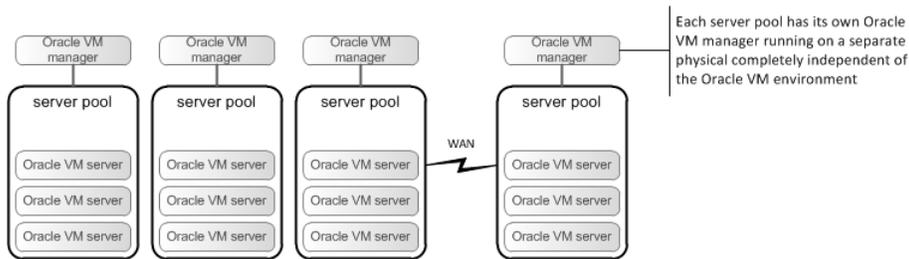


Figure 4: Distributed management topology

Management Network

The next thing to decide is how Oracle VM manager is going to communicate with your Oracle VM servers. This communication has its own dedicated Oracle VM network channel and is configured in the Oracle VM Manager networking screens. The options are:

- In-band network management
- Out-of-band network management

In-band management, illustrated in Figure 5, is pretty typical and simply means the IP address for the Oracle VM Manager is on the same subnet as the primary network interface or virtual bridge. In other words, the IP address for the Oracle VM Manager is on the same VLAN used by the typical end user to access Oracle VM Guests and other servers.

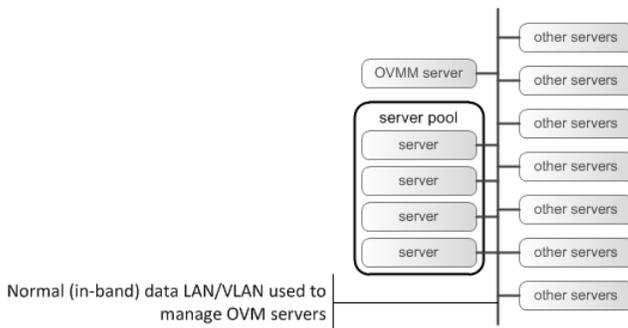


Figure 5: In-band network management

Out-of-band network management, illustrated in Figure 6, means the IP address for the Oracle VM Manager is on a network interface or virtual bridge that is dedicated to just managing servers and is usually restricted to the systems administrators only. This would be a VLAN that is not accessible to the typical end user.

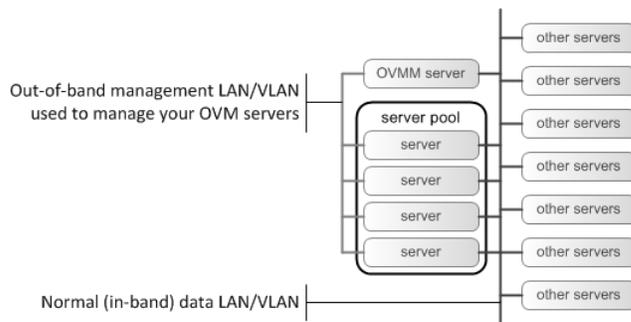


Figure 6: Out-of-band network management

In either model, if you plan to use a *tagged* VLAN for the server management network channel you will need to decide which IP subnet and VLAN segment will be used *before* you install the Oracle VM Servers as this is one of the installation configuration options rather than one that is configured post installation.

System Patching

As part of its centralized management capability, Oracle VM Manager monitors your Oracle VM Servers to see if there are any updates available to be applied to them and, if there are, it can apply them directly. In order to do this you need set up local YUM repositories, registered as such with ULN or Oracle’s public YUM repository, to download the latest updates and patches into a local store and to configure your Oracle VM Manager to use this local YUM repository. Clearly, you will need to incorporate these into your management (and likely, network) design. You will likely find it convenient to use local YUM repositories to manage your local Oracle Linux patches and updates in addition to Oracle VM, as they both use the same systems. Additionally, EM12c can use these same repositories as the source of its Oracle Linux patches. Instructions on how to do this can be found at:

[\[http://www.oracle.com/technetwork/topics/linux/yum-repository-setup-085606.html\]](http://www.oracle.com/technetwork/topics/linux/yum-repository-setup-085606.html)

Oracle VM Server Pool Design Considerations

Oracle VM Manager is supplied with an embedded MySQL database. Enough for most environments, this can still be replaced with a larger Oracle Database, depending on your requirements, operating considerations and number of Oracle VM components (servers, pools and VMs) you plan to manage. Runtime only licenses for Oracle Database are included as part of Oracle VM Manager along with those for the embedded Oracle WebLogic server.

Oracle VM servers can perform one or more of three duties within a server pool:

- **Server Pool Master:** one and only one server will always be the Master Server in a Server Pool. If the designated server pool Master Server fails, one of the remaining nodes in the server pool will automatically assume the role and responsibilities. The server pool Master Server decides upon which node any new VM is started and coordinates the HA facility of any Clustered Server Pool. This is the only duty that is automatically reassigned in the event of a failure. This role can be

specifically assigned to a server in a Server Pool by editing that Server Pool in Oracle VM Manager and selecting the desired Oracle VM Server from the drop-down list.

- Utility Server: servers marked as Utility Servers can perform I/O intensive tasks for the server pool such as VM creation, VM Template creation, cloning, etc. By default, every server in a server pool is assigned the Utility Server role but editing that server in Oracle VM Manager can change this.
- VM Server: servers marked as VM Servers are capable of running VMs within the server pool. By default, every server in a server pool is assigned the VM Server role but editing that server in Oracle VM Manager can change this.

These roles need to be designed according to anticipated demand and operational conditions. With only a few Oracle VM servers in a server pool it is quite common to allocate every server the duties of Utility and VM Server so that every server can perform any job. As you start to have larger server pools you may want to restrict which servers are assigned the Utility Server role so that the remaining VM Servers are not impacted by the creation of VMs, templates and clones (over and above the load it applies to the shared network and storage subsystems themselves). You will need to incorporate these considerations into your management design.

When you create a Server Pool you assign it as either clustered or not. A clustered server pool is capable of supporting highly available VMs, where VMs marked as enabled for high availability will be restarted on alternative servers in the server pool if the original server they are running on fails. You cannot convert a non-clustered Server Pool into a clustered one once it has been created, so you need to plan for this up front. Each Clustered Server Pool requires its own dedicated Server Pool Filesystem, which should be sized at 12GB and can be either SAN or NAS based, including NFS. This Server Pool Filesystem will be used by the cluster to store data on HA enabled VMs and their status as well as a filesystem based cluster heartbeat. The Cluster Heartbeat network channel will be used by the servers in the cluster to maintain their network cluster heartbeat (as you may imagine from the title). Part of your design consideration should be to ensure that these cluster heartbeats are kept apart from high-load devices and networks to ensure that their latency is not compromised to an extent they trigger an HA event (known as fencing in clustering terminology) whereby an apparently healthy server reboots itself thinking it is no longer part of the cluster. You should take special care to isolate the Cluster Heartbeat channel from the Live Migrate channel if you anticipate extensive use of Live Migrations (such as when employing the DRS or DPM server pool policies).

Oracle VM Server Synchronization

In order to keep the cluster synchronized you are required to synchronize the clocks of the Oracle VM servers in each server pool. This is achieved by using a Network Time Protocol server (or servers) and configuring the NTP service on each Oracle VM server to point to those same NTP servers by simply configuring and using the NTP section in the Oracle VM Manager tools and resources section of the UI. There are a number of publicly available Internet NTP servers or you can point to one inside your LAN; you can even configure the Oracle VM Manager Oracle Linux host to be an NTP server (as long as it's running on a physical server, as all VMs are prone to clock-drift to a lesser or greater degree).

Hard Partitioning for Oracle Licenses

If you are planning to use Oracle VM to provide Hard Partitioning for Oracle per-processor licensed products you will need to design how the individual cores in each Oracle VM server will be allocated to those hard partitions. The specifics for hard partitions along with the requirements can be found here:

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

Oracle VM Server Types within Server Pool

When planning your Oracle VM server pools you should take into account the type of machines and storage that you intend to use for the workloads in each Oracle VM server pool. One important consideration for any Oracle VM server pool is that for Live Migration the processors in the source and target servers must be the same to ensure that no corruption can occur within running VMs if they are moved to another Oracle VM server within the server pool. For example, Oracle databases make extensive use of advanced chipset features including SMMD instruction sets based on what they find available when they start up – if they try to execute one of these instructions on a CPU that does not have them the database will crash.

Oracle VM Manager will determine which servers in any server pool are suitable targets for any given Live Migration attempt in realtime; if any target is unsuitable it will tell you why and even possible ways to resolve the issue. Best practice is to ensure that you always add servers to the same server pool in the same CPU family and model pairs at least. In this way there will always be at least one other server in the pool that is Live Migration compatible with any other. There are server side settings that can be modified to force Live Migration compatibility across heterogeneous processors but such modified server pools will become unsupported configurations.

Oracle VM Shared Storage Design

Storage concepts surrounding Oracle VM server and Oracle VM manager implementations are probably the single most important thing to understand, yet the most complex aspect of the overall design.

There are only a few things that need to be taken into account with storage, but there are a multitude of different options and concepts that need to be thoroughly understood before designing a server pool. Basically, the following things need to be considered:

- Preparing your shared storage devices
- Presenting a shared storage repository to Oracle VM servers
- Presenting additional “local” and/or “shared” storage to Oracle VM guests

Storage Connect Framework

All storage devices that can be used by Oracle VM are managed through Oracle VM Manager directly. When you install a new Oracle VM Server and discover it from your Oracle VM Manager, one of the things it does is take a look to see what, if any, storage devices it can see, reporting this back to Oracle VM Manager. Oracle VM sees all its possible storage devices through an open storage architecture

called the Storage Connect Framework. Storage vendors can (and do) write their own Storage Connect Plugins, which when installed according to their design into Oracle VM 3, exposes the set of features and abilities that storage device can provide to Oracle VM and its users. One of the first things to check, when planning a new Oracle VM 3 deployment, is to see if your storage device vendor provides relevant storage connect plugins for you to use; depending on your stage of development you may even choose to examine this before you select your storage devices.

Out of the box, Oracle VM 3 can work with generic SAN, iSCSI and NFS storage devices, and in almost all cases, these will work with whatever storage devices you use. All the abilities detailed in this document are available through these generic storage connect plugins, including zero-block, instant live clones.

To use a SAN device, you will need to decide which servers will be able to see it, which you want to access it and which you want to act as the admin servers for them. At the simplest level, one would zone all LUNs to all servers and allocate all servers as admin servers. This gives the greatest flexibility and allows you to control the ‘virtual zoning’ of the servers, pools and VMs by creating and carefully sharing storage repositories or physical disks according to your design.

It is very important that you carefully review the Oracle VM documentation regarding storage and repositories *before* making any design decisions, as this could save you immeasurable time and effort later in your deployment lifecycle. Note that some decisions are final and irrevocable within the context of an existing deployment; the only way to reverse them is to start again from scratch.

Storage repository

A storage repository is simply a directory structure that resides on a single disk (or NFS export) that can be seen by all the Oracle VM servers in a Server Pool controlled by an Oracle VM manager. The storage repository contains the configuration files and image files for the Oracle VM guests, “local” and “shared” virtual disks (sparse files) and any other resources needed to run and manage Oracle VM virtual machines.

In order to create a resource pool of processors, memory and storage using Oracle VM you simply install Oracle VM on each physical server and allocate at least one shared storage device as a Storage Repository to be shared equally between each of the Oracle VM servers. You tell all the Oracle VM servers that they are a shared pool by creating a Server Pool using Oracle VM Manager. You can have any number of Server Pools but each one should contain no more than 64 Oracle VM servers for unclustered (non-HA) server pool or 32 Oracle VM servers for clustered server pool (HA enabled). All the Oracle VM servers in a Server Pool knows about and talks to all the others to keep a consistent map of what they are doing. In this way, new VMs are started on the Oracle VM server within the pool that has the most available resource (this behavior can be altered if desired). If a server pool is clustered, then any VMs marked as highly available that fail for any reason will be automatically restarted – on a different Oracle VM server if required. Additionally, running VMs within a Server Pool may be moved between Oracle VM servers within that pool without interruption through the use of Live-Migration.

All physical disk based storage repositories, whether shared or local, will be formatted as OCFS2

filesystems and secured so that only those Oracle VM Servers they are allocated to are able to mount them and view the contents thereof. NFS based repositories will rely on the NFS security model. OCFS2 filesystems can be very large but we recommend you create storage repositories of no larger than 8TB for most purposes. One of the nice innovations of Oracle VM 3 is that it is now very flexible about how storage repositories can be shared between both servers and even server pools. You can restrict which servers in a server pool can access shared repositories and you can share NFS repositories across server pools. These ideas are explored further below.

You must decide how you want to present storage to each VM server. Options include network file system (NFS), and block level storage such as direct attached storage (DAS), SCSI over Ethernet (iSCSI) or SCSI over Fibre Channel (FCP). One important aspect of shared repositories and clustered server pools is that non-clustered server pools can only use shared NFS repositories. In fact, non-clustered server pools cannot use anything other than NFS repositories; if a single server wants to use local storage it will need to be created as a singleton clustered server pool.

Also worth noting here is that local storage repositories have a couple of other restrictions that need to be planned for: (i) only whole, unpartitioned, physical disks can be used for local storage, and (ii) local SAS disks are not supported as local storage repositories. This means that you cannot use the unused parts of Oracle VM Server boot disks as any form of storage repository, so turning all the drives in a machine into a single huge RAID device just results in wasted space. You can use a local SAS disk as the boot disk for Oracle VM Server, which might allow you to use the remaining non-SAS disk as a local storage repository. Remember that any time you use a local storage repository that VMs and data located there are permanently fixed to that Oracle VM Server for normal purposes.

Here are some additional considerations to ensure this all works:

- For FCP devices you need to ensure every LUN is presented to every Oracle VM server and configure the Zones to ensure that every possible path is exposed and weighted appropriately; if you ever encounter a path that one Oracle VM server can see but that another cannot, you will have a problem when the system is running. You will also need to mask any LUNs that will be kept for 'private' storage for individual VMs or will be otherwise unused by the Oracle VM Server Pool as shared Storage Repositories. If you are planning to use SAN boot for the Oracle VM servers themselves you will need to set this up too and ensure that you select the relevant installation options when installing Oracle VM Servers.
- When using NAS/NFS storage repositories you will most likely want to use load balancing bonded network links in each Oracle VM Server with Jumbo frames enabled [MTU 9000] to get the most throughput and availability out of the environment. Note that the Storage network channel is redundant at the time of writing – check the release notes of any version of Oracle VM beyond 3.2.1 when implementing to see if it has become functional when planning your deployment.
- For DAS devices you need to ensure they can be access simultaneously by the number of processors you are planning to turn into Oracle VM servers; shared SCSI, for instance, can only be shared by a maximum of two servers.

- If your storage vendor supplies a Storage Connect Plugin for Oracle VM, you will need to install its components on each Oracle VM Server and your Oracle VM Manager according to the instructions supplied with that plugin.
- Storage vendors may even supply additional components to be installed into Oracle VM Servers, such as EMC with its Powerpath add-on to replace dm-multipath for some of its arrays.

Advanced Strategies for Shared Storage Repositories

Taking advantage of the flexible shared storage model of Oracle VM 3 we can conceive new usage models utilizing different shared storage repositories.

One strategy we recommend is to always have at least one NFS repository shared with all the servers in all the server pools. You can place common, base templates here, boot ISOs and similar so that every server pool can create VMs from the same base images. You can also use it as an intermediary location for VMs from any location to any other location. For example, you could have three server pools: development, test and production. You could create a new version of the product in development and make a direct copy of the entire VM into the shared global NFS repository by simply cloning it with that repository as the target. From there you can either move it into the test pool or perhaps even better, use it as the template to create clones into the test repository. Once the tests are complete you could clone the finished VM directly into production, knowing that the thing you are deploying is actually the thing you tested.

Presenting Virtual and Physical Disks to Virtual Machines

Virtual machines can use virtual disks, physical disks or a combination of the two – the choice is up to you but here are some considerations that may help you to decide. One of the important things to understand is that there are no real differences between VMs that use physical or virtual disks, they can all be created, cloned, shared and deleted; as long as you have the physical resources available to do so.

Virtual Disks are the ultimate in flexibility:

- You can create a virtual disk that is 2TB in size but only consumes the physical data blocks required to hold the data actually placed into it.
- You can take an instant copy of a virtual disk by cloning it (this doesn't work on NFS shared repositories), and this copy takes up zero additional physical data blocks until its data is changed or more it put into it.
- You can take clones of running VMs without interrupting them.
- You can deploy hundreds of new VMs almost instantly.

Physical Disks are ultimate in performance and security:

- You can give a collection of raw LUNs to an Oracle DB VM to manage itself through ASM exactly the same as if it was a local server, but you can move that VM between servers without changing anything.
- You can allocate a physical LUN exclusively to a single VM, wherever that VM actually runs.

- You can clone physical disks into virtual disks (as long as you have the space).

Things you can do with virtual and physical disks:

- You can choose to share physical and virtual disks between VMs to make clusters such as RAC.
- Create a Template from an existing VM (as long as you have the space).

Physical and virtual disks are allocated to VMs in exactly the same way; virtual disks can be created on the fly as needed, physical disks can be selected from the pool of available unused or shared physical disks remaining in the SAN/NAS zone.

Basic Sizing Considerations for a Storage Repository

Now that you are familiar with the role of the storage repository and the differences between virtual disks and physical disks, you can think about the size of the disk space needed for the storage repository. The storage repository contains all the files associated with each Oracle VM guest. Sizing requirements will vary widely depending on your specific implementation, but the following table should provide a general idea of the information you will need to gather in order to calculate a repository size that will fit your needs:

There are many variables involved that will impact the size of your repository – too many to cover in this particular document, but an example of calculating size would probably be a good idea. As a very rough guide to calculating the size requirements, let's assume the following in a very simplistic scenario:

SIMPLE EXAMPLE OF CALCULATING REPOSITORY SIZE

QUANTITY	OBJECT	UNIT (MB)	SIZE	EXTENDED (MB)	SIZE
3	Oracle VM servers in a server pool		N/A		N/A
6	Oracle VM guests running any supported operating systems		12,288		73,728
1	Additional 36 gigabyte local virtual disk per Oracle VM guest		36,864		221,184
1	Additional 36 gigabyte shared virtual disk accessible by all six VM guest images		36,864		36,864
10	Raw physical disks per Oracle VM guest to be used by Oracle databases for data files, redo logs, etc (not counted because they are not part of the storage repository)		N/A		N/A
3	Oracle VM guest templates for quick deployment of new guest images		12,288		36,864
2	ISO images of an operating system such as Windows		430		960
6	Additional space to allow for the creation of 6 more Oracle VM guests (including 1 additional local virtual disk per guest)		49,152		294,912
Total repository size needed (rounded up to nearest gigabyte)					664,512 (MB)

650 (GB)

So, in this very simple scenario, you would need to create a 650 Gigabyte (plus overhead for file system metadata) LUN/Disk or NFS file system and make it available to all three Oracle VM servers.

Using DM- Multipath with the storage repository

DM-Multipath is automatically used for shared disks being presented to the Oracle VM servers using DAS, iSCSI or FCP. The default setting should be fine for almost any device you meet but this area is one of the very few times when you may be required to edit a file on an Oracle VM server directly in order to modify the multipath.conf entries for your SAN.

Using DM- Multipath with Virtual Machines

There is usually some confusion over how device mapper Multipath (DM-Multipath) is incorporated into the operating system contained within VMs. The short explanation is that DM-Multipath is not used on the operating systems contained within VMs because the underlying Oracle VM server (physical server) is already providing multiple paths to a single disk using DM- Multipath. Therefore, VMs will only see and use single paths even if those single paths represent multiple physical paths aggregated into the single “physical” disk that is being presented by the Oracle VM server to the OS contained within the VM.

DM-Multipath is relevant only to Oracle VM servers (physical server), and does not pertain in any way to VMs. Just to reinforce a concept discussed earlier, DM-Multipath is only relevant to “physical” disks, not “virtual” disks created within the storage repository and managed by Oracle VM manager. So, the disks used to create the storage repository itself might use DM-Multipath to provide protection from failed controllers but DM-Multipath will not pertain to virtual disks, which as the reader might recall, are simply sparse files contained within the storage repository.

The key to understanding the relationship between DM-Multipath and VMs is illustrated in the following two figures. The first figure shows a single disk being presented to a single VM. Note that you do not attempt to set up DM-Multipath in the VM’s operating system; the OS running inside the VM guest does not need to duplicate the hardware path failover mechanism already being handled by DM- Multipath on the physical Oracle VM server.

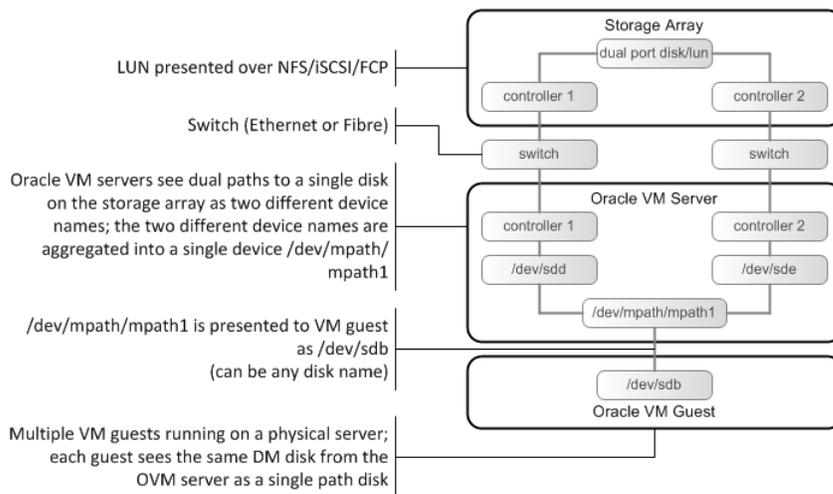


Figure 7: DM-Multipath with a single VM guest

The second figure shows the same single disk being presented to multiple VMs as a shared physical disk. Any disks that you want to present to VMs as shared disks will need marked as such and added to each of those VMs using Oracle VM Manager.

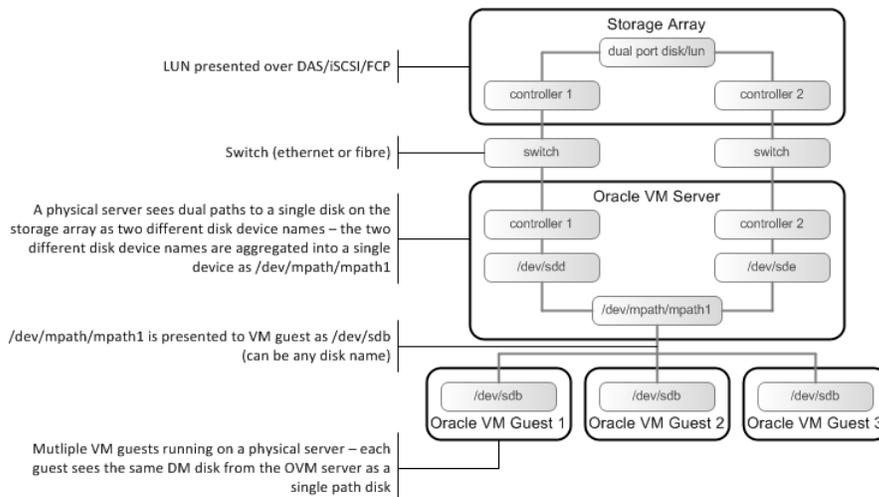


Figure 8: DM-Multipath with multiple VM guests

Best Practices for Storage

The following subsections discuss some Oracle recommended best practices related to storage on your Oracle VM clustered environment.

File Systems for Shared Virtual Disks on Oracle VM Guests

There are some very subtle differences between the various scenarios where it makes sense to use OCFS2; you may want to review this section more than once to get it clear in your own mind. The file system and the way you use shared virtual disks presented to Oracle VM guests will depend on the

storage repository:

- OCFS2 can only be used on shared physical disks.

If you decided to create a storage repository using FCP, iSCSI or DAS, then Oracle VM will have used OCFS2 for the file system of the storage repository. The following figure illustrates shared virtual disks being presented to multiple VMs from a storage repository that is formatted using OCFS2.

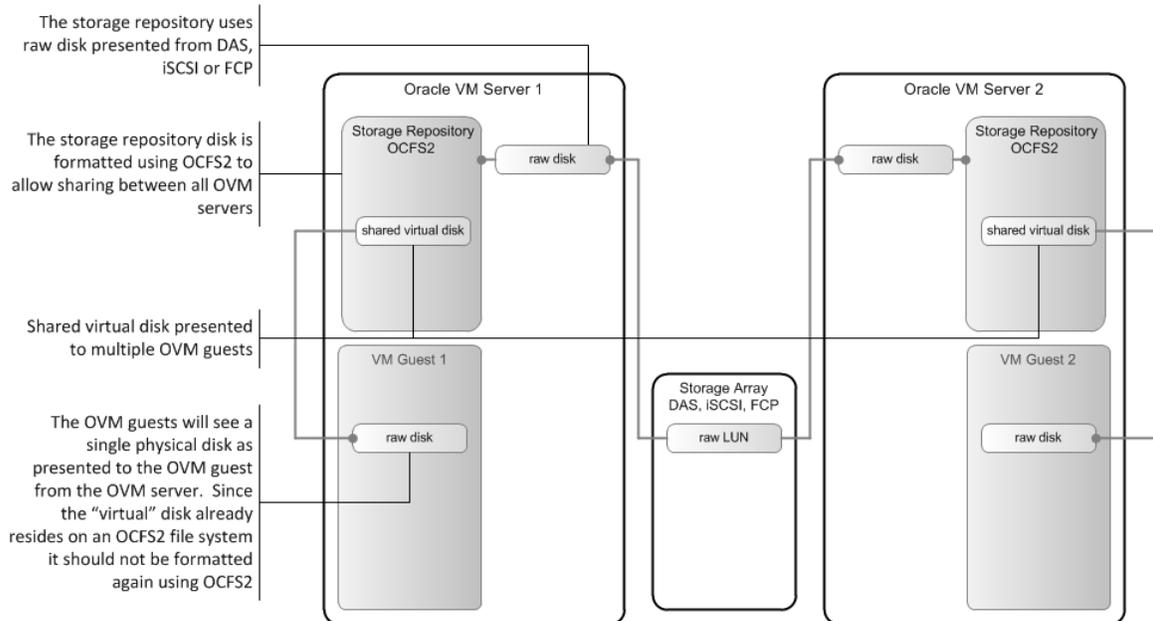


Figure 9: Shared virtual disk from an OCFS2 formatted storage repository

The VMs will see a single physical disk as presented from the Oracle VM server. Since the shared "virtual" disk already resides on an OCFS2 file system it should not be formatted again using OCFS2 on the guest operating system. So, shared virtual disks should only be used by applications in the VM image such as databases that can manage raw disk.

If you decided to create a storage repository using NFS, then you would not use OCFS2 for the file system of the storage repository. The following figure illustrates shared virtual disks being presented to multiple VMs from a storage repository that is using NFS.

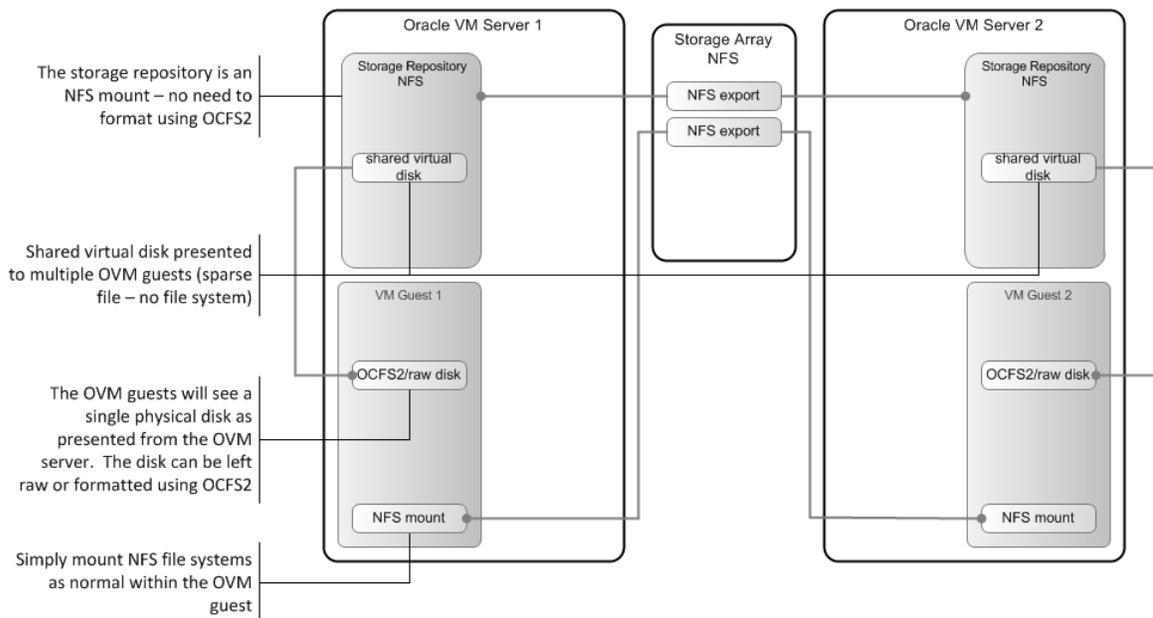


Figure 10: Shared virtual disk from an NFS storage repository

NFS is special in that the volumes being presented by NFS can be any file system on the disk, being solely managed by the NFS servers themselves. As NFS clients, the Oracle VM servers just see the contents of the volumes in exactly the same way as they see those volumes using OCFS2. NFS can be very useful for utilizing specialized storage devices such as ZFS or even collections of distributed storage [such as the unused space of desktop servers] or even providing complex and flexible storage topologies such as geo-clusters or DR; its performance is also very good and it is used in many production Oracle VM environments.

There is one exception to this, which employs dNFS to allow the creating of ‘raw’ disk images on NFS shares that are directly accessed as block devices, and these can then be used with OCFS2 to provide a clustered filesystem. In fact, this is exactly what Oracle VM 3 does when it creates a clustered server pool with an NFS server pool filesystem. See section **6.2 Server Pool Clusters** in the **Oracle VM User’s Guide**.

File Systems for Shared Physical Disks on Oracle VM Guests

When using FCP, iSCSI or DAS as shared storage we have seen that Oracle VM will use the OCFS2 clustered file system to create storage volumes that can be consistently shared between all the Oracle VM servers. OCFS2 is the standard Linux clustered file system and, as such, supported not only by Oracle VM but any standard Linux kernel. Using shared physical disks accessible from the VMs you may choose to use OCFS2 in order to create shared, clustered volumes that are managed and utilised directly and exclusively by those VMs.

The following figure illustrates shared storage being presented using FCP, iSCSI or DAS completely unrelated to the storage repository.

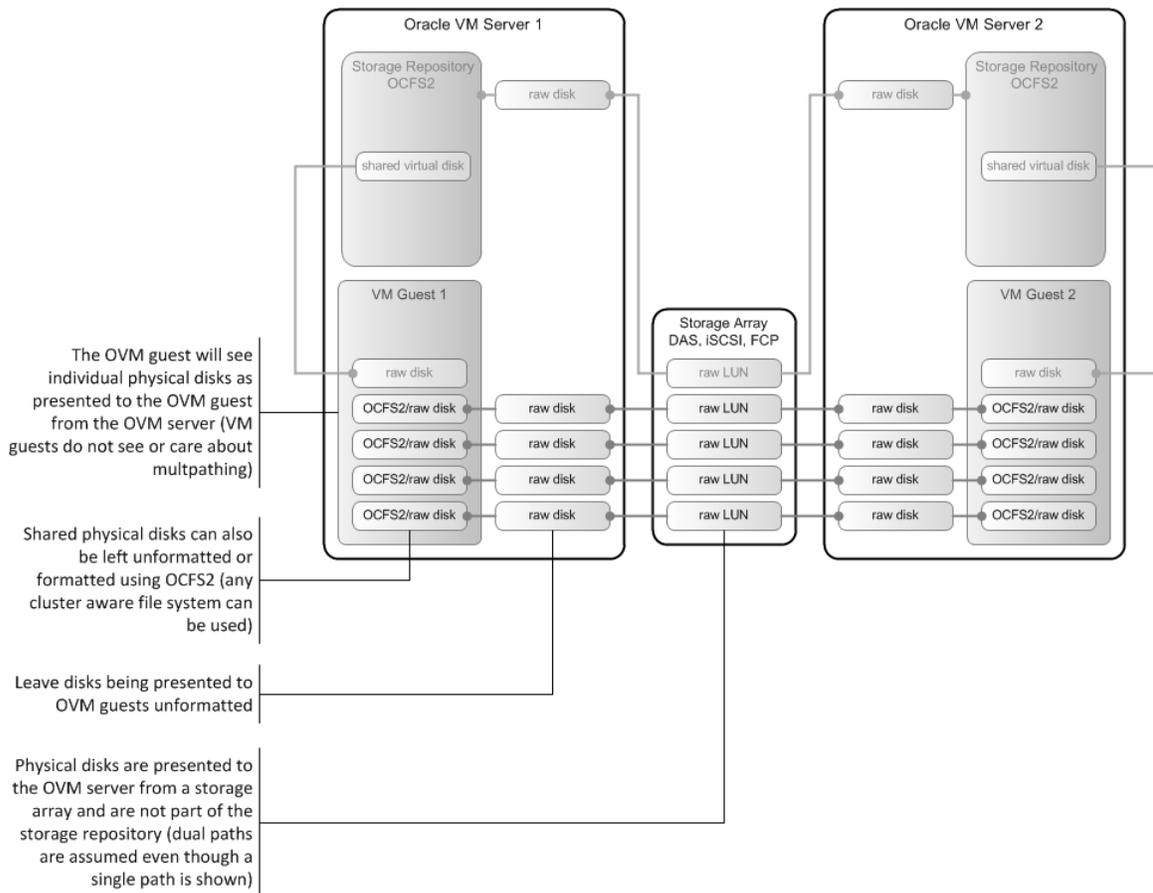


Figure 11: Shared physical disks being presented to VMs and formatted as OCFS2

LUNs are presented to the Oracle VM servers as raw disk and the VMs will see shared physical disks presented from the Oracle VM server. It is important to note that the disks presented to the VMs should remain unformatted. The VMs will see the shared physical disks as unformatted, so any cluster aware file system can be used on the guest operating system including OCFS2.

Oracle VM Networking Design

When you install each new Oracle VM Server, you specify which, of all its available NICs, is to be used as the management interface. By default, it will select eth0 but you may alter this. Additionally, as we have mentioned previously, you can choose to install the management interface as a tagged VLAN by specifying the relevant NIC and segment (VLAN) number for it to tag all management traffic. Such a VLAN configuration is all but permanent (it can be altered via a complex manual procedure) and many choose rather to connect the relevant NIC(s) to an untagged VLAN port (or pairs of ports) on the switch.

There are five different network channels employed by Oracle VM 3 and each of these can be assigned to dedicated NICs, bonds or VLANs or multiples thereof. By default, all network channels bar VM traffic are assigned to the same network as the management channel, which is named after the subnet

identified upon Oracle VM Server discovery by the Oracle VM Manager. These can be separated out and configured independently as desired. As we have noted before, particular attention should be given to the cluster heartbeat network channel. An excellent examination of these different channels and their uses can be found in the whitepaper [Oracle VM 3: Looking "Under the Hood" at Networking \(HTML\)](#).

VLANs deserve particular attention to detail; one of the most common problems with networking is that not every switch in the LAN/WAN has been configured correctly for VLANs. Bad VLAN configurations can appear to work but cause intermittent or weird problems and these are exacerbated in virtual environments where a VM may move from one processor to another or even move its traffic from one switch to another in the network.

We recommend that you draw up a network topology diagram not just between processors, switches, the Oracle VM Manager and storage devices where appropriate but also between virtual networks and the physical LANs and VLANs they will connect to, including the bridges, bonds, teams, etc.

One important consideration for the network design with respect the management of Oracle VM Server Pools is that each Server Pool (irrespective of clustering) will require its own Virtual IP address (VIP) in addition to the IP addresses assigned to the individual Oracle VM servers themselves. This VIP will be used by the Oracle VM Manager and the Server Pools themselves to contact the controlling server (Server Pool Master Server) at any one time; this role is always taken up by one and only one Oracle VM server in every Server Pool even in the event of a server failure.

Stage 2: Creating your Oracle VM environment

Once you have a clear design for your Oracle VM environment, creating that environment can begin.

Hardware Preparation

The first thing to do is install the hardware and plug in the storage and networking connections according to your design. Ensure that all the storage devices (LUNs, Zones, Arrays, etc.) and network devices (NICs, switches, routers, etc.) are correctly configured. If you have a test rig that can be plugged in and engaged for this purpose we advise using it – some customers have a basic Linux installation that can be simply and quickly installed on each host to run through some basic checks like storage probing, network teaming and VLAN connectivity.

Software Preparation

Download the latest version of the Oracle VM software from Oracle Software Delivery Cloud at <https://edelivery.oracle.com/oraclevm>. You can visit [Oracle VM OTN download page](#) for instructions about the latest software updates.

You will need to download at least the Oracle VM Server and Oracle VM Manager files (they are

ORACLE
Oracle Software Delivery Cloud - Oracle Linux and Oracle VM
Sign Out Cloud Portal (Oracle Linux/VM) Language (English) FAQs

Terms & Restrictions Search Download

Oracle VM 3.2.2 Media Pack for x86_64 (64 bit)

TIP View the Readme file(s) to help decide which files you need to download.
Print this page with the list of downloadable files. It contains a list of the part numbers and their corresponding description that you may need to reference during the installation process.

Hi Honglin, by clicking the download button, you agree Oracle's [Terms & Restrictions](#) apply to your use of the software on this portal. Not Honglin? Do not download the software and [login with your account](#).

Oracle VM 3.2.2 Media Pack v1 for x86_64 (64 bit)

Select	Name	Part Number	Size (Bytes)
<input type="button" value="Download"/>	Oracle VM Server 3.2.2 for x86_64 (64 bit)	V37306-01	240M
<input type="button" value="Download"/>	Oracle VM Server 3.2.2 Source for x86_64 (64 bit)	V37307-01	470M
<input type="button" value="Download"/>	Oracle VM Manager 3.2.2	V37304-01	2.5G
<input type="button" value="Download"/>	Oracle VM Manager 3.2.2 - Upgrade only	V37305-01	128M
<input type="button" value="Download"/>	Oracle VM Agent 3.2.1 for SPARC	V36294-01	1.6M
<input type="button" value="Download"/>	Oracle VM Windows Paravirtual (PV) Drivers for Microsoft Windows Guests (XP/Vista/7/2003/2008/2008 R2) 3.0.1 - 32-bit/64-bit (signed by Microsoft for the Windows Logo Program for Windows 2008, Windows 2008 R2, Windows 2003 and Windows 7)	V30673-01	8.9M
Total: 6			

Download Notes

Before You Download:
To ensure that you download the files successfully, first review the Media Pack Readme for download instructions and product information by clicking on the 'Readme' button.

After You Download:
For information regarding file extraction utilities please see our [Frequently Asked Questions](#).
If you encounter an error, please try downloading the file again. If you still have issues, please contact [Software Delivery Customer Service](#).

ZIPPED ISO files) from the list similar to the one shown in the following figure of the Oracle Software Delivery Cloud site for Oracle VM Software.

You only need to burn the Oracle VM Server ISO to disc as the Oracle VM Manager ISO can be simply mounted by the target Linux host when needed.

As we have said, Oracle VM Servers should essentially be stateless and are solely configured and managed centrally using Oracle VM Manager with the exception of rare occasions when SAN multipath configurations may need to be customized according to vendor requirements; such configuration of the Oracle VM server once it is installed needs to be planned for.

This will result in the creation or modification of a set of specific configuration files on each unique Oracle VM server that you will need to either replicate across every server or just duplicate across every server using something like `scp` from the first one you modify.

Figure 12 - Oracle Software Delivery Cloud download screen for Oracle VM

```
[e.g. #scp /etc/multipath.conf ovm_server2:/etc/multipath.conf]
```

Deploying Oracle VM Manager as a VM on Oracle VM

When planning to operate an environment where your Oracle VM Manager is actually virtualized on top of the Oracle VM infrastructure it is managing you need to take some additional steps over and above what is required when using an alternative approach. These steps may require the temporary provisioning of additional hardware during the installation. The full method for deploying Oracle VM Manager in a virtualised environment is documented in the Oracle VM manuals [http://docs.oracle.com/cd/E35328_01/E35330/html/vmiug-manager-as-vm.html] but we outline the key steps here to help you plan your deployment.

- Install Oracle Linux on bare metal (this could even be a laptop for the duration of this initial process)
- Install Oracle VM Manager 3.2 on this bare metal Oracle Linux server
- Install Oracle VM Server on one of the bare metal servers – record the UUID it selects
- Discover this Oracle VM Server from the Oracle VM Manager
- Configure the storage, network and create a Server Pool
- Create a new VM
- Install Oracle Linux into the new VM
- ssh into the new VM
- Get the UUID of your OVM Manager
- Shutdown your OVM Manager (can just stop the service “service ovmm stop”)
- Install Oracle VM Manager into your Oracle Linux VM using the same UUID (“runInstaller.sh — uuid MY_UUID”)
- Connect to this new virtual Oracle VM Manager
- Discover the original Oracle VM Server (this will populate the new Oracle VM Manager database with all the configuration settings that the original Oracle VM Manager has recorded)
- Discover your storage devices (again, this will repopulate the new Oracle VM Manager database)
- The original bare metal Oracle VM Manager server is no longer required and can be decommissioned or reused as desired
- Continue with your remaining Oracle VM infrastructure deployment, using your Oracle VM Manager as usual

Create an Oracle VM Server Pool

Our recommendation is to choose one host as the first one to be installed as an Oracle VM server and follow the steps below:

- Prepare a checklist of hostnames, IP address, VLAN segments and all other requirements for your environment: an example checklist can be found in Appendix B – Deployment Planning Checklist. One key aspect is that every Oracle VM Server in the same server pool must use the same Oracle VM Server agent password. Note that this is independent of the root passwords for those servers, which can all be set independently should you choose.
- On your intended Oracle VM Manager host, install the latest version of Oracle Linux, ensuring you have a minimum of 2x 64bit CPU cores, 4GB RAM and 60GB storage. If you intent to install and run the database locally you will need to account for the additional hardware requirements of whichever database option you have selected; if you are planning to use the standard, “simple” installation of Oracle VM Manager, which includes the local MySQL database, then 8GB RAM is recommended.
- Ensure your host and all intended Oracle VM Servers are configured in your DNS with fully qualified domain names and reverse lookup tables. You can set up with local `/etc/hosts` but you will need to replicate and maintain this manually across all the servers in your Oracle VM environment. We strongly recommend using DNS, even if you have to set up your own local one for the environment, but if you can’t then you will need to follow the below steps on each server once you have installed the base image (Linux or Oracle VM Server, depending on which function they will serve):
 - Check that `/etc/hosts` file does not contain the hostname (or anything other than localhost) for address 127.0.0.1 (see below for how this should look): the hostname can sometimes appear in this line if the DNS settings were not set correctly during installation.

`/etc/hosts`

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1    localhost.localdomain localhost
::1        localhost6.localdomain6 localhost6
```

- Edit the `/etc/hosts` file to ensure that it contains the hostnames and IP addresses for every other Oracle VM server you plan to put in the same Server Pool and any TCP storage hosts.
- Install your Oracle VM Manager software, taking note of the URLs listed at the end of the installation (see example below), as these will be how you use your new Oracle VM Manager. Also, take note of the UUID listed, as this will be the unique identifier of this new Oracle VM Manager used to secure the database and identify ownership of Oracle VM Servers and repositories. You can install a standby copy or completely reinstall a replacement for this Oracle VM Server by supplying this UUID as part of the installation command.

```
#./runInstaller.sh -uuid 0004fb0000010000ce2c25578f0172e4
```

```

Installation Summary
-----
Database configuration:
 Database type       : MySQL
 Database host name  : localhost
 Database name       : ovs
 Database listener port : 49500
 Database user       : ovs

Weblogic Server configuration:
 Administration username : weblogic

Oracle VM Manager configuration:
 Username       : admin
 Core management port : 54321
 UUID          : 0004fb0000010000f40416bae13104a4

Passwords:
There are no default passwords for any users. The passwords to use for Oracle VM Manager, Database,
and Oracle WebLogic Server have been set by you during this installation. In the case of a default i
nstall, all passwords are the same.

Oracle VM Manager UI:
  https://localhost.localdomain:7002/ovm/console
Log in with the user 'admin', and the password you set during the installation.

Please note that you need to install tightvnc-java on this computer to access a virtual machine's co
nsole.

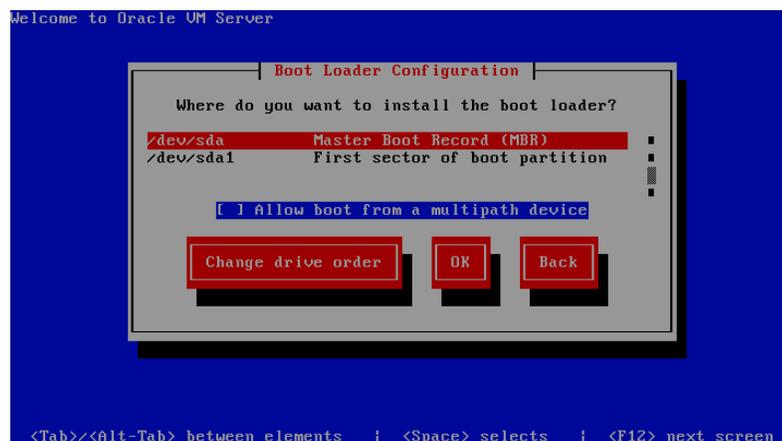
For more information about Oracle Virtualization, please visit:
  http://www.oracle.com/virtualization/

Oracle VM Manager installation complete.

Please remove configuration file /tmp/ovm_config91z0tB.
[root@localhost mnt]#

```

- Select your first machine to become an Oracle VM Server and install Oracle VM 3.2 on it, taking care to enter your allocated hostname, IP address (selecting your chosen management port), VLAN segment if required, hostname and associated networking entries. If you are using a SAN to boot your Oracle VM Server from, be careful to check the “Allow boot from a multipath device” option if you have a multipath boot device (screen shown here for reference). Ensure that you enter the agent password you have selected for the whole server pool.



- At the end of the installation (and its initial reboot) you will see an Oracle VM Server console showing the details you have entered so far, but mostly empty values as they will be filled once the

server has been configured using Oracle VM Manager. An example of the console of an installed but unconfigured Oracle VM Server is shown in the screenshot below.

```
Oracle VM Server 3.2.1 Console [Alt-F2 for login console]

Local hostname      : vml.electriclewis.lan
Manager UUID       : Unowned
Hostname           : None
Server IP          : None
Server Pool        : None
Clustered          : No
Server Pool Virtual IP : None
Cluster state      : Unknown
Master Server      : No
Cluster type       : None
Cluster storage    : None

OVS Agent          : Running
VMs running        : 0
System memory      : 2047
Free memory        : 1470
Uptime             : 0 days, 0 hours, 0 minutes_
```

- You can go ahead and install all the Oracle VM Servers you plan to use as they can all be discovered on one go, just taking care to give them all the same identical agent password.
- Log into your Oracle VM Manager using a browser, the URL you recorded earlier, the username “admin” and the password you allocated during the Oracle VM Manager installation.
- Discover your Oracle VM Servers and, once they have been discovered successfully, configure the patch and NTP management functions of Oracle VM Manager. These will make sure that the Oracle VM Servers are kept in sync and assist the health and integrity of the clusters. Once you have configured the server update management (YUM) your Oracle VM Manager will be able to tell you if any servers are out of date with your mirror of the Oracle public YUM repository for Oracle VM and allow you to update the out of date servers directly from within Oracle VM Manager.
- At this point you can also set up Tags in your Oracle VM Manager to mark various aspects of your environment with appropriate non-configuration data that you can use to manage and identify aspects of your environment.
- Once you have done this you can move to the Storage tab of the Oracle VM Manager and discover your storage devices and allocate them to the servers. Once you have discovered all your storage servers you can use the shared storage devices to create your Oracle VM Server Pools.
- This is also a good time to modify and enhance the default network configuration of your servers to create network bonds or change the MTU of the ports. If you are using physical NICs (rather than the virtual connections typically found in blade servers) we recommend you add at least one additional port to your management bond (bond0) for each of your servers. This is also a good time to set the MTU of the ports if you are using Jumbo Frames.

- Once you have discovered your servers, storage and networks and configured NTP and YUM using Oracle VM Manager you can start to create your Server Pools. Remember that you need to choose whether or not to cluster a server pool at creation time, as this cannot be changed later – almost every other aspect can be modified post creation but the cluster pool file system. Every Server Pool regardless of clustering requires its own virtual IP address (VIP), which you should have pre-allocated as part of your planning.
- At any time you can go to the console of an Oracle VM Server and see its status, including its own configuration data, if it is owned and managed by an Oracle VM Manager and its cluster pool file system if it is part of a cluster. If you can't access the physical console of the server you can ssh into it and type the command, "console". An example of a fully configured Oracle VM Server that is part of clustered server pool is shown here:

```

Oracle VM Server 3.2.1 Console [Alt-F2 for login console]

Local hostname      : ovm2.electriclewis.lan
Manager UUID       : 0004fb0000010000ce2c25578f0172e3
Hostname           : ovm2.electriclewis.lan
Server IP          : 192.168.2.102
Server Pool        : Clustered Pool
Clustered          : Yes
Server Pool Virtual IP : 192.168.2.253
Cluster state      : DLM_Ready
Master Server      : Yes
Cluster type       : lun
Cluster storage    : /dev/mapper/1IET_00010003

OVS Agent          : Running
VMs running        : 1
System memory      : 8013
Free memory        : 3132
Uptime             : 7 days, 5 hours, 31 minutes

```

Stage 3: Testing your Oracle VM Environment

The key stage before signing off your Oracle VM environment for production (even if it is just production in ‘Test & Development’) is to check that everything is working the way it should to identify potential problems that should be fixed before you sign it off.

Oracle has created a TechNote specifically addressing this topic: **VMPinfo3 Diagnostic Capture for Oracle VM 3 troubleshooting [ID 1364933.1]**, which even provides a tool and instructions for automated health-checking of an Oracle VM Server Pool. In addition or as an alternative to that process we present here a simple checklist of elements of functionality to check the status of your Oracle VM Server Pool installation.

BASIC FUNTCIONALITY TESTS

TEST	ACTION	RESULT OF YOUR TEST
Create a clustered Server Pool	Create a clustered server pool with shared storage	
Create a Guest VM	<p>Create a guest VM using Oracle VM Manager (or OEM); set this guest to be HA enabled and configure its native networking so that it has an active IP address on some known network.</p> <p>Once created, start it up and running if not left that way after its creation.</p>	
Test Live-Migration	<p>Open a network connection to the guest VM you created; this can be a continuous <code>ping</code> or a regular <code>ssh</code> session from some other system to this guest VM.</p> <p>Whilst this network connection is open/active initialize a live-migration of this guest VM to one of the other nodes in the server pool.</p> <p>The guest VM should be moved to the other node without any interruption to the active network session; i.e. the continuous ping or ssh session should still be running even after the guest VM has been migrated from the original Oracle VM server to the other within the server pool.</p> <p>If the live-migration fails, look in the log to see what the problem was.</p> <p>If the live-migration succeeds but the live network session breaks (i.e. the continuous ping or ssh session loses connection to the guest VM) then it means that the networking configuration on the two Oracle VM servers is different. Common issues are different VLAN configs, switch configuration or routing errors.</p>	
Test HA failover	<p>Switch off or disconnect the electrical power from the Oracle VM server on which the HA enabled guest VM is running.</p> <p>After three minutes, that same guest VM should automatically start up on one of the other Oracle VM servers in the HA enabled server pool.</p> <p>An even more complete test of this HA mechanism is to perform the same test whilst the Oracle VM Manager (or OEM) is either stopped, or disconnected from the network, as HA failover is independent of active</p>	

management involvement.

If it takes a long time (more than 5 minutes) for the guest VM to restart on another Oracle VM server within the server pool then check the logs to see if any errors were reported.

If the guest VM fails to start on one of the other Oracle VM Servers in the server pool check that both the server pool and the guest VM were started with the HA options enabled. Note: it is not enough to enable the HA option on a running guest VM - it needs to have been enabled when it booted to ensure correct operation.

Test server pool master failover

Only one of the Oracle VM Servers in a server pool can be the acting server pool master and, by the same token, one of the Oracle VM Servers in a server pool must be a server pool master. This is assigned automatically by negotiation between the running nodes in a server pool. Test this is working correctly by shutting down the active server pool master Oracle VM Server; you can perform this shutdown gracefully (by issuing `init 0` or similar) if you wish.

After three minutes, one of the other Oracle VM Servers in the server pool should become the server pool master.

If you have assigned a virtual IP address to the server pool, the new server pool master will be accessible via this IP address.

If you are using OEM10g ensure you have installed the latest patches to ensure this functionality operates correctly.

Test any VLAN networks

Make sure that guest VMs configured to use VLANs can connect correctly when they are running on every Oracle VM Server in the server pool (live-migrate them from node to node to test this).

Common problems for VLANs are errors in the network scripts for the VLANs on the Oracle VM Servers themselves, mistakes in the VLAN configuration within the network switches (VLANs on some switches will appear to work with some operating systems even when incorrectly configured but not with others).

Once you have passed the tests and are happy that all the essential elements of your Oracle VM environment are operating within expected parameters we recommend you take an archive of the configuration files you created or modified for each Oracle VM server so that you can recreate them or add new Oracle VM servers as required.

Appendix A – Essential Reference Information

Configuration Maxima

The configuration maxima for Oracle VM Server 3.2 can be found at

http://docs.oracle.com/cd/E35328_01/E35329/html/vmrns-limits.html

Supported Guest Operating Systems

Oracle VM supports Linux, Oracle Solaris and Microsoft Windows operating systems running in its Oracle VM 3.2 virtual machines. Paravirtualised (PV) VMs are recommended for Oracle software running on Oracle VM for both performance and the ability to dynamically change the amount RAM and CPUs on the fly. Hardware Virtualised (HVM) VMs can run any unmodified OS and may have very high performance on the latest processors but usually do not recognize dynamic changes in RAM or CPU cores. The supported Guest Operating Systems are listed on the Configuration Maxima page above but can change so ensure you always refer to the documentation for the exact version of Oracle VM you are using for the latest list.

Supported Guest Operating Systems for Oracle VM 3.2 are at

http://docs.oracle.com/cd/E35328_01/E35329/html/vmrns-guest-os.html

Supported Hardware

Oracle VM is supported on any x86 hardware according to the policy outlined in the FAQ at https://linux.oracle.com/hcl_faq.html but is specifically tested and certified on a number of hardware vendor's platforms. The list of certified hardware can be found and searched at

<https://linux.oracle.com/hardware-certifications>

Note that there are sections for Servers and Storage.

Important Reference Sites

- The Oracle Technology Network (OTN) page for Oracle VM can be found at <http://www.oracle.com/technetwork/server-storage/vm/overview/index.html>
- The Oracle VM Wiki contains information on the Oracle VM CLI, the Storage Connect Plugs as well as the latest updates, papers and seminars on Oracle VM <https://wikis.oracle.com/display/oraclevm/Home>
- The Unbreakable Linux Network, which is the home of the Oracle Linux and Oracle VM distributions, is found at <http://linux.oracle.com>
- The Oracle Public YUM Repository, which provides all the patches, updates and errata to Oracle Linux and Oracle VM Server for free, is found at <http://public-yum.oracle.com>

Appendix B – Deployment Planning Checklist

Component	Item	Value
Management Network	Subnet	
	Netmask	
	Gateway Address	
	DNS1 Address	
	DNS2 Address	
Server Pool	Agent Password	
	Virtual IP Address	
OVM Manager	IP Address	
	Root Password	
	<i>OVM Manager install password</i>	
	<i>OVM Admin password</i>	
OVM Server 1	Management IP Address	
	Hostname	
	Root password	
	<i>ILOM IP Address <if required></i>	
	<i>ILOM username/password <if required></i>	
OVM Server 2	Management IP Address	
	Hostname	
	Root password	
	<i>ILOM IP Address <if required></i>	
	<i>ILOM username/password <if required></i>	
NAS Filer	IP Address	
	Server Pool Filesystem export name/LUN	
	Repository Filesystem export name/LUN	
HTTP Server	Base URL for ISO Files	
	<i>YUM Repository URL <if required></i>	
VLAN	VLAN one ID	
	VLAN two ID	



Oracle VM: Designing, Creating and Testing an
Oracle VM 3.2 Environment

April 2013

Author: Wayne Lewis

Contributing Authors: Greg King, Honglin Su

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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. 0113

Hardware and Software, Engineered to Work Together