

VDI 3.1 Installation and Configuration Guide

Contents

- VDI 3.1 Installation and Configuration Guide
 - About VDI (All Topics)
 - Features of VDI 3.1
 - System Requirements (All Topics)
 - Sun VirtualBox Virtualization Platform Installation (All Topics)
 - VMware vCenter Virtualization Platform Installation (All Topics)
 - Microsoft Hyper-V Virtualization Platform Installation (All Topics)
 - Microsoft Remote Desktop Platform Installation (All Topics)
 - Single Host Configuration (All Topics)
 - High Availability Configuration (All Topics)
 - VDI 3.1 Update Installation (All Topics)
-

VDI 3.1 Installation and Configuration Guide

Contents

- Features of VDI 3.1
 - Architecture
-

About VDI (All Topics)

Features of VDI 3.1

With Sun Virtual Desktop Infrastructure Software, you can deploy a number of virtual desktop operating systems and access these operating systems from a variety of client devices – such as traditional PCs or Macintoshes, energy-efficient Sun Ray thin clients, or thin clients from other vendors. Sun VDI software also enables you to utilize existing IT assets, increase scalability, and simplify management.

Sun VDI 3.1 includes the following features:

- Support for a wide variety of guest operating systems, including Windows, Ubuntu, OpenSolaris™, and SUSE
- Support for four desktop providers - Sun VirtualBox, VMware vCenter, Microsoft Hyper-V, and Microsoft Remote Desktop Services
- Built-in virtualization capabilities (with Sun VirtualBox for VDI)
- Integration with OpenSolaris and Sun Unified Storage Systems (for Sun VirtualBox and Microsoft Hyper-V desktop providers)
- Support for Active Directory
- Multiple virtual desktops per user
- Streamlined and simplified installation with a common installer for the VDI Core software and Sun Ray Software
- Simplified template management with template creation, revisioning, and System Preparation built into the VDI Manager
- Cold Migration for controlled host and storage outages
- Virtual networking (VLAN) for isolation of pools on networks
- New pool settings, including smart card removal policy, available running desktops cloning policy, and per-pool RDP configuration
- Sun Ray Software 5 support, including Sun Desktop Access Client and USB device redirection

About Microsoft Hyper-V Virtualization Platforms

Sun VDI 3.1 provides users access to virtual machines hosted by Microsoft Hyper-V. Microsoft Hyper-V can be installed either as a free stand-alone product (Microsoft Hyper-V Server), or it can be enabled as the Hyper-V role in Windows Server 2008. You can find out more about the different versions on the [Microsoft website](#).

Like the Sun VirtualBox desktop provider, the Microsoft Hyper-V desktop provider takes advantage of iSCSI and ZFS as a part of OpenStorage (Solaris/OpenSolaris, and the Sun 7000 Series). For a VDI/Hyper-V demo, virtual disks can be stored on the VDI Core host, but for a production environment, as with VirtualBox, a Microsoft Hyper-V desktop provider requires a separate storage host.

The Microsoft Hyper-V desktop provider is remotely managed by VDI. To enable communication between the VDI Core and the Windows Server hosting Microsoft Hyper-V, the Windows Server needs to be prepared using the same preparation as for a Microsoft Remote Desktop provider. See the [How to Prepare a Windows Server for VDI](#) page for more details.

About Microsoft Remote Desktop Platforms



In Windows Server 2008 R2, Terminal Services (TS) has been renamed Remote Desktop Services (RDS).

Sun VDI 3.1 provides users access to Terminal Services or Remote Desktop Services sessions provided by Windows Server 2003 and Windows Server 2008. VDI relies on some Microsoft tools being implemented along with Remote Desktop Services to provide advanced features such as farm or cluster of Remote Desktop Services hosts (RDS hosts) with load balancing and session reconnection.

Microsoft Terminal Services on Windows Server 2003

VDI provides access to Terminal Services sessions delivered by:

- A single server running Windows Server 2003
- A cluster of servers with the following characteristics:
 - All servers running Windows Server 2003, any edition
 - The cluster is a Microsoft Network Load Balancing (NLB) cluster, which provides load balancing among servers

Microsoft Session Directory may be used to enable users to reconnect to existing sessions.

Microsoft reference documentation:

[Session Directory and Load Balancing Using Terminal Server Network Load Balancing Clusters](#)

Microsoft Remote Desktop Services on Windows Server 2008

VDI provides access to Remote Desktop Services sessions delivered by:

- A single server running Windows Server 2008
- A farm of servers with the following characteristics:
 - All servers running Windows Server 2008
 - Microsoft TS Session Broker is used to enable load balancing and enable users to reconnect to existing sessions.

As described in Microsoft documentation, preliminary load balancing may be provided using DNS round robin or Microsoft Network Load Balancing (NLB) or a hardware load balancer.

Microsoft reference documentation:

[TS Session Broker Load Balancing Step-by-Step Guide](#)
[Network Load Balancing Step-by-Step Guide: Configuring Network Load Balancing with Terminal Services](#)

Microsoft Remote Desktop Provider RDS Farm Management

If the Microsoft Remote Desktop provider is composed of RDS hosts belonging to a farm, VDI queries each individual RDS host to determine the sessions running on that host, so that information about these sessions can be displayed by the VDI Manager or CLI. By querying the first RDS host of the provider, VDI detects the information about the farm, and returns the farm name to the Remote Client when users try to access a session so that the session may be started on any host participating in the farm.

Because of this management scheme, administrators should add all the RDS hosts from a farm to the Microsoft Remote Desktop provider, so that VDI can contact each individual RDS host in order to register the local administrator's credentials and the SSL certificate for that host. However, the administrator does not need to enter any information about the farm in the VDI Manager or CLI, VDI detects that RDS hosts belong to a farm by querying them directly.

Limitations of Microsoft Remote Desktop Providers and Pools

Due to the specificity of Remote Desktop Services compared to the other virtualization backends, some restrictions apply to the actions and settings offered by the desktop providers and pools:

- The provider may contain multiple RDS hosts if they are members of the same Network Load Balancing cluster or Session Broker farm. See the section about supported platforms. In this case, Microsoft NLB or Microsoft Session Broker are responsible for the load balancing across the hosts. VDI does not perform any such load balancing.
- The provider may contain a single stand-alone RDS host.
- A given RDS host can only participate in one provider.
- Each Microsoft Remote Desktop provider can have one pool only.
- The pool does not offer any cloning capabilities. The Remote Desktop Services server or farm is responsible for opening new RDS sessions when new users connect.
- The type of assignment in the pool is always flexible. Session reconnection will be offered if the RDS setup is correctly configured for it, but is not the responsibility of VDI. See section about supported platforms.
- The list of desktops displayed for this pool correspond to the aggregated list of RDS sessions from all RDS hosts associated with the provider. All RDP sessions, whether they come from VDI or not, are displayed.
- Groups or users may be assigned to RDS pools but you cannot manually assign a user to an RDS desktop.

About Template Management

In Sun VDI 3, making changes to imported desktops was cumbersome. You either needed to make adaptations outside of Sun VDI and re-import the desktop image again, or you were forced to temporarily assign the desktop to a user, access the desktop on behalf of this user, and then make your changes. Sun VDI 3.1 introduces an Adobe Flash plug-in that enables you to easily access, test, and modify the desktop directly from within the VDI Manager. This feature also includes changing desktop properties as well as mounting ISO images for setting up the operating system.

Any desktop can be used as template for cloning additional desktops. Testing desktop templates and keeping track of any changes before rollout is crucial for large enterprise deployments. Sun VDI now includes support for managing several template revisions. You can create a new template revision at any time, test your changes and declare the new revision as the master used for the cloning process. You can also revert to a previous revision if you are not satisfied with your changes.

Sun VDI leverages the Microsoft System Preparation tool (Sysprep) for preparing Windows desktops for cloning. The use of Sysprep ensures that each desktop clone is assigned its own unique security identifier (SID), which is mandatory if desktops need to join an Active Directory domain. In Sun VDI 3.1 it is now possible to trigger Sysprep from within the VDI Manager. The corresponding template revision is automatically marked as Sysprepped once the preparation has completed.

About Maintenance Mode

On occasion, you might need to off-line a configured host or storage, including maintenance, upgrades, and decommissioning. The Maintenance Mode feature, introduced in VDI 3.1, allows virtual machines to be cleared from a currently used host or storage and moved to a different host or storage so that normal operations may continue while the initial host or storage is unavailable. This process is also considered a "cold" migration because running virtual machines will be suspended to allow the maintenance process to proceed.

Maintenance Mode is available for Sun VirtualBox and Microsoft Hyper-V desktop providers only.

Host Maintenance

The two ways a host can be put in Maintenance mode are:

Migrate Desktops or Shutdown and Restart Desktops on Another Host

- Requires > 1 VirtualBox host.
- Migrate Desktops - Offered only if VDI thinks there are other compatible hosts. Otherwise, "Shutdown and Restart Desktops" is offered.
- Desktops are migrated one after the other. A desktop being migrated will be temporarily unavailable for up to a minute.

Suspend Desktops on a Host

- Always offered.
- Suspends all desktops on the current host.
- If a suspended desktop is requested it will be resumed on another VirtualBox host, if available.



A compatible VirtualBox host must have identical, or very similar, CPU models. Attempting to resume a desktop on a different CPU model will often result in a failure. VDI 3.1 verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible.

VDI 3.1 also checks for valid VirtualBox versions. Migrating from VirtualBox 3.0 to VirtualBox 2.0 is not supported.

Storage Server Maintenance

Suspend Desktops on a Storage Server

- Always offered.
- No data is moved or deleted from the specified storage server.
- Each desktop on the storage server is suspended.
- Desktops will be unavailable until the storage server is re-enabled.
- Desktop hard disk data remains on the storage server.

For details about how to enable the Maintenance Mode feature, refer to the [How to Use Maintenance Mode](#) page.

About VirtualBox in VDI 3.1

VDI 3.1 supports only specific Sun VirtualBox versions. For accurate and up-to-date information about version support, refer to the [VDI 3.1 Release Notes](#). VirtualBox for VDI 3.1 provides the following new VirtualBox desktop provider features.

Desktop Suspend and Resume

VDI 3.1 offers the ability to suspend and resume desktops across VirtualBox hosts. This functionality supports the following new features:

- **Suspended Pool Power State**
Desktops in a pool will be booted and then suspended, ready for fast user access without the resource overhead of keeping your desktops running.
- **Host Migration**
Desktops can be cold migrated from one VirtualBox host to another. This functionality will suspend all desktops on a specified host and resume them on other available hosts in the Desktop Provider with only a minimal outage. For more information about the Host Migration feature, see the [About Maintenance Mode](#) page.
- **Suspend Host**
Suspend all desktops on a specified VirtualBox host, enabling it to be maintained or upgraded easily. After maintenance is complete, simply re-enable the VirtualBox host and all desktops will be resumed to their original state.



To use suspend and resume reliably, all VirtualBox hosts in a desktop provider should contain identical CPU models.

Non-Windows Desktop Recycling

Recycling is now supported for non-Windows desktops including OpenSolaris, Ubuntu, and SUSE Linux Enterprise Desktop.

Desktop Import Improvements

In VDI 3.0, virtual machine templates had to be moved to the VDI Core host in order to be imported to and managed by the VDI Manager. In VDI 3.1, you have the choice of importing templates from the VDI Core host or the Sun VirtualBox host.

About Per Pool Network Configuration

In VDI 3.1 you can take advantage of the Per Pool Network Configuration feature, which enables an administrator to specify the subnet in which

desktops will be placed. For Sun VirtualBox and Microsoft Hyper-V desktop providers, VDI will detect the networks that are configured on the provider's hosts, and the administrator can select which of these networks should be used in specific pools.

Configuration of networks is done at two levels in VDI:

- Desktop Provider - Each subnet available on either a Sun VirtualBox or Microsoft Hyper-V host is identified by a unique label. By default this label is the subnet address, but it can be changed in the Network tab for the desktop provider. When a host is added to a desktop provider, VDI will detect the subnets available on that host and will update the Network table accordingly. If a subnet is not available on any of the hosts in a provider, VDI will display a warning. You can view the list of subnets available for a specific host by selecting that host in the Host tab for the desktop provider. If you make changes to the networking on a host, click the Refresh button in the Network tab so that VDI can rescan the subnets available on the host.
- Desktop pool - A pool can have one or more networks assigned to it. When a pool is created, VDI will check whether any networks are available on all hosts for the desktop provider of the pool, and it will assign one of these networks to the pool. If no networks are available on all hosts for the provider, the administrator must explicitly specify a network to be used by the pool through the Settings tab for the pool. When desktops are imported or cloned in a pool, VDI will create a network device on the desktop and configure that device to be in the networks that have been enabled for the pool. If more than one network has been configured for the pool, VDI will use the network that has been configured as the primary network when trying to establish an RDP connection to the desktop. The primary network for a pool can be configured in the Settings tab.



The Per Pool Network Configuration feature is only available for Sun VirtualBox pools if Host Networking is being used.

About New Pool Settings

In VDI, a pool is defined as a collection of desktops. With VDI 3.1, you can take advantage of configuration settings that are only applied to a selected pool, managed by the VDI Core.

Per Pool RDP Configuration

With VDI 3.1, you can configure the RDP options to be used by Sun Ray sessions when users connect to their desktops. A wide range of options including locale, keyboard layout, color depth, theming, redirection and more are configurable on a per pool basis using the VDI administration console or CLI. These settings will be used by Sun Ray sessions connecting to desktops managed by any of the currently supported VDI desktop provider types.

For details on how to enable this feature, see the [How to Configure RDP Options Per Pool](#) page.

About the Smart Card Removal Policy

With VDI 3.1, you can control what should happen to a user's desktop after a smartcard is removed from a Sun Ray DTU. Using the Smart Card Removal Policy, you can indicate that a user's desktop should be shut down, suspended, or recycled when the smart card has been out of a DTU for a specific length of time. If the user reinserts a smart card before the specified time has elapsed, the associated action on the desktop will be canceled. The Smart Card Removal Policy is configurable per pool and is available for all Sun VirtualBox, Microsoft Hyper-V, and VMware vCenter pools. This policy may be configured using the VDI Manager or CLI.



Recycling is applied only to desktops that have flexible assignments. Choosing the recycle option for your Smart Card Removal Policy will have no effect on personally assigned desktops.

For details on how to enable this feature, see the [How to Configure Smart Card Removal](#) page.

About Running Available Desktops

With VDI 3.1, you can take advantage of faster access to desktops by using the Running Available Desktops feature. For each of your desktop pools, you can ensure that any number of available desktops will be maintained in the running state. This can considerably reduce the time taken to deliver desktop sessions to your users.

About Sun Ray Software in VDI 3.1

Since version 3.0, VDI has included a common installer for the VDI Core Software and the Sun Ray Software components which installs and configures a default version of Sun Ray Software for VDI. The common installer is a significant improvement for installation and configuration of a full VDI solution, especially for administrators who are less familiar with Sun Ray Software. Administrators who have a strong Sun Ray Software background may choose to change the defaults of the configuration that is installed for VDI by reviewing the [VDI Defaults](#) page and using the included links to access relevant Sun Ray Software information.

The Sun Ray Software for VDI 3.1 is based on Sun Ray Software 5, which includes several new features that can also be useful in VDI. The following sections provide an overview of these features and their default configurations with the common installer.

USB Device Redirection

USB redirection is a Sun Ray Software 5 feature that enables users to access USB devices connected to a Sun Ray DTU from their Windows XP sessions, provided that the appropriate device drivers are installed on the Windows server. USB redirection is automatically enabled during VDI Core configuration and is supported as part of the [Per Pool RDP Settings](#). It can be enabled or disabled using the VDI Manager or CLI. See the Feature Overview table on the [How to Configure RDP Options Per Pool](#) page for more information.

For the most up-to-date details about USB Redirection, refer to the [About USB Device Redirection](#) page in the SRWC 2.2 Information Center.

Sun Desktop Access Client

The Sun Desktop Access Client is a software application that runs on common client operating systems and provides the ability to connect to a desktop session running on a Sun Ray server. Users can switch between their Sun Ray DTU and any supported Desktop Access Client enabled PC without using smart cards. In other words, a user can install and run the Sun Desktop Access Client instead of relying only on a Sun Ray Desktop Unit (DTU) for session access. For example, a user could connect to the same Sun Ray session from a PC laptop or desktop at home and a Sun Ray DTU at the office.

For the most up-to-date details about Sun Desktop Access Clients, refer to the [About Sun Desktop Access Clients](#) page in the SRSS 4.2 Information Center.

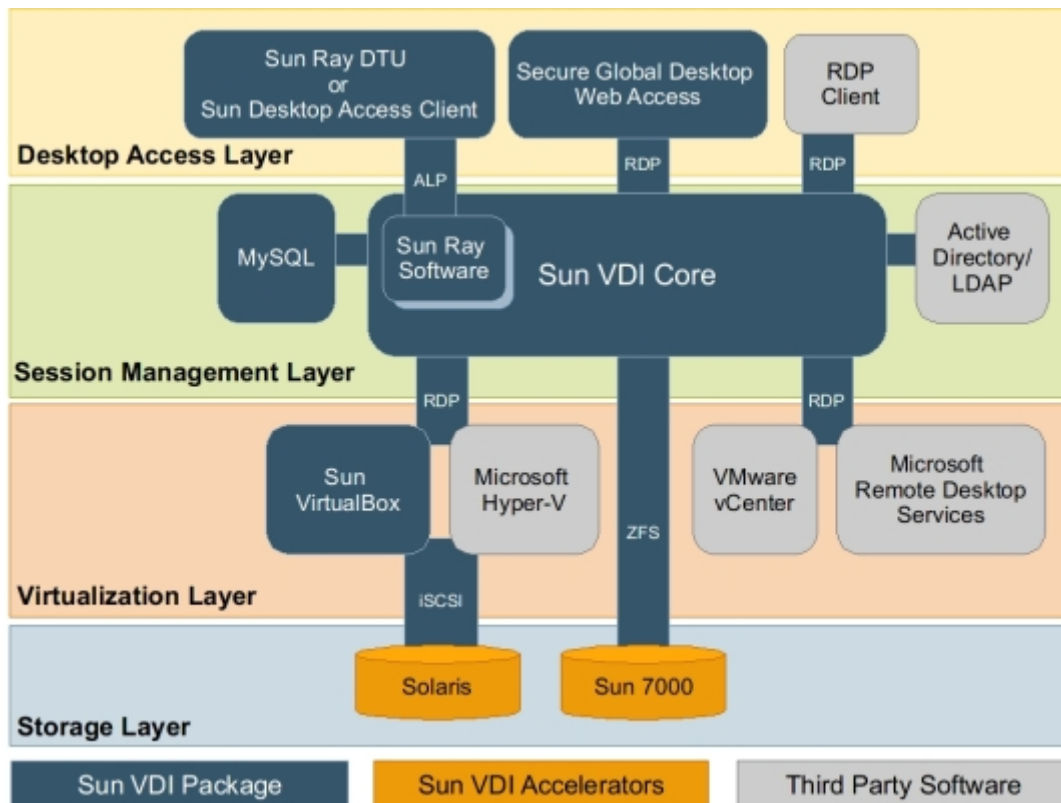
About Sun Service Tags

VDI 3.1 includes support for Sun Service Tags. Sun Service Tags enable you to automatically discover and register systems, software and services on your network. For more details about Sun Service Tags, see the [Sun Inventory Information Site](#).

VDI 3.1 delivers a single service tag. This tag is automatically added to the service tag registry when you configure VDI using `vda-config(1m)`. Once you have configured VDI, the VDI service tag will be available for registration using Sun's Service Tag discovery and registration client software. For more details about registration of service tags, see the [Discovery and Registration](#) page of the Sun Inventory Information Site.

Architecture

Sun Virtual Desktop Infrastructure 3.1 (VDI) is made up of four main components: virtualization platform, session management (VDI Core), desktop access clients, and storage.



Virtualization Platform

The basis for the VDI architecture is the virtualization platform. In addition to creating and storing virtual machines, the virtualization platform offers the core functionality needed for virtual desktop management such as starting, stopping, and snapshotting virtual machines. Sun VDI 3.1 supports the Sun VirtualBox, VMware vCenter, Microsoft Hyper-V, and Microsoft Remote Desktop Services virtualization platforms.

Session Management

The central component of the Sun VDI is the Sun VDI Core. The VDI Core provides all the functionality needed to build and manage large scale virtual machine deployments. In addition to its management capabilities, the VDI Core is also responsible for the brokering of virtual desktops on behalf of desktop access clients.

By integrating with Active Directory, the VDI Core is able to provide support for assignment of virtual desktops to existing users and groups within an organization. The VDI Core configuration data and runtime information is stored in a MySQL database, which may be shared across multiple VDI Core instances on the network. This database configuration ensures access to the VDI Core even in failover scenarios.

Desktop Access

Three distinct mechanisms are supported for access to virtual desktops.

Sun Ray DTU or Sun Desktop Access Client - Users can access their virtual desktops through a Sun Ray Desktop Unit (DTU) or through Sun Desktop Access Client software by authenticating themselves with a user name and password, or a token card can be inserted in a Sun Ray DTU in place of providing a user name. Successful authentication initiates a custom Sun Ray Software Kiosk Session. The custom Kiosk Session uses the Sun VDI Core to request access to a virtual desktop on behalf of the user. Once a virtual desktop has been assigned to the user, a Remote Desktop Protocol (RDP) connection to the desktop is established for the session using the Sun Ray Windows Connector.

Secure Web Access with SGD - In this case, the browser is used to initiate a Sun Secure Global Desktop Software (SGD) session. SGD, in turn, uses the VDI Core's RDP redirection capability to establish a connection to an assigned virtual desktop.

RDP Client Access - (RDP redirection must be supported on the client side to use this mechanism). As with the previous case, the VDI Core's redirection capability is used to establish a connection to an assigned virtual desktop.

Storage

Sun VDI takes advantage of iSCSI paired with ZFS in the Sun Unified Storage 7000 Series (Amber Road) and the Solaris or OpenSolaris OS to provide reliable storage for Sun VirtualBox and Microsoft Hyper-V desktop providers. The ZFS sparse volume and clone features enable efficient usage of storage space and a fast creation of desktops. The virtual disks occupy only the used sectors of disk space on the storage host, regardless of the size of the virtual disk. Because only the differences between the template and the cloned virtual disk are stored, a pristine clone consumes close to no storage space.

Contents



- Supported Software
- Standard System Requirements
- Standard System Requirements
- Supported Configurations
 - Supported Configurations for Sun VirtualBox Virtualization Platforms
 - Supported Configurations for VMware vCenter Virtualization Platforms
 - Supported Configurations for Microsoft Hyper-V Virtualization Platforms
 - Supported Configurations for Microsoft Remote Desktop Platforms
 - Single Host Configuration
 - VirtualBox on Primary Host Configuration
 - Primary Host Virtualized Configuration
 - Explicitly Unsupported Configurations
- Deployment Guide

System Requirements (All Topics)










Supported Software

This section includes support tables for VDI Core host operating systems, virtualization platforms, storage servers, desktop guest systems. For more information about VDI Support, see [Supported Configurations](#).

VDI Core Host Operating Systems

Software	Supported in VDI 3.1
Solaris 10 10/09 SPARC and x86 (64-bit)	
Solaris 10 5/09 SPARC and x86 (64-bit)	

Virtualization Platforms and Desktop Providers


Software	VirtualBox Virtualization Platform	VMware Infrastructure Virtualization Platform	Microsoft Hyper-V Virtualization Platforms	Microsoft Remote Desktop Platform	Not Supported
Sun VirtualBox for VDI (VirtualBox 2.0.10 and 3.0.12)					
All other VirtualBox versions					
VMware VirtualCenter 2.5 (Update 1, 2, 3, 4)					
VMware ESX server 3.5 (Update 1, 2, 3, 4)					
VMware vSphere (ESX server 4.0)					
Microsoft Hyper-V Server 2008					
Microsoft Windows Server 2008					
Microsoft Windows Server 2003					

Storage Servers













Software	VirtualBox Virtualization Platforms	VMware Infrastructure Virtualization Platforms	MS Hyper-V Virtualization Platforms
Solaris 10 10/09		Qualified by VMware	
Solaris 10 5/09		Qualified by VMware	
OpenSolaris 2008.11		Qualified by VMware	
OpenSolaris 2009.06		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q3.2.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q3.1.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q3.0.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.5.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.5.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.4.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.3.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.3.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.2.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.2.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.1.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.1.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2009.Q2.0.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.2.3		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.2.2		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.2.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.2.0		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.1.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.1.0		Qualified by VMware	

Sun Unified Storage 7000 Series 2008.Q4.0.1		Qualified by VMware	
Sun Unified Storage 7000 Series 2008.Q4.0.0		Qualified by VMware	










* With the 2009.Q3.0.0 release a new iSCSI stack (COMSTAR) has been introduced to the Unified Storage 7000 Series which is incompatible with the VirtualBox and Hyper-V desktop providers of the VDI 3.0 and VDI 3.1 releases. COMSTAR will be supported by VDI 3.1.1 which will be released as soon as possible after the 3.1 release.

 **Tip**
For more information about the Sun Unified Storage 7000 Series, see the [Fishworks Documentation](#).

Desktop Guest Systems

Software	VirtualBox Virtualization Platforms	VMware Infrastructure Virtualization Platforms	MS Hyper-V Virtualization Platforms	Not Supported
Windows XP SP2/3				
Windows Vista Enterprise				
Windows 7				
Windows 2000 SP4				
Ubuntu 8.10 (Intrepid Ibex)				
Ubuntu 9.04 (Jaunty Jackalope)				
OpenSolaris 2009.06				
SUSE Linux Enterprise 11				

Web Browser for VDI Manager

	Firefox 3	Firefox 3.5	Internet Explorer 6	Internet Explorer 7	Internet Explorer 8
Windows					
Solaris			N/A	N/A	N/A
Linux			N/A	N/A	N/A

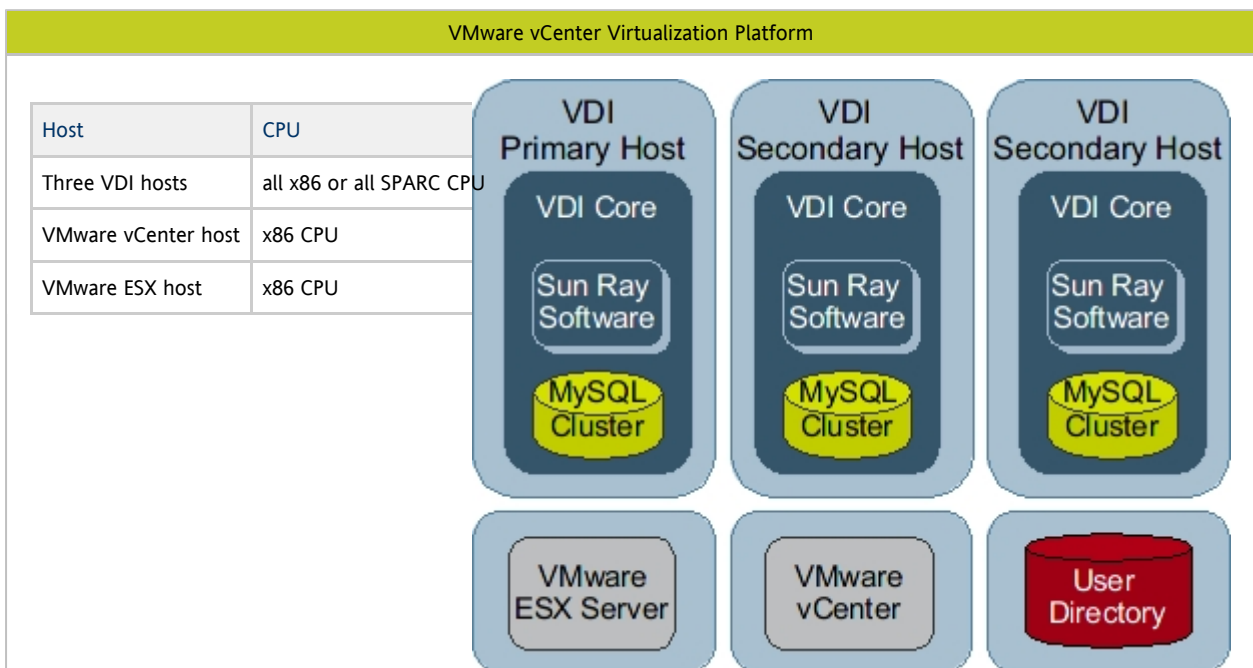
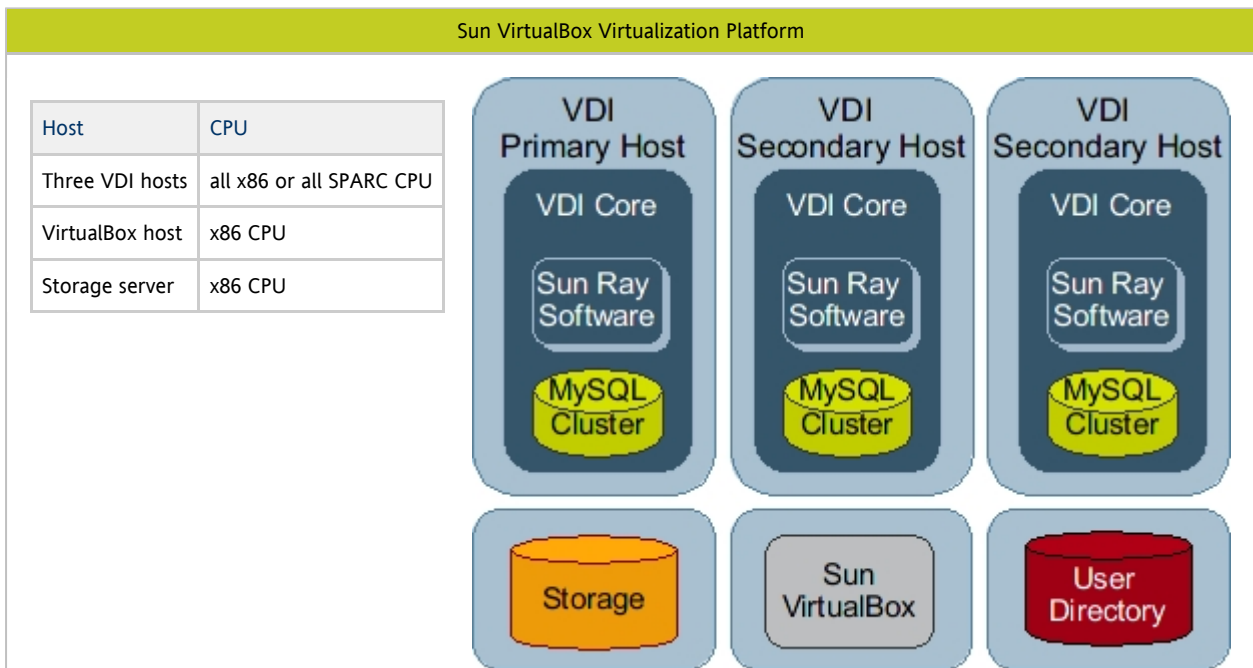
Standard System Requirements

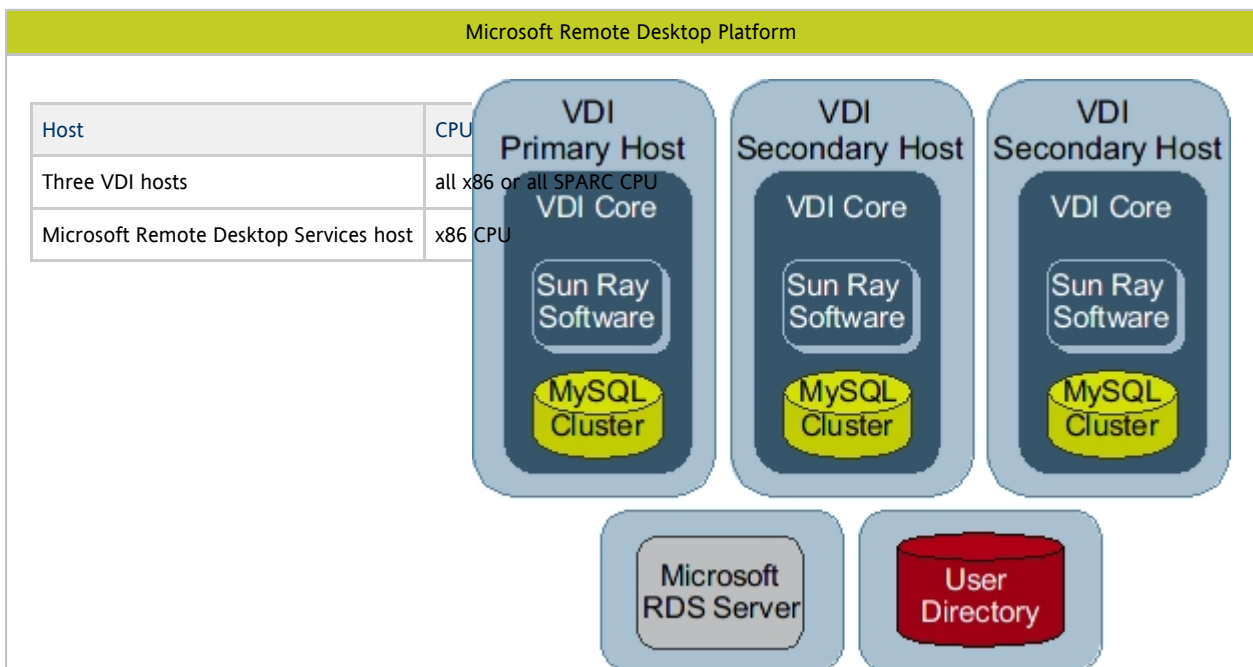
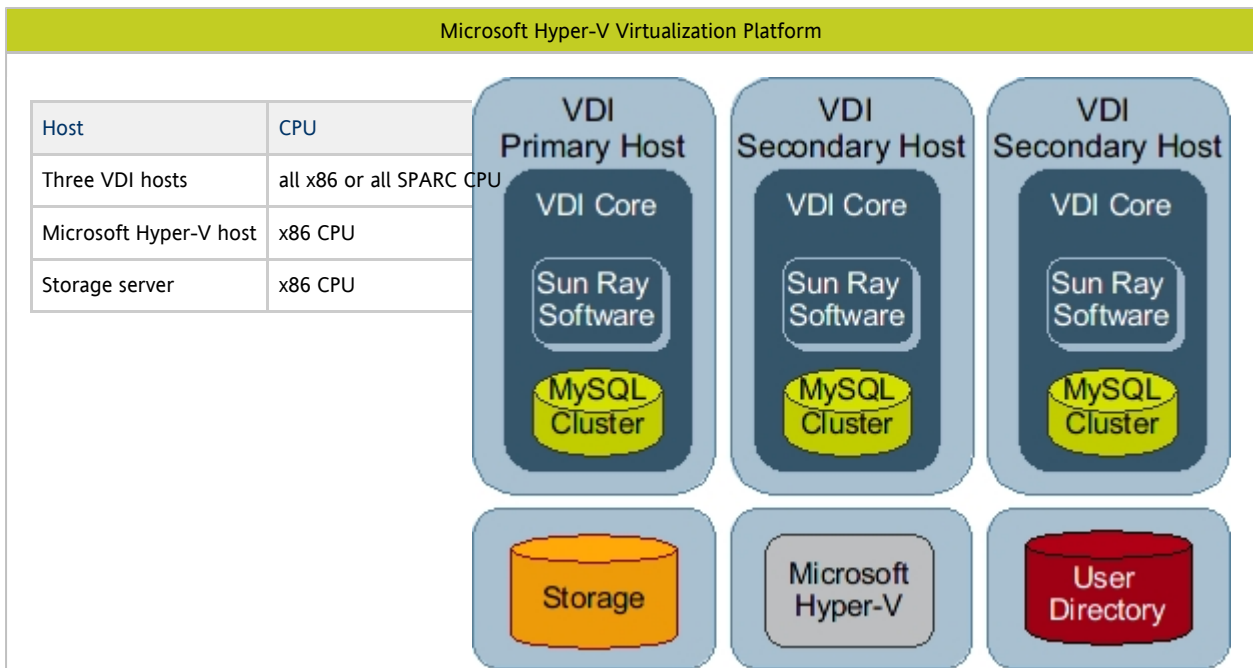
This section outlines the minimum requirements for a standard VDI 3.1 configuration. See the following pages for more information about supported software and VDI deployment options:

- [VDI 3.1 Release Notes](#)
- [Supported Configurations](#)
- [Deployment Guide](#)

Standard System Requirements

A standard (minimum) VDI setup for a production environment requires three VDI hosts and one desktop provider host. Sun VirtualBox or Microsoft Hyper-V virtualization platforms additionally require a storage server. All required VDI components (VDI Core) are installed on the VDI hosts. The desktop provider hosts are used to maintain the virtual machines.





Supported Configurations

There are many possible configurations for the virtualization platform and the VDI Core. Some configurations are supported for production environments, and some will work for evaluation but are not supported in production environments.

Available Configurations

The following tables list available configurations for VDI, and the corresponding option that must be chosen during VDI Core configuration.

Sun VirtualBox Virtualization Platform

Configuration Type	Options Selected During Configuration	Notes
Demo (Evaluation) Configuration	On the Demo Host: 0 Evaluation Sun VDI Host	Not supported for production environments.
High Availability Configuration with Bundled MySQL Database	On the Primary Host: 1 Primary Sun VDI Host On the first Secondary Host: 2 Secondary Sun VDI Host On the second Secondary Host: 2 Secondary Sun VDI Host	
High Availability Configuration with Remote MySQL Database	On the Primary Host: 1 Primary Sun VDI Host (Specify Remote Database) On the first Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database) On the second Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database)	Requires an additional support contract for the database.
Single Host Configuration	On the Single Host: 3 Single Sun VDI Host	Requires an additional support contract for the database.
VirtualBox on Primary Host Configuration	On the Primary Host: 1 Primary Sun VDI Host On the first Secondary Host: 2 Secondary Sun VDI Host On the second Secondary Host: 2 Secondary Sun VDI Host	

VMware vCenter Virtualization Platform

Configuration Type	Options Selected During Configuration	Notes
Demo (Evaluation) Configuration	On the Demo Host: 0 Evaluation Sun VDI Host	Not supported for production environments.
High Availability Configuration with Bundled MySQL Database	On the Primary Host: 1 Primary Sun VDI Host On the first Secondary Host: 2 Secondary Sun VDI Host On the second Secondary Host: 2 Secondary Sun VDI Host	
High Availability Configuration with Remote MySQL Database	On the Primary Host: 1 Primary Sun VDI Host (Specify Remote Database) On the first Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database) On the second Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database)	Requires an additional support contract for the database.
Primary Host Virtualized Configuration	On the Primary Host: 1 Primary Sun VDI Host On the first Secondary Host: 2 Secondary Sun VDI Host On the second Secondary Host: 2 Secondary Sun VDI Host	

Microsoft Hyper-V Virtualization Platform

Configuration Type	Options Selected During Configuration	Notes
Demo (Evaluation) Configuration	On the Demo Host: 0 Evaluation Sun VDI Host	Not supported for production environments.
High Availability Configuration with Bundled MySQL Database	On the Primary Host: 1 Primary Sun VDI Host On the first Secondary Host: 2 Secondary Sun VDI Host On the second Secondary Host: 2 Secondary Sun VDI Host	
High Availability Configuration with Remote MySQL Database	On the Primary Host: 1 Primary Sun VDI Host (Specify Remote Database) On the first Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database) On the second Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database)	Requires an additional support contract for the database.

Microsoft Remote Desktop Platform

Configuration Type	Options Selected During Configuration	Notes
Demo (Evaluation) Configuration	On the Demo Host: 0 Evaluation Sun VDI Host	Not supported for production environments.
High Availability Configuration with Bundled MySQL Database	On the Primary Host: 1 Primary Sun VDI Host On the first Secondary Host: 2 Secondary Sun VDI Host On the second Secondary Host: 2 Secondary Sun VDI Host	
High Availability Configuration with Remote MySQL Database	On the Primary Host: 1 Primary Sun VDI Host (Specify Remote Database) On the first Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database) On the second Secondary Host: 2 Secondary Sun VDI Host (Specify Remote Database)	Requires an additional support contract for the database.

Sun VDI Support Information

- Some VDI configurations are **explicitly unsupported configurations**. See below for more details.
- In order to download the VDI Patches, you must have (at minimum) a Sun Basic Service support plan. For more information about Sun Service support plans, see the [Sun Services](#) site.
- Because VMware and Microsoft software are not included as part of the Sun VDI Software 3.1 package, Sun Support contracts do not cover VMware-related issues. For VMware coverage, you need an additional support plan. For more information about Sun Service plans for VMware, see the [Sun Services for VMware](#) site.

Supported Configurations

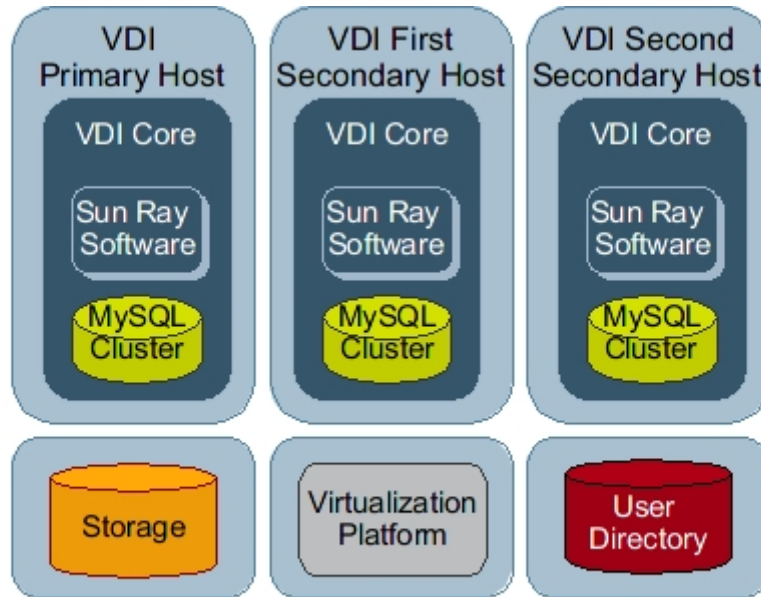
The configurations described in this section are supported as production environment configurations by a Sun VDI contract.

High Availability Configuration with Bundled MySQL Database

The VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one VDI host fails, all users with desktop sessions on that host are kicked back to the VDI Login dialog, and must reconnect to their session, which is restarted on one of the available hosts. The bundled MySQL cluster database requires three hosts to be highly available. There should never be an interruption of database service as long as

no more than one of the three required hosts fails at a time. Therefore, the bundled VDI stack requires a minimum of three VDI hosts to be considered fail-proof. This number of hosts does not include the virtualization platform hosts, which should be considered for failover separately.

For more information about the High Availability (Bundled MySQL Database) configuration, refer to the [Standard System Requirements](#) and [About VDI Core Configuration](#) pages.



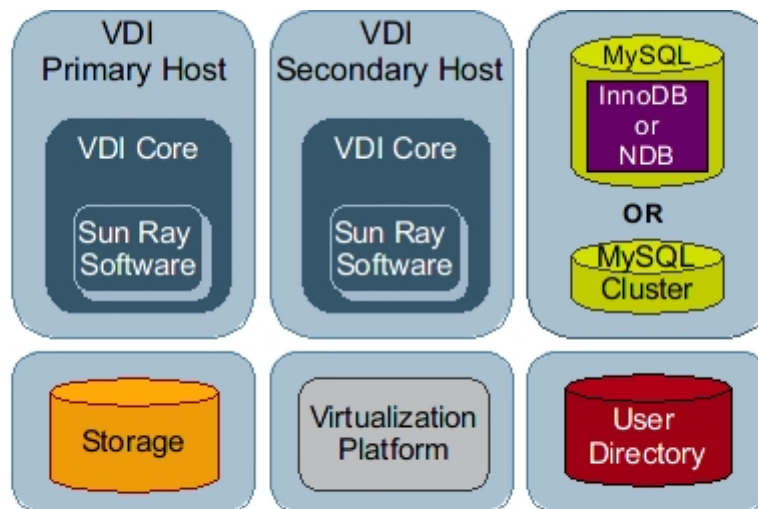
High Availability (Bundled MySQL Database) Configuration

High Availability Configuration with Remote MySQL Database

Instead of using the bundled MySQL database, you can also choose to use an existing database and connect to it by specifying it as a remote database during VDI Core configuration. In this case, the VDI Core only requires two hosts to be highly available. This number of hosts does not include the remote database hosts, or the virtualization platform hosts, which should be considered for failover separately.

The High Availability configuration with the remote MySQL database requires a MySQL version 5.0 or higher, with a transactional storage engine (usually InnoDB or NDB), or a MySQL Cluster version 6.2.15 or higher.

For more information about the High Availability (Remote MySQL Database) configuration, refer to the [About VDI Core Configuration](#) page.



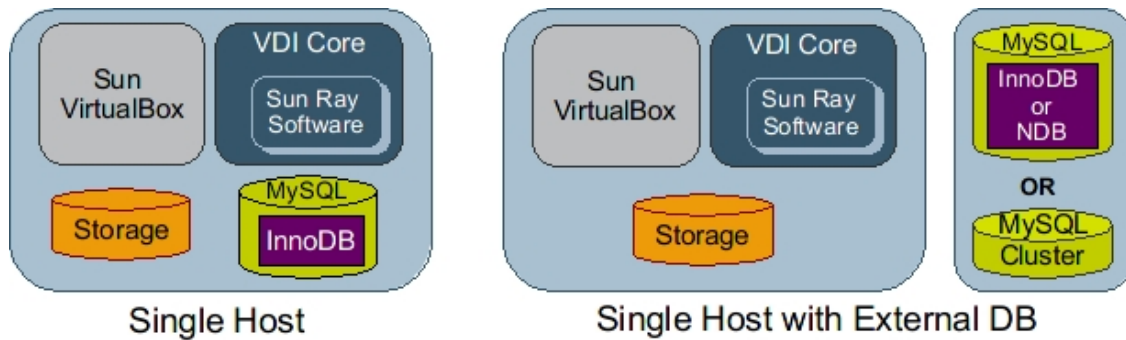
High Availability (Remote MySQL Database) Configuration

Single Host Configuration

The Single Host configuration is similar to a VirtualBox Demo (Evaluation) configuration because all necessary components can be installed on one machine, including a Sun VirtualBox virtualization platform, VDI Core (with MySQL database and Sun Ray Software). The difference between a Single Host configuration and a Demo (VirtualBox) configuration is that the Demo uses the bundled MySQL cluster database, and the Single Host configuration requires a MySQL "remote" database. The MySQL database can be installed on the Single Host machine or on a different machine. Even if the MySQL database is installed on the same machine as all the other components, it is still considered to be "remote" since it's not bundled with the VDI Core.

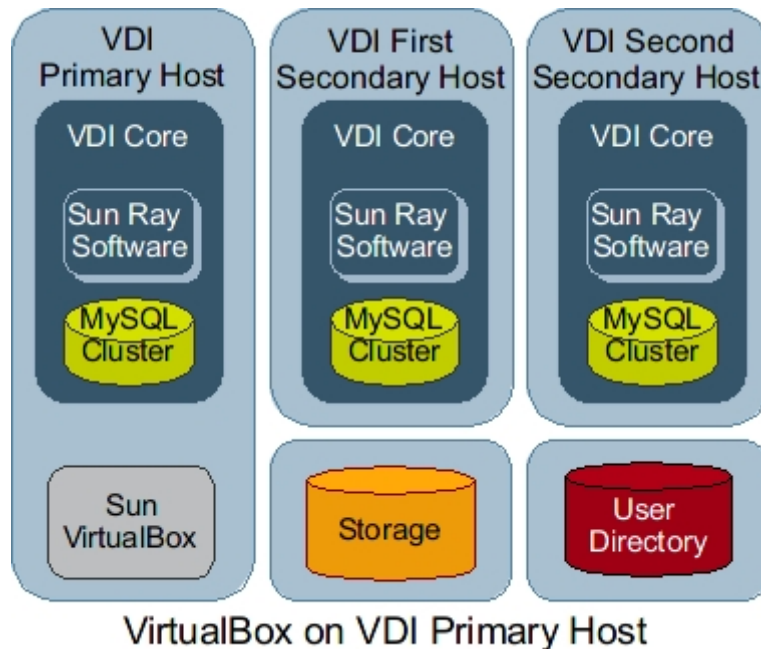
For more information about the Single Host configuration, refer to the [About VDI Core Configuration](#) page.

i The VDI support contracts only cover a VDI configuration with an embedded MySQL database. The Single Host Configuration uses a locally installed MySQL database with a InnoDB engine, which must be configured as a remote database. Therefore, if you want support service for the database component of the Single Host Configuration, you must purchase an additional MySQL service contract. For more information, see the [MySQL Support](#) page.



VirtualBox on Primary Host Configuration

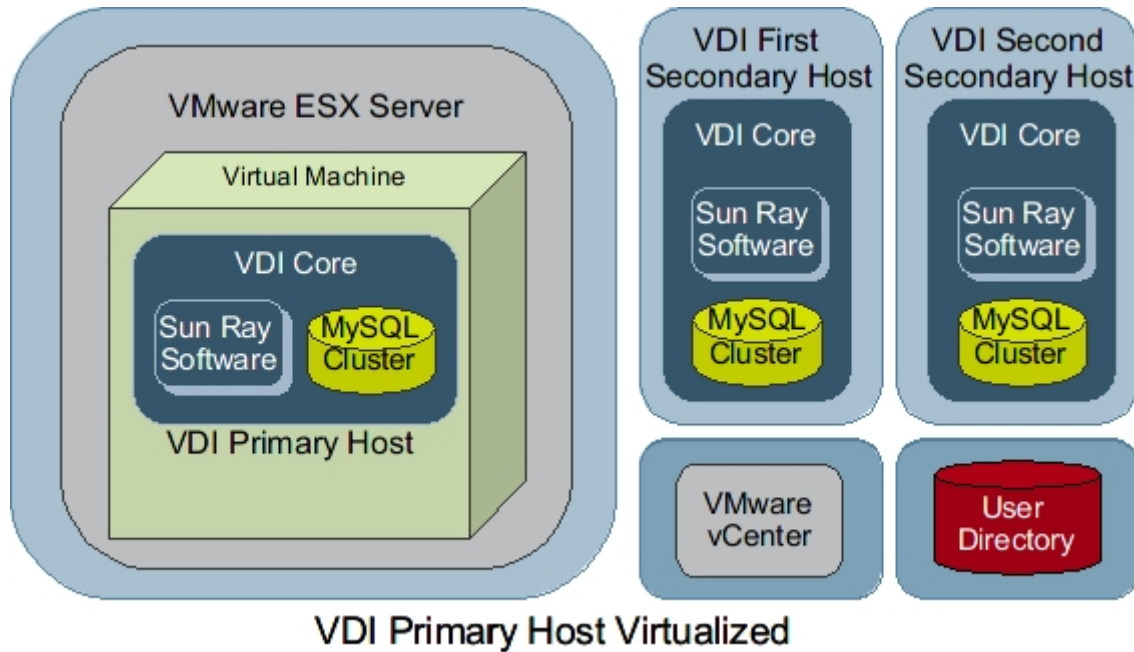
In the VirtualBox on Primary Host Configuration, the Sun VDI Primary node and one of your VirtualBox hosts share one physical machine. Make sure the shared host has enough capacity to deal with these two roles at the same time. For more information about sizing VirtualBox configurations, see the [Deployment Guide](#).



Primary Host Virtualized Configuration

In the Primary Host Virtualized Configuration, the Sun VDI Primary node runs in a virtual machine hosted by a VMware Infrastructure virtualization platform. Running the MySQL Cluster completely in a virtualized environment is not supported. Given the fact that the MySQL

Cluster management node (or Primary node) requires very few resources, you can run it in a virtual machine. The two VDI Secondary hosts running the MySQL Cluster data nodes nevertheless need to run on bare metal.



Explicitly Unsupported Configurations

VDI Demo (Evaluation) Configuration

The VDI Demo Configuration is not supported as a production environment VDI deployment because the embedded database configuration does not comply with MySQL standards. Before committing to a larger deployment, consider trying a VDI Demo Configuration to evaluate the new features. See the [VDI Demo Comparison](#) page to determine which Demo to set up. Consult the [VDI Forum](#) or check the [Troubleshooting and FAQs \(Categorical\)](#) for setup assistance.

Virtualized VDI Core Configuration

In a Virtualized VDI Core Configuration, the Sun VDI Primary and two Secondary nodes could be hosted in virtual machines. The Sun VDI 3.1 Core with embedded database provides high availability out of the box, which requires network and I/O response times that cannot be guaranteed in virtualized environments. Customers who rely on a fully virtualized environment will need to use a configuration utilizing an external database. However, the Primary VDI Core node can be virtualized with the embedded database as long as it is not used for delivering sessions to users. For more information about virtualizing the MySQL database, see the [MySQL FAQs](#).

Mixed Endian Cluster Nodes Configuration

In a Mixed Endian Cluster Nodes Configuration, the Sun VDI Primary and two Secondary nodes could be hosted on machines with dissimilar CPU types (a mix of x86 and SPARC). For a Sun VDI 3.1 Core with embedded database, the management node (Primary VDI node) and first two data nodes (Secondary VDI nodes) used in the cluster must have the same architecture (all x86 or all SPARC). That is, all machines hosting nodes must be either big-endian or little-endian, and you cannot use a mixture of both. Any additional nodes added, because they are MySQL client nodes and not part of the data cluster, can be either architecture. For more information about mixed Endian MySQL nodes, see the [MySQL Limitations](#).

Deployment Guide

i This guide contains the same guidelines for VDI 3.1 as were given for VDI 3. New information regarding sizing will be released in the coming months.

This page provides guidelines for the design of the hardware architecture for VDI 3.1 deployments with a Sun VirtualBox virtualization platform.

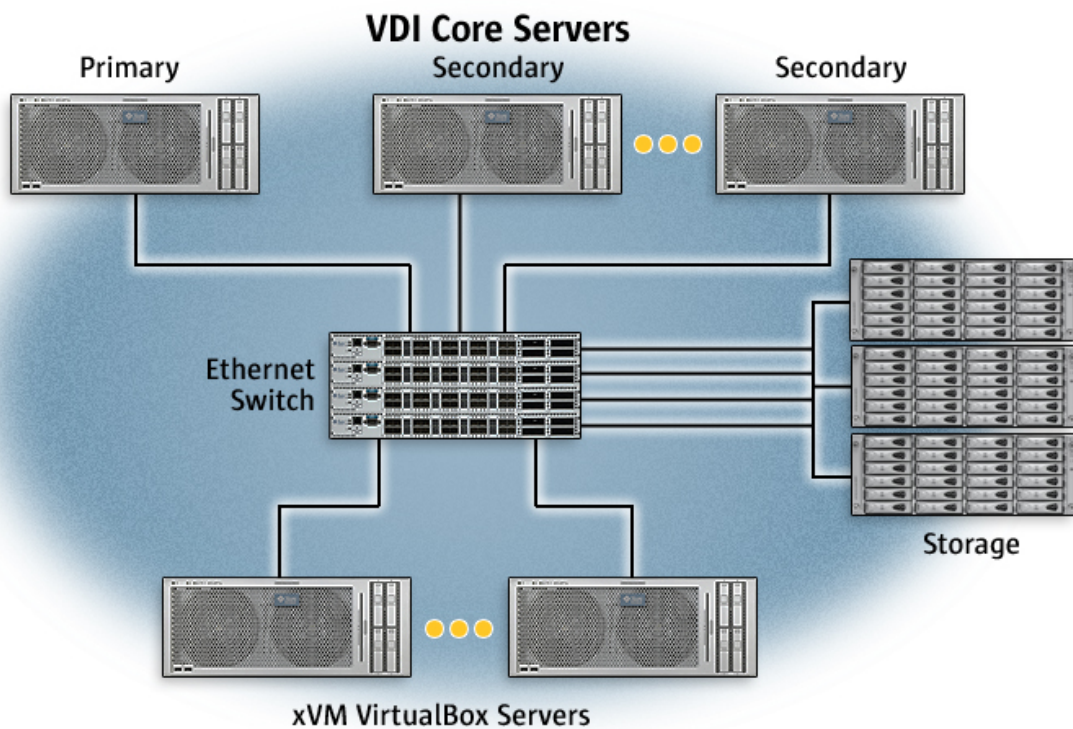
The information provided here is derived from a sizing test with 1000 desktops running a script to simulate an office workload of a "heavy worker" as defined in the VMware [VDI Server Sizing and Scaling](#) Guide. The script runs the following sequence of application actions:

1. Start Microsoft PowerPoint. Load a large presentation and browse the slides. Close Microsoft PowerPoint.
2. Start Internet Explorer. Browse three different web pages. Close Internet Explorer.
3. Start a command prompt. Do a directory listing.
4. Start Microsoft PowerPoint. Load a large presentation and browse the slides. Close Microsoft PowerPoint.
5. Start Microsoft Excel. Open a Microsoft Excel spreadsheet. Close Microsoft Excel.
6. Start Microsoft PowerPoint. Load a large presentation and browse the slides. Close Microsoft PowerPoint.
7. Start Microsoft Word. Type a small document. Close Microsoft Word.

The workload is different for every installation and relatively small changes in the usage patterns can have noticeable effects on the hardware requirements. Therefore, be sure to size every deployment individually. This page provides helpful information for such efforts.

VirtualBox Virtualization Platform Architecture

The hardware environment for a VDI 3.1 deployment typically looks like this:



Every production deployment consists of one primary VDI Core host and at least two secondary VDI Core hosts to provide redundancy. The VDI Core servers host a clustered MySQL database for the VDI data, route information between clients and desktops, and provide the broker functionality which delivers the desktops to the clients. Alternatively, remote databases are also supported. The VirtualBox servers run the virtual machines which provide the desktops. The storage servers provide the virtual disks which are interpreted as physical disks by the operating systems running within the virtual machines. The iSCSI protocol is used to transfer the disk data between the VirtualBox servers and the storages. That iSCSI data creates a major part of the total network traffic of a VDI system. For more information, see the [Sizing Guidelines for Storage Servers](#) section below.

Other consumers of network bandwidth are the clients of VDI 3.1: Sun Rays, Sun Secure Global Desktop, and RDC clients. The clients connect to the VirtualBox servers through the VDI Core servers. In the case of a Sun Ray client, which uses the ALP protocol to transfer the desktop graphics, the VDI Core servers convert the RDP protocol received by the VirtualBox servers to the ALP protocol. So, there is one data stream for each client connection between the client, the VDI Core server, and the VirtualBox server. RDP clients such as the Windows Connector (utsc), connect to the VDI Core server which, in turn, uses the RDP Redirect feature to instruct the clients to connect to the VirtualBox servers directly as there is no need to translate the RDP protocol. In this case, there is a data stream between the soft client and the VirtualBox server.

i The bold terms below are rules of thumb for calculating the according resource requirements.

Sizing Guidelines for VDI Core Servers

The primary VDI Core server requires a dual-core CPU and 2 GB of memory. As long as the VDI services are not configured on that server, these hardware requirements do not change with the number of running desktops.

The secondary VDI Core server requirements for the number of cores and memory size varies with the number of running desktops supported, as well as the required network bandwidth. The bandwidth also varies with the content displayed. The numbers given below are typical for office work. Displaying videos or web pages with Flash content can increase the required bandwidth.

- Number of cores = number of running desktops / 20
Example: Two secondary VDI Core servers with 8 CPUs and 4 cores per CPU can serve $2 * 8 * 4 * 20 = 1280$ running desktops
- Memory size [MB] = number of desktops * 110 MB + 2048 MB
Example: Two secondary VDI Core servers with 64 GB of memory can serve $(2 * 64 * 1024 \text{ MB} - 2 * 2048 \text{ MB}) / 110 \text{ MB} = 1154$ running desktops
- Network bandwidth [Mb/s] = number of running desktops * 0.15 [Mb/s]
Example: One secondary VDI Core server with one 1 Gb Ethernet interface can serve $1024 / 0.15 \text{ Mb/s} = 6827$ running desktops

For more information, see the [Complete Sun Ray Server Sizing Guide](#).

Sizing Guidelines for VirtualBox Servers

VDI supports any x86 server running Solaris to host VirtualBox. Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date version support information.

- Number of cores = number of running desktops / 4
Example: A server with 8 CPUs and 4 cores per CPU can support up to $8 * 4 * 4 = 128$ running desktops
- Memory size [MB] = number of running desktops * memory size of a desktop * 1.2 + 1024 MB
Example: A server with 64 GB of memory can support $64 * 1024 \text{ MB} - 1024 \text{ MB} / (512 \text{ MB} * 1.2) = 105$ running desktops of 512 MB in size

i A rule of thumb for VirtualBox servers is that a server with 32 cores and 64 GB of memory supports 100 desktops. While the CPU power of the server chosen for the examples above enables it to support 128 desktops, do not increase the memory size to do so. At least 20% of the available CPU power should be available as a security margin.

- Network bandwidth [Mb/s] = storage network bandwidth / number of VirtualBox servers
For more details about network bandwidth see the [Sizing Guidelines for Storage Servers](#) section, below.

i **More Than 100 Virtual Machines**
If you want to run more than 100 virtual machines on a single VirtualBox server, the SYSV semaphores on the VirtualBox server need to be increased. You need to set the number of available semaphores to the number of virtual machines you intend to run, including a security margin for other processes. To set the SYSV semaphores for 1000 virtual machines, type as root:

```
prctl -r -n project.max-sem-ids -v 1024
projmod -s -K "project.max-sem-ids=(priv,1024,deny)" user.root
```

The first line changes the available semaphores for the current process, and the second line makes this number a permanent system setting for the root user. If the VBoxSVC process is run by another user, add a `user.myuser` line to the `/etc/project` file and change the second line accordingly.

The maximum number of virtual machines on a single VirtualBox server is 1023.

Sizing Guidelines for Storage Servers

VDI supports most Sun Storage 7000 Unified Storage System firmware and any x86 server running an OpenSolaris OS. Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date version support information.

The recommended disk layout is RAID 10, mirrored sets in a striped set, with ZFS striping the data automatically between multiple sets. This layout is called "mirrored" by the 7000 series. While this disk layout uses 50% of the available disk capacity for redundancy, it is faster than RAID 5 for intense small random read/writes, which is the typical access characteristic for iSCSI.

The storage servers provide the virtual disks that are accessed by VirtualBox through iSCSI. Because iSCSI is a CPU-intensive protocol the number of cores of the storage server are a decisive factor for its performance. Other important factors are the memory size (cache), the number of disks, and the available network bandwidth.

The network bandwidth is very volatile and determined by the relation of desktops starting up (peak network bandwidth) and desktops that have cached the applications in use (average network bandwidth). Starting a virtual machine creates a network load of 150 MB which needs to be satisfied in around 30 seconds. If many desktops are started at the same time, the requested network bandwidth may exceed 1 Gb/s if the CPUs of the storage can handle the load created by the iSCSI traffic. This scenario is typical for shift-work companies. In such a case, set the Pool, Cloning, or Machine State option to Running, which always keeps the desktops running and therefore decouples the OS boot from the login of a user. Another option is to trunk several interfaces to provide more than 1 Gb/s bandwidth through one IP. You can also use Jumbo Frames to speed up iSCSI connections. Jumbo Frames need to be configured for all participants of the network: storage servers, VirtualBox servers, and switches. Note that Jumbo Frames are not standardized so there is a risk of incompatibilities.

VDI, in combination with VirtualBox, uses the Sparse Volume feature of ZFS, which enables it to allocate more disk space for volumes than is physically available as long as the actual data written does not exceed the capacity of the storage. This feature, in combination with the fact that cloned desktops reuse unchanged data of their templates, results in a very effective usage of the available disk space. Therefore, the calculation for disk space below is a worst-case scenario assuming that all volumes are completely used by data which differs from the template.

- Number of cores = number of virtual disks in use / 200
Example: A x7210 storage with 2 CPUs and 4 cores per CPU can serve up to $2 * 4 * 200 = 1600$ virtual disks
- Memory size - The more the better. The free memory can be used as a disk cache, which reduces the access time.
- Average Network bandwidth [Mb/s] = number of virtual disks in use * 0.032 Mb/s
Example: An x7210 storage with one Gigabit Ethernet interface can serve up to $1000 / 0.032 = 31250$ virtual disks
- Peak Network bandwidth [Mb/s] = number of virtual disks in use * 40 Mb/s
Example: An x7210 storage with one Gigabit Ethernet interface can serve up to $1000 / 40 = 25$ virtual disks
- Disk space [GB] = number of desktops * size of the virtual disk [GB]
Example: An x7210 storage with a capacity of 46 TB can support $46 * 1024 \text{ GB} / 2 / 8 \text{ GB} = 2944$ 8 GB disks in a RAID 10 configuration

Helpful Hints

- The graphic performance of desktops is better without background images.
- Avoid processes that generate constant or, burst disk I/O, for example the indexing service of Microsoft Windows or virus scanners with a periodic scheduled scan.

Contents

- [About VirtualBox in VDI 3.1](#)
 - [How to Set Up a VirtualBox Server](#)
 - [How to Set Up a Solaris Storage Server](#)
 - [How to Set Up an OpenSolaris Storage Server](#)
 - [How to Set Up a Sun Storage 7000 Unified Storage System](#)
 - [About Maintenance Mode](#)
 - [How to Use Maintenance Mode](#)
 - [VirtualBox and Hyper-V Host Maintenance](#)
 - [Storage Maintenance](#)
 - [How to Replicate and Replace a Storage Server](#)
-

Sun VirtualBox Virtualization Platform Installation (All Topics)


About VirtualBox in VDI 3.1

VDI 3.1 supports only specific Sun VirtualBox versions. For accurate and up-to-date information about version support, refer to the [VDI 3.1 Release Notes](#). VirtualBox for VDI 3.1 provides the following new VirtualBox desktop provider features.

Desktop Suspend and Resume

VDI 3.1 offers the ability to suspend and resume desktops across VirtualBox hosts. This functionality supports the following new features:

- **Suspended Pool Power State**
Desktops in a pool will be booted and then suspended, ready for fast user access without the resource overhead of keeping your desktops running.
- **Host Migration**
Desktops can be cold migrated from one VirtualBox host to another. This functionality will suspend all desktops on a specified host and resume them on other available hosts in the Desktop Provider with only a minimal outage. For more information about the Host Migration feature, see the [About Maintenance Mode](#) page.
- **Suspend Host**
Suspend all desktops on a specified VirtualBox host, enabling it to be maintained or upgraded easily. After maintenance is complete, simply re-enable the VirtualBox host and all desktops will be resumed to their original state.

 To use suspend and resume reliably, all VirtualBox hosts in a desktop provider should contain identical CPU models.

Non-Windows Desktop Recycling

Recycling is now supported for non-Windows desktops including OpenSolaris, Ubuntu, and SUSE Linux Enterprise Desktop.

Desktop Import Improvements

In VDI 3.0, virtual machine templates had to be moved to the VDI Core host in order to be imported to and managed by the VDI Manager. In VDI 3.1, you have the choice of importing templates from the VDI Core host or the Sun VirtualBox host.

How to Set Up a VirtualBox Server

A Sun VirtualBox virtualization platform requires a storage server to house the virtual machines in addition to the VirtualBox server (an x86 system running a current Solaris OS).

The following servers can be used as storage by VirtualBox:

- Sun Storage 7000 Unified Storage System (the 7210 or 7410 are best suited for production environments)
- Any server running a current OpenSolaris OS (best suited for demo purposes)
- Any server running a current Solaris OS

Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date version support information.


Before You Begin

See the following pages for detailed instructions on storage host preparation:

- [How to Set Up a Sun Storage 7000 Unified Storage System](#)
- [How to Set Up an OpenSolaris Storage Server](#)
- [How to Set Up a Solaris Storage Server](#)

Steps


1. Make sure that the Solaris swap space is adequate, otherwise your installation will fail.

 Solaris hosts running VirtualBox must have swap space equal to or greater than the host's physical memory size. See the [VDI 3.1 Release Notes](#) for detailed information.

2. As root user, unzip the VDI archive if you have not already done so. Then unzip the VirtualBox archive, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
# unzip vbox_3.0.zip
# cd vbox_3.0
# ./vb-install
```

3. Complete installation as prompted by the `vb-install` script.

 All VirtualBox hosts must run VirtualBox as the same UID (User ID) for suspend/resume functionality to work. Running VirtualBox as root is the easiest way to guarantee this.

How to Set Up a Solaris Storage Server

A Solaris storage host must be an x86 system. Each virtual disk is represented by a ZFS volume, and the volumes are stored in a ZFS pool that is accessed by VirtualBox through iSCSI. The management of the ZFS volumes is done by Sun VDI and requires SSH root access to the ZFS storage server and a ZFS pool on that server.

Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date Solaris version support information.

Steps

1. Install the Solaris operating system.

The Solaris installer offers you the option to use UFS or ZFS for the root file system. If the storage server contains multiple disks and the other disks are exclusively used for the VDI ZFS pools, either of the two choices is fine. If only one disk is available, choose ZFS.

2. Enable root access.

- a. Edit the file `/etc/ssh/sshd_config` and change the line `PermitRootLogin no` to `PermitRootLogin yes`
- b. Restart the SSHD service to implement the changes you made to the `sshd_config` file.

```
# svcadm restart ssh
```

3. (Optional) Create a ZFS pool.

If ZFS was selected during installation, the Solaris installer created a pool named `rpool`. This pool contains the root file systems and can also be used by Sun VDI. Create a dedicated pool to separate the Sun VDI data from the OpenSolaris/Solaris file systems.

```
# zpool create <pool name> <disk1> <disk2> <disk3> ...
```

4. Enable iSCSI access.

Type the following CLI command as root on the Solaris storage server.

```
# svcadm enable svc:/system/iscsitgt:default
```

How to Set Up an OpenSolaris Storage Server

An OpenSolaris storage host must be an x86 system. Each virtual disk is represented by a ZFS volume, and the volumes are stored in a ZFS pool

that is accessed by VirtualBox through iSCSI. The management of the ZFS volumes is done by Sun VDI and requires SSH root access to the ZFS storage server and a ZFS pool on that server.

Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date OpenSolaris version support information.

Steps

1. Install the OpenSolaris operating system.

The OpenSolaris installer offers you the option to create a user. It is important to accept this option otherwise several steps of the configuration will fail.

2. Enable root access.

- a. Log into the ZFS storage host. If you are working remotely, use the newly created user account:

```
$ ssh <username>@<ZFS_Storage_Host>
```

- b. Become the root user.

```
$ su
```

- c. Transform the root role to the root user.

```
# rolemod -K type=normal root
```

- d. Edit the file `/etc/ssh/sshd_config` and change the line `PermitRootLogin no` to `PermitRootLogin yes`.

- e. Restart the SSHD service to implement the changes you made to the `sshd_config` file.

```
# svcadm restart ssh
```

3. (Optional) Create a ZFS pool.

The OpenSolaris installer has already created a pool named `rpool`. This pool contains the OpenSolaris/Solaris file systems and can also be used by Sun VDI. Create a dedicated pool to separate the Sun VDI data from the OpenSolaris/Solaris file systems.

```
# zpool create <pool name> <disk1> <disk2> <disk3> ...
```

4. Install the iSCSI packages.

Install the iSCSI packages on the storage server. You can add the packages from the command line of the storage server. Internet connectivity is required.

```
pkg install SUNWiscsi  
pkg install SUNWiscsitgt
```

5. Enable iSCSI access.

Type the following CLI command as root on the Solaris storage server:

```
# svcadm enable svc:/system/iscsitgt:default
```

How to Set Up a Sun Storage 7000 Unified Storage System

To see an up-to-date list of supported Sun Storage 7000 Unified Storage Systems, see the [VDI 3.1 Release Notes](#).

Steps

1. Set up the system.
Follow the instructions provided by the "Quick Setup" manual of the Sun Unified Storage System.
2. (Optional) Update the Sun Unified Storage System Software.
Updating the system to at least version ak-2008.11.20.1.0,1-1.5 provides access to important performance features.
3. (Optional) Create a project.
You can create your own project or use the supplied project called 'default' which can be used by Sun VDI.



Do not update the software of a Sun Storage 7000 Unified Storage System after it has been added to VDI. See the [VDI 3.1 Release Notes](#) for more information.

About Maintenance Mode

On occasion, you might need to off-line a configured host or storage, including maintenance, upgrades, and decommissioning. The Maintenance Mode feature, introduced in VDI 3.1, allows virtual machines to be cleared from a currently used host or storage and moved to a different host or storage so that normal operations may continue while the initial host or storage is unavailable. This process is also considered a "cold" migration because running virtual machines will be suspended to allow the maintenance process to proceed.

Maintenance Mode is available for Sun VirtualBox and Microsoft Hyper-V desktop providers only.

Host Maintenance

The two ways a host can be put in Maintenance mode are:

Migrate Desktops or Shutdown and Restart Desktops on Another Host

- Requires > 1 VirtualBox host.
- Migrate Desktops - Offered only if VDI thinks there are other compatible hosts. Otherwise, "Shutdown and Restart Desktops" is offered.
- Desktops are migrated one after the other. A desktop being migrated will be temporarily unavailable for up to a minute.

Suspend Desktops on a Host

- Always offered.
- Suspends all desktops on the current host.
- If a suspended desktop is requested it will be resumed on another VirtualBox host, if available.



A compatible VirtualBox host must have identical, or very similar, CPU models. Attempting to resume a desktop on a different CPU model will often result in a failure. VDI 3.1 verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible.

VDI 3.1 also checks for valid VirtualBox versions. Migrating from VirtualBox 3.0 to VirtualBox 2.0 is not supported.

Storage Server Maintenance

Suspend Desktops on a Storage Server

- Always offered.
- No data is moved or deleted from the specified storage server.
- Each desktop on the storage server is suspended.
- Desktops will be unavailable until the storage server is re-enabled.
- Desktop hard disk data remains on the storage server.

For details about how to enable the Maintenance Mode feature, refer to the [How to Use Maintenance Mode](#) page.

How to Use Maintenance Mode

In version 3.1, VDI provides one or more mechanisms to clear or suspend running desktops on virtualization hosts and storage servers. Maintenance Mode allows administrators to perform typical maintenance tasks on servers (rebooting, upgrading, etc.) with minimal impact to users.

VirtualBox and Hyper-V Host Maintenance

Desktop providers with a single VirtualBox or Hyper-V host will be able to suspend all running desktops associated with the specified host.

Desktop providers with more than one VirtualBox host allow running desktops to be migrated to other enabled hosts. Depending on host compatibility one of two migration options will be available. If VDI believes there are other compatible hosts, it will attempt to migrate each desktop by suspending it and resuming it on another host. If no compatible hosts are detected, VDI will attempt to shutdown and restart the desktops on other hosts.

A compatible VirtualBox host must have identical, or very similar, CPU models. VDI verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible. VDI also checks for valid VirtualBox versions. Migrating from VirtualBox 3.0 to VirtualBox 2.0 is not supported.

VDI Manager Steps

1. Select the Desktop Providers category, and click the desktop provider containing the host you would like to suspend.
2. Select the Host tab, select the host to migrate and click the Maintenance button.
A pop-up window will appear with two options depending on host compatibility.
 - a. Choose maintenance type.
 - If you will be moving the desktops to a different host, select the Migrate Desktops option.
 - If you will be suspending all desktops on the host, select the Suspend Desktops option.
 - b. Choose a time for the server to begin entering maintenance or click Now to select the current time.
 - c. Click OK to submit the maintenance mode job.

Storage Maintenance

VDI provides a mechanism to put one or more storage servers in Maintenance Mode. Maintenance Mode implies that the storage server is disabled and all running desktops are either shutdown or suspended. At this point, maintenance can take place on the storage server (rebooting, upgrading, etc.). When the storage server is re-enabled, any desktops suspended as a result of entering maintenance mode will be resumed.

VDI Manager Steps

1. Select the Desktop Providers category, and click the desktop provider containing the storage server you would like to suspend.
2. Select the Storage tab, select the storage server, and click the Maintenance button.
 - a. Choose a time for the server to begin entering maintenance, or click Now to select the current time.
 - b. Click OK to submit the maintenance mode job.



General Note for Storage and Virtualization Hosts

All running desktops associated with a host or storage server can be manually shut down or suspended by navigating to the Storage or Host Desktop tab. This tab can be accessed by clicking the Host or Storage server hyperlink.

How to Replicate and Replace a Storage Server

Storage replication is a useful technique to increase storage server availability for lower budget VDI installations. After replicating a Sun Storage 7000 Unified Storage System or OpenSolaris Storage Server, the VDI storage replace feature allows you to easily enable the replicated storage server from the VDI Manager if, for some reason, the storage server fails.

Before You Begin

1. Configure the storage server for replication and replicate it.

- Sun Storage 7000 Unified Storage System

Replication is a built-in feature, and can be configured from the Sun Unified Storage System UI.

- a. Select a project and choose Replication in the menu bar.
- b. Then add a storage as the replication target.

Once the replication is finished, you have two storages with identical ZFS structures and you can use the replication target as the target in the Replace Storage wizard in the VDI Manager.

- OpenSolaris Storage Server

Use the following steps to configure and manually run storage replication.

- a. Snapshot the whole pool, by running the following command.

```
zfs snapshot <pool>@rep
```

- b. Snapshot all volumes, by running the following command for each volume.

```
zfs snapshot <pool>/<volume>@rep
```

- c. Create a replication stream, by running the following command.

```
zfs send -R <pool>@rep | ssh root@<host> zfs receive -dF <newpool>
```

- d. Remove all @rep snapshots on the source and the target storage servers, by running the following command for each volume.

```
zfs destroy <pool>@rep / zfs destroy <pool>/<volume>@rep
```

VDI Manager Steps

If a storage server fails, use the procedure below to replace and re-enable the storage server.

1. Disable the failed storage server.
 - a. Select the Desktop Providers category, and a desktop provider that uses the failed storage server.
 - b. Select the Storage tab, and transition the failed storage to the maintenance mode.
2. Enable the new storage server.
 - a. In the Storage tab, click Replace to activate the Replace Storage wizard. Enter information about the new storage (target of the replication).
 - b. Select the new storage server from Storage overview, and click Enable.

Contents

- [How to Set Up a VMware ESX Server](#)
 - [How to Set Up a VMware vCenter Server](#)
 - [How to Create Virtual Machines \(VMware vCenter\)](#)
 - [How to Test the Platform Setup \(VMware vCenter\)](#)
-

VMware vCenter Virtualization Platform Installation (All Topics)

How to Set Up a VMware ESX Server

VMware ESX server is a Linux-based appliance that provides a virtualization platform by abstracting CPU resources, storage, and memory of a physical host into multiple virtual machines.

Steps

1. Power on the host machine with the VMware ESX Server CD in the CD drive.
If available, you can also use remote management applications such as the Integrated Lights Out Manager (ILOM) to drive the installation.
2. During installation, you can safely rely on the suggested default settings.
Refer to "Chapter 7: Installing VMware ESX Server Software" in the [VMware ESX Server 3 and VirtualCenter Installation Guide](#).
3. After installation, install the VMware Virtual Infrastructure Client so that you can access the VMware ESX server.
Refer to page 74, "Installation Procedure for VMware Infrastructure Management" in the [VMware ESX Server 3 and VirtualCenter Installation Guide](#).

How to Set Up a VMware vCenter Server

VMware vCenter provides central management of several ESX servers, and can be installed on a physical or virtual host. Refer to "Chapter 6: Installing VMware Infrastructure Management" in the [VMware ESX Server 3 and VirtualCenter Installation Guide](#).

Steps

Once VMware vCenter is installed, complete the following configuration steps:

1. Add the VMware ESX server as a managed host.
In VMware vCenter select the datacenter where the host will be added. In the menu bar go to Inventory > Datacenter > Add Host, and follow the instructions.
2. Install the Windows System Preparation Tools for Windows XP.
These tools can be downloaded from the following Microsoft web sites:
 - [Windows XP Service Pack 2 Deployment Tools](#)
 - [Windows XP Service Pack 3 Deployment Tools](#)
3. Extract the Sysprep tools from the CAB into the following directory:

```
C:\Documents and Settings\All Users\Application Data\VMware\VMware VirtualCenter\sysprep\xp
```

For further instructions, see "Appendix B: Installing the Microsoft Sysprep Tools" in the [VMware Basic System Administration Guide](#).

4. Verify that the server is configured for access to the web services API.
Sun VDI takes advantage of the web services API provided by the VMware Infrastructure SDK to communicate through HTTPS with VMware vCenter.
 - a. Verify that the VMware vCenter Webaccess component is installed and configured.
 - b. Verify that Port 443 (HTTPS) is enabled in any firewall that may be active on the system.
 - c. As a simple test, go to `https://<vCenter Host>/mob`. If everything works correctly, you will have access to the VMware Infrastructure SDK browser (after providing the VMware administrator user name and password).



The previous version of Sun VDI required the installation of a Virtual Desktop Connector agent on the VMware vCenter host. Starting with Sun VDI 3, this step is no longer necessary.

How to Create Virtual Machines (VMware vCenter)

Sun VDI presents users with easy access to their virtual desktops, typically instances of Microsoft Windows XP executed in a virtual machine. You

can manually create virtual machines, or you can configure Sun VDI to create or clone additional virtual machines automatically from a template.

Steps

1. Create a virtual machine with Microsoft Windows.

Use your standard process for creating virtual machines. For full details, see "Creating Virtual Machines" in [VMware Basic System Administration](#).

Follow these recommendations:

- Use Microsoft Windows XP SP3 as the baseline. The license must be a volume license.
- Define one disk. It should be as small as possible, for example, 4 GB. The size impacts system performance and overall storage consumption.
- RAM also should be as small as possible, for example, 384 MB.
- A single CPU should be enough.
- One network interface is needed. It should be configured for DHCP. Ensure that the virtual machine obtains a valid IP after powering on.

2. Install the VMware Tools.

Once you have created a virtual machine with Microsoft Windows XP installed on it, install VMware Tools. VMware Tools is a suite of utilities that enhances the performance of the virtual machine's guest operating system and improves management of the virtual machine. Installing VMware Tools in the guest operating system is vital.

The installation can be easily triggered from within the VMware Virtual Infrastructure Client (VIC): Right-click the virtual machine and choose Install VMware Tools. For additional details see Installing and Upgrading VMware Tools in [VMware Basic System Administration](#).

3. Enable remote desktop access.

RDP is the main access method to the Microsoft Windows XP desktop. By default, this access method is disabled and rejected through the firewall. To enable remote desktop access, launch VMware's Virtual Infrastructure Client, with your virtual machine still powered on and logged in, then follow these steps:

- a. Open a console for the virtual machine, and click the virtual machine's Start button.
- b. Right-click on My Computer in the start menu, and select Properties.
- c. In the System Properties window, select the Remote tab.
- d. Under Remote Desktop, check the box marked Enable Remote Desktop on This Computer so that this item is selected.
- e. Make sure that the desired users have been granted remote access rights.
- f. Click OK to save the settings and close the dialog.

Before you try to connect to a virtual desktop remotely, ensure that no firewall blocks the remote access. Make sure that port 3389 is enabled in any firewall that may be active on the system.

4. Install the Sun VDA Tools.

VDI has a tools component that notifies the VDI service when a desktop is in use and handles RDP connections when the guest OS initiates Standby. The VDI Tools must be installed on the guest operating system for recycling to work correctly and so that the RDP connection is correctly closed when the virtual machine goes into Standby or Suspend mode.

- a. Locate the `vda-tools.msi` installer file in the directory where you unzipped the VDI archive.
The `vda-tools.msi` is located in the `./image/vda_3.1/Windows/Packages/` subdirectory. Copy the installer to the desired virtual machine.
- b. Within the virtual machine's console, double-click the installer and follow the prompts to complete installation.
The default target location for the VDI Tools on Windows is `C:\Program Files\Sun\Virtual Desktop Access\Tools`.
- c. The VM Services list should now contain a new service named Sun VDI Tools, running and set to start automatically.

5. Configure power management.

An unused virtual desktop is treated like a laptop that is unplugged to save power. It is suspended automatically to release all its CPU and memory consumption. Suspending virtual machines also affects how they are recycled. A virtual machine will be recycled if it has been suspended for longer than the recycling idle timeout period, even if the user has not logged out of the machine.

- a. Open the Virtual Infrastructure Client.
- b. Select the desired virtual machine and open the console.
- c. Log into the virtual machine.
- d. Go to Start -> Control Panel. Open Power Options.
- e. Set the System Standby time to the desired value.
- f. Verify that the guest OS actually enters standby as configured.
Virtual machines should be configured to be suspended when the OS goes into standby. This is enabled in VMware vCenter.
- g. Open the Virtual Infrastructure Client.
- h. Right-click on the desired virtual machine and go to Edit Settings
- i. Go to Options -> Power Management, select Suspend the Virtual Machine.

6. Convert a virtual machine into a template.

You can clone additional virtual machines manually, or let Sun VDI clone them automatically from a template. Any existing virtual machine can be converted into a template.

- a. Open the Virtual Infrastructure Client.
- b. Right-click the desired virtual machine and power down the machine.
- c. From the commands area or the pop-up menu, click Convert to Template.

For additional details see Chapter 13: Working with Templates and Clones in [VMware Basic System Administration](#).

7. Create a Customization Specification.

It is necessary to customize the identity and network settings of Windows XP after a clone has been created from a template. This can be achieved using a Customization Specification.

- a. Open the Virtual Infrastructure Client.
- b. Click Edit from the menu above the tool bar and select Customization Specifications.
- c. Click the New icon in the Customization Specification Manager to start the wizard.
- d. On the first wizard step, choose Windows as the target virtual machine OS, and give the specification a name and description.
- e. The following steps ask the standard Windows installation questions and should be completed to correspond with your requirements, with the exception of the following.
 - **Computer Name:** Make sure that the Use the Virtual Machine Name item is selected. If not, you may end up with duplicate hostnames.
 - **Windows License:** Enter your Windows XP serial number. The Include Server License Information item should be left unchecked.
 - **Networking:** Make sure the interface is configured for DHCP. If not, your cloned virtual machines will not have unique IP addresses and will not work with Sun VDI.

- f. After completing the wizard and saving your customization specification, close the Customization Specification Manager.

For additional details see Chapter 14: Customizing Guest Operating System in [VMware Basic System Administration](#).

How to Test the Platform Setup (VMware vCenter)

It is highly recommended to test the configuration made so far, before setting up the remaining Sun VDI components. A quick manual test consists of cloning a virtual machine using the desired template and customization specification followed by a remote access to the cloned virtual machine via RDP.

Before You Begin

In order to test the platform setup, you must first create a virtual machine. Refer to the [How to Create Virtual Machines \(VMware vCenter\)](#) page for more details.

Steps

1. Open the Virtual Infrastructure Client.
2. Right-click on the desired template and select Deploy Virtual Machine from this Template.
 - a. The wizard will ask you to specify a name for the new VM - select the desired host/cluster and datastore with sufficient free space.
 - b. On the Guest Customization step, select the Customize Using an Existing Customization Specification option, then choose the customization specification you just created from the list.
 - c. Review your selections, and click Finish to begin cloning.
3. After the cloning has finished, select the new virtual machine and power it on.

After some time you should see its IP address and host name appear in the Virtual Infrastructure Client. Make sure that it has a unique IP address and that the host name corresponds to the virtual machine name.
4. On the VMware vCenter server, open a Remote Desktop Connection by clicking Start > All Programs > Accessories > Communications > Remote Desktop Connection.
 - a. In the Remote Desktop Connection window, enter the IP address of the newly cloned virtual machine, and click Connect.
 - b. If everything is configured correctly, a full-screen remote desktop session to your virtual machine should be displayed.

- [About Microsoft Hyper-V Virtualization Platforms](#)
 - [How to Install Microsoft Hyper-V](#)
 - [How to Prepare a Windows Server for VDI](#)
 - [How to Set Up a Solaris Storage Server](#)
 - [How to Set Up an OpenSolaris Storage Server](#)
 - [How to Set Up a Sun Storage 7000 Unified Storage System](#)
 - [About Maintenance Mode](#)
 - [How to Use Maintenance Mode](#)
 - [VirtualBox and Hyper-V Host Maintenance](#)
 - [Storage Maintenance](#)
 - [How to Replicate and Replace a Storage Server](#)
-

Microsoft Hyper-V Virtualization Platform Installation (All Topics)

About Microsoft Hyper-V Virtualization Platforms

Sun VDI 3.1 provides users access to virtual machines hosted by Microsoft Hyper-V. Microsoft Hyper-V can be installed either as a free stand-alone product (Microsoft Hyper-V Server), or it can be enabled as the Hyper-V role in Windows Server 2008. You can find out more about the different versions on the [Microsoft website](#).

Like the Sun VirtualBox desktop provider, the Microsoft Hyper-V desktop provider takes advantage of iSCSI and ZFS as a part of OpenStorage (Solaris/OpenSolaris, and the Sun 7000 Series). For a VDI/Hyper-V demo, virtual disks can be stored on the VDI Core host, but for a production environment, as with VirtualBox, a Microsoft Hyper-V desktop provider requires a separate storage host.

The Microsoft Hyper-V desktop provider is remotely managed by VDI. To enable communication between the VDI Core and the Windows Server hosting Microsoft Hyper-V, the Windows Server needs to be prepared using the same preparation as for a Microsoft Remote Desktop provider. See the [How to Prepare a Windows Server for VDI](#) page for more details.

How to Install Microsoft Hyper-V

Microsoft Hyper-V can be installed either as a (free) standalone product, or it can be enabled as a role in Windows Server 2008.

- To install the standalone product, please refer to the [Microsoft Hyper-V Server 2008 site](#).
- To install the Hyper-V role, please refer to the [Microsoft Hyper-V Getting Started Guide](#).

How to Prepare a Windows Server for VDI


After installing Microsoft Hyper-V or Microsoft Remote Desktop Services you must prepare your Windows server to communicate with the VDI Core. VDI does not require any agents to be installed on the Windows servers, instead the VDI Core communicates with Windows servers using Windows Remote Management (WinRM) over HTTPS (a secure protocol). For HTTPS, WinRM requires a server certificate to operate properly. This certificate is used for encryption of the communication channel. For more details, see "[Windows Remote Management](#)" or "[Configuration and Security](#)" in the Microsoft documentation.

Preparing the Windows server for communication with the VDI Core is a two-step process. First, you must generate the self-signed certificate using the Microsoft Internet Information Services (IIS) 6.0 Resource Kit Tools (Step 1, below). Then configure `winrm` to listen for HTTPS requests (Step 2, below).



These steps are necessary for Remote Desktop Services (or Terminal Services) Windows servers so that critical information about the server can be displayed in the VDI Manager (including CPU usage, memory usage, and number of user sessions). The delivery of desktop sessions from RDS pools is still provided by a regular RDP connection. For information about how to configure the RDP settings per desktop pool, see the [How to Configure RDP Options Per Pool](#) page.

Before You Begin

 The following commands should be executed in Command shell, not Powershell.

Steps

1. Generate a self-signed certificate on the Windows server.

Use the `selfssl.exe` tool which is part of the IIS 6.0 Resource Kit and can be downloaded from the [Microsoft Support web site](#).

- a. Copy `selfssl.exe` to your Windows Server.
- b. Create a self-signed certificate:

```
selfssl /T /V:<days>
```

There parameter `/V`: dictates the number of days the certificate will be valid. There is no maximum value.

- c. Run the `certutil` command, and make note of the Cert Hash of the new certificate:

```
certutil -store MY
```



If the Windows server and VDI server are not in time sync, you might not be able to connect VDI to the server as the certificate is not valid for the delta between both servers.

2. Configure Windows Remote Management for HTTPS.

The `winrm` tool is used to configure remote management settings on the server. You must specify the certificate hash to be used, and the authentication settings to allow the VDI Core to send requests.

- a. Install WS-Man (WinRM).



This step is for Windows Server 2003 only. Windows Server 2008 and Hyper-V Server 2008 come with WinRM pre-installed.

- b. Download the WS-MAN v1.1. installation file (`WindowsServer2003-KB936059-x86-ENU.exe`) from www.microsoft.com.
- c. Proceed to the installation by running the installation file `WindowsServer2003-KB936059-x86-ENU.exe`.
- d. Create a listener on the Windows Server.

In a command shell run:

```
winrm create winrm/config/listener?Address=IP:<HYPER_IP>+Transport=HTTPS  
@{Hostname="<HOST>";CertificateThumbprint="<CERTHASH>";Port="443" }
```

- Replace `<HYPER_IP>` with the IP address of the Windows Server.
- Replace `<HOST>` with the Computer Name of the Windows Server.
- Replace `<CERTHASH>` with the Cert Hash value, with no spaces, noted from the self-signed certificate created with `selfssl`.

- e. Open that port so that the Windows Server can receive requests from the VDI Core:

```
netsh firewall add portopening TCP 443 "Sun VDI Remote Management"
```

Port 443 is the port the VDI Core listens on by default.

- f. Enable Basic authentication on the server by running the command:

```
winrm set winrm/config/service/auth @{Basic="true" }
```



If you use a port other than 443 for VDI communication with Hyper-V or RDS, you must remember to specify this port when adding the host in VDI Manager.

How to Set Up a Solaris Storage Server

A Solaris storage host must be an x86 system. Each virtual disk is represented by a ZFS volume, and the volumes are stored in a ZFS pool that is accessed by VirtualBox through iSCSI. The management of the ZFS volumes is done by Sun VDI and requires SSH root access to the ZFS storage server and a ZFS pool on that server.

Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date Solaris version support information.

Steps

1. Install the Solaris operating system.

The Solaris installer offers you the option to use UFS or ZFS for the root file system. If the storage server contains multiple disks and the other disks are exclusively used for the VDI ZFS pools, either of the two choices is fine. If only one disk is available, choose ZFS.

2. Enable root access.

- a. Edit the file `/etc/ssh/sshd_config` and change the line `PermitRootLogin no` to `PermitRootLogin yes`
- b. Restart the SSHD service to implement the changes you made to the `sshd_config` file.

```
# svcadm restart ssh
```

3. (Optional) Create a ZFS pool.

If ZFS was selected during installation, the Solaris installer created a pool named `rpool`. This pool contains the root file systems and can also be used by Sun VDI. Create a dedicated pool to separate the Sun VDI data from the OpenSolaris/Solaris file systems.

```
# zpool create <pool name> <disk1> <disk2> <disk3> ...
```

4. Enable iSCSI access.

Type the following CLI command as root on the Solaris storage server.

```
# svcadm enable svc:/system/iscsitgt:default
```

How to Set Up an OpenSolaris Storage Server

An OpenSolaris storage host must be an x86 system. Each virtual disk is represented by a ZFS volume, and the volumes are stored in a ZFS pool that is accessed by VirtualBox through iSCSI. The management of the ZFS volumes is done by Sun VDI and requires SSH root access to the ZFS storage server and a ZFS pool on that server.

Refer to the [VDI 3.1 Release Notes](#) for the most accurate and up-to-date OpenSolaris version support information.

Steps

1. Install the OpenSolaris operating system.

The OpenSolaris installer offers you the option to create a user. It is important to accept this option otherwise several steps of the configuration will fail.

2. Enable root access.

- a. Log into the ZFS storage host. If you are working remotely, use the newly created user account:


```
$ ssh <username>@<ZFS_Storage_Host>
```

b. Become the root user.

```
$ su
```

c. Transform the root role to the root user.

```
# rolemod -K type=normal root
```

d. Edit the file `/etc/ssh/sshd_config` and change the line `PermitRootLogin no` to `PermitRootLogin yes`.

e. Restart the SSHD service to implement the changes you made to the `sshd_config` file.

```
# svcadm restart ssh
```

3. (Optional) Create a ZFS pool.

The OpenSolaris installer has already created a pool named `rpool`. This pool contains the OpenSolaris/Solaris file systems and can also be used by Sun VDI. Create a dedicated pool to separate the Sun VDI data from the OpenSolaris/Solaris file systems.

```
# zpool create <pool name> <disk1> <disk2> <disk3> ...
```

4. Install the iSCSI packages.

Install the iSCSI packages on the storage server. You can add the packages from the command line of the storage server. Internet connectivity is required.

```
pkg install SUNWiscsi  
pkg install SUNWiscsitgt
```

5. Enable iSCSI access.

Type the following CLI command as root on the Solaris storage server:

```
# svcadm enable svc:/system/iscsitgt:default
```

How to Set Up a Sun Storage 7000 Unified Storage System

To see an up-to-date list of supported Sun Storage 7000 Unified Storage Systems, see the [VDI 3.1 Release Notes](#).

Steps

1. Set up the system.

Follow the instructions provided by the "Quick Setup" manual of the Sun Unified Storage System.

2. (Optional) Update the Sun Unified Storage System Software.

Updating the system to at least version `ak-2008.11.20.1.0,1-1.5` provides access to important performance features.

3. (Optional) Create a project.

You can create your own project or use the supplied project called 'default' which can be used by Sun VDI.



Do not update the software of a Sun Storage 7000 Unified Storage System after it has been added to VDI. See the [VDI 3.1 Release Notes](#) for more information.

About Maintenance Mode

On occasion, you might need to off-line a configured host or storage, including maintenance, upgrades, and decommissioning. The Maintenance Mode feature, introduced in VDI 3.1, allows virtual machines to be cleared from a currently used host or storage and moved to a different host or storage so that normal operations may continue while the initial host or storage is unavailable. This process is also considered a "cold" migration because running virtual machines will be suspended to allow the maintenance process to proceed.

Maintenance Mode is available for Sun VirtualBox and Microsoft Hyper-V desktop providers only.

Host Maintenance

The two ways a host can be put in Maintenance mode are:

Migrate Desktops or Shutdown and Restart Desktops on Another Host

- Requires > 1 VirtualBox host.
- Migrate Desktops - Offered only if VDI thinks there are other compatible hosts. Otherwise, "Shutdown and Restart Desktops" is offered.
- Desktops are migrated one after the other. A desktop being migrated will be temporarily unavailable for up to a minute.

Suspend Desktops on a Host

- Always offered.
- Suspends all desktops on the current host.
- If a suspended desktop is requested it will be resumed on another VirtualBox host, if available.



A compatible VirtualBox host must have identical, or very similar, CPU models. Attempting to resume a desktop on a different CPU model will often result in a failure. VDI 3.1 verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible.

VDI 3.1 also checks for valid VirtualBox versions. Migrating from VirtualBox 3.0 to VirtualBox 2.0 is not supported.

Storage Server Maintenance

Suspend Desktops on a Storage Server

- Always offered.
- No data is moved or deleted from the specified storage server.
- Each desktop on the storage server is suspended.
- Desktops will be unavailable until the storage server is re-enabled.
- Desktop hard disk data remains on the storage server.

For details about how to enable the Maintenance Mode feature, refer to the [How to Use Maintenance Mode](#) page.

How to Use Maintenance Mode

In version 3.1, VDI provides one or more mechanisms to clear or suspend running desktops on virtualization hosts and storage servers. Maintenance Mode allows administrators to perform typical maintenance tasks on servers (rebooting, upgrading, etc.) with minimal impact to users.

VirtualBox and Hyper-V Host Maintenance

Desktop providers with a single VirtualBox or Hyper-V host will be able to suspend all running desktops associated with the specified host.

Desktop providers with more than one VirtualBox host allow running desktops to be migrated to other enabled hosts. Depending on host compatibility one of two migration options will be available. If VDI believes there are other compatible hosts, it will attempt to migrate each desktop by suspending it and resuming it on another host. If no compatible hosts are detected, VDI will attempt to shutdown and restart the desktops on other hosts.

A compatible VirtualBox host must have identical, or very similar, CPU models. VDI verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible. VDI also checks for valid VirtualBox versions. Migrating from VirtualBox 3.0 to VirtualBox 2.0 is not supported.

VDI Manager Steps

1. Select the Desktop Providers category, and click the desktop provider containing the host you would like to suspend.
2. Select the Host tab, select the host to migrate and click the Maintenance button.
A pop-up window will appear with two options depending on host compatibility.
 - a. Choose maintenance type.
 - If you will be moving the desktops to a different host, select the Migrate Desktops option.
 - If you will be suspending all desktops on the host, select the Suspend Desktops option.
 - b. Choose a time for the server to begin entering maintenance or click Now to select the current time.
 - c. Click OK to submit the maintenance mode job.

Storage Maintenance

VDI provides a mechanism to put one or more storage servers in Maintenance Mode. Maintenance Mode implies that the storage server is disabled and all running desktops are either shutdown or suspended. At this point, maintenance can take place on the storage server (rebooting, upgrading, etc.). When the storage server is re-enabled, any desktops suspended as a result of entering maintenance mode will be resumed.

VDI Manager Steps

1. Select the Desktop Providers category, and click the desktop provider containing the storage server you would like to suspend.
2. Select the Storage tab, select the storage server, and click the Maintenance button.
 - a. Choose a time for the server to begin entering maintenance, or click Now to select the current time.
 - b. Click OK to submit the maintenance mode job.



General Note for Storage and Virtualization Hosts

All running desktops associated with a host or storage server can be manually shut down or suspended by navigating to the Storage or Host Desktop tab. This tab can be accessed by clicking the Host or Storage server hyperlink.

How to Replicate and Replace a Storage Server

Storage replication is a useful technique to increase storage server availability for lower budget VDI installations. After replicating a Sun Storage 7000 Unified Storage System or OpenSolaris Storage Server, the VDI storage replace feature allows you to easily enable the replicated storage server from the VDI Manager if, for some reason, the storage server fails.

Before You Begin

1. Configure the storage server for replication and replicate it.
 - Sun Storage 7000 Unified Storage System
Replication is a built-in feature, and can be configured from the Sun Unified Storage System UI.
 - a. Select a project and choose Replication in the menu bar.
 - b. Then add a storage as the replication target.
Once the replication is finished, you have two storages with identical ZFS structures and you can use the replication target as the target in the Replace Storage wizard in the VDI Manager.
 - OpenSolaris Storage Server
Use the following steps to configure and manually run storage replication.

- a. Snapshot the whole pool, by running the following command.

```
zfs snapshot <pool>@rep
```

- b. Snapshot all volumes, by running the following command for each volume.

```
zfs snapshot <pool>/<volume>@rep
```

- c. Create a replication stream, by running the following command.

```
zfs send -R <pool>@rep | ssh root@<host> zfs receive -dF <newpool>
```

- d. Remove all @rep snapshots on the source and the target storage servers, by running the following command for each volume.

```
zfs destroy <pool>@rep / zfs destroy <pool>/<volume>@rep
```

VDI Manager Steps

If a storage server fails, use the procedure below to replace and re-enable the storage server.

1. Disable the failed storage server.
 - a. Select the Desktop Providers category, and a desktop provider that uses the failed storage server.
 - b. Select the Storage tab, and transition the failed storage to the maintenance mode.
2. Enable the new storage server.
 - a. In the Storage tab, click Replace to activate the Replace Storage wizard. Enter information about the new storage (target of the replication).
 - b. Select the new storage server from Storage overview, and click Enable.

Contents

- [About Microsoft Remote Desktop Platforms](#)
 - [How to Install Microsoft Remote Desktop Services](#)
 - [How to Prepare a Windows Server for VDI](#)
-

Microsoft Remote Desktop Platform Installation (All Topics)

About Microsoft Remote Desktop Platforms



In Windows Server 2008 R2, Terminal Services (TS) has been renamed Remote Desktop Services (RDS).

Sun VDI 3.1 provides users access to Terminal Services or Remote Desktop Services sessions provided by Windows Server 2003 and Windows Server 2008. VDI relies on some Microsoft tools being implemented along with Remote Desktop Services to provide advanced features such as farm or cluster of Remote Desktop Services hosts (RDS hosts) with load balancing and session reconnection.

Microsoft Terminal Services on Windows Server 2003

VDI provides access to Terminal Services sessions delivered by:

- A single server running Windows Server 2003
- A cluster of servers with the following characteristics:
 - All servers running Windows Server 2003, any edition
 - The cluster is a Microsoft Network Load Balancing (NLB) cluster, which provides load balancing among servers

Microsoft Session Directory may be used to enable users to reconnect to existing sessions.

Microsoft reference documentation:

[Session Directory and Load Balancing Using Terminal Server](#)

[Network Load Balancing Clusters](#)

Microsoft Remote Desktop Services on Windows Server 2008

VDI provides access to Remote Desktop Services sessions delivered by:

- A single server running Windows Server 2008
- A farm of servers with the following characteristics:
 - All servers running Windows Server 2008
 - Microsoft TS Session Broker is used to enable load balancing and enable users to reconnect to existing sessions.

As described in Microsoft documentation, preliminary load balancing may be provided using DNS round robin or Microsoft Network Load Balancing (NLB) or a hardware load balancer.

Microsoft reference documentation:

[TS Session Broker Load Balancing Step-by-Step Guide](#)

[Network Load Balancing Step-by-Step Guide: Configuring Network Load Balancing with Terminal Services](#)

Microsoft Remote Desktop Provider RDS Farm Management

If the Microsoft Remote Desktop provider is composed of RDS hosts belonging to a farm, VDI queries each individual RDS host to determine the sessions running on that host, so that information about these sessions can be displayed by the VDI Manager or CLI. By querying the first RDS host of the provider, VDI detects the information about the farm, and returns the farm name to the Remote Client when users try to access a session so that the session may be started on any host participating in the farm.

Because of this management scheme, administrators should add all the RDS hosts from a farm to the Microsoft Remote Desktop provider, so that VDI can contact each individual RDS host in order to register the local administrator's credentials and the SSL certificate for that host. However, the administrator does not need to enter any information about the farm in the VDI Manager or CLI, VDI detects that RDS hosts belong to a farm by querying them directly.

Limitations of Microsoft Remote Desktop Providers and Pools

Due to the specificity of Remote Desktop Services compared to the other virtualization backends, some restrictions apply to the actions and settings offered by the desktop providers and pools:

- The provider may contain multiple RDS hosts if they are members of the same Network Load Balancing cluster or Session Broker farm. See the section about supported platforms. In this case, Microsoft NLB or Microsoft Session Broker are responsible for the load balancing across the hosts. VDI does not perform any such load balancing.
- The provider may contain a single stand-alone RDS host.
- A given RDS host can only participate in one provider.
- Each Microsoft Remote Desktop provider can have one pool only.
- The pool does not offer any cloning capabilities. The Remote Desktop Services server or farm is responsible for opening new RDS sessions when new users connect.
- The type of assignment in the pool is always flexible. Session reconnection will be offered if the RDS setup is correctly configured for it, but is not the responsibility of VDI. See section about supported platforms.
- The list of desktops displayed for this pool correspond to the aggregated list of RDS sessions from all RDS hosts associated with the provider. All RDP sessions, whether they come from VDI or not, are displayed.
- Groups or users may be assigned to RDS pools but you cannot manually assign a user to an RDS desktop.


How to Install Microsoft Remote Desktop Services

- To install the Remote Desktop Services role on Windows Server 2003 refer to <http://www.microsoft.com/windowsserver2003/technologies/terminalservices/default.mspx>.
- To install the Remote Desktop Services role on Windows Server 2008 refer to


How to Prepare a Windows Server for VDI

After installing Microsoft Hyper-V or Microsoft Remote Desktop Services you must prepare your Windows server to communicate with the VDI Core. VDI does not require any agents to be installed on the Windows servers, instead the VDI Core communicates with Windows servers using Windows Remote Management (WinRM) over HTTPS (a secure protocol). For HTTPS, WinRM requires a server certificate to operate properly. This certificate is used for encryption of the communication channel. For more details, see "[Windows Remote Management](#)" or "[Configuration and Security](#)" in the Microsoft documentation.

Preparing the Windows server for communication with the VDI Core is a two-step process. First, you must generate the self-signed certificate using the Microsoft Internet Information Services (IIS) 6.0 Resource Kit Tools (Step 1, below). Then configure `winrm` to listen for HTTPS requests (Step 2, below).

 These steps are necessary for Remote Desktop Services (or Terminal Services) Windows servers so that critical information about the server can be displayed in the VDI Manager (including CPU usage, memory usage, and number of user sessions). The delivery of desktop sessions from RDS pools is still provided by a regular RDP connection. For information about how to configure the RDP settings per desktop pool, see the [How to Configure RDP Options Per Pool](#) page.

Before You Begin

 The following commands should be executed in Command shell, not Powershell.

Steps

1. Generate a self-signed certificate on the Windows server.

Use the `selfssl.exe` tool which is part of the IIS 6.0 Resource Kit and can be downloaded from the [Microsoft Support](#) web site.


- a. Copy `selfssl.exe` to your Windows Server.
- b. Create a self-signed certificate:

```
selfssl /T /V:<days>
```

There parameter `/V:` dictates the number of days the certificate will be valid. There is no maximum value.

- c. Run the `certutil` command, and make note of the Cert Hash of the new certificate:


```
certutil -store MY
```

 If the Windows server and VDI server are not in time sync, you might not be able to connect VDI to the server as the certificate is not valid for the delta between both servers.

2. Configure Windows Remote Management for HTTPS.

The `winrm` tool is used to configure remote management settings on the server. You must specify the certificate hash to be used, and the authentication settings to allow the VDI Core to send requests.

- a. Install WS-Man (WinRM).

 This step is for Windows Server 2003 only. Windows Server 2008 and Hyper-V Server 2008 come with WinRM pre-installed.

- b. Download the WS-MAN v1.1. installation file (WindowsServer2003-KB936059-x86-ENU.exe) from www.microsoft.com.
- c. Proceed to the installation by running the installation file WindowsServer2003-KB936059-x86-ENU.exe.
- d. Create a listener on the Windows Server.

In a command shell run:

```
winrm create winrm/config/listener?Address=IP:<HYPER_IP>+Transport=HTTPS
@{Hostname="<HOST>";CertificateThumbprint="<CERTHASH>";Port="443" }
```

- Replace <HYPER_IP> with the IP address of the Windows Server.
 - Replace <HOST> with the Computer Name of the Windows Server.
 - Replace <CERTHASH> with the Cert Hash value, with no spaces, noted from the self-signed certificate created with selfssl.
- e. Open that port so that the Windows Server can receive requests from the VDI Core:

```
netsh firewall add portopening TCP 443 "Sun VDI Remote Management"
```

Port 443 is the port the VDI Core listens on by default.

- f. Enable Basic authentication on the server by running the command:

```
winrm set winrm/config/service/auth @{Basic="true" }
```



If you use a port other than 443 for VDI communication with Hyper-V or RDS, you must remember to specify this port when adding the host in VDI Manager.

Contents

- About VDI Core Configuration
 - Single Host Configuration
 - High Availability (HA) Configurations
- How to Install and Configure the VDI Core (Single VDI Host)
 1. Install and configure the remote MySQL database on the single VDI host.
 - How to Install and Configure a Remote MySQL Database (InnoDB)
 2. Create a privileged database administrator.
 - How to Create a Privileged Database Administrator
 3. Install and configure the VDI Core on the single VDI host.
 - How to Install and Configure a Remote MySQL Database (InnoDB)
 - How to Create a Privileged Database Administrator
 - How to Back Up and Restore Data (Remote MySQL Database)
 - How to Check the VDI Core Services and Logs
 - How to Remove the VDI Core

Single Host Configuration (All Topics)

Contents

- Single Host Configuration
 - High Availability (HA) Configurations
 - VDI Core (Bundled MySQL Database)
 - VDI Core (Remote MySQL Database)
-

About VDI Core Configuration

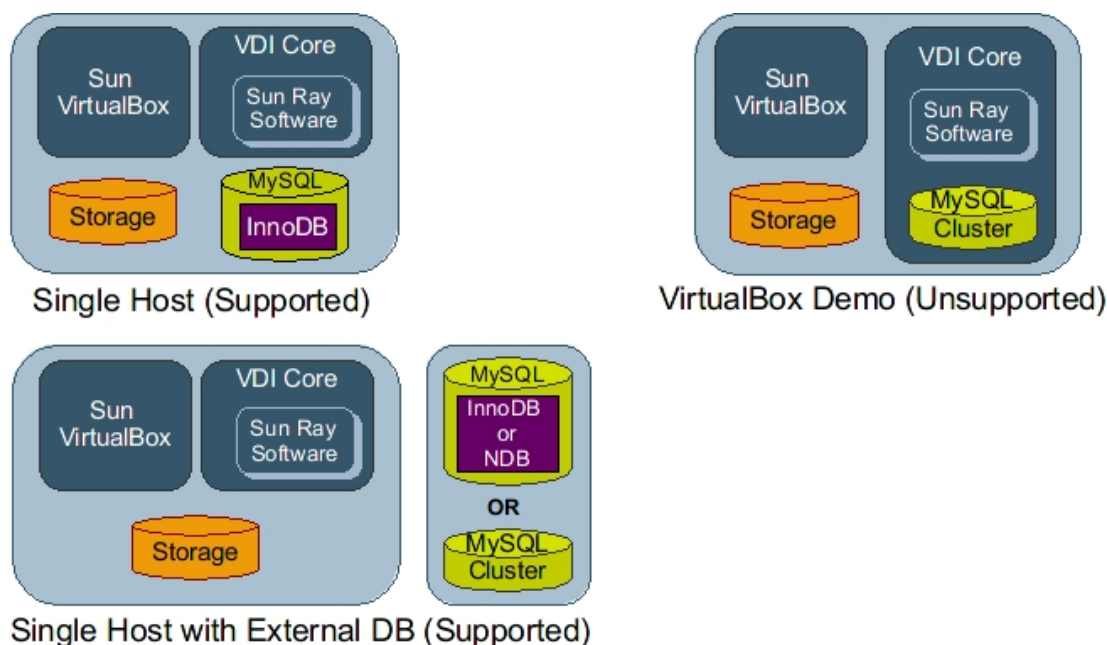
There are many possible configurations for the VDI Core. Some of the configurations are supported for production environments, while others (such as Demo/Evaluation configurations) are not. For the full details about supported and unsupported VDI Core configurations, refer to the [Supported Configurations](#) page.

The VDI Core installation and configuration information focuses on two supported VDI Core configuration types: Single Host, and High Availability. The Single Host configuration is simple to set up and requires minimal hardware while the High Availability configuration provides failover capabilities and better performance.

Single Host Configuration

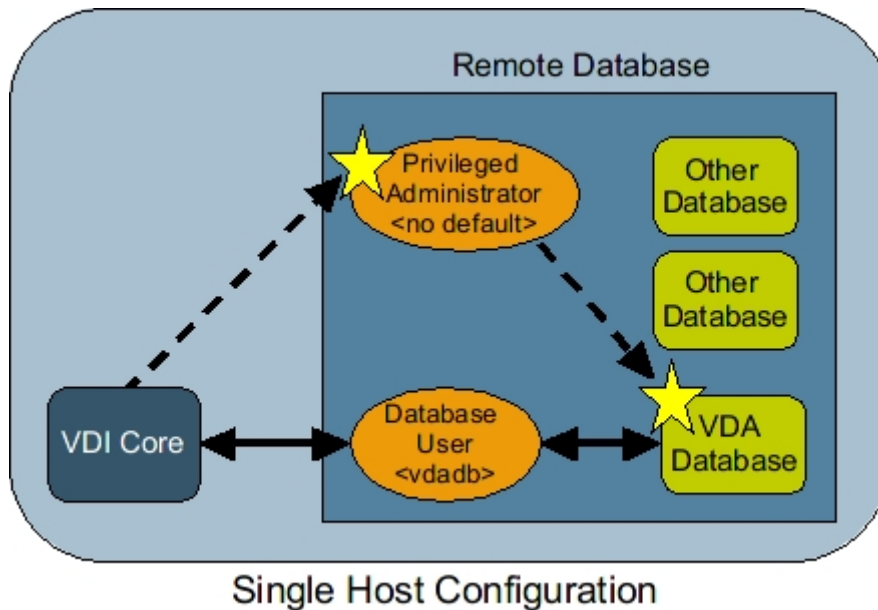
The Single Host configuration is the recommended configuration for deployments that prioritize low cost above availability, since there are no failover capabilities for a one machine setup.

The Single Host configuration is similar to a Demo (VirtualBox) configuration because all necessary components can be installed on one machine, including a Sun VirtualBox virtualization platform, VDI Core (with MySQL database and Sun Ray Software). The difference between a Single Host configuration and a Demo (VirtualBox) configuration is that the Demo uses the bundled MySQL cluster database, and the Single Host configuration requires a MySQL "remote" database. The MySQL database can be installed on the Single Host machine or on a different machine. Even if the MySQL database is installed on the same machine as all the other components, it is still considered to be "remote" since it's not bundled with the VDI Core.



If the MySQL remote database is installed on the same machine as the VDI Core, the database must have an InnoDB storage engine to be considered a supported configuration. This Single Host configuration also requires a separate support contact from MySQL to cover database support. If the MySQL remote database is installed on a different machine than the VDI Core, the requirements for the database are more flexible. It must be a MySQL version 5.0 or higher, with a transactional storage engine (usually InnoDB or NDB), or a MySQL Cluster version 6.2.15 or higher.

A remote database requires a privileged database administrator. The privileged database administrator is used by the VDI Core to create and configure the VDA database during VDI Core configuration. After the VDA database is created, it will be accessed by the VDI core through the database user (default 'vdadb').



Suggested Pages

Refer to the following pages if you plan to set up a Single Host VDI configuration.

- [How to Install and Configure the VDI Core \(Single VDI Host\)](#) - Describes all necessary steps for setting up the VDI Core in a Single Host configuration, including [How to Install and Configure a Remote MySQL Database \(InnoDB\)](#), [How to Create a Privileged Database Administrator](#), and how to install the VDI Core.
- [How to Check the VDI Core Services and Logs](#) - Describes how to check the services and statuses of the VDI Core, bundled MySQL database, and RDP broker.

High Availability (HA) Configurations

High availability offers reliability that if one server fails another one will continue to host VDI desktop sessions with only a minimal interruption to the end-user. High availability configurations are suggested for use cases where fail-safety is prioritized over low price.

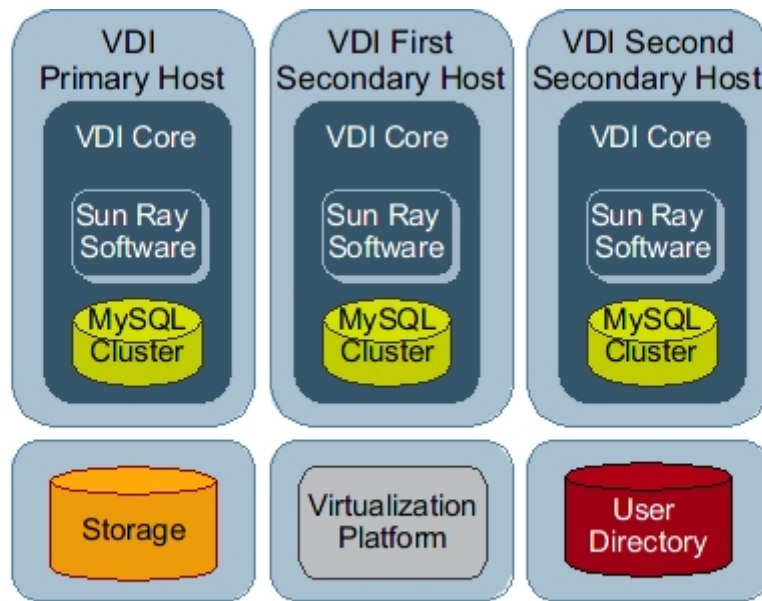
The VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one VDI host fails, all users with desktop sessions on that host are kicked back to the VDI Login dialog, and must reconnect to their session, which is restarted on one of the available hosts. The bundled MySQL cluster database requires three hosts to be highly available. There should never be an interruption of database service as long as no more than one of the three required hosts fails at a time. Therefore, the bundled VDI stack requires a minimum of three VDI hosts to be considered fail-proof. This number of hosts does not include the virtualization platform hosts, which should be considered for failover separately.

VDI also supports the option to connect to a remote MySQL database, instead of the bundled MySQL cluster database. In this case, the VDI Core only requires two hosts to be highly available. This number of hosts does not include the remote database hosts, or the virtualization platform hosts, which should be considered for failover separately.

VDI Core (Bundled MySQL Database)

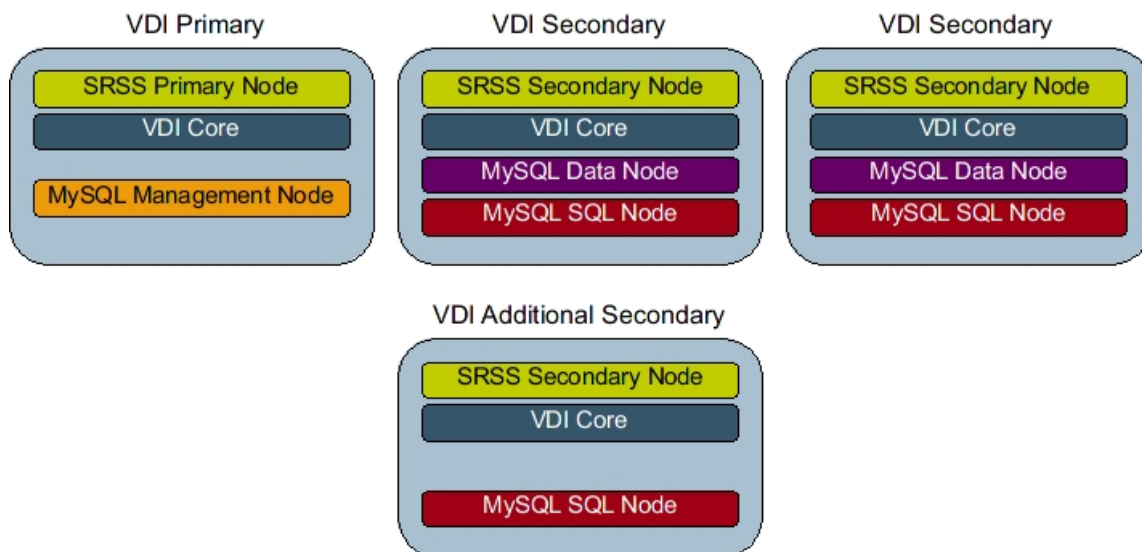
The High Availability configuration with the bundled MySQL database is automatically installed during VDI Core installation, and configured by choosing Primary or Secondary during VDI Core configuration. The VDI Core with bundled MySQL database requires one host to be configured as Primary, and the other two to be configured as Secondary.

i The documentation refer to the two Secondary hosts as "First Secondary" and "Second Secondary", while the configuration script refers to them as "Secondary A" and "Secondary B". Both naming schemes refer to identical hosts, and are meant to help you differentiate between the two during configuration and maintenance.



High Availability (Bundled MySQL Database) Configuration

By selecting VDI Primary during the configuration step, the following will be installed by default: MySQL Management Node, Sun Ray Server Software Primary Node, and a VDI Core Primary Node. By selecting VDI Secondary during the configuration step, the following will be installed by default: MySQL Data Node, MySQL SQL Node, Sun Ray Server Software Secondary Node, and a VDI Core Secondary Node. Any additional VDI Secondary hosts have a MySQL SQL Node, Sun Ray Server Software Secondary Node, and a VDI Core Secondary Node.



Default Configuration for VDI Primary and Secondary Hosts

The MySQL Cluster Nodes perform the following functions:

- Management Node - Manages the other nodes within the MySQL Cluster, performing functions such as providing configuration data, starting and stopping nodes, and running backups.
- Data Node - Stores cluster data.
- SQL Node - Serves as an interface to access the cluster data. This is a traditional MySQL server that uses the NDB Cluster storage engine.

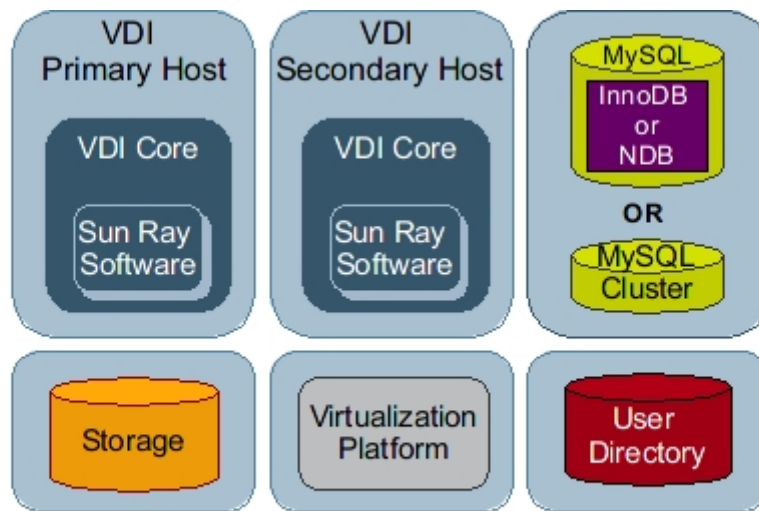
Suggested Pages

Refer to the following pages for information about the High Availability configuration using the bundled MySQL Cluster database.

- [How to Install and Configure the VDI Core \(Bundled MySQL Database\)](#) - Describes all necessary steps for setting up the VDI Core in a High Availability configuration with the bundled MySQL Cluster database, including [How to Prepare a VDI Primary Host](#) and [How to Prepare a VDI Secondary Host](#) procedures.
- [About VDI MySQL Cluster Reconfiguration](#)- Provides an overview of the MySQL Cluster reconfiguration.
- [How to Reconfigure the MySQL Cluster](#) - Explains how to convert any MySQL Node type to a different MySQL Node type.
- [How to Perform a Rolling Restart of the MySQL Cluster](#) - Describes how to start and stop each node of the MySQL Cluster so that the cluster, as a whole, remains operational.
- [How to Prevent Unrestricted SQL Node Joins](#) - Explains how to restrict any non-VDI SQL nodes from joining the bundled MySQL Cluster.
- [How to Check the VDI Core Services and Logs](#) - Describes how to check the services and statuses of the VDI Core, bundled MySQL database, and RDP broker.

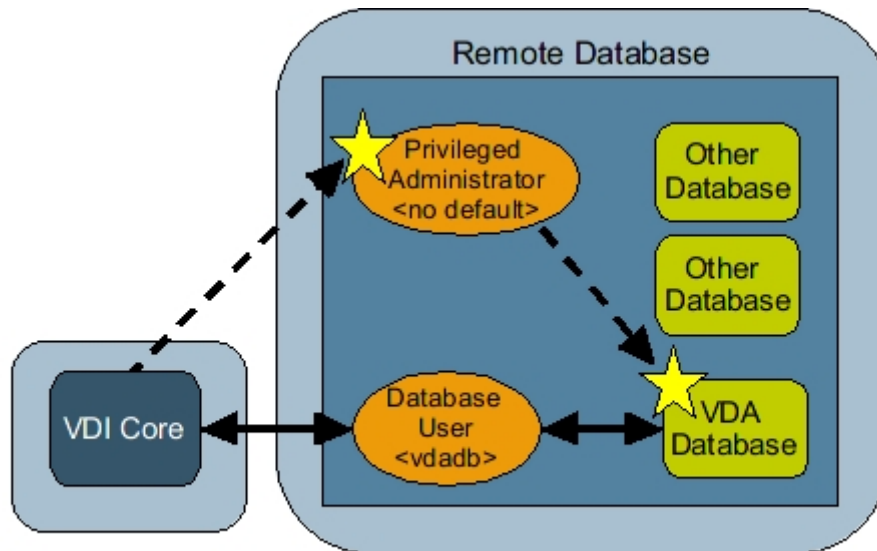
VDI Core (Remote MySQL Database)

The High Availability configuration with the remote MySQL database requires a MySQL version 5.0 or higher, with a transactional storage engine (usually InnoDB or NDB), or a MySQL Cluster version 6.2.15 or higher.



High Availability (Remote MySQL Database) Configuration

Before the VDI Core can be installed, the database must be installed and configured with a privileged database administrator. The privileged database administrator is used by the VDI Core to create and configure the VDA database during VDI Core configuration. After the VDA database is created, it will be accessed by the VDI core through the database user (default 'vdadb').



High Availability (Remote MySQL Database) Configuration

Because a remote database is used instead of the bundled MySQL Cluster, the VDI Core only requires one Primary and one Secondary to provide high availability.

Suggested Pages

Refer to the following pages for information about the High Availability configuration using a remote MySQL database.

- [How to Install and Configure the VDI Core \(Remote MySQL Database\)](#) - Describes all necessary steps for setting up the VDI Core in a High Availability configuration with a remote MySQL database, including [How to Prepare a VDI Primary Host](#) and [How to Prepare a VDI Secondary Host](#) procedures.
- [How to Install and Configure a Remote MySQL Database \(InnoDB\)](#) - Describes how to install a MySQL database with an InnoDB storage engine. Refer to this page if you do not already have a remote database, but are interested in using one with VDI.
- [How to Create a Privileged Database Administrator](#) - Explains how to configure a privileged database administrator so that the VDA database can be created.
- [How to Check the VDI Core Services and Logs](#) - Describes how to check the services and statuses of the VDI Core, bundled MySQL database, and RDP broker.

How to Install and Configure the VDI Core (Single VDI Host)

The [Single Host VDI Configuration](#) should only be used in production environments where fail-over is not required.

1. Install and configure the remote MySQL database on the single VDI host.

How to Install and Configure a Remote MySQL Database (InnoDB)

This page describes how to install MySQL 5.1 (with an InnoDB storage engine) on an x86 platform running Solaris.

Steps

1. Create the file `/etc/my.cnf`, and add the following content.

```

[mysqld]
user=mysql
datadir=/usr/local/mysql/data
basedir=/usr/local/mysql
port=3306
socket=/tmp/mysql.sock
max_allowed_packet=20M
#transaction_isolation=READ-COMMITTED
lower_case_table_names=1
max_connections=1000
skip-locking
key_buffer=16K
table_cache=4
sort_buffer_size=64K
net_buffer_length=2K
thread_stack=64K
wait_timeout=31536000

innodb_data_home_dir=/usr/local/mysql/data
innodb_data_file_path=ibdata1:10M:autoextend
innodb_log_group_home_dir=/usr/local/mysql/data
innodb_buffer_pool_size=50M
innodb_additional_mem_pool_size=10M
innodb_log_file_size=5M
innodb_log_buffer_size=10M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50

```

2. Create a user "mysql" and a group "mysql" by running the following commands.

```

# groupadd mysql
# useradd -g mysql mysql

```

3. Get the MySQL tar file (mysql-5.1.30-solaris10-i386.tar) , untar it, and keep it in the / directory.
4. Create the directory /usr/local, by running the following command.

```

# mkdir /usr/local

```

5. Change to the new directory, and create a symbolic link, called "mysql", that points to the MySQL files in the / directory, by running the following commands.

```

# cd /usr/local
# ln -s /mysql-5.1.30-solaris10-i386 mysql
# ls -lrt

total 2
lrwxrwxrwx  1 root  root           35 Nov 12 17:33 mysql ->
/export/mysql-5.1.30-solaris10-i386
bash-3.00#

```

6. Make sure that the / directory contains the proper owner and group permissions by running the following commands.

```
# chgrp -R mysql /mysql-5.1.30-solaris10-i386
# chown -R mysql /mysql-5.1.30-solaris10-i386
```

7. Check the permissions for the /usr/local/mysql directory as well.

```
# cd /usr/local/mysql
# ls -lrt

-rw-r--r--  1 mysql  mysql  19071 Nov 15 13:07 COPYING
-rw-r--r--  1 mysql  mysql   5139 Nov 15 13:07 EXCEPTIONS-CLIENT
-rw-r--r--  1 mysql  mysql   8767 Nov 15 13:07 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql   1410 Nov 15 13:07 README
drwxr-xr-x  2 mysql  mysql   1536 Nov 15 13:07 bin
drwxr-xr-x  4 mysql  mysql    512 Nov 15 13:07 data
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:05 docs
drwxr-xr-x  2 mysql  mysql   1024 Nov 15 13:05 include
drwxr-xr-x  3 mysql  mysql   1024 Nov 15 13:06 lib
drwxr-xr-x  4 mysql  mysql    512 Nov 15 13:06 man
drwxr-xr-x 10 mysql  mysql    512 Nov 15 13:07 mysql-test
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:07 scripts
drwxr-xr-x 27 mysql  mysql   1024 Nov 15 13:07 share
drwxr-xr-x  5 mysql  mysql   1024 Nov 15 13:07 sql-bench
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:07 support-files
```

8. From the /usr/local/mysql directory, run the following command, and check that it provides the corresponding output.

```
# ./scripts/mysql_install_db --user=mysql
```

To start mysqld at boot time you have to copy support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

```
/usr/local/mysql/bin/mysqladmin -u root password 'new-password'
/usr/local/mysql/bin/mysqladmin -u root -h wipro-33 password 'new-password'
```

Alternatively you can run:
/usr/local/mysql/bin/mysql_secure_installation

which will also give you the option of removing the test databases and anonymous user created by default. This is strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr/local/mysql ; /usr/local/mysql/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl cd /usr/local/mysql/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/local/mysql/scripts/mysqlbug script!

The latest information about MySQL is available at <http://www.mysql.com/> Support MySQL by buying support/licenses from <http://shop.mysql.com/>

9. From the `/usr/local/mysql` directory, run the following command, and check to see that you get the corresponding output.

```
# ./bin/mysqld_safe --defaults-file=/etc/my.cnf --ledir=/usr/local/mysql/bin --user=mysql &
```

```
[1] 15885
# 090323 22:36:26 mysqld_safe Logging to '/usr/local/mysql/data/wipro-33.err'.
090323 22:36:26 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
```

10. Now, leave the terminal just the way it is. To make sure the process you just enabled is running all the time, go to the console and start this process.

```
# cd /usr/local/mysql/bin
# ./mysql --user=root
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.30 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

11. Stop the MySQL daemon by running the following command in a terminal, if you want to stop the daemon.

```
# ./mysqladmin shutdown
```

When the command is run, the terminal, that was left alone, should give the following output.

```
# /usr/local/mysql/bin/mysqld_safe --defaults-file=/etc/my.cnf --ledir=/usr/local/mysql/bin
--user=mysql &
```

```
[1] 16017
# 090323 22:47:38 mysqld_safe Logging to '/usr/local/mysql/data/wipro-33.err'.
090323 22:47:38 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
090323 22:49:31 mysqld_safe mysqld from pid file /usr/local/mysql/data/wipro-33.pid ended
```

2. Create a privileged database administrator.

How to Create a Privileged Database Administrator

VDI requires a privileged database administrator to create the VDI database during VDI Core configuration (default name "vda"). The following procedure describes the creation of a privileged administrator with all privileges.

Steps

1. Use the MySQL command line tool to enter the MySQL interactive mode as root using the following command.

```
# ./mysql --user=root
```

2. Then execute the following statements (replace '<user>' and '<password>' accordingly):

```
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'localhost' IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```



For a Single Host configuration, it is simplest just to use the privilege set given above. If you have an existing remote MySQL database that you want to use with VDI, you may prefer to create a privileged administrator that only has the minimum privileges required to create the VDI database during VDI Core configuration (default name "vda"). One possible privilege set for this would be as follows:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON *.* TO '<db-user>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON *.* TO '<db-user>'@'<db-host-dns>' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```

For more information about MySQL user privileges, refer to the document [Privileges Provided by MySQL](#).

3. Install and configure the VDI Core on the single VDI host.

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip  
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip  
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:


```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Choose the **3 Single Sun VDI Host** configuration type.

- a. Accept the default local host DNS, or specify the DNS name of your MySQL server if it's on a non-local host.
- b. Accept the default port (3306) or specify the port on which your MySQL server is listening.
- c. Specify a privileged database administrator, see above.
- d. Specify the password for the database administrator that you have specified.
- e. Specify whether you want to connect to your MySQL server via SSL or not.
- f. Specify the name of the VDI database that will be created or just accept the default 'vda'.
- g. Specify the name of a user that will be associated with and used to access the VDI database. Alternatively you can simply accept the default 'vdadb'.
- h. Specify a password for the VDI database user.
- i. Specify the maximum number of users to be hosted.
- j. Specify the user ID range start.

This information is useful to avoid user ID conflicts and to comply with company regulations regarding user IDs.

At the end of the configuration script you will be given a path of the configuration log file. Or, on Solaris platforms, you can find it at `/var/adm/log/vda-config.<date and time>.log`

How to Install and Configure a Remote MySQL Database (InnoDB)

This page describes how to install MySQL 5.1 (with an InnoDB storage engine) on an x86 platform running Solaris.

Steps

1. Create the file `/etc/my.cnf`, and add the following content.

```
[mysqld]
user=mysql
datadir=/usr/local/mysql/data
basedir=/usr/local/mysql
port=3306
socket=/tmp/mysql.sock
max_allowed_packet=20M
#transaction_isolation=READ-COMMITTED
lower_case_table_names=1
max_connections=1000
skip-locking
key_buffer=16K
table_cache=4
sort_buffer_size=64K
net_buffer_length=2K
thread_stack=64K
wait_timeout=31536000

innodb_data_home_dir=/usr/local/mysql/data
innodb_data_file_path=ibdata1:10M:autoextend
innodb_log_group_home_dir=/usr/local/mysql/data
innodb_buffer_pool_size=50M
innodb_additional_mem_pool_size=10M
innodb_log_file_size=5M
innodb_log_buffer_size=10M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
```

2. Create a user "mysql" and a group "mysql" by running the following commands.

```
# groupadd mysql
# useradd -g mysql mysql
```

3. Get the MySQL tar file (mysql-5.1.30-solaris10-i386.tar) , untar it, and keep it in the / directory.
4. Create the directory /usr/local, by running the following command.

```
# mkdir /usr/local
```

5. Change to the new directory, and create a symbolic link, called "mysql", that points to the MySQL files in the / directory, by running the following commands.

```
# cd /usr/local
# ln -s /mysql-5.1.30-solaris10-i386 mysql
# ls -lrt

total 2
lrwxrwxrwx  1 root  root           35 Nov 12 17:33 mysql ->
/export/mysql-5.1.30-solaris10-i386
bash-3.00#
```

6. Make sure that the / directory contains the proper owner and group permissions by running the following commands.

```
# chgrp -R mysql /mysql-5.1.30-solaris10-i386
# chown -R mysql /mysql-5.1.30-solaris10-i386
```

7. Check the permissions for the /usr/local/mysql directory as well.

```
# cd /usr/local/mysql
# ls -lrt

-rw-r--r--  1 mysql  mysql  19071 Nov 15 13:07 COPYING
-rw-r--r--  1 mysql  mysql   5139 Nov 15 13:07 EXCEPTIONS-CLIENT
-rw-r--r--  1 mysql  mysql   8767 Nov 15 13:07 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql   1410 Nov 15 13:07 README
drwxr-xr-x  2 mysql  mysql   1536 Nov 15 13:07 bin
drwxr-xr-x  4 mysql  mysql    512 Nov 15 13:07 data
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:05 docs
drwxr-xr-x  2 mysql  mysql   1024 Nov 15 13:05 include
drwxr-xr-x  3 mysql  mysql   1024 Nov 15 13:06 lib
drwxr-xr-x  4 mysql  mysql    512 Nov 15 13:06 man
drwxr-xr-x 10 mysql  mysql    512 Nov 15 13:07 mysql-test
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:07 scripts
drwxr-xr-x 27 mysql  mysql   1024 Nov 15 13:07 share
drwxr-xr-x  5 mysql  mysql   1024 Nov 15 13:07 sql-bench
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:07 support-files
```

8. From the /usr/local/mysql directory, run the following command, and check that it provides the corresponding output.

```
# ./scripts/mysql_install_db --user=mysql
```

To start mysqld at boot time you have to copy support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

```
/usr/local/mysql/bin/mysqladmin -u root password 'new-password'
/usr/local/mysql/bin/mysqladmin -u root -h wipro-33 password 'new-password'
```

Alternatively you can run:
/usr/local/mysql/bin/mysql_secure_installation

which will also give you the option of removing the test databases and anonymous user created by default. This is strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr/local/mysql ; /usr/local/mysql/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl cd /usr/local/mysql/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/local/mysql/scripts/mysqlbug script!

The latest information about MySQL is available at <http://www.mysql.com/> Support MySQL by buying support/licenses from <http://shop.mysql.com/>

9. From the `/usr/local/mysql` directory, run the following command, and check to see that you get the corresponding output.

```
# ./bin/mysqld_safe --defaults-file=/etc/my.cnf --ledir=/usr/local/mysql/bin --user=mysql &

[1] 15885
# 090323 22:36:26 mysqld_safe Logging to '/usr/local/mysql/data/wipro-33.err'.
090323 22:36:26 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
```

10. Now, leave the terminal just the way it is. To make sure the process you just enabled is running all the time, go to the console and start this process.

```
# cd /usr/local/mysql/bin
# ./mysql --user=root

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.30 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

11. Stop the MySQL daemon by running the following command in a terminal, if you want to stop the daemon.

```
# ./mysqladmin shutdown
```

When the command is run, the terminal, that was left alone, should give the following output.

```
# /usr/local/mysql/bin/mysqld_safe --defaults-file=/etc/my.cnf --ledir=/usr/local/mysql/bin
--user=mysql &
[1] 16017
# 090323 22:47:38 mysqld_safe Logging to '/usr/local/mysql/data/wipro-33.err'.
090323 22:47:38 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
090323 22:49:31 mysqld_safe mysqld from pid file /usr/local/mysql/data/wipro-33.pid ended
```

How to Create a Privileged Database Administrator

VDI requires a privileged database administrator to create the VDI database during VDI Core configuration (default name "vda"). The following procedure describes the creation of a privileged administrator with all privileges.

Steps

1. Use the MySQL command line tool to enter the MySQL interactive mode as root using the following command.

```
# ./mysql --user=root
```

2. Then execute the following statements (replace '<user>' and '<password>' accordingly):

```
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'localhost' IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```



For a Single Host configuration, it is simplest just to use the privilege set given above. If you have an existing remote MySQL database that you want to use with VDI, you may prefer to create a privileged administrator that only has the minimum privileges required to create the VDI database during VDI Core configuration (default name "vda"). One possible privilege set for this would be as follows:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON *.* TO '<db-user>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON *.* TO '<db-user>'@'<db-host-dns>' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```

For more information about MySQL user privileges, refer to the document [Privileges Provided by MySQL](#).

How to Back Up and Restore Data (Remote MySQL Database)

The following information should be used when backing up data on a remote database in either a Single Host or High Availability (Remote MySQL Database) configuration.

Steps

1. Make a backup of the VDI database.
 - a. Log into the remote database.

```
# zlogin <MySQL server>
```

- b. Change to the `usr/local/mysql/bin/` directory and run `mysqldump`.

```
# cd usr/local/mysql/bin/  
# ./mysqldump --user=root -u root --opt checkdb | gzip > /dumptest1.sql.gz
```

2. (Optional) Perform a VDI update.



During the un-configuration, answer 'yes' to deleting the remote database. Use the same values for "privileged administrator", "VDA database", "user name for the VDA database" used while executing `vda-config` for the re-configuration.

For more information about updating VDI, refer to the [About Updating to VDI 3.1](#) page.

3. Restore the backed up VDI database on the new VDI installation.
 - a. Log into the remote database.

```
# zlogin <MySQL server>
```

- b. Change to the `usr/local/mysql/bin/` directory and run the following command.


```
# cd usr/local/mysql/bin/  
# ./mysql --user=root checkdb < /primary-dump/dumptest1.sql
```

Contents

- [Contents](#)
 - [How to Check the Status of the Common Agent Container](#)
 - [How to Restart the Common Agent Container](#)
 - [How to Check the Status of the VDI Core Service Module](#)
 - [How to Increase or Decrease the Logging Level of the VDI Core Service](#)
 - [How to Check the Status of the Database](#)
 - [How to Check the Database Service Status on an Evaluation \(Demo\) Configuration](#)
 - [How to Check the Database Service Status on a High Availability \(Bundled MySQL Database\) Configuration](#)
 - [How to Check the Status of the \(VDI Manager\) Web Service](#)
 - [How to Check that the RDP Broker Service is Running](#)
-

How to Check the VDI Core Services and Logs

It is important to know how to check the status of the different services provided by Sun VDI for troubleshooting or reconfiguration purposes. Most services run under the control of the Solaris Service Management Facility (SMF).

 Starting with Sun VDI 3, the main VDI Core service now runs as a module within the Common Agent Container (Cacao). This Java-based agent is an integrated part of Solaris 10 and is already used in a wide range of Sun products. If you encounter any issues, you should first check the status of Cacao as well as the status of the VDI Core service module.

How to Check the Status of the Common Agent Container


- As root, run the following command.

```
# cacaoadm status
```

- Alternatively, use the Solaris Service Management Facility.

```
# svcs svc:/application/management/common-agent-container-1:default
```

The corresponding log file is located at `/var/cacao/instances/default/logs/cacao.0`.

 If you would like to maintain a longer Cacao history, edit the properties `log.file.limit` and `log.file.count` in `/etc/cacao/instances/default/private/cacao.properties`. You can change both count and the limit (Max allowed 2147483647). Then restart Cacao for changes to become effective.

How to Restart the Common Agent Container

- As root, run the following command.

```
# cacaoadm stop -f
# cacaoadm start
```


How to Check the Status of the VDI Core Service Module

The VDI Core Service runs within the Common Agent Container.

- As root, run the following command.

```
# cacaoadm status com.sun.vda.service_module
```

The corresponding log file is located at `/var/cacao/instances/default/logs/cacao.0`.

 Log messages at error or warning level will also be forwarded to the `syslog` daemon.

How to Increase or Decrease the Logging Level of the VDI Core Service

If you are troubleshooting, you may want to increase the level of detail in the logs.

- To increase the logging level, run the following command as root.

```
# cacaoadm set-filter -p com.sun.vda.service=ALL
```

- To decrease the logging level, run the following command as root.

```
# cacaoadm set-filter -p com.sun.vda.service=NULL
```

- Restart Cacao after changing the logging level.

How to Check the Status of the Database

Sun VDI offers the option of using the bundled MySQL cluster database or connecting to a remote MySQL database. You can check the status of either database type with the information below.

- Check the status of the database, by running the following command as root.

```
# /opt/SUNWvda/sbin/vda-db-status status
```

How to Check the Database Service Status on an Evaluation (Demo) Configuration

The VDI Database Service is available if you are using the bundled MySQL cluster database. It is not available for remote databases. The database service runs under the Solaris Service Management Facility.

- On a Demo host, the status of the database service can be checked by running the following command as root.

```
# svcs svc:/application/database/vdadb:sql
```

The corresponding log file is located at `/var/svc/log/application-database-vdadb:sql.log`.

How to Check the Database Service Status on a High Availability (Bundled MySQL Database) Configuration

The VDI Database Service is available if you are using the bundled MySQL cluster database. It is not available for remote databases. The database service runs under the Solaris Service Management Facility.

- On any host with a MySQL management node or data node (the Primary host and first two Secondary hosts), run the following command as root to check the status of the database service.

```
# svcs svc:/application/database/vdadb:core
```

The corresponding log file is located at `/var/svc/log/application-database-vdadb:core.log`.

- On any host with a MySQL SQL node (the first two Secondary host and all additional Secondary hosts in a standard configuration), run the following command as root to check the status of the database service.

```
# svcs svc:/application/database/vdadb:sql
```

The corresponding log file is located at `/var/svc/log/application-database-vdadb:sql.log`.

How to Check the Status of the (VDI Manager) Web Service

- As root, run the following command.

```
# /opt/SUNWvda/sbin/vda-webadmin status
```

The corresponding log file is located at `/var/opt/SUNWvda/log/webadmin0.log`.

How to Check that the RDP Broker Service is Running

The RDP broker service supplied by Sun VDI also runs under the Solaris Service Management Facility.

- Ensure that the RDP broker service is running, by running the following command as root.


```
# svcs svc:/application/rdpbroker:default
```

The corresponding log file is located at `/var/svc/log/application-rdpbroker:default.log`.

How to Remove the VDI Core

Un-configure and uninstall the VDI Core:

```
# /opt/SUNWvda/sbin/vda-install -u
```

Contents

- About VDI Core Configuration
 - Single Host Configuration
 - High Availability (HA) Configurations
- How to Install and Configure the VDI Core (Bundled MySQL Database)
 - 1. Install and configure the VDI Core on the VDI Primary host.
 - How to Prepare a VDI Primary Host
 - 2. Install and configure the VDI Core on the first VDI Secondary host.
 - How to Prepare a VDI Secondary Host
 - 3. Install and configure the VDI Core on the second VDI Secondary host.
 - How to Prepare a VDI Secondary Host
- How to Install and Configure the VDI Core (Remote MySQL Database)
 - 1. Install and configure the VDI Core on the VDI Primary host.
 - How to Prepare a VDI Primary Host
 - How to Complete the Remote Database Configuration
 - 2. Install and configure the VDI Core on the VDI Secondary hosts.
 - How to Prepare a VDI Secondary Host
 - How to Complete the Remote Database Configuration
 - How to Prepare a VDI Primary Host
 - How to Prepare a VDI Secondary Host
- How to Install and Configure a Remote MySQL Database (InnoDB)
- How to Create a Privileged Database Administrator
- How to Back Up and Restore Data (Bundled MySQL Database)
- How to Back Up and Restore Data (Remote MySQL Database)
- About VDI MySQL Cluster Reconfiguration
- How to Reconfigure the MySQL Cluster
- How to Perform a Rolling Restart of the MySQL Cluster
- How to Prevent Unrestricted SQL Node Joins
- How to Check the VDI Core Services and Logs
- How to Remove the VDI Core

High Availability Configuration (All Topics)

Contents

- Single Host Configuration
 - High Availability (HA) Configurations
 - VDI Core (Bundled MySQL Database)
 - VDI Core (Remote MySQL Database)
-

About VDI Core Configuration

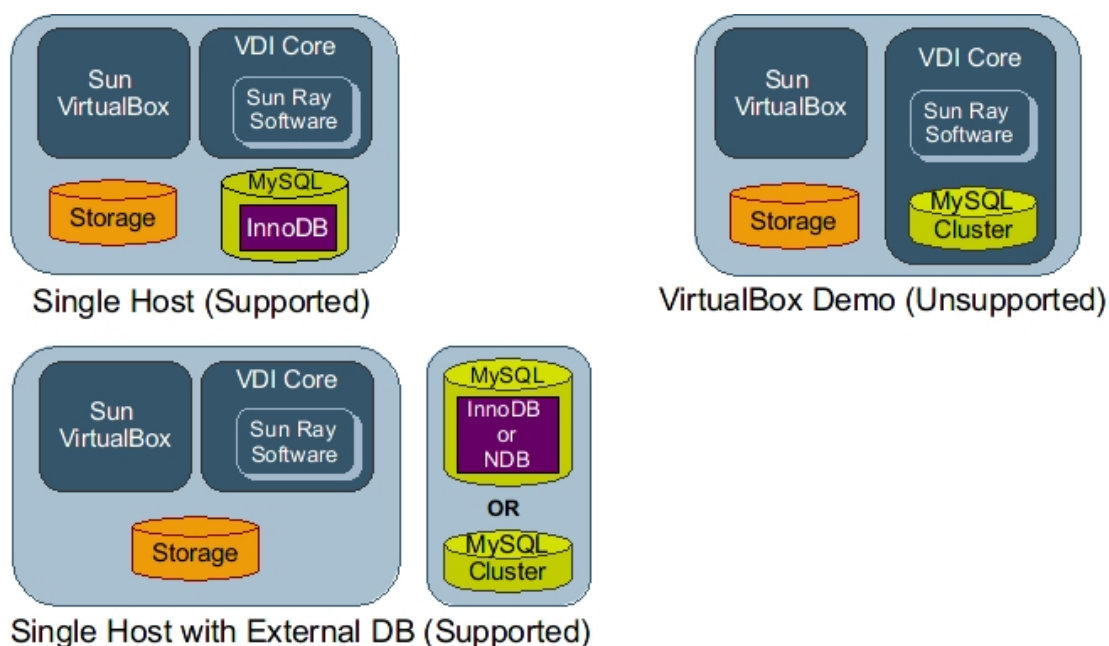
There are many possible configurations for the VDI Core. Some of the configurations are supported for production environments, while others (such as Demo/Evaluation configurations) are not. For the full details about supported and unsupported VDI Core configurations, refer to the [Supported Configurations](#) page.

The VDI Core installation and configuration information focuses on two supported VDI Core configuration types: Single Host, and High Availability. The Single Host configuration is simple to set up and requires minimal hardware while the High Availability configuration provides failover capabilities and better performance.

Single Host Configuration

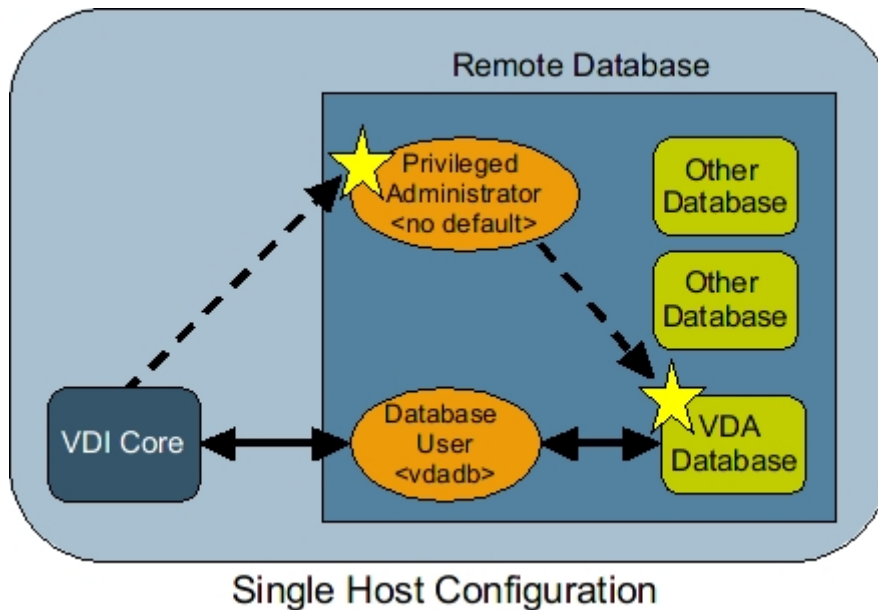
The Single Host configuration is the recommended configuration for deployments that prioritize low cost above availability, since there are no failover capabilities for a one machine setup.

The Single Host configuration is similar to a Demo (VirtualBox) configuration because all necessary components can be installed on one machine, including a Sun VirtualBox virtualization platform, VDI Core (with MySQL database and Sun Ray Software). The difference between a Single Host configuration and a Demo (VirtualBox) configuration is that the Demo uses the bundled MySQL cluster database, and the Single Host configuration requires a MySQL "remote" database. The MySQL database can be installed on the Single Host machine or on a different machine. Even if the MySQL database is installed on the same machine as all the other components, it is still considered to be "remote" since it's not bundled with the VDI Core.



If the MySQL remote database is installed on the same machine as the VDI Core, the database must have an InnoDB storage engine to be considered a supported configuration. This Single Host configuration also requires a separate support contact from MySQL to cover database support. If the MySQL remote database is installed on a different machine than the VDI Core, the requirements for the database are more flexible. It must be a MySQL version 5.0 or higher, with a transactional storage engine (usually InnoDB or NDB), or a MySQL Cluster version 6.2.15 or higher.

A remote database requires a privileged database administrator. The privileged database administrator is used by the VDI Core to create and configure the VDA database during VDI Core configuration. After the VDA database is created, it will be accessed by the VDI core through the database user (default 'vdadb').



Suggested Pages

Refer to the following pages if you plan to set up a Single Host VDI configuration.

- [How to Install and Configure the VDI Core \(Single VDI Host\)](#) - Describes all necessary steps for setting up the VDI Core in a Single Host configuration, including [How to Install and Configure a Remote MySQL Database \(InnoDB\)](#), [How to Create a Privileged Database Administrator](#), and how to install the VDI Core.
- [How to Check the VDI Core Services and Logs](#) - Describes how to check the services and statuses of the VDI Core, bundled MySQL database, and RDP broker.

High Availability (HA) Configurations

High availability offers reliability that if one server fails another one will continue to host VDI desktop sessions with only a minimal interruption to the end-user. High availability configurations are suggested for use cases where fail-safety is prioritized over low price.

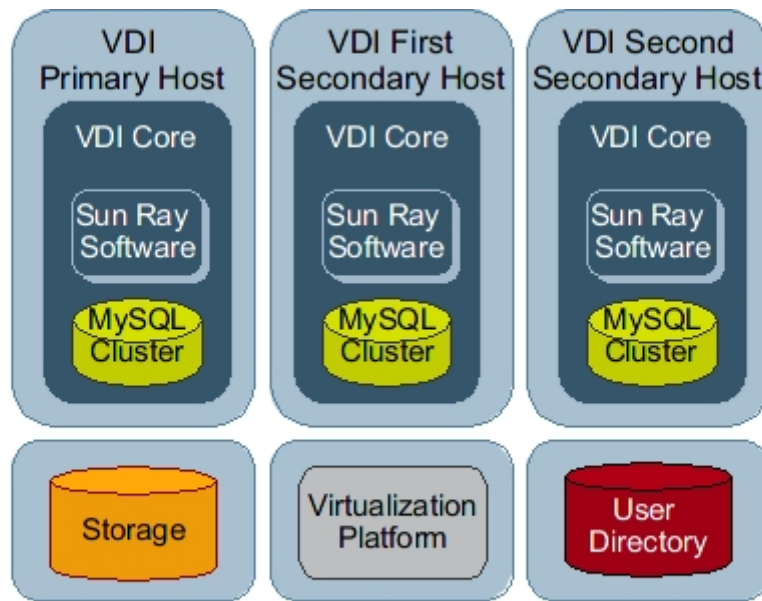
The VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one VDI host fails, all users with desktop sessions on that host are kicked back to the VDI Login dialog, and must reconnect to their session, which is restarted on one of the available hosts. The bundled MySQL cluster database requires three hosts to be highly available. There should never be an interruption of database service as long as no more than one of the three required hosts fails at a time. Therefore, the bundled VDI stack requires a minimum of three VDI hosts to be considered fail-proof. This number of hosts does not include the virtualization platform hosts, which should be considered for failover separately.

VDI also supports the option to connect to a remote MySQL database, instead of the bundled MySQL cluster database. In this case, the VDI Core only requires two hosts to be highly available. This number of hosts does not include the remote database hosts, or the virtualization platform hosts, which should be considered for failover separately.

VDI Core (Bundled MySQL Database)

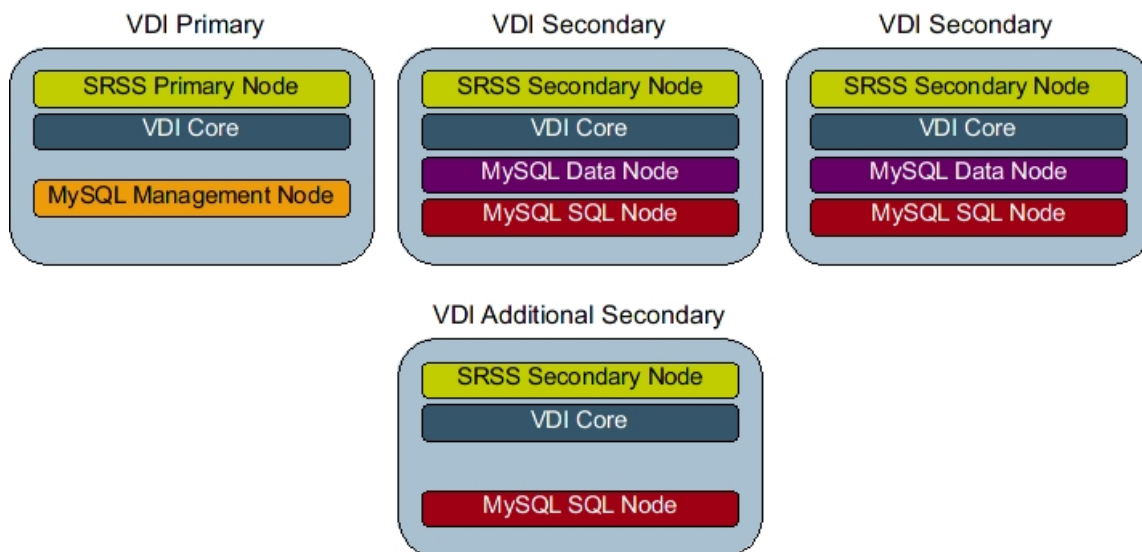
The High Availability configuration with the bundled MySQL database is automatically installed during VDI Core installation, and configured by choosing Primary or Secondary during VDI Core configuration. The VDI Core with bundled MySQL database requires one host to be configured as Primary, and the other two to be configured as Secondary.

i The documentation refer to the two Secondary hosts as "First Secondary" and "Second Secondary", while the configuration script refers to them as "Secondary A" and "Secondary B". Both naming schemes refer to identical hosts, and are meant to help you differentiate between the two during configuration and maintenance.



High Availability (Bundled MySQL Database) Configuration

By selecting VDI Primary during the configuration step, the following will be installed by default: MySQL Management Node, Sun Ray Server Software Primary Node, and a VDI Core Primary Node. By selecting VDI Secondary during the configuration step, the following will be installed by default: MySQL Data Node, MySQL SQL Node, Sun Ray Server Software Secondary Node, and a VDI Core Secondary Node. Any additional VDI Secondary hosts have a MySQL SQL Node, Sun Ray Server Software Secondary Node, and a VDI Core Secondary Node.



Default Configuration for VDI Primary and Secondary Hosts

The MySQL Cluster Nodes perform the following functions:

- Management Node - Manages the other nodes within the MySQL Cluster, performing functions such as providing configuration data, starting and stopping nodes, and running backups.
- Data Node - Stores cluster data.
- SQL Node - Serves as an interface to access the cluster data. This is a traditional MySQL server that uses the NDB Cluster storage engine.

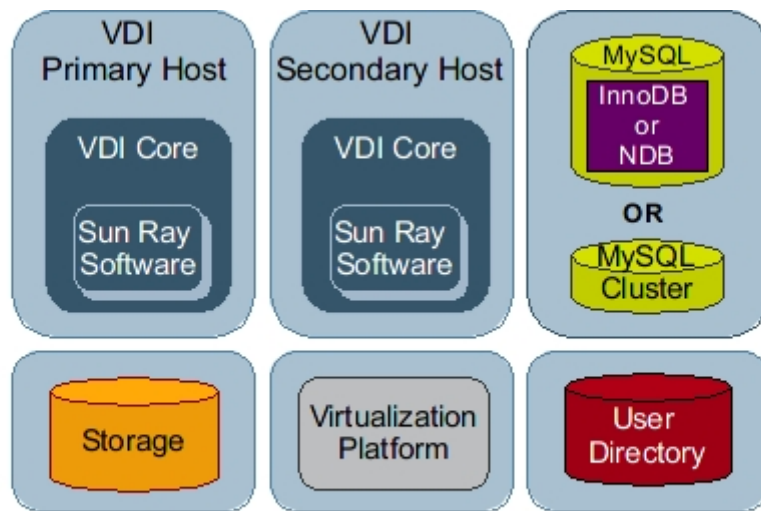
Suggested Pages

Refer to the following pages for information about the High Availability configuration using the bundled MySQL Cluster database.

- [How to Install and Configure the VDI Core \(Bundled MySQL Database\)](#) - Describes all necessary steps for setting up the VDI Core in a High Availability configuration with the bundled MySQL Cluster database, including [How to Prepare a VDI Primary Host](#) and [How to Prepare a VDI Secondary Host](#) procedures.
- [About VDI MySQL Cluster Reconfiguration](#)- Provides an overview of the MySQL Cluster reconfiguration.
- [How to Reconfigure the MySQL Cluster](#) - Explains how to convert any MySQL Node type to a different MySQL Node type.
- [How to Perform a Rolling Restart of the MySQL Cluster](#) - Describes how to start and stop each node of the MySQL Cluster so that the cluster, as a whole, remains operational.
- [How to Prevent Unrestricted SQL Node Joins](#) - Explains how to restrict any non-VDI SQL nodes from joining the bundled MySQL Cluster.
- [How to Check the VDI Core Services and Logs](#) - Describes how to check the services and statuses of the VDI Core, bundled MySQL database, and RDP broker.

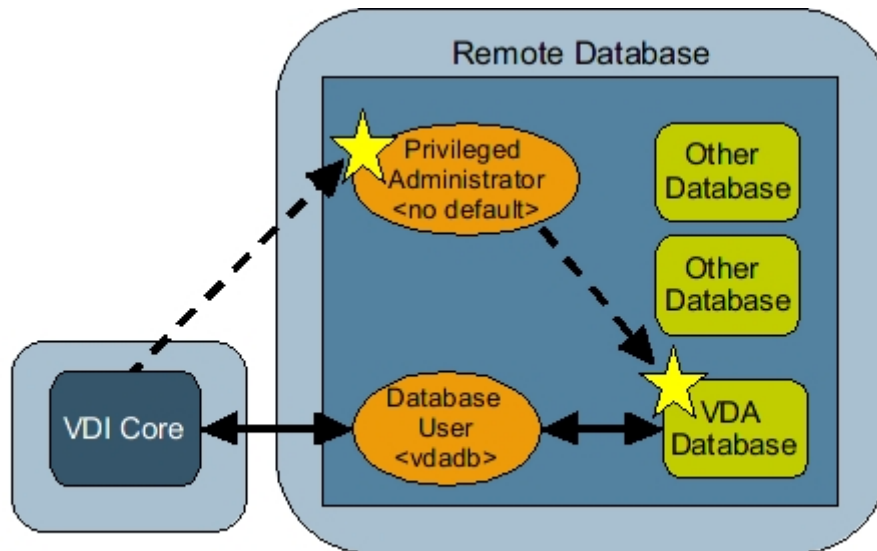
VDI Core (Remote MySQL Database)

The High Availability configuration with the remote MySQL database requires a MySQL version 5.0 or higher, with a transactional storage engine (usually InnoDB or NDB), or a MySQL Cluster version 6.2.15 or higher.



High Availability (Remote MySQL Database) Configuration

Before the VDI Core can be installed, the database must be installed and configured with a privileged database administrator. The privileged database administrator is used by the VDI Core to create and configure the VDA database during VDI Core configuration. After the VDA database is created, it will be accessed by the VDI core through the database user (default 'vdadb').



High Availability (Remote MySQL Database) Configuration

Because a remote database is used instead of the bundled MySQL Cluster, the VDI Core only requires one Primary and one Secondary to provide high availability.

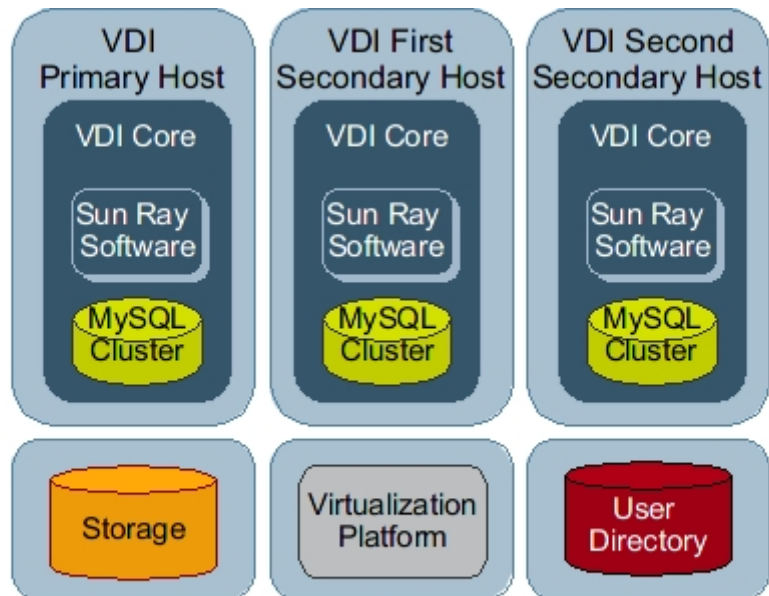
Suggested Pages

Refer to the following pages for information about the High Availability configuration using a remote MySQL database.

- [How to Install and Configure the VDI Core \(Remote MySQL Database\)](#) - Describes all necessary steps for setting up the VDI Core in a High Availability configuration with a remote MySQL database, including [How to Prepare a VDI Primary Host](#) and [How to Prepare a VDI Secondary Host](#) procedures.
- [How to Install and Configure a Remote MySQL Database \(InnoDB\)](#) - Describes how to install a MySQL database with an InnoDB storage engine. Refer to this page if you do not already have a remote database, but are interested in using one with VDI.
- [How to Create a Privileged Database Administrator](#) - Explains how to configure a privileged database administrator so that the VDA database can be created.
- [How to Check the VDI Core Services and Logs](#) - Describes how to check the services and statuses of the VDI Core, bundled MySQL database, and RDP broker.

How to Install and Configure the VDI Core (Bundled MySQL Database)

To configure VDI 3.1 for a high available production environment, you need to set up one physical Primary Host and a minimum of two physical Secondary Hosts. The installation and configuration of the first and second Secondary hosts is the same procedure.



High Availability (Bundled MySQL Database) Configuration

Before you Begin

i If your VDI installation will consist of more than 20 secondary hosts, add more [MYSQLD] sections to the end of the file `/etc/opt/SUNWvda/config.clustered.ini`.

1. Install and configure the VDI Core on the VDI Primary host.

How to Prepare a VDI Primary Host

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Choose the **1 Primary Sun VDI Host** configuration type.
6. Specify an administrator password.
This is the password that will be used to secure the MySQL database.
7. Specify a cluster signature.
This password will be used to encrypt messages that will be exchanged among the Sun Ray hosts forming a Fail-Over-Group (FOG). This password must be the same on all hosts that will be added to the multi-host group. It needs to be at least 8 characters long.
8. Choose whether to use the MySQL Cluster database bundled with VDI 3.1 or connect to a remote MySQL database.
 - If you choose to use the VDI MySQL Cluster, you must specify the DNS names of your first two secondary hosts, which will also run the MySQL Cluster data nodes.
 - If you choose to connect to a remote MySQL database, the remote database must be MySQL 5.0 or higher with InnoDB or MySQL Cluster 6.2.15 or higher.

2. Install and configure the VDI Core on the first VDI Secondary host.

How to Prepare a VDI Secondary Host



Always wait until the configuration of one VDI Secondary host has completed before configuring the next one, otherwise you could spoil the MySQL Cluster.

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```


or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Select the 2 **Secondary Sun VDI Host** configuration type and specify an administrator password.

6. Enter the cluster signature.

Must be the same as for the primary host.

7. Specify the maximum number of users to be hosted.

8. Specify the user ID range start.

This information is useful to avoid user ID conflicts and to comply with company regulations regarding user IDs.

9. Enter the DNS names of the primary host and the secondary host you are configuring.

10. Choose whether to use the MySQL database of the Sun VDI cluster or connect to a remote MySQL database.

This selection must be the same as for the primary host.

Once configuration is complete, go to `http://<server name>:1800` (or `http://localhost:1800` if remote administration has been disabled). Use root user credentials to log into the VDI Manager. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

3. Install and configure the VDI Core on the second VDI Secondary host.

How to Prepare a VDI Secondary Host



Always wait until the configuration of one VDI Secondary host has completed before configuring the next one, otherwise you could spoil the MySQL Cluster.

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

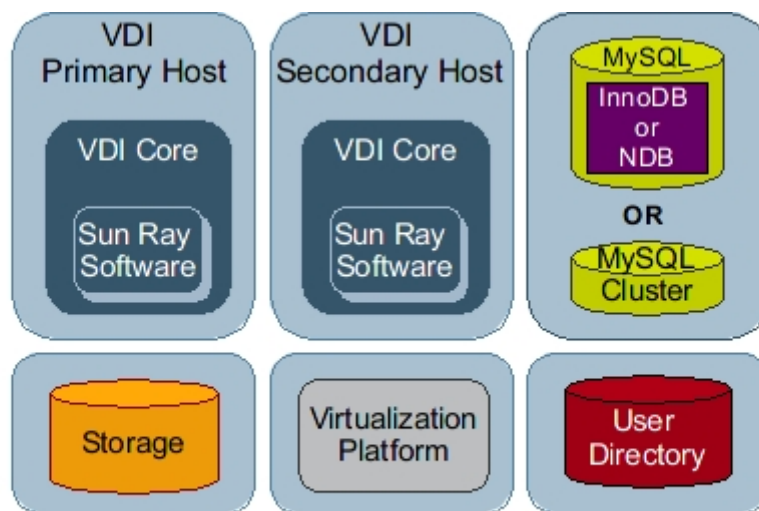
Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Select the 2 **Secondary Sun VDI Host** configuration type and specify an administrator password.
6. Enter the cluster signature.
Must be the same as for the primary host.
7. Specify the maximum number of users to be hosted.
8. Specify the user ID range start.
This information is useful to avoid user ID conflicts and to comply with company regulations regarding user IDs.
9. Enter the DNS names of the primary host and the secondary host you are configuring.
10. Choose whether to use the MySQL database of the Sun VDI cluster or connect to a remote MySQL database.
This selection must be the same as for the primary host.

Once configuration is complete, go to <http://<server name>:1800> (or <http://localhost:1800> if remote administration has been disabled). Use root user credentials to log into the VDI Manager. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

How to Install and Configure the VDI Core (Remote MySQL Database)

As an alternative to the VDI Cluster database, it is possible to use a remote MySQL database. This needs to be a MySQL version 5.0 (or higher) or a MySQL Cluster version 6.2.15 (or higher). You can use either the 32-bit or the 64-bit version. It is important that a transactional storage engine is available, which will usually be InnoDB or NDB.



High Availability (Remote MySQL Database) Configuration

Before You Begin

- Create a privileged database administrator, if you have not done so already. The VDI Core requires a privileged administrator user name and password to create, during VDI Core configuration, the database used by VDI (called by default "vda").

For information about how to create a privileged administrator, refer to the [How to Create a Privileged Database Administrator](#) page.

1. Install and configure the VDI Core on the VDI Primary host.

How to Prepare a VDI Primary Host

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip  
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip  
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation  
+ Installing Sun VDI Core...  
+ Installing MySQL Database...  
+ Installing Web Administration...  
+ Installing Apache Tomcat...  
+ Installing RDP Broker...  
+ Installing Sun Ray Client...  
+ Installing Java Runtime Environment...  
+ Installing Sun Ray Server Software...  
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Choose the **1 Primary Sun VDI Host** configuration type.
6. Specify an administrator password.
This is the password that will be used to secure the MySQL database.
7. Specify a cluster signature.
This password will be used to encrypt messages that will be exchanged among the Sun Ray hosts forming a Fail-Over-Group (FOG). This password must be the same on all hosts that will be added to the multi-host group. It needs to be at least 8 characters long.
8. Choose whether to use the MySQL Cluster database bundled with VDI 3.1 or connect to a remote MySQL database.
 - If you choose to use the VDI MySQL Cluster, you must specify the DNS names of your first two secondary hosts, which will also run the MySQL Cluster data nodes.
 - If you choose to connect to a remote MySQL database, the remote database must be MySQL 5.0 or higher with InnoDB or MySQL Cluster 6.2.15 or higher.

How to Complete the Remote Database Configuration

- Choose **2 Remote Database**.
 1. Enter the DNS name of your MySQL server.
 2. Enter the port on which your MySQL server is listening.
 3. Specify a privileged database administrator. This user needs to have the privileges to create databases and add users. If you do not have such a user yet follow the instructions below [How to Create a Privileged Database User](#) in order to add one.
 4. Specify the password for the database administrator that you have specified.
 5. Specify whether you want to connect to your MySQL server via SSL or not.
 6. Specify the name of the VDI database that will be created or just accept the default 'vda'.
 7. Specify the name of a user that will be associated with and used to access the VDI database. Alternatively you can simply accept the default 'vdadb'.
 8. Specify a password for the VDI database user.

2. Install and configure the VDI Core on the VDI Secondary hosts.

How to Prepare a VDI Secondary Host



Always wait until the configuration of one VDI Secondary host has completed before configuring the next one, otherwise you could spoil the MySQL Cluster.

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Select the **2 Secondary Sun VDI Host** configuration type and specify an administrator password.
6. Enter the cluster signature.
Must be the same as for the primary host.
7. Specify the maximum number of users to be hosted.
8. Specify the user ID range start.
This information is useful to avoid user ID conflicts and to comply with company regulations regarding user IDs.
9. Enter the DNS names of the primary host and the secondary host you are configuring.
10. Choose whether to use the MySQL database of the Sun VDI cluster or connect to a remote MySQL database.
This selection must be the same as for the primary host.

Once configuration is complete, go to <http://<server name>:1800> (or <http://localhost:1800> if remote administration has been disabled). Use root user credentials to log into the VDI Manager. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

How to Complete the Remote Database Configuration

- Choose **2 Remote Database**.
 1. Enter the DNS name of your MySQL server.
 2. Enter the port on which your MySQL server is listening.
 3. Specify whether you want to connect to your MySQL server via SSL or not.
 4. Enter the name of your VDI database which has been specified when configuring your Primary Sun VDI host.
 5. Enter the name of the user that has access to the VDI database. This is the user you have specified when configuring your Primary Sun VDI host (the default is 'vdadb').
 6. Enter the password for your VDI database user.

Next Steps

Go to [How to Check the VDI Core Services and Logs, or Managing Desktops](#).

How to Prepare a VDI Primary Host

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Choose the **1 Primary Sun VDI Host** configuration type.
6. Specify an administrator password.
This is the password that will be used to secure the MySQL database.
7. Specify a cluster signature.
This password will be used to encrypt messages that will be exchanged among the Sun Ray hosts forming a Fail-Over-Group (FOG). This password must be the same on all hosts that will be added to the multi-host group. It needs to be at least 8 characters long.
8. Choose whether to use the MySQL Cluster database bundled with VDI 3.1 or connect to a remote MySQL database.
 - If you choose to use the VDI MySQL Cluster, you must specify the DNS names of your first two secondary hosts, which will also run the MySQL Cluster data nodes.
 - If you choose to connect to a remote MySQL database, the remote database must be MySQL 5.0 or higher with InnoDB or MySQL Cluster 6.2.15 or higher.

How to Prepare a VDI Secondary Host



Always wait until the configuration of one VDI Secondary host has completed before configuring the next one, otherwise you could spoil the MySQL Cluster.

1. As root user, unzip the VDI archive if you have not already done so, and run the installation.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

2. Run the installation.

```
# ./vda-install
```

The files will be installed to `/opt/SUNWvda/`.

The installation script displays the text of the Sun Software License Agreement and prompts you to accept its terms and conditions. After the license confirmation, the installation process begins, and all VDI components are installed.

After accepting the license agreement, the installation process begins, and all VDI components are installed. These components include:

```
Sun VDI 3.1 Installation
+ Installing Sun VDI Core...
+ Installing MySQL Database...
+ Installing Web Administration...
+ Installing Apache Tomcat...
+ Installing RDP Broker...
+ Installing Sun Ray Client...
+ Installing Java Runtime Environment...
+ Installing Sun Ray Server Software...
+ Installing Sun Ray Connector for Windows Operating Systems...
```


3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

```
# /opt/SUNWvda/sbin/vda-config
```

Refer to the [VDI Defaults](#) page for more information about the configuration script.

5. Select the 2 **Secondary Sun VDI Host** configuration type and specify an administrator password.
6. Enter the cluster signature.
Must be the same as for the primary host.
7. Specify the maximum number of users to be hosted.
8. Specify the user ID range start.
This information is useful to avoid user ID conflicts and to comply with company regulations regarding user IDs.
9. Enter the DNS names of the primary host and the secondary host you are configuring.
10. Choose whether to use the MySQL database of the Sun VDI cluster or connect to a remote MySQL database.
This selection must be the same as for the primary host.

Once configuration is complete, go to <http://<server name>:1800> (or <http://localhost:1800> if remote administration has been disabled). Use root user credentials to log into the VDI Manager. You will be re-directed to https and the browser will ask you to accept the security certificate. After confirmation, you should get the login screen.

How to Install and Configure a Remote MySQL Database (InnoDB)

This page describes how to install MySQL 5.1 (with an InnoDB storage engine) on an x86 platform running Solaris.

Steps

1. Create the file `/etc/my.cnf`, and add the following content.

```

[mysqld]
user=mysql
datadir=/usr/local/mysql/data
basedir=/usr/local/mysql
port=3306
socket=/tmp/mysql.sock
max_allowed_packet=20M
#transaction_isolation=READ-COMMITTED
lower_case_table_names=1
max_connections=1000
skip-locking
key_buffer=16K
table_cache=4
sort_buffer_size=64K
net_buffer_length=2K
thread_stack=64K
wait_timeout=31536000

innodb_data_home_dir=/usr/local/mysql/data
innodb_data_file_path=ibdata1:10M:autoextend
innodb_log_group_home_dir=/usr/local/mysql/data
innodb_buffer_pool_size=50M
innodb_additional_mem_pool_size=10M
innodb_log_file_size=5M
innodb_log_buffer_size=10M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50

```

2. Create a user "mysql" and a group "mysql" by running the following commands.

```

# groupadd mysql
# useradd -g mysql mysql

```

3. Get the MySQL tar file (mysql-5.1.30-solaris10-i386.tar) , untar it, and keep it in the / directory.
4. Create the directory /usr/local, by running the following command.

```

# mkdir /usr/local

```

5. Change to the new directory, and create a symbolic link, called "mysql", that points to the MySQL files in the / directory, by running the following commands.

```

# cd /usr/local
# ln -s /mysql-5.1.30-solaris10-i386 mysql
# ls -lrt

total 2
lrwxrwxrwx  1 root  root           35 Nov 12 17:33 mysql ->
/export/mysql-5.1.30-solaris10-i386
bash-3.00#

```

6. Make sure that the / directory contains the proper owner and group permissions by running the following commands.

```
# chgrp -R mysql /mysql-5.1.30-solaris10-i386
# chown -R mysql /mysql-5.1.30-solaris10-i386
```

7. Check the permissions for the /usr/local/mysql directory as well.

```
# cd /usr/local/mysql
# ls -lrt

-rw-r--r--  1 mysql  mysql  19071 Nov 15 13:07 COPYING
-rw-r--r--  1 mysql  mysql   5139 Nov 15 13:07 EXCEPTIONS-CLIENT
-rw-r--r--  1 mysql  mysql   8767 Nov 15 13:07 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql   1410 Nov 15 13:07 README
drwxr-xr-x  2 mysql  mysql   1536 Nov 15 13:07 bin
drwxr-xr-x  4 mysql  mysql    512 Nov 15 13:07 data
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:05 docs
drwxr-xr-x  2 mysql  mysql   1024 Nov 15 13:05 include
drwxr-xr-x  3 mysql  mysql   1024 Nov 15 13:06 lib
drwxr-xr-x  4 mysql  mysql    512 Nov 15 13:06 man
drwxr-xr-x 10 mysql  mysql    512 Nov 15 13:07 mysql-test
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:07 scripts
drwxr-xr-x 27 mysql  mysql   1024 Nov 15 13:07 share
drwxr-xr-x  5 mysql  mysql   1024 Nov 15 13:07 sql-bench
drwxr-xr-x  2 mysql  mysql    512 Nov 15 13:07 support-files
```

8. From the /usr/local/mysql directory, run the following command, and check that it provides the corresponding output.

```
# ./scripts/mysql_install_db --user=mysql
```

To start mysqld at boot time you have to copy support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

```
/usr/local/mysql/bin/mysqladmin -u root password 'new-password'
/usr/local/mysql/bin/mysqladmin -u root -h wipro-33 password 'new-password'
```

Alternatively you can run:
/usr/local/mysql/bin/mysql_secure_installation

which will also give you the option of removing the test databases and anonymous user created by default. This is strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr/local/mysql ; /usr/local/mysql/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl cd /usr/local/mysql/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/local/mysql/scripts/mysqlbug script!

The latest information about MySQL is available at <http://www.mysql.com/> Support MySQL by buying support/licenses from <http://shop.mysql.com/>

9. From the `/usr/local/mysql` directory, run the following command, and check to see that you get the corresponding output.

```
# ./bin/mysqld_safe --defaults-file=/etc/my.cnf --ledir=/usr/local/mysql/bin --user=mysql &

[1] 15885
# 090323 22:36:26 mysqld_safe Logging to '/usr/local/mysql/data/wipro-33.err'.
090323 22:36:26 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
```

10. Now, leave the terminal just the way it is. To make sure the process you just enabled is running all the time, go to the console and start this process.

```
# cd /usr/local/mysql/bin
# ./mysql --user=root

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.30 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

11. Stop the MySQL daemon by running the following command in a terminal, if you want to stop the daemon.

```
# ./mysqladmin shutdown
```

When the command is run, the terminal, that was left alone, should give the following output.

```
# /usr/local/mysql/bin/mysqld_safe --defaults-file=/etc/my.cnf --ledir=/usr/local/mysql/bin
--user=mysql &
[1] 16017
# 090323 22:47:38 mysqld_safe Logging to '/usr/local/mysql/data/wipro-33.err'.
090323 22:47:38 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
090323 22:49:31 mysqld_safe mysqld from pid file /usr/local/mysql/data/wipro-33.pid ended
```

How to Create a Privileged Database Administrator

VDI requires a privileged database administrator to create the VDI database during VDI Core configuration (default name "vda"). The following procedure describes the creation of a privileged administrator with all privileges.

Steps

1. Use the MySQL command line tool to enter the MySQL interactive mode as root using the following command.

```
# ./mysql --user=root
```

2. Then execute the following statements (replace '<user>' and '<password>' accordingly):

```
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'localhost' IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```



For a Single Host configuration, it is simplest just to use the privilege set given above. If you have an existing remote MySQL database that you want to use with VDI, you may prefer to create a privileged administrator that only has the minimum privileges required to create the VDI database during VDI Core configuration (default name "vda"). One possible privilege set for this would be as follows:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON *.* TO '<db-user>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;  
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER ON *.* TO '<db-user>'@'<db-host-dns>' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```

For more information about MySQL user privileges, refer to the document [Privileges Provided by MySQL](#).

How to Back Up and Restore Data (Bundled MySQL Database)

In a typical VDI instance, using the bundled MySQL database, a back up is only necessary when updating to a new release of VDI. The integration of the bundled MySQL database already provides an assurance of fail-safety, that reduces the need for backups as a means for disaster recovery. Find further details about this topic in the official [MySQL documentation](#).



Please keep in mind that flexible desktop assignments existing at the time when the backup will be created might not be valid anymore when the VDI configuration has to be restored from this backup. This may lead to some unexpected side effects. For this reason you should consider replication as an alternative to regularly taken backups. Master-Slave replication is supported with the MySQL Cluster version coming along with VDI. Find details on the topic here: [MySQL Cluster Replication](#)

Steps

The following procedure assumes that you have a functioning (installed and configured) VDI instance, using the bundled MySQL database.

1. Make a backup of the VDI database.

a. Open the `ndb_mgm` console by running the following command on the VDI Primary host.

```
# /opt/SUNWvda/mysql/bin/ndb_mgm
```

b. Start the back up by running the following command at the `ndb_mgm` prompt.

```
ndb_mgm> START BACKUP
```

c. Verify that backups have been created by checking the following directory on the two VDI Secondary hosts:

```
/var/opt/SUNWvda/mysql-cluster/BACKUP.
```

2. (Optional) Perform a VDI update.

For more information about updating VDI, refer to the [About Updating to VDI 3.1](#) page.

3. Restore the backed up VDI database on the new VDI installation.

- a. On the first VDI Secondary host, run the following command.

```
/opt/SUNWvda/mysql/bin/ndb_restore -b <backup #> -n <nodeid #> -r  
--backup_path=<path>
```

- b. On the second VDI Secondary host, run the following command.

```
/opt/SUNWvda/mysql/bin/ndb_restore -b <backup #> -n <nodeid #> -r  
--backup_path=<path>
```

- c. Log into the VDI Manager, and check that all the data is restored.

Go to <http://<server name>:1800> (or <http://localhost:1800> if remote administration has been disabled), and use root user credentials to log in.

How to Back Up and Restore Data (Remote MySQL Database)

The following information should be used when backing up data on a remote database in either a Single Host or High Availability (Remote MySQL Database) configuration.

Steps

1. Make a backup of the VDI database.

- a. Log into the remote database.

```
# zlogin <MySQL server>
```

- b. Change to the `usr/local/mysql/bin/` directory and run `mysqldump`.

```
# cd usr/local/mysql/bin/  
# ./mysqldump --user=root -u root --opt checkdb | gzip > /dumptest1.sql.gz
```

2. (Optional) Perform a VDI update.



During the un-configuration, answer 'yes' to deleting the remote database. Use the same values for "privileged administrator", "VDA database", "user name for the VDA database" used while executing `vda-config` for the re-configuration.

For more information about updating VDI, refer to the [About Updating to VDI 3.1](#) page.

3. Restore the backed up VDI database on the new VDI installation.

- a. Log into the remote database.

```
# zlogin <MySQL server>
```

b. Change to the `usr/local/mysql/bin/` directory and run the following command.

```
# cd usr/local/mysql/bin/  
# ./mysql --user=root checkdb < /primary-dump/dumptest1.sql
```

About VDI MySQL Cluster Reconfiguration



The following tips and procedures require a profound knowledge of the VDI configuration in general and the MySQL Cluster database configuration in particular.

Be sure have have this level of knowledge before continuing. Familiarize yourself with MySQL Cluster. Detailed information can be found in the official MySQL documentation [MySQL Cluster Overview](#). Failures made when executing the following procedures may seriously spoil your VDI installation or make it completely unusable.

When the MySQL Cluster database option has been chosen during the VDI configuration a MySQL Cluster database will be installed under the hood. It has been mentioned already that for this option at least three physical hosts are required each of which will assume a different role from the perspective of the MySQL Cluster database. A detailed overview about MySQL Cluster node types and core concepts can be found here: [MySQL Cluster Core Concepts](#). In particular there will exist:

1. A Primary hosts which runs the MySQL Cluster Management node
2. The 1st Secondary which runs the first MySQL Cluster data node as well as a SQL node
3. The 2nd Secondary which runs the second MySQL Cluster data node as well as a SQL node
4. Further Secondaries which will run a SQL node each

This is a rather static MySQL Cluster configuration consisting always only of one Management node, two Data nodes and up to 99 SQL nodes. Besides that several compromises have been made favoring ease of installation and configuration over absolute security. Find detailed information about security related aspects in conjunction with MySQL Cluster here: [MySQL Cluster Security Issues](#). Several reasons may exist forcing you to adapt the MySQL Cluster database setup for instance:

- Security, you want to make the MySQL Cluster installation more secure
- Scalability/Fail-safety, you want to increase the level of fail-safety of the MySQL Cluster database by adding more Data nodes
- Error recovery, one of your hosts suffers from an outage and a new or another hosts need to take over his role e.g. one of your Data node hosts is broken which is a dangerous situation as in that moment you lack fail-safety (one Data node alone doesn't provide fail-safety). In a situation like this you are maybe forced to promote one of your other secondary hosts to be a Data node.

What follows are some procedures that can be followed in order to do some reconfigurations to the MySQL Cluster configuration none of which is support out-of-the-box by the 'vda-config' script. Most of these reconfiguration procedures imply a complete outage of the system. Be aware also that you have to comply with the MySQL Cluster rules regarding reconfiguration as published here [Performing Rolling Restart of MySQL Cluster](#).

- [How to Perform a Rolling Restart of the MySQL Cluster](#)
- [How to Reconfigure the MySQL Cluster](#)
- [How to Prevent Unrestricted SQL Node Joins](#)

How to Reconfigure the MySQL Cluster




The following tips and procedures require a profound knowledge of the VDI configuration in general and the MySQL Cluster database configuration in particular. Be sure have have this level of knowledge before continuing. See the [About VDI MySQL Cluster Reconfiguration](#) page for more information.

The following table show the different host types from the perspective of the MySQL Cluster database and the possible transformation from one type to another. The following terms will be used:

- Non-VDI host - a host which is not yet a VDI host e.g. a completely new host
- Primary-Management host - the host which is running the MySQL Cluster Management node
- Secondary-Data host - a secondary host which is running one of the MySQL Cluster data nodes as well as a SQL node
- Secondary-SQL host - a secondary host which is running a SQL node only

From/To	Non-VDI host	Primary-Management host	Secondary-Data host	Secondary-SQL host
Non-VDI host	-	From Non-VDI to Primary-Management host	From Non-VDI to Secondary-Data host	From Non-VDI to Secondary-SQL host
Primary-Management host	From Primary-Management to Non-VDI host	-	From Primary-Management to Secondary-Data host	From Primary-Management to Secondary-SQL host
Secondary-Data host	From Secondary-Data to Non-VDI host	From Secondary-Data to Primary-Management host	-	From Secondary-Data to Secondary-SQL host
Secondary-SQL host	From Secondary-SQL to Non-VDI host	From Secondary-SQL to Primary-Management host	From Secondary-SQL to Secondary-Data host	-

From Non-VDI to Primary-Management host

 This reconfiguration implies a complete outage of the system.

1. Prepare your new Primary-Management host by installing and configuring it as Primary VDI host. Refer to the [How to Prepare a VDI Primary Host](#) page for more information.
2. Prepare your two Secondary-Data hosts.
 - a. Stop the `vdadb:core` service by running the following command.

```
svcadm disable svc:/application/database/vdadb:core
```

- b. Stop the `vdadb:sql` service by running the following command.

```
svcadm disable svc:/application/database/vdadb:sql
```

- c. If your original Primary-Management host is still running, un-configure it now by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

- d. On both Secondary-Data hosts edit `/etc/opt/SUNWvda/my.cnf` exchange the ip address of the original Primary-Management host with that of your new one.
- e. Edit `/etc/opt/SUNWvda/vdadbconnection.properties` and exchange the ip address of the original Primary-Management host with that of your new one.
- f. On both Secondary-Data hosts change the `svc:/application/database/vdadb:core` SMF configuration by running the following command.

```
svccfg -s svc:/application/database/vdadb:core setprop config/ndbd_connectstring = astring: <management-host>
```

- g. Refresh the `svc:/application/database/vdadb:core` SMF service description: `svcadm refresh svc:/application/database/vdadb:core`
- h. Check that your `svc:/application/database/vdadb:sql` SMF service is in 'disabled' state. Start it again by running the following command. This can take a couple of minutes.

```
svcadm enable svc:/application/database/vdadb:core
```

- i. Start the `svc:/application/database/vdadb:sql` SMF service again by running the following command.


```
svcadm enable svc:/application/database/vdadb:sql
```

From Non-VDI to Secondary-Data host



This reconfiguration implies a complete outage of the system.

1. Stop the Data node as well as the SQL node on your two Secondary-Data hosts (or the remaining one in case one data node is broken etc.). On all Secondary-SQL hosts stop the SQL node.
 - a. On your Secondary Data hosts, run the following command.

```
svcadm disable svc:/application/database/vdadb:core
```

- b. Wait until the service has been stopped (this can take a couple of minutes).
- c. Verify that it has been stopped by running the following command.

```
svcs svc:/application/database/vdadb:core
```

This can take a couple of minutes. When the service has been stopped correctly you will see something similar to this:

```
STATE          STIME      FMRI
disabled      Dez_09    svc:/application/database/vdadb:core
```

- d. On your Secondary Data hosts as well as on all Secondary SQL hosts stop the SQL node by running the following command.

```
svcadm disable svc:/application/database/vdadb:sql
```

Wait until the service has been stopped (this can take a couple of minutes).

- e. Verify that the SQL node has been stopped by running the following command.

```
svcs svc:/application/database/vdadb:sql
```

When the service has been stopped correctly you will see something similar to this:

```
STATE          STIME      FMRI
disabled      Dez_09    svc:/application/database/vdadb:sql
```

2. On your Primary host stop the `svc:/application/database/vdadb:core` service by running the following command.

```
svcadm disable svc:/application/database/vdadb:core
```

Wait until the service has been stopped.

- a. Verify that the service has been stopped by running the following command.

```
svcs svc:/application/database/vdadb:core
```

When the service has been stopped correctly you will see something similar to this:

```
STATE      STIME      FMRI
disabled   Dez_09     svc:/application/database/vdadb:core
```

3. On your Primary host:

- a. Change the file `/etc/opt/SUNWvda/config.ini` and exchange the ip/hostname of the data node that you want to retire with that of the new one.

Be sure not to mix hostnames and ip addresses in this file! Follow the existing convention in this file.

- b. Start the `svc:/application/database/vdadb:core` service again by running the following command.

```
svcadm enable svc:/application/database/vdadb:core
```

- c. Wait a couple of minutes, and check that the service has been started correctly again by running the following command.

```
svcs svc:/application/database/vdadb:core
```

When the service has been started correctly you will see something similar to this:

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:core
```

4. On your remaining "old" Secondary-Data host:

- a. Start the data node again by running the following command.

```
svcadm enable svc:/application/database/vdadb:core
```

Wait until the service has been started (this can take a couple of minutes).

- b. Verify that it has been started by running the following command.

```
svcs svc:/application/database/vdadb:core
```

This can take a couple of minutes. When the service has been started correctly you will see something similar to this:

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:core
```

- c. Start the SQL node again by running the following command.

```
svcadm enable svc:/application/database/vdadb:sql
```

Wait until the service has been started (this can take a couple of minutes).

- d. Verify that it has been started by running the following command.

```
svcs svc:/application/database/vdadb:sql
```

This can take a couple of minutes. When the service has been started correctly you will see something similar to this:

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:sql
```

5. Configure your new Secondary Data host by running the following command.

```
/opt/SUNWvda/sbin/vda-config
```

6. On your Secondary SQL hosts start the SQL node again by running the following command.

```
svcadm enable svc:/application/database/vdadb:sql
```

Wait until the service has been started. This can take a couple of minutes.

a. Verify that it has been started by running the following command.

```
svcs svc:/application/database/vdadb:sql
```

This can take a couple of minutes. When the service has been started correctly you will see something similar to this:

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:sql
```

From Non-VDI to Secondary-SQL host

As long as there are still free [MYSQLD] slots on the Primary available, you can add a new SQL node by using the instructions on the [How to Prepare a VDI Secondary Host](#) page.

From Primary-Management to Non-VDI host

1. Un-configure your Primary-Management host.
2. Configure a new Primary-Management host using the [instructions](#) above.

From Primary-Management to Secondary-Data host

1. Convert the Primary-Management host to a Non-VDI host using the [instructions](#) above.
2. Now configure the host to run as a Secondary-Data host following these [instructions](#).

From Primary-Management to Secondary-SQL host

1. Convert the management node to a nothing node using the [instructions](#) above.
2. Now configure the host to run an SQL node using these [instructions](#).

From Secondary-Data to Non-VDI host

1. Un-configure the Secondary-Data host by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

2. Set up a new Secondary-Data host using the [instructions](#) above.

From Secondary-Data to Primary-Management host

1. Un-configure the Secondary Data host by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

2. Set up a new Secondary Data host follow the [instructions](#) above.
3. Un-configure your existing Primary-Management host following these [instructions](#)
4. Reconfigure your former Secondary-Data host as a Primary-Management hosts following these [instructions](#).

From Secondary-Data to Secondary-SQL host

1. Un-configure the data node by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

2. Set up a new data node using these [instructions](#).
3. Convert the new data node into an SQL node using the [instructions](#) above.

From Secondary-SQL to Non-VDI host

1. Un-configure the SQL node by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

From Secondary-SQL to Primary-Management host

1. Unconfigure the SQL node by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

2. Replace your existing management node by the un-configured SQL node using these [instructions](#).

From Secondary-SQL to Secondary-Data host

1. Unconfigure your data node by running the following command.

```
/opt/SUNWvda/sbin/vda-config -u
```

2. To set up a new data node, use the [instructions](#) above.

How to Perform a Rolling Restart of the MySQL Cluster

This section is only for the embedded/bundled MySQL.

A "rolling restart" of the MySQL Cluster refers to stopping and starting (or restarting) each node in turn, so that the cluster, as a whole, remains operational. This is often done as part of a rolling upgrade or rolling downgrade, where high availability of the cluster is required.

There are three main reasons why a rolling restart of the VDI MySQL cluster:

- Cluster configuration change - To add an SQL node to the cluster, or set a configuration parameter to a new value without experiencing a complete outage. The page [How to Reconfigure the MySQL Cluster](#) explains how to add a host to the MySQL Cluster with complete outage.
- Change on VDI Core host - To make changes in the hardware or operating system.
- Freeing of resources - To allow memory, allocated to a table by successive INSERT and DELETE operations, to be freed for re-use by other MySQL Cluster tables.

The general process for performing a rolling restart can be generalized as follows:

1. Stop all cluster management nodes (`ndb_mgmd` process), reconfigure them, then restart them.
2. Stop, reconfigure, then restart each cluster data node (`ndbd` process) in turn.
3. Stop, reconfigure, then restart each cluster SQL node (`mysqld` process) in turn.

Be sure to comply with the MySQL Cluster rules regarding reconfiguration as published here: [Performing Rolling Restart of MySQL Cluster](#).



The following tips and procedures require a profound knowledge of the VDI configuration in general and the MySQL Cluster database configuration in particular.

Be sure have have this level of knowledge before continuing. Familiarize yourself with MySQL Cluster. Detailed information can be found in the official MySQL documentation [MySQL Cluster Overview](#). Failures made when executing the following procedures may seriously spoil your VDI installation or make it completely unusable.

Steps

1. Stop the `vdadb:core` service on the Primary host.

Execute `svcadm disable vda:/application/database/vdadb:core`. Verify that it has been stopped by executing `svcs svc:/application/database/vdadb:core` (this can take a couple of minutes). You should see an output similar to this one:

```
STATE      STIME      FMRI
disabled   Dez_09     svc:/application/database/vdadb:core
```

2. Start the `vdadb:core` service again on the Primary host.

Execute `svcadm enable svc:/application/database/vdadb:core`. This makes the new MySQL Cluster configuration effective. Verify that the service has been started by executing `svcs svc:/application/database/vdadb:core` (again, this may take a couple of minutes). You should see an output similar to this one:

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:core
```

3. Stop the `vdadb:core` service on the first Secondary data node.

Execute `svcadm disable svc:/application/database/vdadb:core` on your first Secondary host. Verify that it has been stopped by executing `svcs svc:/application/database/vdadb:core` (this can take a couple of minutes).

```
STATE      STIME      FMRI
disabled   Dez_09     svc:/application/database/vdadb:core
```

4. Once the data node has been stopped, start it again.

Execute `svcadm enable svc:/application/database/vdadb:core` and wait until it has been started (again, this might take a couple of minutes). Verify that the service has been started by executing `svcs svc:/application/database/vdadb:core`. You should see output similar to this one:

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:core
```

5. Repeat the last two steps on your second Secondary host.

6. Stop the SQL node on every secondary host.

Execute `svcadm disable svc:/application/database/vdadb:sql`. Verify that it has been stopped by executing `svcs svc:/application/database/vdadb:sql` (this can take a couple of minutes).

```
STATE      STIME      FMRI
disabled   Dez_09     svc:/application/database/vdadb:sql
```

7. Start the SQL node on every secondary host.

Execute `svcadm enable svc:/application/database/vdadb:sql`. Verify that it has been started by executing `svcs svc:/application/database/vdadb:sql` (this can take a couple of minutes).

```
STATE      STIME      FMRI
online     Dez_09     svc:/application/database/vdadb:sql
```

How to Prevent Unrestricted SQL Node Joins



The following tips and procedures require a profound knowledge of the VDI configuration in general and the MySQL Cluster database configuration in particular. Be sure you have this level of knowledge before continuing. See the [About VDI MySQL Cluster Reconfiguration](#) page for more information.

The default MySQL Cluster configuration allows up to 20 SQL nodes joining the MySQL Cluster. In security sensitive environment however one might want to prevent unrestricted SQL node joins. This is just one step toward a more secure MySQL Cluster configuration other steps can be taken by following the suggestions from the official MySQL site [MySQL Cluster Security Issues](#). Restricting SQL nodes from joining the MySQL Cluster will be accomplished by changing the file `/etc/opt/SUNWvda/config.ini` on the Primary VDI host. At the end of this file there are a couple of `[MYSQLD]` sections. For every SQL node that wants to join the MySQL Cluster there has to exist a free `[MYSQLD]` slot. For VDI every Secondary host (and the Primary if configured to serve sessions as well) runs its own SQL node hence one `[MYSQLD]` slot has to exist for every Secondary and the Primary in case. Restricted access of SQL nodes will be accomplished by exactly specifying the hosts that are allowed to join. Augment the `[MYSQLD]` slots in your `/etc/opt/SUNWvda/config.ini` file like this:

```
...
[MYSQLD]
HostName=<ip_or_dns_of_the_host_running_an_sql_node>
...
```

Please follow the existing convention in this file regarding the use of IPs vs. host names. Mixing of IPs and host names in `/etc/opt/SUNWvda/config.ini` is not allowed. Remove all unnecessary `[MYSQLD]` slots. Example: Imagine you have 3 secondary hosts with the following host names: `my-1st-secondary`, `my-2nd-secondary`, `my-3rd-secondary`. Initially your `/etc/opt/SUNWvda/config.ini` will look like this:

```
...
[MYSQLD]
[MYSQLD]
[MYSQLD]
[MYSQLD]
[MYSQLD]
[MYSQLD]
[MYSQLD]
...
[MYSQLD]
```

Change it to look like this:


```
...
[MYSQLD]
HostName=my-1st-secondary
[MYSQLD]
HostName=my-2nd-secondary
[MYSQLD]
HostName=my-3rd-secondary
```

Contents

- Contents
 - How to Check the Status of the Common Agent Container
 - How to Restart the Common Agent Container
 - How to Check the Status of the VDI Core Service Module
 - How to Increase or Decrease the Logging Level of the VDI Core Service
 - How to Check the Status of the Database
 - How to Check the Database Service Status on an Evaluation (Demo) Configuration
 - How to Check the Database Service Status on a High Availability (Bundled MySQL Database) Configuration
 - How to Check the Status of the (VDI Manager) Web Service
 - How to Check that the RDP Broker Service is Running
-

How to Check the VDI Core Services and Logs

It is important to know how to check the status of the different services provided by Sun VDI for troubleshooting or reconfiguration purposes. Most services run under the control of the Solaris Service Management Facility (SMF).

 Starting with Sun VDI 3, the main VDI Core service now runs as a module within the Common Agent Container (Cacao). This Java-based agent is an integrated part of Solaris 10 and is already used in a wide range of Sun products. If you encounter any issues, you should first check the status of Cacao as well as the status of the VDI Core service module.

How to Check the Status of the Common Agent Container


- As root, run the following command.

```
# cacaoadm status
```

- Alternatively, use the Solaris Service Management Facility.

```
# svcs svc:/application/management/common-agent-container-1:default
```

The corresponding log file is located at `/var/cacao/instances/default/logs/cacao.0`.

-  If you would like to maintain a longer Cacao history, edit the properties `log.file.limit` and `log.file.count` in `/etc/cacao/instances/default/private/cacao.properties`. You can change both count and the limit (Max allowed 2147483647). Then restart Cacao for changes to become effective.

How to Restart the Common Agent Container

- As root, run the following command.

```
# cacaoadm stop -f
# cacaoadm start
```


How to Check the Status of the VDI Core Service Module

The VDI Core Service runs within the Common Agent Container.

- As root, run the following command.

```
# cacaoadm status com.sun.vda.service_module
```

The corresponding log file is located at `/var/cacao/instances/default/logs/cacao.0`.

-  Log messages at error or warning level will also be forwarded to the `syslog` daemon.

How to Increase or Decrease the Logging Level of the VDI Core Service

If you are troubleshooting, you may want to increase the level of detail in the logs.

- To increase the logging level, run the following command as root.

```
# cacaoadm set-filter -p com.sun.vda.service=ALL
```

- To decrease the logging level, run the following command as root.

```
# cacaoadm set-filter -p com.sun.vda.service=NULL
```

- Restart Cacao after changing the logging level.

How to Check the Status of the Database

Sun VDI offers the option of using the bundled MySQL cluster database or connecting to a remote MySQL database. You can check the status of either database type with the information below.

- Check the status of the database, by running the following command as root.

```
# /opt/SUNWvda/sbin/vda-db-status status
```

How to Check the Database Service Status on an Evaluation (Demo) Configuration

The VDI Database Service is available if you are using the bundled MySQL cluster database. It is not available for remote databases. The database service runs under the Solaris Service Management Facility.

- On a Demo host, the status of the database service can be checked by running the following command as root.

```
# svcs svc:/application/database/vdadb:sql
```

The corresponding log file is located at `/var/svc/log/application-database-vdadb:sql.log`.

How to Check the Database Service Status on a High Availability (Bundled MySQL Database) Configuration

The VDI Database Service is available if you are using the bundled MySQL cluster database. It is not available for remote databases. The database service runs under the Solaris Service Management Facility.

- On any host with a MySQL management node or data node (the Primary host and first two Secondary hosts), run the following command as root to check the status of the database service.

```
# svcs svc:/application/database/vdadb:core
```

The corresponding log file is located at `/var/svc/log/application-database-vdadb:core.log`.

- On any host with a MySQL SQL node (the first two Secondary host and all additional Secondary hosts in a standard configuration), run the following command as root to check the status of the database service.

```
# svcs svc:/application/database/vdadb:sql
```

The corresponding log file is located at `/var/svc/log/application-database-vdadb:sql.log`.

How to Check the Status of the (VDI Manager) Web Service

- As root, run the following command.

```
# /opt/SUNWvda/sbin/vda-webadmin status
```

The corresponding log file is located at `/var/opt/SUNWvda/log/webadmin0.log`.

How to Check that the RDP Broker Service is Running

The RDP broker service supplied by Sun VDI also runs under the Solaris Service Management Facility.

- Ensure that the RDP broker service is running, by running the following command as root.

```
# svcs svc:/application/rdpbroker:default
```

The corresponding log file is located at `/var/svc/log/application-rdpbroker:default.log`.

How to Remove the VDI Core

Un-configure and uninstall the VDI Core:

```
# /opt/SUNWvda/sbin/vda-install -u
```

Contents

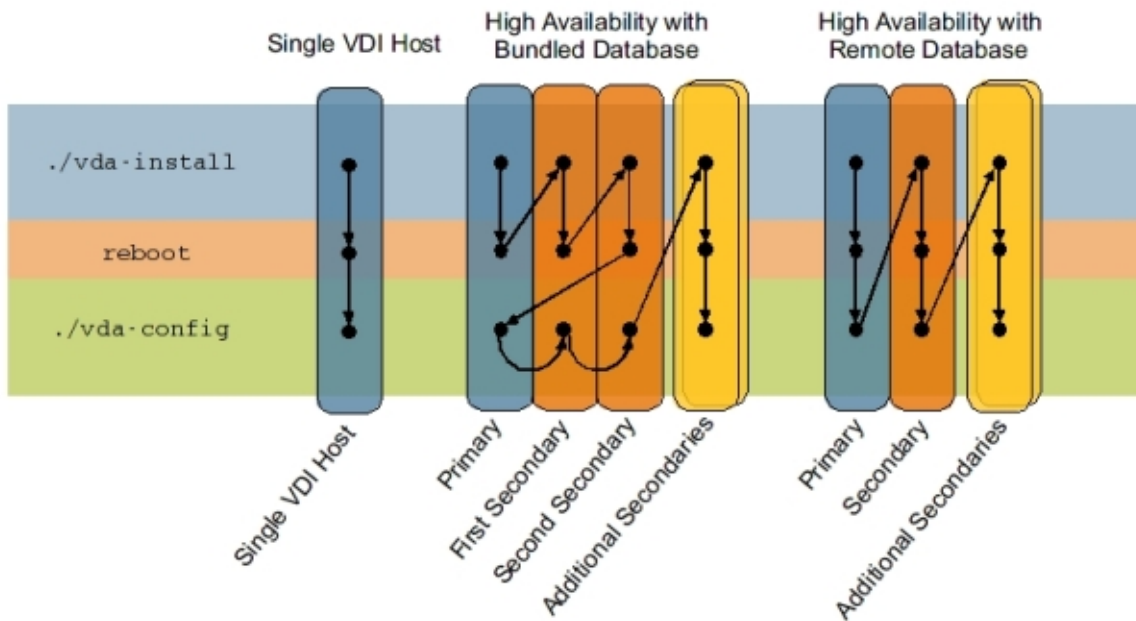
- [About Updating to VDI 3.1](#)
 - [How to Update to VDI 3.1 \(Single VDI Host\)](#)
 - [How to Update to VDI 3.1 \(High Availability with Bundled MySQL Database\)](#)
 - [How to Update to VDI 3.1 \(High Availability with Remote Database\)](#)
 - [How to Update to VDI 3.1 \(VirtualBox and vCenter Virtualization Platforms\)](#)
 - [How to Back Up and Restore Data \(Bundled MySQL Database\)](#)
 - [How to Back Up and Restore Data \(Remote MySQL Database\)](#)
-

VDI 3.1 Update Installation (All Topics)

About Updating to VDI 3.1

Because of the various supported databases and virtualization platforms in VDI 3, it's necessary to follow specific instructions for updating to version 3.1. The following update paths are currently supported:

- [VDI 3 to VDI 3.1 \(Single Host Configuration\)](#)
- [VDI 3 to VDI 3.1 \(High Availability with Bundled MySQL Database\)](#)
- [VDI 3 to VDI 3.1 \(High Availability with Remote Database\)](#)




In addition to updating the VDI Core setup, the virtualization platform will also require an update.

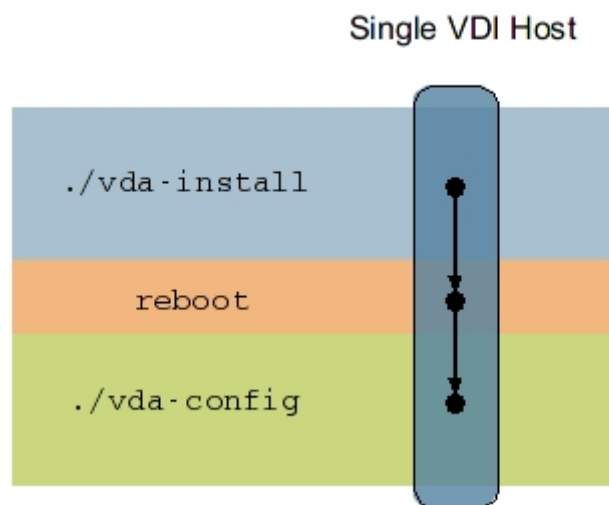
If your VDI 3 installation has a VirtualBox virtualization platform, it's required that you update to the [supported version](#) of VirtualBox for VDI 3.1. The VirtualBox server should be updated after updating the VDI Core hosts to avoid communication issues with the VDI Core. Besides updating the VirtualBox server, you'll also need to update the VirtualBox Guest Additions on all the desktops to correspond to the VirtualBox version on the VirtualBox host.

If your VDI 3 installation has a VMware vCenter virtualization platform, you'll need to update the VDA Tools on each of the desktops.

For more details about how to update VirtualBox and VMware vCenter virtualization platforms for VDI 3.1, refer to the [How to Update to VDI 3.1 \(VirtualBox and vCenter Virtualization Platforms\)](#) page.

How to Update to VDI 3.1 (Single VDI Host)

 **Outage Implications**
Updating the VDI single host will cause a full VDI Core outage.



Before You Begin



Back up!

Use the following checklist to be sure all important data has been backed up properly.



Database - It is highly recommended to back up the database before starting the update process if, for some reason, the update process were to fail. For more information about backing up data on a remote database, refer to the [How to Back Up and Restore Data \(Remote MySQL Database\)](#) page.



Customized `pam.cnf` files - The update process will regenerate the content of the file `/etc/pam.cnf` (an SRSS access configuration file). If you have customized the file, you will need to back it up before updating, and re-add the customization to the newly generated file.



Making Changes with VDI Manager and CLI

Making changes to the VDI Core through the VDI Manager or CLI is strictly forbidden while updating to VDI 3.1.

Steps

1. Run the update installation and configuration on the single VDI host.

a. As root user, unzip the VDI archive, and change to the corresponding directory.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

b. Run the installation.

```
# ./vda-install
```

The `vda-install` script will recognize if VDI 3 is installed and will ask if you want to update. By answering 'y', the installation part of the update process will be started. The old VDI 3 packages will be un-installed and replaced by the new VDI 3.1 packages.

c. Once the `vda-install` script has finished, reboot the single VDI host, and wait until the host is up again.

```
# reboot
```

d. Run the configuration.

```
/opt/SUNWvda/sbin/vda-config
```

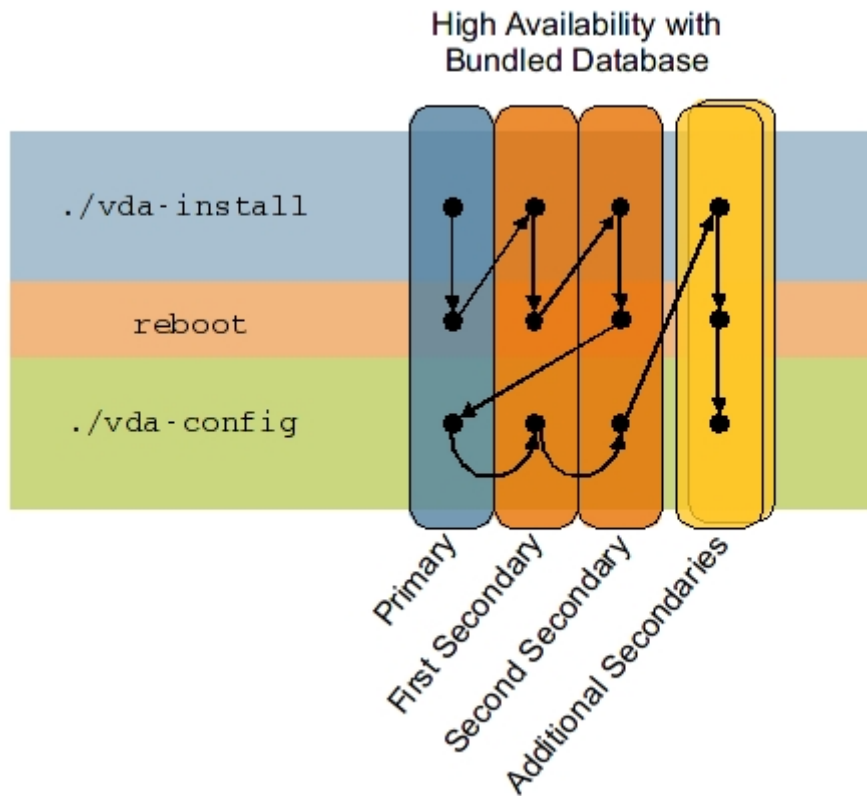
The `vda-config` script will recognize if the VDI Single Host is in the middle of the update process, and will conclude the update process by running the update configuration. The configuration will turn on all VDI-related services again.

How to Update to VDI 3.1 (High Availability with Bundled MySQL Database)



Outage Implications

Updating the VDI Primary host and the first two VDI Secondary hosts will cause a VDI Core outage on these hosts. Additional secondary VDI hosts will stay functional throughout the update process. Once the first phase of the migration has been started you must not restart any of the SQL nodes running on any of the VDI hosts.



Before You Begin



Back up!

Use the following checklist to be sure all important data has been backed up properly.



Database - It is highly recommended to back up the database before starting the update process if, for some reason, the update process were to fail. For more information about backing up data on the bundled MySQL Cluster database, refer to the [How to Back Up and Restore Data \(Bundled MySQL Database\)](#) page.



Customized `my.cnf` files - The update process for any host within a MySQL (embedded) cluster setup will regenerate the content of the file `/etc/opt/SUNWvda/my.cnf` (a database configuration file). If you have customized the file, you will need to back it up before updating, and re-add the customization to the newly generated file.



Customized `pam.cnf` files - The update process will regenerate the content of the file `/etc/pam.cnf` (an SRSS access configuration file). If you have customized the file, you will need to back it up before updating, and re-add the customization to the newly generated file.



Making Changes with VDI Manager and CLI

Making changes to the VDI Core through the VDI Manager or CLI is strictly forbidden while updating to VDI 3.1.

Steps

1. Run the update installation on the Primary VDI host.

- a. As root user, unzip the VDI archive, and change to the corresponding directory.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

- b. Run the installation.

```
# ./vda-install
```

The `vda-install` script will recognize if VDI 3 is installed and will ask if you want to update. By answering 'y' phase 1 of the update process will be started. The installation part of the update process will back up several settings on your Primary and prepare the VDI database to support new VDI 3.1 features. The first update phase will also shut down the MySQL Cluster management node and, because the Primary has been configured to host VDI sessions, will turn down all VDI related services. Next, the old VDI 3 packages will be un-installed and replaced by the new VDI 3.1 packages.

- c. Once the `vda-install` script has finished, reboot your Primary, and wait until the host is up again.

```
# reboot
```

2. Run the update installation on the first VDI Secondary host.

In order to safely update your two Secondary hosts, install the VDI Core on the first Secondary completely, and then install the VDI Core on your second Secondary completely. Installing the VDI Core on both Secondary hosts at the same time is not supported.

- a. As root user, unzip the VDI archive, and change to the corresponding directory.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

- b. Run the installation.

```
# ./vda-install
```

The `vda-install` script will recognize if VDI 3 is installed and will ask you whether you want to update or not. By answering 'y' phase 1 of the update process will be started. The installation part of the update process will back up several settings on your Secondary and prepare the VDI database to support new VDI 3.1 features. The first update phase will also turn down all VDI-related services on the host. Next, the old VDI 3 packages will be un-installed and replaced by the new VDI 3.1 packages.

- c. Once the `vda-install` script has finished, reboot your first Secondary and wait until the host is up again.

```
# reboot
```

- d. Check that the MySQL database has been fully started again by running the following command.

```
/opt/SUNWvda/sbin/vda-db-status
```

All nodes must be in the 'up' state.

3. Run the update installation on the second VDI Secondary host by following Step 2 above.

4. Run the update configuration on the VDI Primary host.

Once the installation part of the update process is complete on the VDI Primary and first two Secondaries, you are ready to start the configuration part of the update process.

- As root user, run the following command.

```
/opt/SUNWvda/sbin/vda-config
```

The `vda-config` script will recognize if the VDI Primary is in the middle of the update process, and will conclude the update process by running the update configuration. If the VDI Primary host was configured to act as a full functional VDI host, the configuration will turn on all VDI-related services again.

5. Run the update configuration on the first VDI Secondary host.

After the update configuration has finished on the VDI Primary, you can start the update configuration on the first two VDI Secondaries. Like the update installation, you must completely configure the first VDI Secondary, then completely configure the second VDI Secondary. Configuring the VDI 3.1 Core on both Secondary hosts at the same time is not supported.

- As root user, run the following command.

```
/opt/SUNWvda/sbin/vda-config
```

The `vda-config` script will recognize if the VDI Secondary is in the middle of the update process, and will conclude the update process by running the update configuration. The configuration will turn on all VDI-related services again.

6. Run the update configuration on the second VDI Secondary host by following Step 5 above.

7. Run the update installation and configuration on additional (optional) VDI Secondary hosts.

After you have completely updated the VDI Primary host and first two VDI Secondary hosts, you can update the additional VDI Secondaries in any sequence. To start the update process, run the following steps.

- a. As root user, unzip the VDI archive, and change to the corresponding directory.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

- b. Run the installation.

```
# ./vda-install
```

The `vda-install` script will recognize if VDI 3 is installed and will ask you whether you want to update or not. By answering 'y' the migration process will be started. It will un-install the old VDI 3 packages, and replace them with the new VDI 3.1 packages.

- c. Once `vda-install` script has finished, reboot your VDI host, and wait until the host is up again.

- d. Run the configuration.

```
/opt/SUNWvda/sbin/vda-config
```

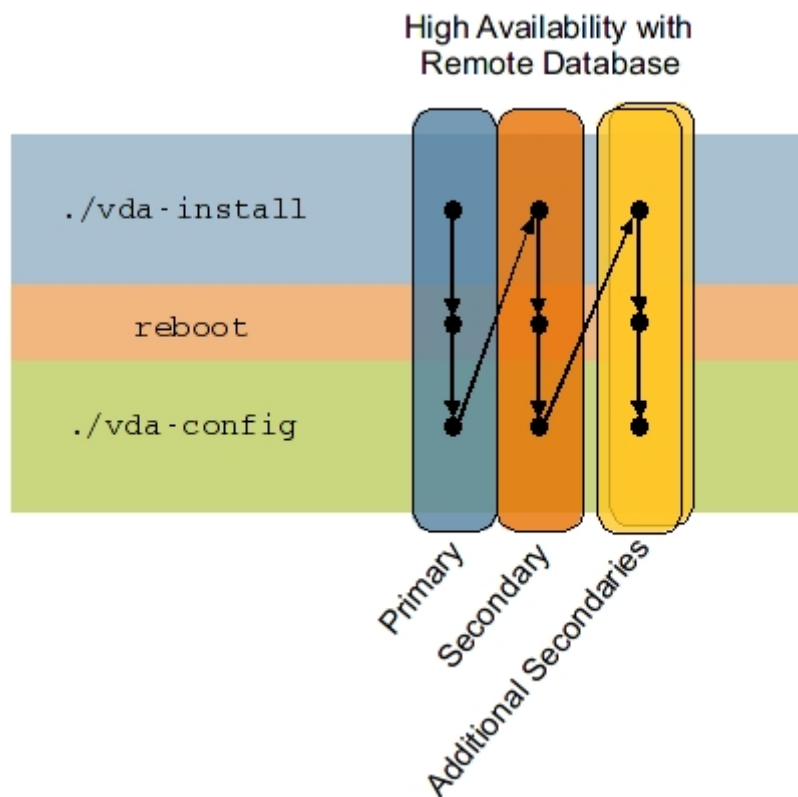
The `vda-config` script will recognize if the VDI Secondary is in the middle of the update process, and will conclude the update process by executing the update configuration. The update configuration will turn on all VDI related services again.

How to Update to VDI 3.1 (High Availability with Remote Database)



Outage Implications

Updating the VDI Primary host and the first two VDI Secondary hosts will cause a VDI Core outage on these hosts. Additional secondary VDI hosts will stay functional throughout the update process. Once the first phase of the migration has been started you must not restart any of the SQL nodes running on any of the VDI hosts.



Before You Begin



Back up!

Use the following checklist to be sure all important data has been backed up properly.



Database - It is highly recommended to back up the database before starting the update process if, for some reason, the update process were to fail. For detailed information about backing up data on a remote database, refer to the [How to Back Up and Restore Data \(Remote MySQL Database\)](#) page.



Customized `pam.cnf` files - The update process will regenerate the content of the file `/etc/pam.cnf` (an SRSS access configuration file). If you have customized the file, you will need to back it up before updating, and re-add the customization to the newly generated file.



Making Changes with VDI Manager and CLI

Making changes to the VDI Core through the VDI Manager or CLI is strictly forbidden while updating to VDI 3.1.

Steps

1. Run the update installation and configuration on the VDI Primary host.

a. As root user, unzip the VDI archive, and change to the corresponding directory.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

b. Run the installation.

```
# ./vda-install
```

The `vda-install` script will recognize if VDI 3 is installed and will ask if you want to update. By answering 'y' phase 1 of the update process will be started. The installation part of the update process will back up several settings on your Primary and prepare the VDI database to support new VDI 3.1 features. The first update phase will also shut down the MySQL Cluster management node and, because the Primary has been configured to host VDI sessions, will turn down all VDI related services. Next, the old VDI 3 packages will be un-installed and replaced by the new VDI 3.1 packages.

c. Once the `vda-install` script has finished, reboot your Primary, and wait until the host is up again.

```
# reboot
```

d. Run the configuration.

```
/opt/SUNWvda/sbin/vda-config
```

The `vda-config` script will recognize if the VDI Primary is in the middle of the update process, and will conclude the update process by running the update configuration. If the VDI Primary host was configured to act as a full functional VDI host, the

configuration will turn on all VDI-related services again.

2. Run the update installation and configuration on the VDI Secondary host.

In order to safely update your two Secondary hosts, install the VDI Core on the first Secondary completely, and then install the VDI Core on your second Secondary completely. Installing the VDI Core on both Secondary hosts at the same time is not supported.

a. As root user, unzip the VDI archive, and change to the corresponding directory.

```
# unzip vda_3.1_amd64.zip
# cd vda_3.1_amd64
```

or

```
# unzip vda_3.1_sparc.zip
# cd vda_3.1_sparc
```

b. Run the installation.

```
# ./vda-install
```

The `vda-install` script will recognize if VDI 3 is installed and will ask you whether you want to update or not. By answering 'y' the migration process will be started. It will un-install the old VDI 3 packages, and replace them with the new VDI 3.1 packages.

c. Once `vda-install` script has finished, reboot your VDI host, and wait until the host is up again.

d. Run the configuration.

```
/opt/SUNWvda/sbin/vda-config
```

The `vda-config` script will recognize if the VDI Secondary is in the middle of the update process, and will conclude the update process by executing the update configuration. The update configuration will turn on all VDI related services again.

How to Update to VDI 3.1 (VirtualBox and vCenter Virtualization Platforms)

When updating from VDI 3 to VDI 3.1, it will necessary to perform some tasks on the virtualization platform. VDI installations with a VirtualBox virtualization platform will require a newer version of VirtualBox to be to use the new features in VDI 3.1, and the corresponding Guest Additions will need to be installed on the VirtualBox virtual machines. VMware vCenter virtualization platforms will not require updating since the VDI 3.1 supports all the versions of vCenter and ESX Server as in VDI 3.

Before You Begin

The VDI Core hosts should be updated before the virtualization platform. If you have not already updated the VDI Core hosts, refer to the [About Updating to VDI 3.1](#) page for more details.

Steps for Sun VirtualBox Virtualization Platforms

1. Update the VirtualBox version on the VirtualBox host.

- a. Un-install the previous version of VirtualBox by changing to the VirtualBox installation directory, and running the following command as root user.

```
# ./vb-install -u
```

- b. When VirtualBox has been successfully removed, install the [currently supported](#) version of VirtualBox by running the following command.

```
# ./vb-install
```

2. Update the Guest Additions on VirtualBox virtual machines.

Steps for VMware vCenter Virtualization Platforms

The only requirement for updating the vCenter virtualization platform to work with VDI 3.1 is to update the VDA tools on each of the vCenter desktops.

- Update the VDA Tools on vCenter desktops.
 1. Remove the existing VDA Tools file from the virtual machine. The default location for the VDA Tools on Windows is `C:\Program Files\Sun\Virtual Desktop Access\Tools`.
 2. Copy the `vda-tools.msi` installer file from the `/var/tmp/vda_3.1_amd64/vda_3.1/Windows/Packages` directory to the virtual machine.
 3. Within the virtual machine's console, double-click the installer and follow the prompts to complete installation. The default target location for the VDA Tools on Windows is `C:\Program Files\Sun\Virtual Desktop Access\Tools`.
 4. The VM Services list should now contain a new service named Sun VDI Tools, running and set to start automatically.

How to Back Up and Restore Data (Bundled MySQL Database)

In a typical VDI instance, using the bundled MySQL database, a back up is only necessary when updating to a new release of VDI. The integration of the bundled MySQL database already provides an assurance of fail-safety, that reduces the need for backups as a means for disaster recovery. Find further details about this topic in the official [MySQL documentation](#).



Please keep in mind that flexible desktop assignments existing at the time when the backup will be created might not be valid anymore when the VDI configuration has to be restored from this backup. This may lead to some unexpected side effects. For this reason you should consider replication as an alternative to regularly taken backups. Master-Slave replication is supported with the MySQL Cluster version coming along with VDI. Find details on the topic here: [MySQL Cluster Replication](#)

Steps

The following procedure assumes that you have a functioning (installed and configured) VDI instance, using the bundled MySQL database.

1. Make a backup of the VDI database.
 - a. Open the `ndb_mgm` console by running the following command on the VDI Primary host.

```
# /opt/SUNWvda/mysql/bin/ndb_mgm
```

- b. Start the back up by running the following command at the `ndb_mgm` prompt.

```
ndb_mgm> START BACKUP
```

- c. Verify that backups have been created by checking the following directory on the two VDI Secondary hosts:
`/var/opt/SUNWvda/mysql-cluster/BACKUP`.

2. (Optional) Perform a VDI update.
For more information about updating VDI, refer to the [About Updating to VDI 3.1](#) page.

3. Restore the backed up VDI database on the new VDI installation.
 - a. On the first VDI Secondary host, run the following command.

```
/opt/SUNWvda/mysql/bin/ndb_restore -b <backup #> -n <nodeid #> -r  
--backup_path=<path>
```

- b. On the second VDI Secondary host, run the following command.

```
/opt/SUNWvda/mysql/bin/ndb_restore -b <backup #> -n <nodeid #> -r  
--backup_path=<path>
```

- c. Log into the VDI Manager, and check that all the data is restored.
Go to <http://<server name>:1800> (or <http://localhost:1800> if remote administration has been disabled), and use root user credentials to log in.

How to Back Up and Restore Data (Remote MySQL Database)

The following information should be used when backing up data on a remote database in either a Single Host or High Availability (Remote MySQL Database) configuration.

Steps

1. Make a backup of the VDI database.
 - a. Log into the remote database.

```
# zlogin <MySQL server>
```

- b. Change to the `usr/local/mysql/bin/` directory and run `mysqldump`.

```
# cd usr/local/mysql/bin/  
# ./mysqldump --user=root -u root --opt checkdb | gzip > /dumptest1.sql.gz
```

2. (Optional) Perform a VDI update.



During the un-configuration, answer 'yes' to deleting the remote database. Use the same values for "privileged administrator", "VDA database" , "user name for the VDA database" used while executing `vda-config` for the re-configuration.

For more information about updating VDI, refer to the [About Updating to VDI 3.1](#) page.

3. Restore the backed up VDI database on the new VDI installation.
 - a. Log into the remote database.

```
# zlogin <MySQL server>
```

- b. Change to the `usr/local/mysql/bin/` directory and run the following command.

```
# cd usr/local/mysql/bin/  
# ./mysql --user=root checkdb < /primary-dump/dumptest1.sql
```