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

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

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.
Oracle Virtual Desktop Infrastructure 3.2 Installation and Configuration Guide

Contents

- Oracle Virtual Desktop Infrastructure 3.2 Installation and Configuration Guide
- About VDI (All Topics)
- New Features of Oracle Virtual Desktop Infrastructure 3.2
- System Requirements (All Topics)
- Oracle VDI Hypervisor 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 Oracle VDI Core Host Configuration (All Topics)
- High Availability with MySQL Cluster Configuration (All Topics)
- High Availability with Remote MySQL Configuration (All Topics)

Oracle Virtual Desktop Infrastructure 3.2 Installation and Configuration Guide

Contents

- Architecture

About VDI (All Topics)

New Features of Oracle Virtual Desktop Infrastructure 3.2

With Oracle Virtual Desktop Infrastructure, you can deploy a number of virtualized 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. Oracle Virtual Desktop Infrastructure also enables you to utilize existing IT assets, increase scalability, and simplify management.

Oracle Virtual Desktop Infrastructure 3.2 introduces the following new features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Oracle VDI Centers</td>
<td>Extends the basic &quot;hot desking&quot; experience known from a single Oracle Virtual Desktop Infrastructure environment to encompass multiple Oracle Virtual Desktop Infrastructure environments. It helps when users travel from one site to another site and need access to their desktops. For more information, refer to the following pages:</td>
</tr>
<tr>
<td></td>
<td>• About Global Oracle VDI Centers</td>
</tr>
<tr>
<td></td>
<td>• How to Prepare a User Directory for Global Oracle VDI Centers</td>
</tr>
</tbody>
</table>

For more information, refer to the following pages:
| **Companies** | Enable multiple user directories to be configured for one Oracle Virtual Desktop Infrastructure environment. This feature provides privacy between multiple groups within the same Oracle VDI environment.  
For more information, refer to the following pages:  
- About Companies  
- How to Create a Company |
| **Windows Deployment Enhancements** | Improve Windows desktop deployment with the Personal Hard Drive and Oracle VDI Fast Preparation features.  
For more information, refer to the following pages:  
- About Personal Hard Drives  
- How to Enable Oracle VDI Fast Preparation for Windows Templates (Oracle VDI Hypervisor and Microsoft Hyper-V) |
| **Oracle VDI Hypervisor Enhancements** | Run Oracle VM VirtualBox desktops more efficiently with the Shared Memory and Memory Paging features.  
For more information, refer to the following page:  
- About Oracle VDI Hypervisor Enhancements |
| **Administration Enhancements** | Manage your Oracle Virtual Desktop Infrastructure installation more easily with the new Back Up and Restore, Clone and Recycle Job Management, Desktop Search, and Desktop Provider Alarm features.  
For more information, refer to the following pages:  
- How to Back Up and Restore the VDI Core Database  
- About Clone and Recycle Job Management  
- How to Search for Desktops  
- About Desktop Provider Alarms |
| **Remote Desktop Protocol Enhancements** | Improve the user desktop experience with the Video Redirection, Auto-Logon, and Audio Input remote desktop protocol features.  
For more information, refer to the following pages:  
- About RDP Enhancements |
| **Storage Enhancements** | Take advantage of storage enhancements including, Active/Active OpenStorage Clustering, Dedicated iSCSI Networking, OpenStorage Write Cache Support (including Logzilla), Orphan Desktop Deletion, and more robustness (including finegrained storage outage detection).  
For more information, refer to the following pages:  
- About OpenStorage Clustering  
- About Dedicated iSCSI Networking  
- About OpenStorage Write Cache  
- How to Delete Orphan Disks |
| **Better Active Directory Support** | Configure Companies with multiple forests, with multiple domains.  
For more information, refer to the following page:  
- About Complex Forest Configurations |
| **Generic Desktop Providers** | Utilize existing PC hardware in your Oracle Virtual Desktop Infrastructure installation. For more information, refer to the following pages:  
- About Generic Desktop Providers  
- How to Import Individual Windows PCs |
| **Sun Ray Desktop Screen Locking** | Increase desktop security for Sun Ray Thin Clients with the Desktop Screen Locking feature. For more information, refer to the following page:  
- How to Enable Desktop Screen Locking on Sun Ray Clients |

**Architecture**

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

**Desktop Access**  
Three distinct mechanisms are supported for access to virtual desktops.

Sun Ray Thin Client or Oracle Virtual Desktop Client - Users can access their virtual desktops through a Sun Ray Thin Client (DTU) or through Oracle Virtual Desktop Client software by authenticating themselves with a user name and password, or a token card can be inserted in a Sun Ray Thin Client in place of providing a user name. Successful authentication initiates a custom Sun Ray Software Kiosk Session. The custom Kiosk Session uses the Oracle 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 an Oracle Secure Global Desktop (SGD) session. SGD, in turn, uses the Oracle 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 Oracle VDI Core's redirection capability is used to establish a connection to an assigned virtual desktop.

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

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

**Virtualization Platform**  
The basis for the Oracle Virtual Desktop Infrastructure 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. Oracle Virtual Desktop Infrastructure 3.2 supports Oracle VM VirtualBox (the Oracle VDI Hypervisor), VMware vCenter, Microsoft Hyper-V, and Microsoft Remote Desktop Services as virtualization platforms.

**Storage**  
Oracle Virtual Desktop Infrastructure takes advantage of iSCSI paired with ZFS in the Sun Unified Storage 7000 Series (Amber Road) or the Oracle Solaris OS to provide reliable storage for Oracle VM 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.
System Requirements (All Topics)

Configurations

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

Configuration Options

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

<table>
<thead>
<tr>
<th>Configuration Type</th>
<th>Options Selected During Configuration</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware vCenter</th>
<th>Microsoft Hyper-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo (Evaluation) Configuration</td>
<td>- On the Demo Host: 0 Evaluation VDI Host</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High Availability Configuration</td>
<td>- On the Primary Host: 1 Primary Oracle VDI Host</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MySQL Cluster</td>
<td>- On the first Secondary Host: 2 Secondary Oracle VDI Host</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>- On the second Secondary Host: 2 Secondary Oracle VDI Host</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>High Availability Configuration</td>
<td>- On the Primary Host: 1 Primary Oracle VDI Host</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>with Remote MySQL</td>
<td>- On the Secondary Host: 2 Secondary Oracle VDI Host (Specify Remote Database)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Single Oracle VDI Core Host</td>
<td>- On the Single Host: 3 Single Oracle VDI Host</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Oracle VM VirtualBox on Primary Host Configuration

- On the Primary Host: 1 Primary Oracle VDI Host
- On the first Secondary Host: 2 Secondary Oracle VDI Host
- On the second Secondary Host: 2 Secondary Oracle VDI Host

Primary Host Virtualized Configuration

- On the Primary Host: 1 Primary Oracle VDI Host
- On the first Secondary Host: 2 Secondary Oracle VDI Host
- On the second Secondary Host: 2 Secondary Oracle VDI Host

Support Information

- Some Oracle Virtual Desktop Infrastructure configurations are unsupported configurations. See below for more details.
- Because VMware and Microsoft software are not included as part of the Oracle Virtual Desktop Infrastructure 3.2 package, Oracle Support contracts do not cover VMware-related issues. For VMware coverage, you need an additional support plan.

Supported Configurations

The configurations described in this section are supported as production environment configurations by an Oracle Virtual Desktop Infrastructure contract.

High Availability Configuration with MySQL Cluster

The Oracle VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one Oracle VDI Core host fails, all users with desktop sessions on that host are kicked back to the Oracle 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 Oracle Virtual Desktop Infrastructure stack requires a minimum of three Oracle VDI Core 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 (MySQL Cluster) configuration, refer to the Oracle Virtual Desktop Infrastructure 3.2 Software Requirements and About Oracle VDI Core Configuration pages.
Instead of using the bundled MySQL cluster database, you can also choose to use an existing database and connect to it by specifying it as a remote database during Oracle VDI Core configuration. In this case, the Oracle 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) configuration, refer to the About Oracle VDI Core page.
For more information about the Single Oracle VDI Core Host configuration, refer to the About Oracle VDI Core Configuration page.

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

Oracle VM VirtualBox on the Primary Host Configuration

In the Oracle VM VirtualBox on the Primary Host Configuration, the Primary node and one of your Oracle VM VirtualBox hosts share one physical machine. Make sure the shared host has enough capacity to deal with these two roles at the same time.

Primary Host Virtualized Configuration

In the Primary Host Virtualized Configuration, the Primary node runs in a virtual machine hosted by the 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 Secondary hosts running the MySQL Cluster data nodes nevertheless need to run on bare metal.
Unsupported Configurations

Oracle Virtual Desktop Infrastructure Demo (Evaluation) Configuration

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

Virtualized Oracle VDI Core Configuration

In a Virtualized Oracle VDI Core Configuration, the Primary and two Secondary nodes could be hosted in virtual machines. The Oracle VDI 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 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.

Oracle Virtual Desktop Infrastructure 3.2.1 Software Requirements

This section includes support tables for Oracle VDI Core host operating systems, virtualization platforms, storage servers, desktop guest systems. For more information about how to configure the Oracle Virtual Desktop Infrastructure components, refer to the Configurations page.

Oracle VDI Core Host and Oracle VM VirtualBox Virtualization Host Operating Systems

<table>
<thead>
<tr>
<th>Software</th>
<th>Supported in Oracle Virtual Desktop Infrastructure 3.2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris 10 5/09 x86 (64-bit) or later</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Virtualization Platforms and Desktop Providers

<table>
<thead>
<tr>
<th>Software</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware Infrastructure</th>
<th>Microsoft Hyper-V</th>
<th>Microsoft Remote Desktop</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypervisor</td>
<td>VMware Infrastructure</td>
<td>Microsoft Hyper-V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------</td>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle VM VirtualBox 3.0.12, 3.0.14, 3.2.8, 3.2.10</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All other Oracle VM VirtualBox versions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware VirtualCenter server 2.5 (Including Updates 1.2,3.4)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware vCenter server 4.0 (Including Updates 1.2), 4.1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware ESX server 3.5, 4, 4.1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware ESXi server 3.5, 4, 4.1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Hyper-V Server 2008 R2</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Server 2008 R2</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Server 2003</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Storage

<table>
<thead>
<tr>
<th>Software</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware Infrastructure</th>
<th>Microsoft Hyper-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris 10 10/09 x86 (64-bit) or later</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q3.x</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.2.1</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.2.0</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.1.0</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.0.2</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.0.1</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.0.0</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2009.Q2.5.1</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Solaris 10 10/09 with Sun Modular Storage (ST2530, ST2540, ST6140, ST6180, ST6540, ST6580, ST6780)</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
</tbody>
</table>

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

### User Directories

<table>
<thead>
<tr>
<th>Versions of Active Directory</th>
<th>Supported</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2003</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2003 R2</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Oracle Virtual Desktop Infrastructure 3.2 Software Requirements

This section includes support tables for Oracle VDI Core host operating systems, virtualization platforms, storage servers, desktop guest systems. For more information about how to configure the Oracle Virtual Desktop Infrastructure components, refer to the Configurations page.

Oracle VDI Core Host and Oracle VM VirtualBox Virtualization Host Operating Systems

| Windows Server 2008 R2 | Supported
|------------------------|------------
| LDAP Directories       | Supported  | Not Supported
| Oracle Directory Server Enterprise Edition 6.3.1 | ✓          |          |
| Oracle Directory Server Enterprise Edition 7.0 | ✓          |          |
| Novell eDirectory 8.8  | ✓          |          |
| OpenLDAP 2.4.19        | ✓          |          |

Desktop Guest Systems

<table>
<thead>
<tr>
<th>Software</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware Infrastructure</th>
<th>Microsoft Hyper-V</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows XP SP2/3 (32bit and 64bit)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows Vista Enterprise</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows 7 (32bit and 64bit)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows 2000 SP4</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Solaris 10 (10/09) or later</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 8.10 (Intrepid Ibex)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 9.04 (Jaunty Jackalope)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 9.10 (Karmic Koala)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 10.04 (Lucid Lynx)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux Enterprise 11</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Enterprise Linux 5.5</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Web Browser for Oracle VDI Manager

<table>
<thead>
<tr>
<th>Web Browser</th>
<th>Firefox 3.6</th>
<th>Internet Explorer 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Solaris</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>Linux</td>
<td>✓</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Software | Supported in Oracle Virtual Desktop Infrastructure 3.2
---|---
Oracle Solaris 10 5/09 x86 (64-bit) or later | ✓

Virtualization Platforms and Desktop Providers

<table>
<thead>
<tr>
<th>Software</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware Infrastructure</th>
<th>Microsoft Hyper-V</th>
<th>Microsoft Remote Desktop</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle VM VirtualBox 3.0.12, 3.0.14, 3.2.8</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All other Oracle VM VirtualBox versions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>VMware VirtualCenter server 2.5 (Including Updates 1,2,3,4)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware vCenter server 4.0 (Including Updates 1,2)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware ESX server 3.5, 4</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware ESXi server 3.5, 4</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Hyper-V Server 2008 R2</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Server 2008 R2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft Windows Server 2003</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Storage

<table>
<thead>
<tr>
<th>Software</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware Infrastructure</th>
<th>Microsoft Hyper-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris 10 10/09 x86 (64-bit) or later</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q3.x</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.2.1</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.2.0</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.1.0</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.0.2</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.0.1</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2010.Q1.0.0</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Sun Unified Storage 7000 Series 2009.Q2.5.1</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
<tr>
<td>Solaris 10 10/09 with Sun Modular Storage (ST2530, ST2540, ST6140, ST6180, ST6540, ST6580, ST6780)</td>
<td>✓</td>
<td>Qualified by VMware</td>
<td>✓</td>
</tr>
</tbody>
</table>

Tip
For more information about the Sun Unified Storage 7000 Series, see the Fishworks Documentation.

User Directories
<table>
<thead>
<tr>
<th>Versions of Active Directory</th>
<th>Supported</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2003</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Windows Server 2003 R2</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LDAP Directories</th>
<th>Supported</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Directory Server Enterprise Edition 6.3.1</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Oracle Directory Server Enterprise Edition 7.0</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Novell eDirectory 8.8</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>OpenLDAP 2.4.19</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

**Desktop Guest Systems**

<table>
<thead>
<tr>
<th>Software</th>
<th>Oracle VDI Hypervisor</th>
<th>VMware Infrastructure</th>
<th>Microsoft Hyper-V</th>
<th>Not Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows XP SP2/3 (32bit and 64bit)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows Vista Enterprise</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows 7 Enterprise/Ultimate (32bit and 64bit)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Windows 2000 SP4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Ubuntu 8.10 (Intrepid Ibex)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 9.04 (Jaunty Jackalope)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 9.10 (Karmic Koala)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu 10.04 (Lucid Lynx)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux Enterprise 11</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Enterprise Linux 5.5</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Web Browser for Oracle VDI Manager**

<table>
<thead>
<tr>
<th>OS</th>
<th>Firefox 3.6</th>
<th>Internet Explorer 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Solaris</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>Linux</td>
<td>✓</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Hardware Requirements**
Oracle Virtual Desktop Infrastructure has very basic hardware requirements. Customers can use new or existing hardware as long as it meets the CPU requirements. Hardware sizing is a very important part of planning an Oracle Virtual Desktop Infrastructure installation. Refer to an Oracle Sales or Support representative for more details about sizing.

All Oracle VM VirtualBox servers must have the virtualization extensions from AMD (AMD-V) and Intel (VT-x).

<table>
<thead>
<tr>
<th>Host</th>
<th>CPU Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle VDI Core hosts</td>
<td>all x86 CPU</td>
</tr>
<tr>
<td>Oracle VM VirtualBox host</td>
<td>x86 CPU</td>
</tr>
<tr>
<td>Microsoft Hyper-V host</td>
<td>x86 CPU</td>
</tr>
<tr>
<td>Oracle VM VirtualBox or Hyper-V storage server</td>
<td>x86 CPU</td>
</tr>
<tr>
<td>VMware Infrastructure hosts</td>
<td>x86 CPU</td>
</tr>
<tr>
<td>Microsoft Remote Desktop Server</td>
<td>x86 CPU</td>
</tr>
</tbody>
</table>

As of Oracle Virtual Desktop Infrastructure 3.2, Solaris SPARC is no longer supported as an Oracle VDI Core host platform.

Deployment Guide

New features of Oracle Virtual Desktop Infrastructure 3.2 such as memory paging or VRDP RCA are not reflected below. Storage enhancements are also not factored into the recommendations and need extra evaluation.

This page provides guidelines for the design of the hardware architecture for Oracle Virtual Desktop Infrastructure deployments with an Oracle VM 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.
4. Start Microsoft PowerPoint. Load a large presentation and browse the slides. Close Microsoft PowerPoint.

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.

Oracle VM VirtualBox Virtualization Platform Architecture

The hardware environment for an Oracle Virtual Desktop Infrastructure deployment typically looks like this:
Every production deployment consists of one primary Oracle VDI Core host and at least two secondary Oracle VDI Core hosts to provide redundancy. The Oracle VDI Core servers host a clustered MySQL database for the Oracle VDI Core 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 Oracle VM 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 Oracle VM VirtualBox servers and the storages. That iSCSI data creates a major part of the total network traffic of a Oracle Virtual Desktop Infrastructure system. For more information, see the Sizing Guidelines for Storage Servers section below.

Other consumers of network bandwidth are the clients of Oracle Virtual Desktop Infrastructure: Sun Rays, Oracle Secure Global Desktop, and RDC clients. The clients connect to the Oracle VM VirtualBox servers through the Oracle VDI Core servers. In the case of a Sun Ray client, which uses the ALP protocol to transfer the desktop graphics, the Oracle VDI Core servers convert the RDP protocol received by the Oracle VM VirtualBox servers to the ALP protocol. So, there is one data stream for each client connection between the client, the Oracle VDI Core server, and the Oracle VM VirtualBox server. RDP clients such as the Windows Connector (uttsc), connect to the Oracle VDI Core server which, in turn, uses the RDP Redirect feature to instruct the clients to connect to the Oracle VM 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 Oracle VM VirtualBox server.

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

### Sizing Guidelines for Oracle VDI Core Servers

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

The secondary Oracle 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 Oracle VDI Core servers with 8 CPUs and 4 cores per CPU can serve $2 \times 8 \times 4 \times 20 = 1280$ running desktops

- **Memory size [MB]** = number of desktops \* 32 MB + 2048 MB  
  Example: Two secondary Oracle VDI Core servers with 64 GB of memory can serve $(2 \times 64 \times 1024 \text{ MB} - 2 \times 2048 \text{ MB}) / 32$
Network bandwidth [Mb/s] = number of running desktops * 0.15 [Mb/s]
Example: Two secondary Oracle VDI Core server with one 1 Gb Ethernet interface can serve 2 * 1024 / 0.15 Mb/s = 13653 running desktops

For more information, see the Complete Sun Ray Server Sizing Guide.

Sizing Guidelines for Oracle VM VirtualBox Servers

We found that the 'VMs/core' unit, while being striking, is a fuzzy statement as the available CPUs today differ by at least a factor of 2 in performance and that even ignores older CPUs customers may want to reuse. Therefore we decided to also provide the 'SPEC CINT2006 Rate (peak) / VM' value. Statements made based on this unit are valid for a longer time as they abstract from a concrete CPU, while statements based on 'VMs/core' hold true for cores showing roughly the same performance only.

CINT values for a vast number of CPUs can be looked up from the database of the Standard Performance Evaluation Corporation (SPEC) at http://www.spec.org/cpu2006/results/rint2006.html or by running the provided test suite.

The numbers for this section have been updated based on a new test run. We were able to run 100 VMs on a X4170 with two E5520 CPUs having 4 cores each. The SPEC CINT2006 Rate (peak) for E5520 CPUs is ~200 which results in a cint / VM value of 2.

Number of cores = number of running desktops / 12.5
Example: A server roughly equivalent to a X4170 with two E5520 CPUs can support up to 2 * 4 * 12.5 = 100 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 MB - 1024 MB / (512 MB * 1.2) = 105 running desktops of 512 MB in size

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

At least 20% of the available CPU power, memory size and network bandwidth should be available as security margin.

Sizing Guidelines for Storage Servers

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.

OpenStorage servers can use flash write cache (aka Logzilla or ZeusIOPS) as well as standard in-memory write cache for iSCSI LUNs. In-memory write cache is faster, cheaper and puts less load on the CPU than flash write cache. Flash write cache on the other hand is able to maintain unwritten data even in case of a power outage. If the in-memory write cache is activated, which is the default setting of VDI, and flash write cache is present, the flash write cache is not used. We recommend to use in-memory write cache.

The storage servers provide the virtual disks that are accessed by Oracle VM 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 (XP guest) 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, Oracle VM VirtualBox servers, and switches. Note that Jumbo Frames are not standardized so there is a risk of incompatibilities.

The Oracle VDI Core, in combination with Oracle VM 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.

- **Number of disks** = number of desktops / 10

- **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 GB / 2 / 8 GB = 2944 disks in a RAID 10 configuration

For details about how to improve desktop performance, refer to the Best Practices for Desktop Images page.

### Contents

- About Oracle VDI Hypervisor Enhancements
- How to Install Oracle VM VirtualBox
- How to Set Up an Oracle Solaris Storage Server
- How to Set Up a Sun Storage 7000 Unified Storage System
- About Dedicated iSCSI Networking
- About OpenStorage Clustering
- About OpenStorage Write Cache
- About Maintenance Mode
- How to Use Maintenance Mode
  
  - Oracle VM VirtualBox and Microsoft Hyper-V Host Maintenance
  - Storage Maintenance
- How to Replicate and Replace a Sun Unified Storage System
- How to Improve Networking Availability and Performance
  
  - Link Aggregation
  - VLANs

### Oracle VDI Hypervisor Platform Installation (All Topics)

**About Oracle VDI Hypervisor Enhancements**

Oracle Virtual Desktop Infrastructure bundles and supports a specific Oracle VM VirtualBox release, which is also referred to as the Oracle VM VirtualBox for Oracle VDI 3.2. For accurate and up-to-date information about release support, refer to the Oracle Virtual Desktop Infrastructure 3.2.2 Release Notes. The Oracle VDI Hypervisor provides the following new Oracle VDI desktop provider features.

**Shared Memory**

Shared Memory (also known as memory ballooning) is a feature that enables more desktops to run on Oracle VDI Hypervisor hosts. By specifying an amount of memory to be shared between desktops, the Oracle VDI hypervisor host's memory can be
automatically redistributed between desktops as required. The Shared Memory feature can be activated on a per-pool basis on the Pool category, Settings tab by specifying a value greater than 0% (up to 75%).

The memory sharing percentage is the amount of memory that can be used for other desktops if a desktop does not require the full amount of memory for itself. For instance, if the desktop memory size is 1 GB and memory sharing is set to 40%, the desktop will initially have around 600 MB of real memory. The other 400 MB will be made available to the desktop on demand.

Oracle Virtual Desktop Infrastructure constantly monitors desktops with memory sharing enabled to ensure they don’t run out of memory. If a desktop's free memory drops below 64 MB more usable memory will be provided. If a desktop has excessive amounts of free memory, some memory will be gradually taken away until the memory sharing percentage is reached. The changes to desktop memory will not be apparent to the guest OS.

Memory Paging

Memory Paging (also known as de-duplication of memory) is a feature that enables more desktops to run on Oracle VDI Hypervisor hosts. If several desktops have identical content in memory, pages will be used to consume real memory on the hypervisor only once. The desktops will reference the page and do not need physical memory for identical pages any more.

The Memory Paging feature can be activated on a per-pool basis on the Pool category, Settings tab.

How to Install Oracle VM VirtualBox

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

The following servers can be used as storage by Oracle VM VirtualBox:

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

Refer to the Oracle Virtual Desktop Infrastructure 3.2.2 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 Oracle Solaris Storage Server

Steps

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

   The following restriction is applicable only to Oracle Virtual Desktop Infrastructure 3.2 and earlier.

   Oracle Solaris hosts running Oracle VM VirtualBox must have swap space equal to or greater than the host's physical memory size. See Bug ID 1225025 in the Oracle Virtual Desktop Infrastructure 3.2 Release Notes for detailed information.

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

```
# unzip vda_3.2.2_amd64.zip
# cd vda_3.2.2_amd64
# unzip vbox_3.2.zip
# cd vbox_3.2
# ./vb-install
```
3. Complete installation as prompted by the `vb-install` script.

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

How to Set Up an Oracle Solaris Storage Server

An Oracle 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 Oracle VM VirtualBox through iSCSI. The management of the ZFS volumes is done by Oracle Virtual Desktop Infrastructure and requires SSH root access to the ZFS storage server and a ZFS pool on that server.

Refer to the Oracle Virtual Desktop Infrastructure 3.2.2 Release Notes for the most accurate and up-to-date Oracle Solaris version support information.

Steps

1. Install the Oracle Solaris operating system.
   The Oracle 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 Oracle Virtual Desktop Infrastructure 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 Oracle Virtual Desktop Infrastructure. Create a dedicated pool to separate the Oracle VDI Core 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 Oracle 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 Oracle Virtual Desktop Infrastructure 3.2.2 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.
   Update the Sun Storage 7000 software to get important performance enhancements. Refer to the Oracle Virtual Desktop Infrastructure 3.2.2 Release Notes to make sure the Sun Unified Storage System Software version is supported with your configuration.

3. Create a project.
   Project names must be unique. Using a project name that exists on both heads of a Sun Storage 7000 cluster will lead to application failure in case of cluster failover.

About Dedicated iSCSI Networking

Oracle Virtual Desktop Infrastructure creates several types of traffic, which are roughly sorted by bandwidth requirements:

- iSCSI traffic between the storage and virtualization hosts
- RDP traffic between the virtualization hosts and Oracle VDI Core servers
- ALP traffic between the Oracle VDI Core servers and Sun Rays clients
- Database traffic between the MySQL nodes on the Oracle VDI Core servers or, in the case of an external database, the traffic between the Oracle VDI Core servers and the external database
- SSH/HTTPS traffic between the Oracle VDI Core servers on one side and the storages as well as the virtualization hosts on the other side

The best practice is to use trunking (aggregation) for the virtualization hosts as well as for the storages. This results in a balanced usage of the physical network interfaces as well as in the ability to keep the trunked interface active even if a physical interface goes down.

There may be situations where the iSCSI traffic needs to be separated (security considerations, routing requirements, traffic shaping). To achieve that separation on the storage side, select a storage in Oracle VDI Manager and click the Edit button to open the Edit Storage wizard. This wizard contains a page with an iSCSI Interface menu of all available interfaces of the storage. The default setting is the interface selected in the wizard, which means the management traffic as well as the iSCSI traffic are served by the same interface by default. By changing the interface, you instruct the VMs on the virtualization hosts to mount their virtual disks via the selected interface, resulting in separated traffic.

A dedicated interface must be selected in the Oracle VDI Manager before the first VDI LUN is created on the storage.

About OpenStorage Clustering

Oracle Virtual Desktop Infrastructure 3.2 supports active/passive as well as active/active OpenStorage clusters starting with firmware 2010.Q1.0.0. Clustered storage is managed by the Oracle VDI Core the same way as normal storage, simply add the storage via the Add Storage wizard. The following paragraphs provide some insight to the necessary management on the OpenStorage side.

A storage cluster introduces redundancy for the server components of a storage including CPU, memory, main board, network cards, but does not increase the redundancy of the disks or their controllers. That is taken care of by the JBODS and the RAID levels used.

Two Sun Unified Storage 7310’s or 7410’s can be configured to run as a cluster. The two storage servers in the cluster (called ‘heads’) are connected over a special card, the Clustron, which allows the heads to exchange state and configuration information and to detect a failed head.

A resource is a core concept of clustering, and is typically either a network interface or a storage pool. To ensure availability, the resource it is taken over by a head if the other head fails.

The main configuration steps when setting up a cluster are to define the resources, which is performed the same way as in a single setup (Configuration, Storage or Configuration, Network), and to assign a head as the resource owner (Configuration, Cluster).

A cluster is called "active/passive" if only one head owns all resources. A cluster is called "active/active" if both heads own
resources. While the performance of an active/passive cluster does not degrade if one head fails, both heads of an active/active
cluster are actively processing requests during normal operation resulting in a better utilization of the available hardware.

Identical hardware available on both storages can only be used to create one resource owned by one head. For example, if you
configure a 192.168.100.100 interface using the nge0 device and assign head1 as it's owner, head2 will use its nge0 device to take
over the 192.168.100.100 interface in case head1 fails. To be able to do so the nge0 device must be unused on head2.

Another constraint to be noted about clustered interfaces – they must be statically configured, you cannot use DHCP.

A typical setup for two 7310 storages with 4 network devices each and two J4400 JBODS with 24 disks each:

<table>
<thead>
<tr>
<th></th>
<th>Head1</th>
<th>Head2</th>
</tr>
</thead>
<tbody>
<tr>
<td>nge0</td>
<td>owner</td>
<td>-</td>
</tr>
<tr>
<td>nge1</td>
<td>owner</td>
<td>-</td>
</tr>
<tr>
<td>nge2</td>
<td>-</td>
<td>owner</td>
</tr>
<tr>
<td>nge3</td>
<td>-</td>
<td>owner</td>
</tr>
<tr>
<td>J4400</td>
<td>owner</td>
<td>-</td>
</tr>
<tr>
<td>J4400</td>
<td>-</td>
<td>owner</td>
</tr>
</tbody>
</table>

nge0 and 1 as well as nge2 and 3 are typically trunked / aggregated.

For more information about OpenStorage clustering, refer to the Administration Guide:


### About OpenStorage Write Cache

OpenStorage can use Flash write cache, like Logzilla or ZeusiOPS SSD, as well as standard in-memory write cache for iSCSI LUNs. It
is recommended to use in-memory write cache with Oracle Virtual Desktop Infrastructure because it is faster, less expensive, and
puts less load on the CPU than Flash write cache. However, one advantage of Flash write cache is that it is able to maintain
unwritten data even in the case of a power outage.

In-memory write cache is activated by default for every volume used by Oracle Virtual Desktop Infrastructure. If in-memory write
cache is activated and Flash write cache is present in the system, the Flash write cache is not used. If you want to minimize the
risk of data loss in case of a storage failure and decide to use Flash write cache, you need to switch off the in-memory write
cache.

To switch off the in-memory write cache, select a storage in the Oracle VDI Manager, click Edit to open the Edit Storage wizard
and unselect the Cache check box. The change will be applied to newly created desktops for Oracle VDI Hypervisors and to newly
started desktops for Microsoft Hyper-V virtualization platforms.

### 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 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 Oracle VDI 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 Oracle VM VirtualBox host.
Migrate Desktops - Offered only if the Oracle VDI Core 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 Oracle VM VirtualBox host, if available.

A compatible Oracle VM 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. The Oracle VDI Core verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible.

The Oracle VDI Core also checks for valid Oracle VM VirtualBox versions. Migrating from Oracle VM VirtualBox 3.0 to Oracle VM 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 Oracle Virtual Desktop Infrastructure you can clear or suspend running desktops on virtualization hosts and storage servers. Maintenance mode allows administrators to perform typical maintenance tasks on servers (rebooting, upgrading) with minimal impact to users.

It is required that all desktops are powered off when putting a Hyper-V host or storage into maintenance mode. For more information refer to Bug ID 6919755 in the Oracle Virtual Desktop Infrastructure 3.2 Release Notes.

Oracle VM VirtualBox and Microsoft Hyper-V Host Maintenance

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

Desktop providers with more than one Oracle VM 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 the Oracle VDI Core detects 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, the Oracle VDI Core will attempt to shutdown and restart the desktops on other hosts. A compatible Oracle VM VirtualBox host must have identical, or very similar, CPU models. The Oracle VDI Core verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible. The Oracle VDI Core also checks for valid Oracle VM VirtualBox versions.

With Microsoft Hyper-V, the desktops cannot not be migrated to another host. They are suspended and will be restarted on the same host. In the event that the disk number associated with a desktop has changed during the host maintenance period (this may happen if the Hyper-V host is rebooted), the Oracle VDI Core will power off the desktop before it can be restarted, causing any suspend data to be lost.
Oracle 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

Oracle Virtual Desktop Infrastructure provides a mechanism to 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). When the storage server is re-enabled, any desktops suspended as a result of entering maintenance mode will be resumed.

Oracle 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 Sun Unified Storage System

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

Preparation

Configure the Sun Storage 7000 Unified Storage System for replication and replicate it. Replication is a built-in feature, and can be configured from the Sun Unified Storage System UI. The steps below are valid for 2010.Q1 firmware and above.

1. Add the target storage for the replication to the Remote Replication service. Go to Configuration, then Services.
2. Add a replication action to the project. Go to Shares, Projects, <project>, then Replication. The "Include Snapshots" option must be selected.

The ZFS structures are now replicated to the target storage as replication package.

Disaster Recovery

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

1. In the Oracle VDI Manager, 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 maintenance mode.
2. In the Unified Storage System UI, convert the replication package to a local project.
Sever the replication connection of the replication target. Go to Shares, Projects, Replica, <replication package>, then Replication.

3. In the Oracle VDI Manager, enable the new storage server.
   a. Navigate to the Storage tab.
   b. Select the storage server to be replaced and click Replace to activate the Replace Storage wizard. Enter information about the new storage (replication target).
   c. Select the new storage and click Edit to activate the Edit Storage wizard. Enter additional information about the new storage.
   d. Select the new storage and click Enable.

How to Improve Networking Availability and Performance

The sections below describe the necessary steps to configure link aggregation and/or VLANs on Oracle VM VirtualBox hosts. The two features can be used independently or in conjunction with each other by tagging an aggregated link with a VLAN ID.

Link aggregation provides better network throughput and failover capabilities by aggregating two or more network interfaces into a single aggregated interface. In the event the one interface fails, the aggregated interface will continue to work in a degraded mode.

VLANs provide a way to tag and isolate network traffic and can improve performance and security.

Link Aggregation

Solaris provides a mechanism to aggregate (or trunk) one or more network interfaces providing better throughput and failover capabilities. To use link aggregation you will need a switch that supports LACP.

In our example, we will be aggregating devices e1000g0 and e1000g1. You can list available devices on your system using the `dladm` command:

```
# dladm show-dev
  e1000g0   link: up  speed: 1000  Mbps  duplex: full
  e1000g1   link: up  speed: 1000  Mbps  duplex: full
  e1000g2   link: down speed: 0    Mbps  duplex: half
  e1000g3   link: down speed: 0    Mbps  duplex: half
```

Interfaces e1000g0 and e1000g1 are connected to ports 0 and 1 on our switch respectively.

Steps

1. Identify the switch ports that each network interface in the aggregation will use.

   In the example, ports 0 and 1 are used. Configure the switch to use aggregation (LACP) on ports 0 and 1. Consult the switch’s documentation for instructions on how to do this.

2. Create the aggregation.

   Consult the `dladm` man page for more information on the parameters below. The policy (-P L3) must match the policy you configured for the switch ports. The last parameter, “1”, indicates the aggregation key.

```
# dladm create-aggr -P L3 -l active -T short -d e1000g0 -d e1000g1 1
```

You can view the aggregated device with `dladm show-link` and `dladm show-aggr`. 
# dladm show-link

e1000g0        type: non-vlan  mtu: 1500       device: e1000g0

e1000g1        type: non-vlan  mtu: 1500       device: e1000g1

e1000g2        type: non-vlan  mtu: 1500       device: e1000g2

e1000g3        type: non-vlan  mtu: 1500       device: e1000g3

aggr1          type: non-vlan  mtu: 1500       aggregation: key 1

# dladm show-aggr

e1000g0        address: 0:14:4f:40:d2:4a (auto) speed: 0 Mbps  duplex: half  link: down
standby

e1000g1        address: 80:9c:4c:0:80:fe speed: 0 Mbps  duplex: half  link: down
standby

3. To make the device persistent, create a hostname file with the IP address that should be assigned to the device and reboot.

    # echo "192.168.1.101" > /etc/hostname.aggr1
    # reboot -- -r

4. After the system is rebooted, verify that the device is plumbed and available.

    # ifconfig -a

5. If this is an existing Oracle VM VirtualBox host, refresh the networks in the Oracle VDI Manager.

    Navigate to the Oracle VDI desktop provider's Network tab and click Refresh. If you have more than one network/subnet ensure that the correct network is selected in the Settings tab for each pool.

    For further information, refer to the Solaris 10 documentation.

VLANs

VLANs provide a way to tag and isolate network traffic and can improve performance and security. Either a physical network interface or an aggregation can be tagged with a VLAN ID.

Solaris currently supports the following interface types for VLANs: ce, bge, xge, e1000g.

Steps

1. Configure the switch ports used by the interfaces in the machine for the corresponding VLAN IDs (VIDs). Consult your switch documentation for instructions on doing this.

2. Calculate the physical point of attachment (PPA).

    Each VLAN interface has a physical point of attachment (PPA) which needs to be calculated using the following formula: driver-name + VID * 1000 + device-instance.

    To calculate the PPA for e1000g0:
    
    driver-name = e1000g
    VID = 123
    device-instance = 0

    e1000g + 123 * 1000 + 0 = e1000g123000
To calculate the PPA for aggr1:
\[ \text{driver-name} = \text{aggr} \]
\[ \text{VID} = 123 \]
\[ \text{device-instance} = 1 \]
\[ \text{aggr} + 123 \times 1000 + 1 = \text{aggr123001} \]

3. With the PPA at hand, plumb the interface.

```bash
# ifconfig e1000g123000 plumb 192.168.1.101 up
```

4. Make the changes persistent.

```bash
# echo "192.168.1.101" > /etc/hostname.e1000g123000
# ifconfig -a
```

5. If this is an existing Oracle VM VirtualBox host, refresh the networks in the Oracle VDI Manager. Navigate to the Oracle VDI desktop provider’s Network tab and click Refresh. If you have more than one network/subnet ensure that the correct network is selected in the Settings tab for each pool.

For further information, refer to the Solaris 10 documentation.

---

**Contents**

- How to Set Up a VMware ESX Server
- How to Set Up a VMware vCenter Server
- 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 the VMware ESX Server 3 and VirtualCenter Installation Guide for more details about installing VMware ESX Server.

3. After installation, install the VMware Virtual Infrastructure Client so that you can access the VMware ESX server. Refer the VMware ESX Server 3 and VirtualCenter Installation Guide for more details about installing the VMware Virtual Infrastructure Client.
How to Set Up a VMware vCenter Server

VMware vCenter provides central management of several VMware ESX servers, and can be installed on a physical or virtual host. Refer to 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, then 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 more information about installing the Microsoft Sysprep Tools, see VMware Basic System Administration Guide.

4. Verify that the server is configured for access to the web services API.
   Oracle Virtual Desktop Infrastructure 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).

How to Test the Platform Setup (VMware vCenter)

It is highly recommended to test the virtualization platform configuration, before setting up the Oracle VDI Core 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 virtual machine. 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, then 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.

---

Microsoft Hyper-V Virtualization Platform Installation (All Topics)

About Microsoft Hyper-V Virtualization Platforms

Oracle VDI 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 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.

⚠️ Errors will be seen if the Hyper-V host is running a locale other than English. For more information refer to Bug ID 6918323 in the VDI 3.2 Release Notes.
How to Prepare a Windows Server

After installing Microsoft Hyper-V or Microsoft Remote Desktop Services you must prepare your Windows server to communicate with the Oracle VDI Core. Oracle Virtual Desktop Infrastructure does not require any agents to be installed on the Windows servers, instead the Oracle 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 Oracle 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).

**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:

   ```
   C:\Program Files\IIS Resources\SelfSSL\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 Oracle VDI Core server are not in time sync, you might not be able to connect the Oracle VDI Core to the server because 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 Oracle VDI Core to send requests.
   a. Install WS-Man (WinRM).

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 Oracle VDI Core:

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

   Port 443 is the port the Oracle 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 Oracle VDI Core communication with Microsoft Hyper-V or RDS, you must remember to specify this port when adding the host in Oracle VDI Manager.

---

How to Set Up an Oracle Solaris Storage Server

An Oracle 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 Oracle VM VirtualBox through iSCSI. The management of the ZFS volumes is done by Oracle Virtual Desktop Infrastructure and requires SSH root access to the ZFS storage server and a ZFS pool on that server.

Refer to the Oracle Virtual Desktop Infrastructure 3.2.2 Release Notes for the most accurate and up-to-date Oracle Solaris version support information.

Steps

1. Install the Oracle Solaris operating system.

   The Oracle 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 Oracle Virtual Desktop Infrastructure 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 Oracle Virtual Desktop Infrastructure. Create a dedicated pool to separate the Oracle VDI Core 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 Oracle 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 Oracle Virtual Desktop Infrastructure 3.2.2 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.
   Update the Sun Storage 7000 software to get important performance enhancements. Refer to the Oracle Virtual Desktop Infrastructure 3.2.2 Release Notes to make sure the Sun Unified Storage System Software version is supported with your configuration.

3. Create a project.
   Project names must be unique. Using a project name that exists on both heads of a Sun Storage 7000 cluster will lead to application failure in case of cluster failover.

About Dedicated iSCSI Networking

Oracle Virtual Desktop Infrastructure creates several types of traffic, which are roughly sorted by bandwidth requirements:

- iSCSI traffic between the storage and virtualization hosts
- RDP traffic between the virtualization hosts and Oracle VDI Core servers
- ALP traffic between the Oracle VDI Core servers and Sun Rays clients
- Database traffic between the MySQL nodes on the Oracle VDI Core servers or, in the case of an external database, the traffic between the Oracle VDI Core servers and the external database
- SSH/HTTPS traffic between the Oracle VDI Core servers on one side and the storages as well as the virtualization hosts on the other side

The best practice is to use trunking (aggregation) for the virtualization hosts as well as for the storages. This results in a balanced usage of the physical network interfaces as well as in the ability to keep the trunked interface active even if a physical interface goes down.

There may be situations where the iSCSI traffic needs to be separated (security considerations, routing requirements, traffic shaping). To achieve that separation on the storage side, select a storage in Oracle VDI Manager and click the Edit button to open the Edit Storage wizard. This wizard contains a page with an iSCSI Interface menu of all available interfaces of the storage. The default setting is the interface selected in the Add Storage wizard, which means the management traffic as well as the iSCSI traffic are served by the same interface by default. By changing the interface, you instruct the VMs on the virtualization hosts to mount their virtual disks via the selected interface, resulting in separated traffic.
A dedicated interface must be selected in the Oracle VDI Manager before the first VDI LUN is created on the storage.

About OpenStorage Clustering

Oracle Virtual Desktop Infrastructure 3.2 supports active/passive as well as active/active OpenStorage clusters starting with firmware 2010.Q1.0.0. Clustered storage is managed by the Oracle VDI Core the same way as normal storage, simply add the storage via the Add Storage wizard. The following paragraphs provide some insight to the necessary management on the OpenStorage side.

A storage cluster introduces redundancy for the server components of a storage including CPU, memory, main board, network cards, but does not increase the redundancy of the disks or their controllers. That is taken care of by the JBODS and the RAID levels used.

Two Sun Unified Storage 7310's or 7410's can be configured to run as a cluster. The two storage servers in the cluster (called 'heads') are connected over a special card, the Clustron, which allows the heads to exchange state and configuration information and to detect a failed head.

A resource is a core concept of clustering, and is typically either a network interface or a storage pool. To ensure availability, the resource it is taken over by a head if the other head fails.

The main configuration steps when setting up a cluster are to define the resources, which is performed the same way as in a single setup (Configuration, Storage or Configuration, Network), and to assign a head as the resource owner (Configuration, Cluster).

A cluster is called "active/passive" if only one head owns all resources. A cluster is called "active/active" if both heads own resources. While the performance of an active/passive cluster does not degrade if one head fails, both heads of an active/active cluster are actively processing requests during normal operation resulting in a better utilization of the available hardware.

Identical hardware available on both storages can only be used to create one resource owned by one head. For example, if you configure a 192.168.100.100 interface using the nge0 device and assign head1 as it's owner, head2 will use its nge0 device to take over the 192.168.100.100 interface in case head1 fails. To be able to do so the nge0 device must be unused on head2.

Another constraint to be noted about clustered interfaces – they must be statically configured, you cannot use DHCP.

A typical setup for two 7310 storages with 4 network devices each and two J4400 JBODS with 24 disks each:

<table>
<thead>
<tr>
<th></th>
<th>Head1</th>
<th>Head2</th>
</tr>
</thead>
<tbody>
<tr>
<td>nge0</td>
<td>owner</td>
<td>-</td>
</tr>
<tr>
<td>nge1</td>
<td>owner</td>
<td>-</td>
</tr>
<tr>
<td>nge2</td>
<td>-</td>
<td>owner</td>
</tr>
<tr>
<td>nge3</td>
<td>-</td>
<td>owner</td>
</tr>
<tr>
<td>J4400</td>
<td>owner</td>
<td>-</td>
</tr>
<tr>
<td>J4400</td>
<td>-</td>
<td>owner</td>
</tr>
</tbody>
</table>

nge0 and 1 as well as nge2 and 3 are typically trunked / aggregated.

For more information about OpenStorage clustering, refer to the Adminstration Guide:


About OpenStorage Write Cache
OpenStorage can use Flash write cache, like Logzilla or ZeusiOPS SSD, as well as standard in-memory write cache for iSCSI LUNs. It is recommended to use in-memory write cache with Oracle Virtual Desktop Infrastructure because it is faster, less expensive, and puts less load on the CPU than Flash write cache. However, one advantage of Flash write cache is that it is able to maintain unwritten data even in the case of a power outage.

In-memory write cache is activated by default for every volume used by Oracle Virtual Desktop Infrastructure. If in-memory write cache is activated and Flash write cache is present in the system, the Flash write cache is not used. If you want to minimize the risk of data loss in case of a storage failure and decide to use Flash write cache, you need to switch off the in-memory write cache.

To switch off the in-memory write cache, select a storage in the Oracle VDI Manager, click Edit to open the Edit Storage wizard and unselect the Cache check box. The change will be applied to newly created desktops for Oracle VDI Hypervisors and to newly started desktops for Microsoft Hyper-V virtualization platforms.

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 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 Oracle VDI 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 Oracle VM VirtualBox host.
- Migrate Desktops - Offered only if the Oracle VDI Core 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 Oracle VM VirtualBox host, if available.

A compatible Oracle VM 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. The Oracle VDI Core verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible.

The Oracle VDI Core also checks for valid Oracle VM VirtualBox versions. Migrating from Oracle VM VirtualBox 3.0 to Oracle VM 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 Oracle Virtual Desktop Infrastructure you can clear or suspend running desktops on virtualization hosts and storage servers. Maintenance mode allows administrators to perform typical maintenance tasks on servers (rebooting, upgrading) with minimal impact to users.

It is required that all desktops are powered off when putting a Hyper-V host or storage into maintenance mode. For more information refer to Bug ID 6919755 in the Oracle Virtual Desktop Infrastructure 3.2 Release Notes.

Oracle VM VirtualBox and Microsoft Hyper-V Host Maintenance

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

Desktop providers with more than one Oracle VM 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 the Oracle VDI Core detects 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, the Oracle VDI Core will attempt to shutdown and restart the desktops on other hosts. A compatible Oracle VM VirtualBox host must have identical, or very similar, CPU models. The Oracle VDI Core verifies that hosts have the correct CPU manufacturer. The administrator is responsible for ensuring that the CPU models are compatible. The Oracle VDI Core also checks for valid Oracle VM VirtualBox versions.

With Microsoft Hyper-V, the desktops cannot not be migrated to another host. They are suspended and will be restarted on the same host. In the event that the disk number associated with a desktop has changed during the host maintenance period (this may happen if the Hyper-V host is rebooted), the Oracle VDI Core will power off the desktop before it can be restarted, causing any suspend data to be lost.

Oracle 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

Oracle Virtual Desktop Infrastructure provides a mechanism to 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). When the storage server is re-enabled, any desktops suspended as a result of entering maintenance mode will be resumed.

Oracle 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 Sun Unified Storage System

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

Preparation
Configure the Sun Storage 7000 Unified Storage System for replication and replicate it. Replication is a built-in feature, and can be configured from the Sun Unified Storage System UI. The steps below are valid for 2010.Q1 firmware and above.

1. Add the target storage for the replication to the Remote Replication service. Go to Configuration, then Services.
2. Add a replication action to the project. Go to Shares, Projects, <project>, then Replication. The "Include Snapshots" option must be selected.

The ZFS structures are now replicated to the target storage as replication package.

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

1. In the Oracle VDI Manager, 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 maintenance mode.

2. In the Unified Storage System UI, convert the replication package to a local project.
   Sever the replication connection of the replication target. Go to Shares, Projects, Replica, <replication package>, then Replication.

3. In the Oracle VDI Manager, enable the new storage server.
   a. Navigate to the Storage tab.
   b. Select the storage server to be replaced and click Replace to activate the Replace Storage wizard. Enter information about the new storage (replication target).
   c. Select the new storage and click Edit to activate the Edit Storage wizard. Enter additional information about the new storage.
   d. Select the new storage and click Enable.

Contents

- About Microsoft Remote Desktop Platforms
- How to Install Microsoft Remote Desktop Services
- How to Prepare a Windows Server

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).

Oracle VDI provides users access to Terminal Services or Remote Desktop Services sessions provided by Windows Server 2003 and Windows Server 2008. Oracle 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

Oracle 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

Oracle 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, Oracle 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 Oracle VDI Manager or CLI. By querying the first RDS host of the provider, Oracle 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 Oracle 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 Oracle 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. Oracle 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 Oracle 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 Oracle 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


How to Prepare a Windows Server

After installing Microsoft Hyper-V or Microsoft Remote Desktop Services you must prepare your Windows server to communicate with the Oracle VDI Core. Oracle Virtual Desktop Infrastructure does not require any agents to be installed on the Windows servers, instead the Oracle 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 Oracle 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 Oracle 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:

   ```
   C:\Program Files\IIS Resources\SelfSSL\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 Oracle VDI Core server are not in time sync, you might not be able to connect the Oracle VDI Core to the server because 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 Oracle VDI Core to send requests.

   a. Install WS-Man (WinRM).


   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:

   ```plaintext
   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 Oracle VDI Core:

   ```plaintext
   netsh firewall add portopening TCP 443 "Oracle VDI Remote Management"
   ```

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

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

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

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

Contents

- About Oracle VDI Core Configuration
  - Single Oracle VDI Core Host Configuration
  - High Availability (HA) Configurations
- How to Install and Configure a Single Oracle VDI Core Host
  1. Install and configure the remote MySQL database on the single Oracle VDI Core host.
  2. Create a privileged database administrator.
  3. Install and configure the Oracle VDI Core on the single Oracle VDI Core host.
- How to Install and Configure a Remote MySQL Database (InnoDB)
- How to Install and Configure a Remote MySQL Database (InnoDB)
Single Oracle VDI Core Host Configuration (All Topics)

About Oracle VDI Core Configuration

There are many possible configurations for the Oracle 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 Oracle VDI Core configurations, refer to the Configurations page.

The Oracle VDI Core installation and configuration information focuses on three supported Oracle VDI Core configuration types: Single Oracle VDI Core Host, High Availability (MySQL Cluster), and High Availability (Remote MySQL). The Single Oracle VDI Core Host configuration is simple to set up and requires minimal hardware while the High Availability configurations provide failover capabilities and better performance.

Single Oracle VDI Core Host Configuration

The Single Oracle VDI Core 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 Oracle VDI Core Host configuration is similar to a Demo (Oracle VDI Hypervisor) configuration because all necessary components can be installed on one machine, including an Oracle VDI Hypervisor, Oracle VDI Core (with MySQL database and Sun Ray Software). The difference between a Single Oracle VDI Core Host configuration and a Demo (Oracle VDI Hypervisor) configuration is that the Demo uses the bundled MySQL Cluster database, and the Single Oracle VDI Core Host configuration requires a MySQL "remote" database. The MySQL database can be installed on the Single Oracle VDI Core 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 Oracle VDI Core.
If the MySQL remote database is installed on the same machine as the Oracle VDI Core, the database must have an InnoDB storage engine to be considered a supported configuration. This Single Oracle VDI Core 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 Oracle 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 Oracle VDI Core to create and configure the VDA database during Oracle VDI Core configuration. After the VDA database is created, it will be accessed by the Oracle VDI Core through the database user (default ‘vdadb’).

Suggested Pages

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

- How to Install and Configure a Single Oracle VDI Core Host - Describes all necessary steps for setting up the Oracle VDI Core on one machine, including How to Install and Configure a Remote MySQL Database (InnoDB), How to Create a Privileged Database Administrator, and how to install the Oracle VDI Core.
How to Check the Oracle VDI Core Services and Logs - Describes how to check the services and statuses of the Oracle 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 desktop sessions with only a minimal interruption to the user. High availability configurations are suggested for use cases where fail-safety is prioritized over low price.

The Oracle VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one Oracle VDI Core host fails, all users with desktop sessions on that host are kicked back to the Oracle 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 Oracle Virtual Desktop Infrastructure stack requires a minimum of three oracle VDI Core hosts to be considered fail-proof. This number of hosts does not include the virtualization platform hosts, which should be considered for failover separately.

Oracle Virtual Desktop Infrastructure also supports the option to connect to a remote MySQL database, instead of the bundled MySQL cluster database. In this case, the Oracle 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.

Oracle VDI Core (MySQL Cluster)

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

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 (MySQL Cluster) Configuration

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

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 Oracle VDI Core (MySQL Cluster)** - Describes all necessary steps for setting up the Oracle 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-Oracle VDI SQL nodes from joining the bundled MySQL Cluster.
- **How to Check the Oracle VDI Core Services and Logs** - Describes how to check the services and statuses of the Oracle VDI Core, bundled MySQL database, and RDP broker.

**Oracle 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.
Before the Oracle 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 Oracle VDI Core to create and configure the VDA database during Oracle VDI Core configuration. After the VDA database is created, it will be accessed by the Oracle VDI Core through the database user (default 'vdadb').

Because a remote database is used instead of the bundled MySQL Cluster, the Oracle 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 Oracle VDI Core (Remote MySQL)** - Describes all necessary steps for setting up the Oracle 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 your Oracle Virtual Desktop Infrastructure.

- **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 Oracle VDI Core Services and Logs** - Describes how to check the services and statuses of the Oracle VDI Core, bundled MySQL database, and RDP broker.
How to Install and Configure a Single Oracle VDI Core Host

The Single Oracle VDI Core Host Configuration should only be used in production environments where fail-over is not required.

Before You Begin

- Install the Oracle VDI Hypervisor.
  1. Set up the Oracle Solaris storage. Refer to the How to Set Up an Oracle Solaris Storage Server page.

Although the Oracle VDI Core is supported on Solaris 10 5/09 (Update 7) and above, ZFS storage is only supported on Solaris 10 10/09 (Update 8) and above. Therefore, for Single Host configurations using ZFS storage, the complete installation will require Solaris 10 10/09 (Update 8) or above.

When Oracle VM VirtualBox and the Oracle VDI Core are installed on the same machine, ZFS uses any memory available (up to the limit) for a the ARC cache. This can cause issues, because the Oracle VDI Core may falsely report that there is not enough memory for starting up any virtual machine.

To resolve this issue, adapt the max value used for the ARC cache. For example, to restrict the memory to 2GB, add the following line in `/etc/system`:

```
set zfs:zfs_arc_max = 2147483648
```

2. Install Oracle VM VirtualBox. Refer to the How to Install Oracle VM VirtualBox page.

Back up!
If you are reinstalling an existing Oracle VDI Core host, make sure all the existing data is backed up and available for restore. Use the following checklist to be sure all important data has been backed up properly.

- Database - You must backup the database before starting the reinstall process. This will enable you to restore the current system once you finish the reinstall. For more information about backing up the Oracle VDI Core database, refer to How to Back Up and Restore the VDI Core Database.
- Customized `pam.conf` files - The reinstall creates a new `/etc/pam.conf` file (an SRSS access configuration file). If you have customized this file, you will need to back it up before reinstalling, and re-add the customization to the new file.

1. Install and configure the remote MySQL database on the single Oracle
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.

```ini
[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.
5. Make sure that the `/usr/local` directory contains the proper owner and group permissions by running the following commands.

   ```bash
   # 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.

   ```bash
   # 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.

   ```bash
   # 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       1024 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.

   ```bash
   # ./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!


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 --basedir=/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;
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'<localhost DNS name>' 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 Oracle VDI Core on the single Oracle VDI Core host.

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

   ```bash
   # unzip vda_3.2_amd64.zip
   # cd vda_3.2_amd64
   ```

2. Run the installation.

   ```bash
   # ./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 the Oracle VDI Core components are installed.

   After accepting the license agreement, the installation process begins, and all Oracle VDI Core components are installed.

3. After successful installation reboot your machine.

   ```bash
   # reboot
   ```

4. As root, run the configuration.

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

   Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Choose the 3 Single Oracle 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 Oracle 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 Oracle VDI Database. Alternatively you can simply accept the default 'vdadb'.
   h. Specify a password for the Oracle 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 Oracle 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.
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
```

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!


From the directory, run the following command, and check to see that you get the corresponding output.

```
# ./bin/mysqld_safe --defaults-file=/etc/my.cnf --ldir=/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
```

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;
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'<localhost DNS name>' 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 the VDI Core Database

As with all user-level data, it is important to back up the VDI Core database periodically. This is also a crucial step if you plan to reinstall an Oracle VDI Core host.

The following information should be used when backing up data for both MySQL Cluster and Remote MySQL database types. To learn more about Oracle VDI Core configurations and the corresponding databases, refer to the About Oracle VDI Core Configuration page.

This backup task archives only the content of the VDI Core database. The volumes of desktops and templates as well as configuration and settings values are not backed up.

Before You Begin

Here is a list of important notes when backing up and restoring the VDI Core database.

- To back up and restore the Sun VDI 3.1 and 3.1.1 databases, refer to How to Back Up and Restore Data (Bundled MySQL Database) and How to Back Up and Restore Data (Remote MySQL Database).
- For multi-host setups, the backup and restore process should only be done on one host. It does not need to be done on every host.
- You can perform the backup and restore tasks on different hosts as long as the archived backup is accessible.
- You can perform the backup and restore tasks on the primary host if you have configured it to host VDI sessions (you ran the vda-config command twice).
- The backup job stops all other VDI jobs to run. Jobs are automatically started again after the backup finishes.
- The restore job stops the entire VDI system to run, but active sessions will continue to run. When a restore job finishes, you must restart the VDI system (through the Common Agent Container (cacao)) on all secondary hosts and the primary host if you have configured it to host VDI sessions.

Steps

1. Make a backup of the database.
   - From the CLI, call the backup command.
     ```bash
     # /opt/SUNWvda/sbin/vda-backup -p </path/to/directory> -o <output-file-name>
     ```
     While the backup job is running, all other jobs are stopped or put in queue in all hosts. A backup archive is created, that includes one file, with an .db extension and a timestamp-based name. For a detailed view of the backup command syntax use the following command.
     ```bash
     # /opt/SUNWvda/sbin/vda-backup -h
     ```
   - From the Oracle VDI Manager:
     a. Select the Settings category, then the VDI Center subcategory.
     b. Select the Summary tab, and click Backup in the VDI Hosts table.

2. Restore the backed up database on the new Oracle Virtual Desktop Infrastructure installation.
   - From the CLI, call the restore command.
     ```bash
     # /opt/SUNWvda/sbin/vda-restore -i </path/to/backup.zip>
     ```
     For a detailed view of the restore command syntax use the following command.
     ```bash
     # /opt/SUNWvda/sbin/vda-restore -h
     ```

3. After the restore job finishes, restart the VDI system on all secondary hosts and the primary host if you have configured it
Oracle VDI 3.2 Installation and Configuration Guide

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
```
When restarting the Common Agent Container not only will the `vda.service.module` be restarted, but all other modules that are registered in the container. This can have undesirable side effects on your system.

How to Check the Status of the Oracle VDI Core Service Module

The Oracle VDI Core Service runs within the Common Agent Container.

- As root, run the following command.

```bash
# 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 Oracle 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.

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

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

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

- Restart Cacao after changing the logging level.

How to Check the Status of the Database

Oracle Virtual Desktop Infrastructure 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.

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

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

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

The Oracle 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 (Oracle 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 Oracle Virtual Desktop Infrastructure 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:
High Availability with MySQL Cluster Configuration (All Topics)

About Oracle VDI Core Configuration

There are many possible configurations for the Oracle 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 Oracle VDI Core configurations, refer to the Configurations page.

The Oracle VDI Core installation and configuration information focuses on three supported Oracle VDI Core configuration types: Single Oracle VDI Core Host, High Availability (MySQL Cluster), and High Availability (Remote MySQL). The Single Oracle VDI Core Host configuration is simple to set up and requires minimal hardware while the High Availability configurations provide failover capabilities and better performance.

Single Oracle VDI Core Host Configuration

The Single Oracle VDI Core 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 Oracle VDI Core Host configuration is similar to a Demo (Oracle VDI Hypervisor) configuration because all necessary components can be installed on one machine, including an Oracle VDI Hypervisor, Oracle VDI Core (with MySQL database and Sun Ray Software). The difference between a Single Oracle VDI Core Host configuration and a Demo (Oracle VDI Hypervisor) configuration is that the Demo uses the bundled MySQL Cluster database, and the Single Oracle VDI Core Host configuration...
requires a MySQL "remote" database. The MySQL database can be installed on the Single Oracle VDI Core 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 Oracle VDI Core.

If the MySQL remote database is installed on the same machine as the Oracle VDI Core, the database must have an InnoDB storage engine to be considered a supported configuration. This Single Oracle VDI Core 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 Oracle 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 Oracle VDI Core to create and configure the VDA database during Oracle VDI Core configuration. After the VDA database is created, it will be accessed by the Oracle VDI Core through the database user (default 'vdadb').
• **How to Install and Configure a Single Oracle VDI Core Host** - Describes all necessary steps for setting up the Oracle VDI Core on one machine, including How to Install and Configure a Remote MySQL Database (InnoDB), How to Create a Privileged Database Administrator, and how to install the Oracle VDI Core.

• **How to Check the Oracle VDI Core Services and Logs** - Describes how to check the services and statuses of the Oracle 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 desktop sessions with only a minimal interruption to the user. High availability configurations are suggested for use cases where fail-safety is prioritized over low price.

The Oracle VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one Oracle VDI Core host fails, all users with desktop sessions on that host are kicked back to the Oracle 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 Oracle Virtual Desktop Infrastructure stack requires a minimum of three oracle VDI Core hosts to be considered fail-proof. This number of hosts does not include the virtualization platform hosts, which should be considered for failover separately.

Oracle Virtual Desktop Infrastructure also supports the option to connect to a remote MySQL database, instead of the bundled MySQL cluster database. In this case, the Oracle 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.

Oracle VDI Core (MySQL Cluster)

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

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

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.

**Oracle 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.
Before the Oracle 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 Oracle VDI Core to create and configure the VDA database during Oracle VDI Core configuration. After the VDA database is created, it will be accessed by the Oracle VDI Core through the database user (default 'vdadb').

Because a remote database is used instead of the bundled MySQL Cluster, the Oracle 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 Oracle VDI Core (Remote MySQL)** - Describes all necessary steps for setting up the Oracle 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 your Oracle Virtual Desktop Infrastructure.

- **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 Oracle VDI Core Services and Logs** - Describes how to check the services and statuses of the Oracle VDI Core, bundled MySQL database, and RDP broker.
How to Install and Configure the Oracle VDI Core (MySQL Cluster)

To configure the Oracle VDI Core 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.

Before you Begin

- **Back up!**
  If you are reinstalling an existing Oracle VDI Core host, make sure all the existing data is backed up and available for restore. Use the following checklist to be sure all important data has been backed up properly.

  - **Database** - You must backup the database before starting the reinstall process. This will enable you to restore the current system once you finish the reinstall. For more information about backing up the Oracle VDI Core database, refer to [How to Back Up and Restore the VDI Core Database](#).
  - **Customized my.cnf files** - The reinstall creates a new `/etc/opt/SUNWvda/my.cnf` file (a database configuration file). If you have customized this file, you will need to back it up before reinstalling, and re-add the customization to the new file.
  - **Customized pam.conf files** - The reinstall creates a new `/etc/pam.conf` file (an SRSS access configuration file). If you have customized this file, you will need to back it up before reinstalling, and re-add the customization to the new file.

1. Install and configure the Oracle VDI Core on the 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.
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.

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Choose the `1 Primary Oracle 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 Oracle VDI 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 Oracle VDI Core on the first 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_3.2_amd64.zip
# cd vda_3.2_amd64
```
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.

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Select the 2 Secondary Oracle 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 Oracle 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 Oracle VDI Core on the second 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_3.2_amd64.zip
# cd vda_3.2_amd64
```

2. Run the installation.
2. Run the installation.

```bash
# ./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.

3. After successful installation reboot your machine.

```bash
# reboot
```

4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Select the 2 Secondary Oracle 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 Oracle 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 Prepare a VDI Primary Host

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

```bash
# unzip vda_3.2_amd64.zip
# cd vda_3.2_amd64
```

2. Run the installation.

```bash
# ./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.

3. After successful installation reboot your machine.

```bash
# reboot
```

4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Choose the 1 Primary Oracle 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 Oracle VDI 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.

```bash
# unzip vda_3.2_amd64.zip
# cd vda_3.2_amd64
```

2. Run the installation.

```bash
# ./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.

3. After successful installation reboot your machine.

```bash
# reboot
```
4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Select the 2 Secondary Oracle 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 Oracle 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 Configure Additional Oracle VDI Core Secondary Hosts**

By default, an Oracle Virtual Desktop Infrastructure (MySQL Cluster) configuration will allow up to 20 SQL nodes to join the MySQL Cluster. SQL nodes are only present in the Oracle VDI Core Secondary hosts.

The procedure for configuring additional Oracle VDI Core Secondary hosts varies depending on the state of the Cluster.

**Configuration for Oracle VDI Core Secondary Hosts 1 to 20**

Configuring Oracle VDI Core Secondary hosts 1 and 2 is a mandatory part of every Oracle Virtual Desktop Infrastructure (MySQL Cluster) configuration. Additional Secondary hosts can be configured as needed. For more details refer to the How to Prepare a VDI Secondary Host page.
Configuration for Oracle VDI Core Secondary Hosts 21+ Before Oracle VDI Core Configuration

If you plan to configure more than 20 Oracle VDI Core Secondary hosts before you have started Oracle VDI Core configuration, add more [MYSQLD] sections to the end of the file /etc/opt/SUNWvda/config.clustered.ini.

Configuration for Oracle VDI Core Secondary Hosts 21+ After Oracle VDI Core Configuration

If you would like to configure more than 20 Oracle VDI Core Secondary hosts after you have already configured the MySQL Cluster, add more [MYSQLD] sections to the end of the file /etc/opt/SUNWvda/config.ini. Then perform a rolling restart of the MySQL Cluster. For more details about rolling restarts, refer to the How to Perform a Rolling Restart of the MySQL Cluster page.

How to Back Up and Restore the VDI Core Database

As with all user-level data, it is important to back up the VDI Core database periodically. This is also a crucial step if you plan to reinstall an Oracle VDI Core host.

The following information should be used when backing up data for both MySQL Cluster and Remote MySQL database types. To learn more about Oracle VDI Core configurations and the corresponding databases, refer to the About Oracle VDI Core Configuration page.

This backup task archives only the content of the VDI Core database. The volumes of desktops and templates as well as configuration and settings values are not backed up.

Before You Begin

Here is a list of important notes when backing up and restoring the VDI Core database.

• To back up and restore the Sun VDI 3.1 and 3.1.1 databases, refer to How to Back Up and Restore Data (Bundled MySQL Database) and How to Back Up and Restore Data (Remote MySQL Database).

• For multi-host setups, the backup and restore process should only be done on one host. It does not need to be done on every host.

• You can perform the backup and restore tasks on different hosts as long as the archived backup is accessible.

• You can perform the backup and restore tasks on the primary host if you have configured it to host VDI sessions (you ran the vda-config command twice).

• The backup job stops all other VDI jobs to run. Jobs are automatically started again after the backup finishes.

• The restore job stops the entire VDI system to run, but active sessions will continue to run. When a restore job finishes, you must restart the VDI system (through the Common Agent Container (cacao)) on all secondary hosts and the primary host if you have configured it to host VDI sessions.

Steps

1. Make a backup of the database.
   • From the CLI, call the backup command.

\[
\texttt{# /opt/SUNWvda/sbin/vda-backup -p </path/to/directory> -o <output-file-name>}
\]

While the backup job is running, all other jobs are stopped or put in queue in all hosts. A backup archive is created, that includes one file, with an .db extension and a timestamp-based name.

For a detailed view of the backup command syntax use the following command.

\[
\texttt{# /opt/SUNWvda/sbin/vda-backup -h}
\]

• From the Oracle VDI Manager:
a. Select the Settings category, then the VDI Center subcategory.
b. Select the Summary tab, and click Backup in the VDI Hosts table.

2. Restore the backed up database on the new Oracle Virtual Desktop Infrastructure installation.
   • From the CLI, call the restore command.

   ```
   # /opt/SUNWvda/sbin/vda-restore -i </path/to/backup.zip>
   ```

   For a detailed view of the restore command syntax use the following command.

   ```
   # /opt/SUNWvda/sbin/vda-restore -h
   ```

3. After the restore job finishes, restart the VDI system on all secondary hosts and the primary host if you have configured it to host VDI sessions.

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

**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 you 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 host 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 host needs 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

<table>
<thead>
<tr>
<th>From/To</th>
<th>Non-VDI host</th>
<th>Primary-Management host</th>
<th>Secondary-Data host</th>
<th>Secondary-SQL host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-VDI host</td>
<td>-</td>
<td>From Non-VDI to Primary-Management host</td>
<td>From Non-VDI to Secondary-Data host</td>
<td>From Non-VDI to Secondary-SQL host</td>
</tr>
<tr>
<td>Primary-Management host</td>
<td>From Primary-Management to Non-VDI host</td>
<td>-</td>
<td>From Primary-Management to Secondary-Data host</td>
<td>From Primary-Management to Secondary-SQL host</td>
</tr>
<tr>
<td>Secondary-Data host</td>
<td>From Secondary-Data to Non-VDI host</td>
<td>From Secondary-Data to Primary-Management host</td>
<td>-</td>
<td>From Secondary-Data to Secondary-SQL host</td>
</tr>
<tr>
<td>Secondary-SQL host</td>
<td>From Secondary-SQL to Non-VDI host</td>
<td>From Secondary-SQL to Primary-Management host</td>
<td>From Secondary-SQL to Secondary-Data host</td>
<td>-</td>
</tr>
</tbody>
</table>

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.
   
   ```
   scvadm 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:

<table>
<thead>
<tr>
<th>STATE</th>
<th>STIME</th>
<th>FMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>Dez_09</td>
<td>svc:/application/database/vdadb:core</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>STATE</th>
<th>STIME</th>
<th>FMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>Dez_09</td>
<td>svc:/application/database/vdadb:sql</td>
</tr>
</tbody>
</table>

On your Primary host stop the service by running the following command.

```bash
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.

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

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

<table>
<thead>
<tr>
<th>STATE</th>
<th>STIME</th>
<th>FMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>Dez_09</td>
<td>svc:/application/database/vdadb:core</td>
</tr>
</tbody>
</table>

On your Primary host:

3. 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 service again by running the following command.

```bash
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.

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

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

<table>
<thead>
<tr>
<th>STATE</th>
<th>STIME</th>
<th>FMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>online</td>
<td>Dez_09</td>
<td>svc:/application/database/vdadb:core</td>
</tr>
</tbody>
</table>

On your remaining "old" Secondary-Data host:

4. Start the data node again by running the following command.

```bash
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.

```bash
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:
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.
   
   ```bash
   /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.
   
   ```bash
   /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.
   
   ```bash
   /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.
   
   ```bash
   /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 Shut Down and Restart an Oracle Virtual Desktop Infrastructure Installation

The following procedure should be used to shut down and restart an Oracle Virtual Desktop Infrastructure installation with a MySQL Cluster database.

Before You Begin

If an Oracle VDI Core service is running on any of the Oracle VDI Core hosts, run the following lines before you shut down the host machine.

```
# /opt/SUNWvda/sbin/vda-webadmin stop
# /opt/SUNWvda/sbin/vda-service stop
```

To check that the services have stopped, run the following commands.

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

Steps

1. Shut down the Oracle Virtual Desktop Infrastructure installation.
   a. Set all hosts and storages to ‘disabled’.
   b. Transfer all desktops to the “Power Off” state. Then wait 1 minute.
   c. Check that there are no jobs running and no desktops running.
   d. Shut down Oracle VDI Core additional secondaries
   e. Shut down Oracle VDI Core Second Secondary. Wait until it is down.
   f. Shut down Oracle VDI Core First Secondary. Wait until it is down.
   g. Shut down Oracle VDI Core Primary.
   h. Shut down the virtualization hosts.
      i. Shut down the storage. If clustered, passive head first.

2. Restart the Oracle Virtual Desktop Infrastructure installation.
   a. Boot the storage. If clustered, active head first.
   b. Wait until you can access the storage's Web UI. If cluster, wait until both heads have established communication.
   c. Boot the virtualization hosts. If Oracle VM VirtualBox hosts, wait until you can browse
      `https://<host>/webservice`.  

d. Boot the Oracle VDI Core Primary host and start the Oracle VDI Core service by running the following.

```bash
# /opt/SUNWvda/sbin/vda-service start
```

e. Boot the Oracle VDI Core First Secondary and start the Oracle VDI Core service.
   Wait until it has been started.

f. Boot the Oracle VDI Core Second Secondary and start the Oracle VDI Core service.
   Wait until it has been started.

g. Boot the Oracle VDI Core Additional Secondaries and rand start the Oracle VDI Core service.

```bash
# /opt/SUNWvda/sbin/vda-service start
```

h. Check that all database nodes have established communication by running the following

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

i. Log in to the Oracle VDI Manager and enable the hosts and storages.

---

**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 svc:/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:
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:

\[
\begin{array}{lll}
\text{STATE} & \text{STIME} & \text{FMRI} \\
\text{online} & \text{Dez}_09 & \text{svc:/application/database/vdadb:core}
\end{array}
\]

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).

\[
\begin{array}{lll}
\text{STATE} & \text{STIME} & \text{FMRI} \\
\text{disabled} & \text{Dez}_09 & \text{svc:/application/database/vdadb:core}
\end{array}
\]

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:

\[
\begin{array}{lll}
\text{STATE} & \text{STIME} & \text{FMRI} \\
\text{online} & \text{Dez}_09 & \text{svc:/application/database/vdadb:core}
\end{array}
\]

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).

\[
\begin{array}{lll}
\text{STATE} & \text{STIME} & \text{FMRI} \\
\text{disabled} & \text{Dez}_09 & \text{svc:/application/database/vdadb:sql}
\end{array}
\]

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).

\[
\begin{array}{lll}
\text{STATE} & \text{STIME} & \text{FMRI} \\
\text{online} & \text{Dez}_09 & \text{svc:/application/database/vdadb:sql}
\end{array}
\]
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 to 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 environments, 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] [MYSQLD] [MYSQLD] [MYSQLD] [MYSQLD] [MYSQLD] [MYSQLD]
HostName=my-1st-secondary 
HostName=my-2nd-secondary 
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 Oracle VDI Core Service Module
- How to Increase or Decrease the Logging Level of the Oracle 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 Oracle VDI Core Services and Logs

It is important to know how to check the status of the various services provided by Oracle Virtual Desktop Infrastructure for troubleshooting or reconfiguration purposes. Most services run under the control of the Solaris Service Management Facility (SMF).

The main Oracle VDI Core service 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 Oracle products. If you encounter any issues, you should first check the status of Cacao as well as the status of the Oracle 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

When restarting the Common Agent Container not only will the vda.service_module be restarted, but all other modules that are registered in the container. This can have undesirable side effects on your system.

How to Check the Status of the Oracle VDI Core Service Module

The Oracle 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 Oracle 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.

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

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

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

- Restart Cacao after changing the logging level.

How to Check the Status of the Database

Oracle Virtual Desktop Infrastructure 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.

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

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

The Oracle 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.

  ```bash
  # 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 Oracle 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 (Oracle 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 Oracle Virtual Desktop Infrastructure 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 Oracle VDI Core Configuration
  - Single Oracle VDI Core Host Configuration
  - High Availability (HA) Configurations
- How to Install and Configure the Oracle VDI Core (Remote MySQL)
  - 1. Install and configure the Oracle VDI Core on the Primary Host.
  - How to Prepare a VDI Primary Host
  - How to Complete the Remote Database Configuration
  - 2. Install and configure the Oracle VDI Core on the Secondary Hosts.
High Availability with Remote MySQL Configuration (All Topics)

About Oracle VDI Core Configuration

There are many possible configurations for the Oracle 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 Oracle VDI Core configurations, refer to the Configurations page.

The Oracle VDI Core installation and configuration information focuses on three supported Oracle VDI Core configuration types: Single Oracle VDI Core Host, High Availability (MySQL Cluster), and High Availability (Remote MySQL). The Single Oracle VDI Core Host configuration is simple to set up and requires minimal hardware while the High Availability configurations provide failover capabilities and better performance.

Single Oracle VDI Core Host Configuration

The Single Oracle VDI Core 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 Oracle VDI Core Host configuration is similar to a Demo (Oracle VDI Hypervisor) configuration because all necessary components can be installed on one machine, including an Oracle VDI Hypervisor, Oracle VDI Core (with MySQL database and Sun Ray Software). The difference between a Single Oracle VDI Core Host configuration and a Demo (Oracle VDI Hypervisor) configuration is that the Demo uses the bundled MySQL Cluster database, and the Single Oracle VDI Core Host configuration requires a MySQL "remote" database. The MySQL database can be installed on the Single Oracle VDI Core 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 Oracle VDI Core.
If the MySQL remote database is installed on the same machine as the Oracle VDI Core, the database must have an InnoDB storage engine to be considered a supported configuration. This Single Oracle VDI Core 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 Oracle 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 Oracle VDI Core to create and configure the VDA database during Oracle VDI Core configuration. After the VDA database is created, it will be accessed by the Oracle VDI Core through the database user (default 'vdadb').

Suggested Pages

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

- How to Install and Configure a Single Oracle VDI Core Host - Describes all necessary steps for setting up the Oracle VDI Core on one machine, including How to Install and Configure a Remote MySQL Database (InnoDB), How to Create a Privileged Database Administrator, and how to install the Oracle VDI Core.
How to Check the Oracle VDI Core Services and Logs - Describes how to check the services and statuses of the Oracle 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 desktop sessions with only a minimal interruption to the user. High availability configurations are suggested for use cases where fail-safety is prioritized over low price.

The Oracle VDI Core and bundled Sun Ray Software require two hosts to be highly available. If one Oracle VDI Core host fails, all users with desktop sessions on that host are kicked back to the Oracle 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 Oracle Virtual Desktop Infrastructure stack requires a minimum of three oracle VDI Core hosts to be considered fail-proof. This number of hosts does not include the virtualization platform hosts, which should be considered for failover separately.

Oracle Virtual Desktop Infrastructure also supports the option to connect to a remote MySQL database, instead of the bundled MySQL cluster database. In this case, the Oracle 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.

Oracle VDI Core (MySQL Cluster)

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

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.

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

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 NDDB 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 Oracle VDI Core (MySQL Cluster)** - Describes all necessary steps for setting up the Oracle 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-Oracle VDI SQL nodes from joining the bundled MySQL Cluster.
- **How to Check the Oracle VDI Core Services and Logs** - Describes how to check the services and statuses of the Oracle VDI Core, bundled MySQL database, and RDP broker.

**Oracle 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 NDDB), or a MySQL Cluster version 6.2.15 or higher.
Before the Oracle 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 Oracle VDI Core to create and configure the VDA database during Oracle VDI Core configuration. After the VDA database is created, it will be accessed by the Oracle VDI Core through the database user (default 'vdadb').

Because a remote database is used instead of the bundled MySQL Cluster, the Oracle 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 Oracle VDI Core (Remote MySQL)** - Describes all necessary steps for setting up the Oracle 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 your Oracle Virtual Desktop Infrastructure.

- **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 Oracle VDI Core Services and Logs** - Describes how to check the services and statuses of the Oracle VDI Core, bundled MySQL database, and RDP broker.
How to Install and Configure the Oracle VDI Core (Remote MySQL)

As an alternative to the MySQL 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.

Before You Begin

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

  For information about how to create a privileged administrator, refer to the How to Create a Privileged Database Administrator page.

  Back up!
  If you are reinstalling an existing Oracle VDI Core host, make sure all the existing data is backed up and available for restore. Use the following checklist to be sure all important data has been backed up properly.

  - Database - You must backup the database before starting the reinstall process. This will enable you to restore the current system once you finish the reinstall. For more information about backing up the Oracle VDI Core database, refer to How to Back Up and Restore the VDI Core Database.
  - Customized `pam.conf` files - The reinstall creates a new `/etc/pam.conf` file (an SRSS access configuration file). If you have customized this file, you will need to back it up before reinstalling, and re-add the customization to the new file.

1. Install and configure the Oracle VDI Core on the 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_3.2_amd64.zip
   # cd vda_3.2_amd64
   ```

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.

3. After successful installation reboot your machine.

   # reboot

4. As root, run the configuration.

   # /opt/SUNWvda/sbin/vda-config

   Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Choose the 1 Primary Oracle 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 Oracle VDI 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 Oracle 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 Oracle VDI Database. Alternatively you can simply accept the default ‘vdadb’.
  8. Specify a password for the Oracle VDI Database User.

2. Install and configure the Oracle VDI Core on the 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_3.2_amd64.zip
# cd vda_3.2_amd64
```

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.

3. After successful installation reboot your machine.

```
# reboot
```

4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Select the 2 Secondary Oracle 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 Oracle 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 Oracle VDI Database which has been specified when configuring your Primary Oracle VDI Core host.
5. Enter the name of the user that has access to the Oracle VDI Database. This is the user you have specified when configuring your Primary Oracle VDI Core host (the default is 'vdadb').
6. Enter the password for your Oracle VDI Database User.

Next Steps

Go to How to Check the Oracle 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_3.2_amd64.zip
# cd vda_3.2_amd64
```

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.

3. After successful installation, reboot your machine.

```
# reboot
```

4. As root, run the configuration.

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

Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Choose the 1 Primary Oracle 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 Oracle VDI 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.

   ```bash
   # unzip vda_3.2_amd64.zip
   # cd vda_3.2_amd64
   ```

2. Run the installation.

   ```bash
   # ./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.

3. After successful installation reboot your machine.

   ```bash
   # reboot
   ```

4. As root, run the configuration.

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

   Refer to the Oracle Virtual Desktop Infrastructure Defaults page for more information about the configuration script.

5. Select the 2 Secondary Oracle 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 Oracle 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.

```plaintext
[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_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.

   ```bash
   # 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.

   ```bash
   # 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.

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

   ```bash
   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.

```bash
# 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.

```bash
# 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.

```bash
# ./scripts/mysql_install_db --user=mysql
```
8. 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/mysqld --user=mysql &
```

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 --ledger=/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;
mysql> GRANT ALL PRIVILEGES ON *.* TO '<user>'@'<localhost DNS name>' 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 the VDI Core Database

As with all user-level data, it is important to back up the VDI Core database periodically. This is also a crucial step if you plan to reinstall an Oracle VDI Core host.

The following information should be used when backing up data for both MySQL Cluster and Remote MySQL database types. To learn more about Oracle VDI Core configurations and the corresponding databases, refer to the About Oracle VDI Core Configuration page.

Before You Begin

Here is a list of important notes when backing up and restoring the VDI Core database.

- To back up and restore the Sun VDI 3.1 and 3.1.1 databases, refer to How to Back Up and Restore Data (Bundled MySQL Database) and How to Back Up and Restore Data (Remote MySQL Database).

- For multi-host setups, the backup and restore process should only be done on one host. It does not need to be done on every host.

- You can perform the backup and restore tasks on different hosts as long as the archived backup is accessible.

- You can perform the backup and restore tasks on the primary host if you have configured it to host VDI sessions (you ran the `vda-config` command twice).

- The backup job stops all other VDI jobs to run. Jobs are automatically started again after the backup finishes.

- The restore job stops the entire VDI system to run, but active sessions will continue to run. When a restore job finishes, you must restart the VDI system (through the Common Agent Container (cacao)) on all secondary hosts and the primary host if you have configured it to host VDI sessions.

Steps

1. Make a backup of the database.
   - From the CLI, call the `backup` command.
     ```bash
     # /opt/SUNWvda/sbin/vda-backup -p </path/to/directory> -o <output-file-name>
     ```
     While the backup job is running, all other jobs are stopped or put in queue in all hosts. A backup archive is created, that includes one file, with an `.db` extension and a timestamp-based name.
     For a detailed view of the `backup` command syntax use the following command.
     ```bash
     # /opt/SUNWvda/sbin/vda-backup -h
     ```
     - From the Oracle VDI Manager:
       a. Select the Settings category, then the VDI Center subcategory.
       b. Select the Summary tab, and click Backup in the VDI Hosts table.

2. Restore the backed up database on the new Oracle Virtual Desktop Infrastructure installation.
   - From the CLI, call the `restore` command.
     ```bash
     # /opt/SUNWvda/sbin/vda-restore -i </path/to/backup.zip>
     ```
     For a detailed view of the `restore` command syntax use the following command.
     ```bash
     # /opt/SUNWvda/sbin/vda-restore -h
     ```

3. After the restore job finishes, restart the VDI system on all secondary hosts and the primary host if you have configured it.
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
```
When restarting the Common Agent Container not only will the `vda.service_module` be restarted, but all other modules that are registered in the container. This can have undesirable side effects on your system.

How to Check the Status of the Oracle VDI Core Service Module

The Oracle 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 Oracle 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

Oracle Virtual Desktop Infrastructure 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 Oracle 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 Oracle 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 (Oracle 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 Oracle Virtual Desktop Infrastructure 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