



An Oracle White Paper
February 2013

Oracle VM Enabling Rapid Migration to Private Cloud

Introduction	2
Migrating from Traditional Datacenter to Private Cloud: Challenges and Solutions	2
Application Deployment with Oracle VM Templates.....	2
Under the hood: Key Components of a Template	3
How Can You Use Oracle VM Templates?	4
What are the benefits of Oracle VM Templates?.....	6
Test Validation Illustrating the Benefits of Oracle VM Templates ...	7
Creating and Deploying Multi-Tier Application Environments with Oracle Virtual Assembly Builder	7
What is Oracle Virtual Assembly Builder?	8
How to create an Assembly with Oracle Virtual Assembly Builder? 9	
What are the benefits of Oracle Virtual Assembly Builder?	10
Difference between an Oracle VM Template and an Assembly ...	10
Pulling It All Together - Oracle Enterprise Manager	11
Conclusion	11

Introduction

The intended audience for this white paper is Oracle Applications Administrators and Application Architects interested to learn how Oracle Application-Driven architecture can help accelerate private cloud deployments. This white paper explains how to deploy complex Oracle applications in a few hours rather than weeks, while still allowing customization to suit the unique requirements of the particular deployment, and how to migrate a full application stack into a private cloud environment by using Oracle VM Templates and Oracle Virtual Assembly Builder.

Migrating from Traditional Datacenter to Private Cloud: Challenges and Solutions

Traditional Datacenters are typically fragmented environments, with various applications installed on different types of operating systems, often running on specialized hardware. Server virtualization is the first step to simplify this but typically when it comes to application deployment this is still done the old-fashioned way, meaning these application are installed and configured manually. This is not only a time consuming process, its also error-prone.

Multi-tier enterprise applications are increasingly being deployed in private cloud environments to realize the benefits of consolidation, flexibility and reduced deployment time. However, as mentioned, administrators still follow the same time-consuming practices for installing, configuring, and deploying all software (operating system, database, middleware, applications, etc.) in virtual machines that have traditionally been prescribed for physical environments. When migrating enterprise applications to a private cloud, a new approach is required. Software components must be standardized and pre-configured in the form of software appliances to be rapidly deployed. These appliances must be easily customizable to suit the unique requirements of the particular deployment.

With Oracle VM Templates and Oracle Virtual Assembly Builder Oracle offers a new approach by uniquely combining all the capabilities necessary to enable this transformation.

Application Deployment with Oracle VM Templates

Oracle VM Templates is an innovative approach to deploying a fully configured software stack by offering pre-installed and pre-configured software images. Use of Oracle VM Templates eliminates the installation and configuration costs, and reduces the ongoing maintenance costs helping organizations achieve faster time to market and lower cost of operations. Oracle VM Templates are part of many key Oracle products available for download, including Oracle Linux, Oracle Solaris, Oracle Database, Fusion Middleware, and many more.

[Oracle VM Guest Additions](#) were introduced with Oracle VM 3 that allow the guest software to pass information back and forth through the Oracle VM Manager to the virtual machine, and thus provide

direct integration between guest software and the virtualization layer, to assist in orchestration of complex, multi-VM deployments.

So how does this work in practice? The rest of this chapter will focus on the key components of Templates, how they can be downloaded and deployed, and finally how they can be customized.

Under the hood: Key Components of a Template

An Oracle VM Template contains various components such as the virtual machine configuration information, virtual disks that contain the operating system and any application software. These components are packaged together as an Oracle VM Template file. Older Oracle VM Template files were packaged as compressed 'tgz' files, while the more recent Templates are packaged according to the industry-standard Open Virtualization Format (OVF) standard as an 'ova' file (archived OVF file).

In order to get a complete understanding of the template capabilities, let's look first at the first generation Oracle VM Templates. Oracle VM Templates as available from Oracle Software Delivery Cloud all have the following components in common:

- One or more virtual machine images. For example, your template may simply contain a single VM with your standard OS build, or, if it is a more complex composite application, it may contain several different VMs, with many products. For example, it may include a VM for the web server, a VM for the app server, and a third for the database, etc.
- Each virtual machine in-turn includes a minimum of 2 or 3 files:
 - A binary system.img file containing the disk image with the boot OS
 - Optionally a binary file that is a disk image containing Oracle or other 3rd party software that is not a part of the OS. Often, this is dictated by software licensing. For example, Oracle Linux is under a GPL license while other software may be under a proprietary license and thus it may be desirable to separate non-GPL software on to a different disk from that containing software under a proprietary license.
 - A configuration text file called the vm.cfg file that contains the configuration parameters for that virtual machine
- For each virtual machine in the Template, there is also typically two or more script files:
 - One script to perform a one-time configuration of the OS the first time the VM is booted. This can be used to interactively customize or set network parameters based on the environment for that VM. For example, to set-up dhcp or static IP address and related parameters.
 - One (or more) script(s) to configure the Oracle-, or other-, products at first boot of the VM. If there are multiple products to be configured in the same VM, it is at the developer's discretion as to whether there will be one script per product or a single combined script. Again, this script can prompt the administrator to configure parameters for that particular VM in that particular environment.

All of the above files are packaged together in a compressed archive file (a 'tgz' file) that represents the Oracle VM Template for that product or solution. This archive file is then downloaded and imported into Oracle VM to be used as the basis for creating guest VMs.

As mentioned, newer Oracle VM Templates are OVF compliant. Their content is similar as outlined above with the difference there is no vm.cfg file but an OVF descriptor file which contains the metadata about the package and its contents. This is an extensible XML document for encoding information, such as product details, virtual hardware requirements, and licensing. Next to the OVF descriptor file, there are one or more disk image files and a manifest file (a 'mf' file), which contains SHA-1 digests of the OVF file and the disk image files. These files are packaged together in a compressed archive file (an 'ova' file). OVA is also the format used by Oracle Virtual Assembly Builder (OVAB) as we will see later on. In that context an 'ova' file is called an Assembly.

How Can You Use Oracle VM Templates?

Downloading and Importing Templates to Create Guest VMs

For many key Oracle products Oracle VM Templates are available and these can be downloaded from [Oracle Software Delivery Cloud](#). After downloading the Template file, it needs to be unzipped (uncompressed) and depending if it is an older or newer Template you get a 'tgz' file or an 'ova' file. A 'tgz' file can be directly imported in Oracle VM Manager as a Template, while an 'ova' file is interpreted as an Assembly, so it needs to be first imported as an Assembly and then converted to a Template.

To import a Template:

- Click on the Repositories tab in the Oracle VM Manager,
- select the Repository in which to store the Template,
- select 'VM Templates' in the navigation tree and select 'Import VM Template...' in the toolbar in the management pane,
- the VM Template wizard will start now guiding you through the import process.

To import an Assembly and create a Template out of it:

- Click on the Repositories tab in the Oracle VM Manager,
- select the Repository in which to store the Assembly,
- select 'Assemblies' in the navigation tree and select 'Import VM Assembly' in the toolbar in the management pane,
- the VM Assembly wizard will start now guiding you through the import process.
- Once the Assembly is imported you can easily create a Template from it by selecting the imported VM Template and select 'Create VM Template...' in the toolbar in the management pane.

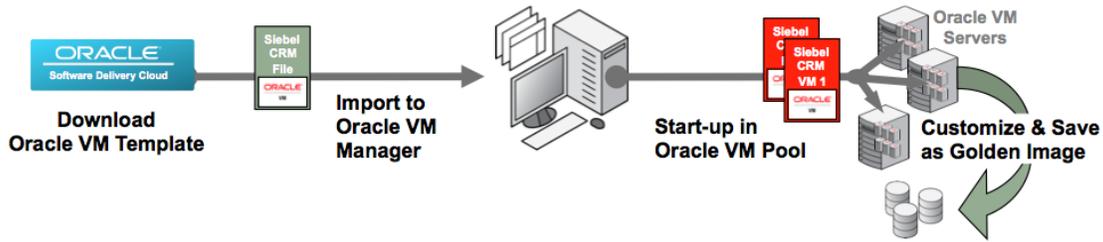


Figure 1. Download and Deploy Oracle VM Templates from Oracle Software Delivery Cloud

Initial VM Creation and Configuration

Creating VMs from Templates is very straight forward as the VM creation wizard initiated from the ‘Servers and VMs’ tab in the Oracle VM Manager UI walks you through the steps. Simply select the option to clone a VM from an existing VM Template and follow the steps.

Note that the creation process may take some time depending on the size of the VM in the Template and the speed of your pool storage as the image files are cloned between directories as part of the creation process. This is done so that the original Template images will remain as pristine source images in one location, while the created VM can then be customized in another location without overwriting the source. However, if you wish to make changes and save those changes to the Template itself, that can be done using the Save As Template option described later in this paper.

First Power-On of VMs Created from a Template

Once the VM is created, it is associated with the Server with most available resources in the Server Pool and it has the status Stopped. When the VM is started, Oracle VM Manager places the VM either on the current Server or on the “best” Server, meaning the Server with most available resources at that time. This VM start policy can be generally configured for a Pool and also overruled for an individual VM. Next to these VM start policies the user has also the option to create anti-affinity groups to specify for example that specific virtual machines should never run on the same Server.

From the Servers and VMs page, click on the VM and select Start in the toolbar in the management pane. As was discussed briefly earlier in the paper, the VM created from the Template will execute any one-time OS configuration and product configuration scripts on the first boot of the VM. Not all Templates require power-on configuration – it depends on what product(s) is/are contained in the Template - so a given Template may boot normally without any prompting at all.

For scripts that do contain user prompts to enter information, the user can easily access the VM console from Oracle VM Manager to respond to the prompts to fully configure the VM and permit the boot cycle to complete. To access the console, select the VM in Oracle VM Manager and click on the Launch Console button in the toolbar in the management pane.

Once the console is available, the user may be prompted for any range of information. Typically, the user will be prompted on the desired network configuration (DHCP or static) and possibly other information that depends largely on the product(s) contained in the Template, but could include information such as hostnames, other account passwords, or other information necessary to fully

configure the contained products. Once the information has been provided, the VM may need to restart for the changes to take effect.

Tip: The password for the *root* user of all Oracle VM Templates is *ovsroot*.

Customizing Templates to Create “Golden Images”

By their nature, Templates typically reflect the general best practices for the products they contain, but any one configuration cannot cover all scenarios, especially in complex, enterprise environments. As a result, you may wish to update the Templates and then save those changes in your Template VMs so that any future VMs created from that Template would automatically reflect those customizations. This can be easily done by making the changes and then clone it as a Template.

Oracle VM Manager also allows you to modify a Template (memory, number of processors,...) and define "Clone Customizers" to specify storage and network mappings. For more details about this consult the Oracle VM User's Guide.

What are the benefits of Oracle VM Templates?

Faster Application Deployment, Lower Cost

Oracle VM eliminates wasted time and expense by providing the ability to rapidly and easily deploy a pre-built, pre-configured, pre-patched guest virtual machine (or multiple machines depending on the application) in the form of an Oracle VM Template. Templates can contain a complete Oracle software solution, such as Siebel CRM or JD Edwards EnterpriseOne, including the operating system (Oracle Linux) and even your internally developed or third party software. A complete solution: ready to start-up.

Oracle VM Templates produced by Oracle are available from [Oracle Software Delivery Cloud](#), while third party developed Templates would be made available via their corresponding channels, ready to download and start-up in an Oracle VM infrastructure. Already licensed for production use, Oracle VM Templates can save users days or weeks learning to install and configure a sophisticated product such as Oracle Enterprise Manager Cloud Control 12c or Oracle Real Application Clusters.

Instead, users can focus on simply downloading and starting the VM(s) to begin using the product right away with the confidence that they are deploying an exact, validated copy of their virtual machines and not a virtual machine hand-built from scratch where the risk of human error such as missing a patch dependency, or misconfiguring a parameter is far greater.

Easy to Maintain and Customize

While Oracle VM Templates are pre-built to reflect the best practices for the included product(s), you may still want to tailor the solution to your particular environment. Oracle Virtual Machines created from Templates are still fully user-accessible and directly customizable in your environment. Some software appliance packaging tools create only a “black box” VM that cannot be directly modified by users once it is deployed, creating an inflexible environment and necessitating new change request/implementation processes. Templates, however, can be fully customized or tuned after installation and then saved again as “golden image” templates to serve as a user’s enterprise

deployment standard to minimize risks and variation across multiple instance deployments in the enterprise.

Fully Supported for Production Deployments

Within these Templates, Oracle software is laid-out in the same manner as the software would be if it were installed and patched manually “the old fashioned way”, meaning the exact same directories and Oracle “homes” are used, and the package and patch inventories are completely standard. Oracle VM Templates are fully supported for production environments.

Test Validation Illustrating the Benefits of Oracle VM Templates

To illustrate how much easier and faster the deployment is with Oracle VM Templates we want to refer to the test validation done by The Elevator Group.

The Evaluator Group assessed Oracle VM with Oracle Real Application Cluster (RAC) 11g Release 2 and E-Business Suite 12.1.1 in June 2012. There were 2 deployment scenarios, Oracle VM and VMware vSphere 5 2-node cluster environments. They concluded deploying RAC and E-Business Suite Templates with Oracle VM was much faster than deploying into a VMware vSphere 5 virtualized environment installing the same applications with traditional rapid installer on a 2-node cluster. Each Template prompts for configuration information (i.e. node names, IP addresses,...) and the Template does the rest in less than half a day. The Oracle RAC Template automatically provides a 2-node cluster with the ability to add additional nodes. E-Business Suite was just as easy to install with filling out a few questions for the application database and application server and it was installed in less than 3 hours. On the VMware vSphere platform the installation of Oracle RAC took almost 40 hours (10 times longer) and E-Business Suite took 15 hours and 15 minutes (almost 7 times longer).

They also noted that Oracle VM Templates are different from how VMware offers templates. Oracle VM Templates are pre-built, pre-configured, pre-tested and ready for deployment once downloaded. VMware templates require the end-user to manually install the application on a guest OS and then through scripts, tools and cloning generates his/her own VMware template. Then that clone can be customized for new virtual machines.

For more details, consult the [Test Report page](#).

Creating and Deploying Multi-Tier Application Environments with Oracle Virtual Assembly Builder

Oracle Virtual Assembly Builder provides the capability, which is not available with any other virtualization solution, to allow users to design, package and deploy multi-tier applications. It provides a framework for capturing the configuration of existing software components and packaging them as self-contained building blocks known as appliances. These appliances can be easily connected to form ready-to-run application blueprints known as Assemblies. Assemblies can then be deployed onto virtualized and cloud environments with automated configuration and late-binding parameters.

What is Oracle Virtual Assembly Builder?

Oracle Virtual Assembly Builder Studio is a graphical tool intended for application administrators to quickly create and configure entire multi-tier application topologies and provision them onto virtualized environments or cloud environments. This can be accomplished in two ways: (a) capturing the configuration of an existing reference application environment (what is called "introspection") and packaging all its components into a collection of customized software appliances, or (b) starting with pre-created, general purpose software appliances representing the various components in an application topology and then customizing these to your own needs.

In the example below we see how an Oracle Weblogic Server is configured into a multi-tier application including a Database backend and Oracle Traffic Director as frontend:

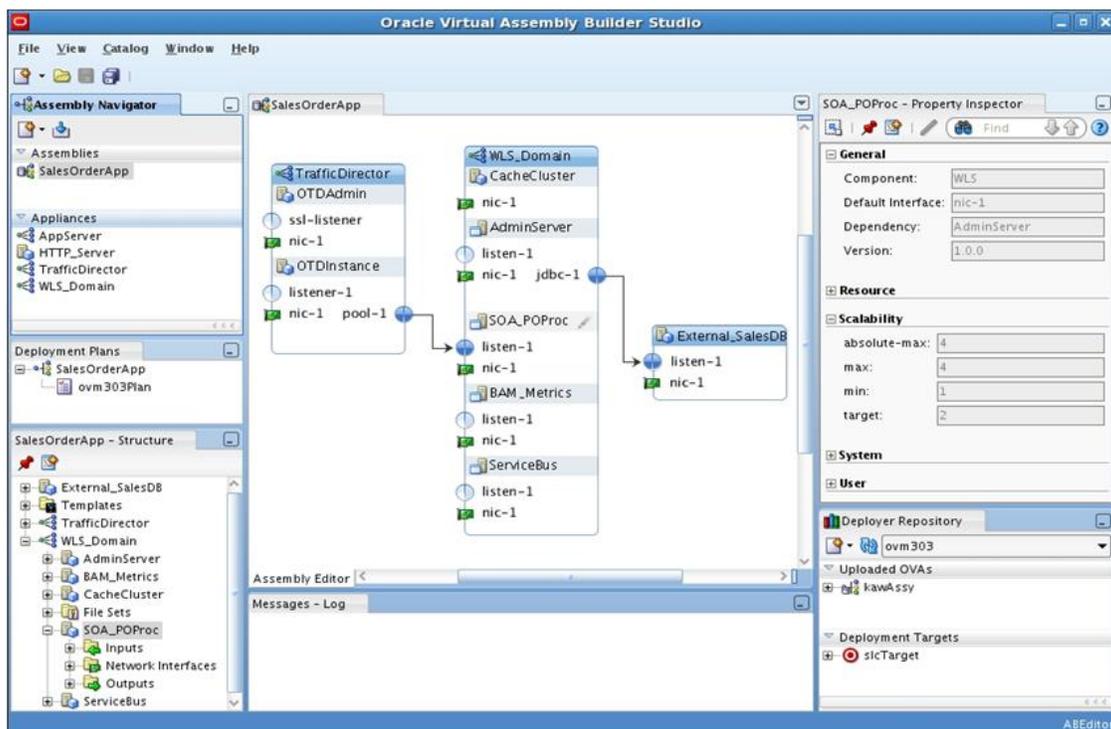


Figure 2. Oracle Virtual Assembly Builder Studio

By incorporating additional metadata into each software appliance, intelligence is provided about starting up and configuring late binding parameters for those components upon virtual machine startup. This allows appliances to be treated like standardized building blocks that can be connected to each other to form Assemblies, thus reducing time and eliminating errors associated with making multiple components work with each other. Because each appliance exposes a small set of configurable properties, Assemblies can be uniquely customized, providing flexibility for administrators to deploy multiple application instances using the same base blueprint contained in the Assembly.

This capture and configuration process can also be done by using a powerful scripting interface, allowing administrators to automate the lifecycle of Assemblies and incorporate them into a broader process flow involving additional data center components.

Oracle Virtual Assembly Builder Deployer complements the Studio by providing a robust server-side application that is designed for handling concurrent access by multiple users to conduct all deployment and lifecycle operations against Assemblies created in the Studio.

Through role-based access control, users can access and share Assembly archives within the Deployer repository. All Assembly archives uploaded by users are versioned upon upload and are stored in industry standard Open Virtual Archive (OVA) format.

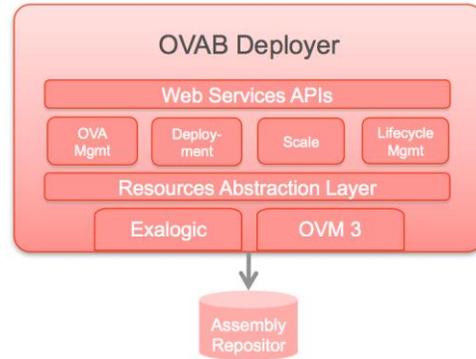


Figure 3. Oracle Virtual Assembly Builder Deployer

Users can then deploy, scale, and perform all lifecycle operations against Assemblies deployed from within the repository either via the Web Services APIs or a command line client utility. Assembly deployment is automatically done into the appropriate resource pool via the resource connector (e.g. Oracle VM), and all appliances within the Assembly are instantiated as a collection of inter-connected virtual machines running on the resource pool.

How to create an Assembly with Oracle Virtual Assembly Builder?

Capturing and deploying an entire multi-tier application topology only takes 4 steps:

1. Introspect

During this initial phase the configuration metadata for individual software components or multiple distributed components will be captured. These target components may reside locally or remotely on multiple distributed systems that could either be physical or virtual.

2. Assemble

Via the visual drag-and-drop interface, complex Assemblies using appliances maintained in a navigation catalog can be created. The relationships and connections between appliances can be established by using a wiring tool that automatically checks for protocol compatibility. Connections from appliances to external resources (e.g. database, security provider, messaging, etc.) not included within Assembly can be created as well.

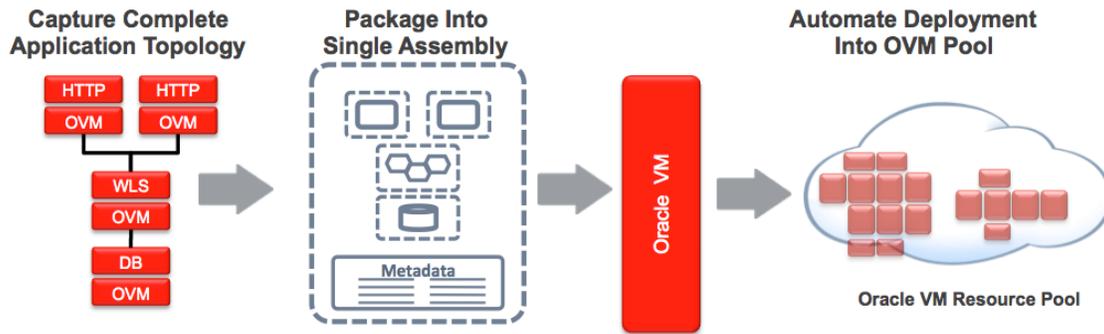


Figure 4. Creating an Assembly with Oracle Virtual Assembly Builder

3. Package

Automatically creates bootable virtual machine disk images with configurable metadata that allow for deploy-time customization of the software component contained in the appliance. A customized operating system distribution (e.g. Oracle Linux) will be packaged into the appliance along with the software component, including additional application binaries and configuration scripts to further customize the appliances.

4. Deploy

By establishing an authenticated connection directly with the Oracle VM Manager, the Assembly archive will be imported and deployed. Oracle Virtual Assembly Builder will start each VM and deployment-specific customization will be done through automatic fix-up of late binding properties within the appliances.

What are the benefits of Oracle Virtual Assembly Builder?

Not only does Oracle Virtual Assembly Builder allow you to create Assemblies using existing appliances but also you have the ability to introspect existing physical environments, analyze their configuration and turn them into appliances. Those appliances can be then packaged as an Assembly or be used as building blocks to create more sophisticated Assemblies. Once your Assembly is ready-to-run Oracle Virtual Assembly Builder will deploy this into your private cloud and this in a fully automated way.

Difference between an Oracle VM Template and an Assembly

An Assembly is very similar to an Oracle VM Template except that an additional set of configuration information and management policies are packaged along with the set of multiple virtual machines, their virtual disks, and the inter-connectivity between them.

Oracle Virtual Assembly Builder is specialized for creating (multi-tier) appliances containing specific Oracle products (Oracle Database, Oracle HTTP Server, Oracle Weblogic Server,...), while Oracle VM Templates can be used for any arbitrary application.

Pulling It All Together - Oracle Enterprise Manager

Oracle Enterprise Manager 12c helps Oracle customers to deliver enterprise private cloud services. It leverages its integration with Oracle VM Manager to manage the Oracle VM based infrastructure and offers complete management of entire software and systems stack - ranging from infrastructure to business applications.

Oracle Enterprise Manager provides an out-of-the-box self-service portal that allows self-service users to access cloud services (provisioning applications) without IT intervention. An administrator can use a guided cloud setup wizard to define allowable virtual machine sizes, assign quotas to users and roles, define access boundaries and placement rules, set up chargeback plans, and publish Oracle VM Templates and Assemblies (multi-tier applications) in the software library of the Oracle Enterprise Manager. The self-service capability of Enterprise Manager allows application owners to select, test and deploy applications on-demand and reduces the enterprise application deployment time from months and days to hours.

For example, when an application needs to be deployed requiring a guest virtual machine to power on, Enterprise Manager automatically selects the physical Oracle VM Server to use from a pool of healthy, available servers using a pool load-balancing algorithm. This, along with performance and resource monitoring offered by Enterprise Manager, is the kind of automation organizations need.

Finally, Enterprise Manager chargeback gives organizations a mechanism to meter cloud resources and charge internal users based on their consumption. This means that now users pay only for the resources they use, allowing them to monitor their usage accurately and efficiently.

Conclusion

Both Oracle VM Templates and Oracle Virtual Assembly Builder help IT administrators to reliably migrate their application stack into a private cloud environment. The innovative approach makes it practical for complex enterprise applications to be easily customized and provisioned with minimal to no manual intervention.

Oracle VM Templates and Oracle Virtual Assembly Builder provide many benefits to:

- Improve operational efficiency without sacrificing flexibility
- Significantly lower risk of mistakes
- Capture and provision complete multi-tier application environments for development, testing, or production
- Faster and easily create and deploy software solutions, resulting in lower costs
- Leverage the most complete enterprise technology stack including Oracle Linux, server virtualization, Oracle Database, ... to create a total solution backed by Oracle
- Leverage best practices for deploying these applications

For the latest information on Oracle VM Templates and Oracle Virtual Assembly Builder, consult the [Oracle VM Templates webpage](#) and the [Oracle Virtual Assembly Builder webpage](#).



White Paper Title
February 2013
Author: Robbie De Meyer
Contributing Authors: Adam Hawley

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com



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

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

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

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0113

Hardware and Software, Engineered to Work Together