Oracle Maximum
Availability Architecture

Running Middleware Applications on Oracle Database Appliance Machines

Process Document

ORACLE WHITE PAPER | AUGUST 2015
VERSION 1.0

Table of Contents

Introduction

This document describes the basic steps for bringing up and tearing down Oracle Fusion Middleware deployments in an Oracle Database Appliance (ODA) environment.

Because ODA is a virtualized environment, the various Oracle Fusion Middleware artifacts take the form of virtual machines.

This document covers the ODA aspects of deploying applications. This includes the specification and starting of the virtual machines. It also covers, in part, the configuring of the WebLogic Server (WLS) and Oracle Traffic Director (OTD) applications. For additional information about configuring WLS and OTD in the deployment, refer to the following documents:

Oracle WebLogic Server Documentation

Oracle Traffic Director Documentation

Sequencing of a deployment

Perform the following operations in the prescribed sequence.

1. Create CPU pools on each node
2. Create an ODA shared repository
3. Import templates for WLS and (if needed) OTD
4. Create and configure each middleware VM
5. Start each middleware VM
6. Configure the middleware applications

All the above steps are covered in the following subsections. In addition, the appendices contain walk-throughs of a VM deployment (Appendix A) and a VM teardown (Appendix B) of a simple installation (minimal WLS and OTD) in a specific target environment (ODA X4-2, X5-2).

Appendix C contains a walk-through of the WLS configuration for the above simple deployment. And Appendix E contains a walk-through of the OTD deployment for the same example.

Depending on your needs, you might want to start with the examples.

Deployment Instructions

Note that the instructions listed here assume that you always start from a clean environment, and return to a clean environment during cleanup. You may want to intentionally do a partial teardown of an earlier deployment, and thus can skip some steps when redeploying. Such special use-cases are not covered here.

Create CPU Pools on each node

The following commands (one per node) are used to create the CPU pools:

```
create cpupool <cpu-pool-name> -numcpu <virt-cpu-count> -node 0

create cpupool <cpu-pool-name> -numcpu <virt-cpu-count> -node 1
```

The pool name should be the same in each command invocation. It is recommended that you allocate the same size pools for each node. Note that the CPU pool sizes should be consistent with your licensing allotments.

Hint: To determine the number of CPUs available on each ODA node, enter the following command:

```
oakcli show cpupool -node <node-number>
```

In the output that appears, count the entries in the section “default-unpinned-pool”. These are the CPUs which are not already reserved by ODA.

You can put all the available CPUs into your pools, or you can create pools that are just large enough to satisfy the needs of your expected deployment.

Create an ODA shared repository

A shared repository must exist to house the virtual artifacts (templates and VMs) for your installation.

Note: If you are doing multiple deployments it is recommended that a separate repository be created for each one.

To create a shared repository use the following command:

```
oakcli create, repo, <repo-name> -dg data -size <repo-size-GBs>
```

The repo size should be large enough to house all the expected middleware artifacts. The following logic is recommended to calculate the size:

1. Get the number of VMs required to support WLS (=> <wls-vms>)
    a. Add up all the separate VMs required to support WLS (including admin servers, managed servers, coherence servers).
    b. Add 1 to account for the WLS template.
2. Get the number of VMS required to support OTD (=> <otd-vms>)
    a. Start with 0
    b. If OTD is being deployed then add 3 to the above (for the OTD admin server and the two OTD nodes).
    c. If OTD is being deployed then add 1 to account for the OTD template.

Use the following to calculate the repo size needed:

```
repo-size-GBs = <wls-vms> * 17 + <otd-vms> * 15
```

It is recommended to include a fudge factor of 10% to the above:

```
repo-size-GBs = <repo-size-GBs> * 1.1
```

You can use the following command to verify that the repo was created successfully:

```
oakcli show repo
```

Note: If it later turns out that your repo is not large enough you can increase it using the following command:

```
oakcli configure repo <repoName> -incsize <new-size>
```

Import the templates for WLS and (if needed) OTD

You must import the VM templates for the WLS, and (if desired) OTD VMs, into the shared repository. You can use the standard template for both in which case you will need to configure WLS and OTD into their respective VMs manually. An alternative approach is to create custom templates for WLS and OTD (based on the standard template) which include configured versions of WLS and OTD.

Use the following command to import a template:

```
oakcli import vmtemplate <unique-template-name> -files <template-file> -repo, <repo-name> -
node 0
```

As you are using a shared repository, you only need to import a template to one of the two ODA nodes.

Note: To help avoid name conflicts, it is recommended that your chosen imported template name must reflect the original template file name. For example,

```
oakcli import vmtemplate WLS_1213_VMT -files /OVS/staging/WLS_1213_VMT.tar.gz, -repo,
wlsodarepo, -node, 0
```

Create and configure the VMs

After importing the templates, you can instantiate the VMs for your designed Oracle Fusion Middleware artifacts.

For each VM, use the following commands in sequence:

1. oakcli clone <parameters>
2. oakcli configure <parameters>
3. oakcli assign ip <parameters>
4. oakcli modify <parameters> (NOTE: this only for hardware >= X5-2)

Warning: The format of the *clone* and *configure* commands (i.e. the <parameters>) varies with the version of ODA and server hardware. Also the *modify* command is used only for server hardware with IB support (i.e. >= X5-2).

Details of each command are provided in the following subsections. These are followed by a description of all the parameters in a final subsection.

oakcli clone command

For ODA version 12 or above, use the following format:

```
oakcli clone vm <vm-name> -vmtemplate <template-name> -snap
```

If the ODA version is earlier than version 12, use:

```
oakcli clone vm <vm-name> -vmtemplate <template-name> -repo <repo-name> -node 0
```

Note that the node provided can be either 0 or 1. The clone command needs to be issued only once since the repository is shared across the nodes.

Hint: You can find your ODA version using the following command:

```
oakcli show version
```

oakcli configure command

If the underlying machine hardware is X5-2 or later, enter the following format:

```
oakcli configure vm <vm-name> -vcpu <num-virt-cpus> -memory <mem-size> -maxmemory <mem-size>
-cpupool <cpu-pool-name> -prefnode <oda-node>
```

If the machine hardware is earlier than X5-2, enter:

```
oakcli configure vm <vm-name> -vcpu <num-virt-cpus> -memory <mem-size> -network <net-config>
-cpupool <cpu-pool-name> -prefnode <oda-node>
```

Note: In either case, it is recommended that you distribute your VMs evenly over the two ODA nodes.

oakcli assign ip command

```
oakcli assign ip -vm <vm-name>
```

This command assigns an available IP address to your new VM and prints it.

One issued, if you later lose the value you can use this command to display the value:

```
oakcli show vm <vm-name>
```

oakcli modify command

Note: These commands are only used for platforms that support IB.

```
oakcli modify vm <vm-name> -addnetwork priv1
```

```
oakcli modify vm <vm-name> -addnetwork net1
```

Notes on the parameters to all the above commands

The parameters to the above commands are described together here since some of them are used in more than one command.

<mem-size>

Memory requirements vary with the type of VM. The following are the current recommendations:

WLS Admin server	2048
WLS managed server	4096
Coherence server	8192
OTD Admin Server	1024
OTD Admin Node	6144

<net-config>

For ODA V1 (SUN_FIRE_X4370_M2 hardware) use:

```
['type=netfront,bridge=priv1','type=netfront,bridge=net1','type=netfront,bridge=net2'
'type=netfront,bridge=net3','type=netfront,bridge=net4']
```

For ODA V2-V3 (X3-2, X4-2, X5-2) use:

```
['type=netfront,bridge=priv1','type=netfront,bridge=net1','type=netfront,bridge=net2']
```

<num-virt-cpus>

CPU requirements vary with the type of VM. The following are the current recommendations:

For ODA V1 (SUN_FIRE_X4370_M2 hardware):

WLS Admin server	2
WLS managed server	2
Coherence server	1
OTD Admin Server	2
OTD Admin Node	2

For ODA versions 2-4 (X3-2, X4-2, X5-2):

WLS Admin server	2
WLS managed server	4
Coherence server	2
OTD Admin Server	2
OTD Admin Node	4

<oda-node>

For each VM, you can request which ODA node (0 or 1) the VM be deployed to.

It is recommended that you evenly distribute your VMs across nodes.

<cpu-pool-name>

The name of the CPU pool you created for the node.

<repo-name>

The name of the shared repo you created.

<template-name>

The name of the template file.

If using custom templates for WLS and OTD make sure that you reference the appropriate template.

<vm-name>

For WLS VMs it is recommended that your VM name include your WLS domain name and indicate whether the VM is an admin or managed server. For example:

```
MyDomain_AS
MyDomain_MS1
MyDomain_MS2
MyDomain_COH1
etc.
```

There is only a single OTD deployment for an installation, which is represented by an admin server and two admin nodes. Therefore for OTD VMs it is recommended that they only be named something to make them easily recognizable. For example:

```
OTD_AS
OTD_AN1
OTD_AN2
```

If you plan to have multiple installations deployed, you should also consider including a unique identifier for the installation to each VM name. For example:

```
CS_MyDomain_AS
CS_OTD_AS
```

There is a limit on the number of characters for VM names. So if you use the above conventions (or similar) and include the domain name in the names, you might need to limit the size of your domain names to avoid exceeding the VM naming limitation.

Start the VMs

Once the VMs have been created and configured, each VM should be started using the following command:

```
oakcli start vm <vm-name>
```

Configure the applications

Configuration (in part) of WLS and OTD are covered in later chapters.

Tearing down a deployment

The steps to completely tear down an existing deployment are:

1. Do any required WLS and/or OTD cleanup.
2. Stop all associated VMs
3. Release the allocated IP for each VM
4. Delete each VM
5. Delete the loaded templates
6. Delete the CPU pools
7. Delete the shared repository

Here are the associated commands to use for each step:

1. WLS and OTD teardown are covered in subsequent sections.
2. oakcli stop vm <vm-name> -force   (note: do per VM in the deployment)
3. oakcli release ip -vm <vm-name>   (note: do per VM in the deployment)
4. oakcli delete vm <vm-name>   (note: do per VM in the deployment)
5. oakcli delete vmtemplate <template-name>   (note: do per loaded template)
6. oakcli delete cpupool <cpu-pool-name> -node <node>   (note: do for both nodes, 0 and 1)
7. oakcli stop repo <repo-name> -node <node>   (note: do for both nodes, 0 and 1)
8. oakcli delete repo <repo-name>

Hints: The following commands are useful if you have forgotten the names of the artifacts. They can also be issued after each step to verify that the step worked as expected.

- to obtain a list of VMs use the command:
  ```
  oakcli show vm
  ```
- to obtain a list of loaded templates use the command:
  ```
  oakcli show vmtemplate
  ```
- to obtain a list of cpu pools for each node use the command:
  ```
  oakcli show cpupool -node <node-num>
  ```
- to obtain a list of repositories on both nodes use the command:
  ```
  oakcli show repo
  ```

Note that above steps assume that you are tearing down the entire deployment. If you are redeploying, you might want to do a partial teardown. That use-case is not covered here, but you can extrapolate from the instructions provided.

Also, if you are doing multiple deployments make sure that you do not delete resources that are being used by the other deployments (for example loaded templates).

Configuring WLS application(s)

This section describes the commands used to configure the WebLogic Server VMs.

This section uses a number of variables that you must substitute for your environment and configuration into the provided commands. You should have these values available before you begin each section.

A few of these variables are:

<wls-version>: The version of WLS that you installed (for example 12.1.3)

<wls-home>: The WebLogic home directory.

<wls-domains-root>: The directory containing the data for your WLS domains.

<domain-name>: The domain name of the domain which contains the server being configured.

<domain-password >: The password associated with the domain for the server being configured.

<domain-id>: A unique identifier (for example 1, 2, 3...) that you assign to each domain.

Other variables are listed in the sections they are relevant to.

Note: It is recommended that you configure the admin server first and wait till it comes online before configuring the managed servers.

Insure WLS is installed on all the admin and managed server VMs.

WLS must be installed into all your WLS VMs. You have two options for making this happen:

1. Manually install WLS in your VMs after they are created.
2. Create a custom template which already has WLS installed and use that template when creating your VMs.

Set up the user accounts

```
echo <domain-password> | passwd --stdin root

echo <domain-password> | passwd --stdin oracle
```

Optimize the environment on each WLS VM

If the VM being configured is a Coherence cluster of managed servers, enter these commands:

```
grep_sed_util "net.core.rmem_max" "net.core.rmem_max=2097152" "/etc/sysctl.conf"

grep_sed_util "net.core.wmem_max" "net.core.wmem_max=2097152" "/etc/sysctl.conf"
```

If the VM being configured supports IB (must be at least X5-2), enter these commands:

```
grep_sed_util "MLX4_XEN_FMR_LOAD" "MLX4_XEN_FMR_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "MLX4_VNIC_LOAD" "MLX4_VNIC_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "SDP_LOAD" "SDP_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "SRP_LOAD" "SRP_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "ISER_LOAD" "ISER_LOAD=no" "/etc/ofed/openib.conf"

echo "alias ib0 mlx4_en">>/etc/modprobe.conf

echo "alias ib1 mlx4_en">>/etc/modprobe.conf
```

```
grep_sed_util "tsc" "      kernel \/vmlinuz-2.6.39-400.245.1.el5uek ro root=LABEL=\/ nohpet
nopmtimer tsc=reliable hda=noprobe hdb=noprobe ide0=noprobe numa=off pci=noaer,nocrs
console=tty console=ttyS0,115200n8 selinux=0 nohz=off crashkernel=256M@64M loglevel=7
panic=60 ipv6.disable=1" "/boot/grub/grub.conf"
```

Set up the network

A public and a private interface are configured for each server. In the IB case the private interface is IB. For non-IB systems, eth0 is the private interface.

You should have the following values available before you start.

Values that apply across all servers in all domain:

<private-netmask>: This is the netmask for the internal network.

<private-gateway>: This is the gateway for the internal network.

<dns-server>: The DNS server used for WLS name resolution. There may be more than one.

Values that apply to all servers in a domain:

<public-network-intf>: This is the interface for he external network you are using for WLS (For exampleeth1).

<public-netmask>: This is the netmask for the public network.

<public-gateway>: This is the gateway for the public network.

<dns-server>: The DNS server used for WLS name resolution. There may be more than one.

Values that are unique per server:

<private-ip>: this is the IP address returned from the *oakcli assign ip* command.

<public-ip>: this is the IP address you assigned to the server.

<host-name>:The fully qualified hostname you have assigned to the server.

Issue the following commands described below for each of the three servers. Text in < > braces are the values described above, and must be provided by you.

Set up the private network. If your system supports IB then issue the following commands to configure IB and the internal IB (private) network:

First create the private network:

```
 . /etc/sysconfig/network-scripts/network-functions

service openibd start

chkconfig openibd on

service openibd status

configure_slave "ib0"

configure_slave "ib1"

configure_master <private-ip> <private-netmask>

createNetworkConfigFile eth0 <private-ip> <private-netmask> <private-gateway>
```

If your system does not support IB, enter the following command to configure the private network:

```
createNetworkConfigFile eth0 <private-ip> <private-netmask>
```

Admin servers only

Configure the public network:

```
createNetworkConfigFile <public-network-intf> <public-ip> <public-netmask> <public-gateway>

set_hostname <host-name>

export HOSTNAME=<host-name>

echo "<private-ip> $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "<public-ip>  $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts"
```

For all managed servers in the domain:

Note for this you will need the following values for each managed server:

<this-managed-internal-ip>: This is the IP address returned from the *oakcli assign ip* command.

<this-managed-hostname>: This is the fully qualified host name you assigned to the managed server,

Enter the following commands in the admin server for each managed server in the domain.

```
MANAGED_HOSTNAME=<this-managed-hostname>

echo "<this-managed-internal-ip> $MANAGED_HOSTNAME ${MANAGED_HOSTNAME%%.*}" >> /etc/hosts
```

Managed servers only

Note for this you will need the following values:

<admin-internal-ip>: This is the IP address returned from the *oakcli assign ip* command for the admin server

<admin-hostname>: This is the fully-qualified host name you assigned to the admin server

Configure the public network:

```
createNetworkConfigFile <public-network> <public-ip> <public-netmask> <public-gateway>

set_hostname <host-name>

export HOSTNAME=<host-name>

export ADMIN_HOSTNAME="<admin-hostname>"

echo "<public-ip> $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "<private-ip> $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "<admin-internal-ip> $ADMIN_HOSTNAME ${ADMIN_HOSTNAME%%.*}" >> /etc/hosts
```

Continue for all servers (admin and managed)

```
sed -i -e "s/^HOSTNAME=.*/HOSTNAME=$HOST_NAME/g" /etc/sysconfig/network

// configure name server(s)
//    Note - if more than one DNS server do this for each

echo "nameserver " <dns-server> > /etc/resolv.conf

echo "search ${HOSTNAME#*.}" >> /etc/resolv.conf

/sbin/service network restart
```

Set up time services (all servers)

Enter the following commands:

```
ntpdate <ntp-server> 2>&1
```

where <ntp-server> is the IP address of the NTP server you are using.

Create the domain (admin servers only)

```
<wls-home>/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --
createDomain <wls-domains-root> "<wls-version>" <domain-id>
```

Generate Demo Cert (all servers)

These commands use the WLS java utilities classes to generate a demo certificate.

```
. <wls-home>/server/bin/setWLSEnv.sh

java utils.CertGen -keyfile privatekeyfile -keyfilepass DemoIdentityPassPhrase -certfile
certfile -cn $HOSTNAME

java utils.der2pem <wls-home>/server/lib/CertGenCA.der

java utils.ImportPrivateKey -keystore DemoIdentity.jks -storepass
DemoIdentityKeyStorePassPhrase -keyfile privatekeyfile.pem -keyfilepass
DemoIdentityPassPhrase -certfile newcerts.pem -alias DemoIdentity

cp DemoIdentity.jks <wls-home>/server/lib
```

Set up log files (admin servers only)

```
mkdir -p <wls-domains-root>/<domain-name>/servers/AdminServer/logs
```

Set up boot.properties (admin server only)

```
mkdir -p <wls-domains-root>/<domain-name>/servers/AdminServer/security

echo "username=system" >> <wls-domain-root>/<domain-
name>/servers/AdminServer/security/boot.properties

echo "password=<domain-password>" >> <wls-domains-root>/<domain-
name>/servers/AdminServer/security/boot.properties

chown -R oracle:oinstall /u01
```

Set up the node manager (managed servers only)

This section assumes the following value, which you should have available:

<node-manager-home>: The directory to contain the node manager data (for example /u01/<domain-name>).

Enter the following commands:

```
mkdir -p <node-manager-home>
```

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust -Dssl.debug=true -
Dweblogic.StdoutDebugEnabled=true -Dweblogic.security.SSL.verbose=true"
```

```
<wls-home>/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --
setupNodeManager <wls-domains-root> <node-manager-home> <domain-id>
```

If WLS version is 12.1.3 issue this command:

```
export JAVA_OPTIONS="-Dverbose=true -DStartScriptEnabled=false -DCrashRecoveryEnabled=true -
DListenPort=5556"
```

If the WLS version is not 12.1.3 issue this command:

```
export JAVA_OPTIONS="-Dverbose=true -DStartScriptEnabled=false -DCrashRecoveryEnabled=true -
DDomainRegistrationEnabled=true -DListenPort=5556"
```

Then:

```
cp "$WL_HOME/server/bin/startNodeManager.sh" <node-manager-home>
```

```
sed -i "s/NODEMGR_HOME=.*\/common\/nodemanager.*/NODEMGR_HOME=<node-manager-home>/g" <node-
manager-home>/startNodeManager.sh
```

> *Warning: The above sed command syntax requires that you escape the path separator characters in the quoted <node-manager-home>. For example \/u01\/<domain-name>*

For WLS versions 12.1.2 and 12.1.3 only:

```
cp $WL_HOME/server/lib/DemoIdentity.jks <node-manager-home>/security
```

Start the node manager:

```
chown -R oracle:oinstall /u01
```

```
su oracle -c "<node-manager-home>/startNodeManager.sh  > <node-manager-home>/nodemanager.out
2>&1&"
```

Start the server

This section assumes the following value which you should have available for each server:

<server-id>: A unique identifier for each managed server (For example 1, 2, 3…) which is used by the node manager.

To start an admin server:

```
su - oracle -c "nohup <wls-domains-root>/<domain-name>/bin/startWebLogic.sh > <wls-domains-
root>/<domain-name>/servers/AdminServer/logs/AdminServer.out 2>&1&"
```

```
sed -i "s/<insert script>/startAdmin <domains-name>/g" /usr/sbin/wls-vm-start
```

To start a managed server:

```
<wls-home>/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --
startManagedServer <server-id> <domain-id>
```

```
sed -i 's/<insert script>/startNodeManager/g' /usr/sbin/wls-vm-start
```

Set up node manager (admin servers only)

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust"
```

```
<wls-home>/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --
modifyNodeManager <domain-password> <domain-name> "system" <domain-password> "<wls-version>"
<domain-id>
```

Configure JMS distributed distributions (admin servers only)

```
nohup $WL_HOME/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --
setupDistributedDestinations <domain-password> <domain-id>
```

Tearing down WLS application(s)

You should have the following values available:

<wls-version>: The version of WLS that you installed (for example 12.1.3)

<wls-home>: The WebLogic home directory.

<domain-password >: The password associated with the domain for the server being configured.

<domain-id>: A unique identifier (for example 1, 2, 3...) that you assign to each domain.

Do the following:

```
 WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust"

 export WLST_PROPERTIES

 <wls-home>/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --
shutdownDomain <domain-password> <wls-version> <domain-id>
```

Configuring OTD deployment(s)

Note: if not using OTD skip this step.

This section describes the commands used to configure Oracle Traffic Director.

Insure OTD is installed on the three OTM VMs

OTD must be installed into your OTD VMs. You have two options for making this happen:

1. Manually install OTD in your VMs after they are created.
2. Create a custom template which already has OTD installed, and use that template when creating your VMs.

Set up the user accounts

You should have the following values available before you start.

<otd-password>: The password for OTD accounts.

root users. For example

```
echo <otd-password> | passwd --stdin root

echo <otd-password> | passwd --stdin oracle
```

Optimize the environment on each OTD VM

The following commands should be used to optimize the OTM servers.

```
  /etc/sysctl.conf parameters:


grep_sed_util "net.ipv4.ip_nonlocal_bind" "net.ipv4.ip_nonlocal_bind = 1"

  "/etc/sysctl.conf"

grep_sed_util "fs.file-max" "fs.file-max=6291456" "/etc/sysctl.conf"

grep_sed_util "fs.nr_open" "fs.nr_open=2097152" "/etc/sysctl.conf"

grep_sed_util "net.ipv4.tcp_fin_timeout" "net.ipv4.tcp_fin_timeout=30"

  "/etc/sysctl.conf"

   /sbin/sysctl-p
```

/etc/security/limits.conf parameters:

```
grep_sed_util "oracle hard nofile" "oracle hard nofile 2097152"

  "/etc/security/limits.conf"

grep_sed_util "oracle soft nofile" "oracle soft nofile 1024"

  "/etc/security/limits.conf"
```

User limits:

```
  echo "ulimit -n 131072">> ~oracle/.bash_profile

  echo "ulimit -n 131072">> ~root/.bash_profile
```

Note the following settings are only for systems that support IB (must be at least X5-2):

/etc/ofed/openib.conf parameters:

```
grep_sed_util "MLX4_XEN_FMR_LOAD" "MLX4_XEN_FMR_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "MLX4_VNIC_LOAD" "MLX4_VNIC_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "SDP_LOAD" "SDP_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "SRP_LOAD" "SRP_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "ISER_LOAD" "ISER_LOAD=no" "/etc/ofed/openib.conf"
```

/etc/ofed/openib.conf parameters:

```
echo "alias ib0 mlx4_en">>/etc/modprobe.conf

echo "alias ib1 mlx4_en">>/etc/modprobe.conf
```

Set up the network

A public and a private interface are configured for each server. In the IB case the private interface is IB. For non-IB systems, eth0 is the private interface.

You should have the following values available before you start.

Values that apply to all three servers:

<private-netmask>: This is the netmask for the internal network

<private-gateway>: This is the gateway for the OTD network

<dns-server>: The DNS server used for OTD name resolution. There may be more than one.

Values that are unique per server:

<public-network>: This is the external network you are using for OTD (for example eth1).

<public-netmask>: This is the netmask for the OTD network

<public-gateway>: This is the gateway for the OTD network

<internal ip>: This is the IP address returned from the *oakcli assign ip* command

<public-ip>: This is the IP address you assigned to the server

<host-name>:The fully qualified hostname you have assigned to the server.

Issue the following commands for each of the three servers. Text in < > braces are the values described above, and must be provided by you.

If your system supports IB then issue the following commands to configure IB and the internal IB (private) network:

Configure the private network:

```
 . /etc/sysconfig/network-scripts/network-functions

service openibd start

chkconfig openibd on
```

```
service openibd status

configure_slave "ib0"

configure_slave "ib1"

configure_master <private-ip> <private-netmask>
```

If your system does not support IB then issue the following command to configure the private network:

```
createNetworkConfigFile eth0 <private-ip> <private-netmask>
```

Now continue with these remaining commands on all OTD servers.

Configure the public network:

```
createNetworkConfigFile <public-network> <public-ip> <public-netmask> <public-gateway>
```

Configure the hostname and name resolution

```
set_hostname <host-name>

export HOSTNAME=<host-name>

sed -i -e "s/^HOSTNAME=.*/HOSTNAME=<host-name>/g" /etc/sysconfig/network


   // Note - if more than one DNS server do this for each
echo "nameserver " <dns-server> > /etc/resolv.conf

echo "search ${HOSTNAME#*.}" >> /etc/resolv.conf


echo "<public-ip> $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

/sbin/service network restart
```

Set up time services

Issue the following commands:

```
ntpdate <ntp-server> 2>&1
```

where <ntp-server> is the IP address of the NTP server you are using.

Create password file

The password file contains your OTD password. This file is used to authenticate by a number of the commands which will follow. This file is only used during installation. It is recommended that the file be removed when done with the VM's initial configuration.

```
echo "tadm_password=<otd-password>" > <otd-pswd-file>

chmod 0600 <otd-pswd-file>
```

where <otd-pswd-file> is the temporary file used to save the password during installation.

Recommended password location: /tmp/.otdpasswd

Configure the OTD Servers

The configuration of the OTD admin server differs considerably from the configuration of the nodes. So these are described in separate subsections.

The configuration of the admin server is dependent in part on the nodes being up, and vice versa.

The approach is to start the admin server first and partially configure it (thus meeting the nodes' dependencies). Then start the nodes. When they are up, the admin server configuration can be completed. This is discussed in greater detail in the subsections.

In general OTD server configuration requires the following values, which you should have available.

<config-name>: A name to identify your OTD configuration (for example "otd-config"). This name is used as is for the node configurations, and is used as a base for the per domain configurations in the admin server.

<public-ip>: This is the IP address you assigned to the server being configured.

<otd-admin-port>: This is the port for OTD administration (for example 8989).

<otd-admin-user>: This is the OTD admin user name (for example admin).

<otd-home>: The directory where OTD is installed.

<otd-inst-home>: The directory representing the artifacts for this version of OTD.

Configure the OTD admin server

First issue the following command on the admin server:

```
<otd-home>/bin/tadm configure-server -i=<otd-inst-home> -r=root -j=<otd-home>/jdk -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-port> -evQ

<otd-inst-home>/admin-server/bin/startserv
```

After the above completes, the dependencies that the admin nodes have on the admin server are met. So the next step is to configure both admin nodes. After the admin nodes are up and running, continue with the following steps on the admin server.

Per domain logic:

The following needs to be done for each WLS domain. Each domain should be assigned a number which will be used to generate domain-specific names, for example for (int i =0; i < domain-count; i++). This number will be referred to in the following commands as:

<domain-num>

For each domain issue the following commands.

This part requires the following values, which you should have available.

<wls-domain-vip>: The unique public IP address you assigned to the current domain for OTD communication.

<wls-start-listen-port>: The port corresponding to the first cluster in the domain (or the domain itself if not clustered). This value is incremented to get values for subsequent clusters, so make sure the range of expected values is available.

<wls-domain-config-name>: A name identifying the OTD configuration corresponding to each domain. It is recommended that a common base OTD config name be used in conjunction with the <domain-num> for constructing the domain config names (for example "otd-config1").

<origin-servers>: A list of internal addresses of the form <ip>:<port> for all servers in the first cluster in the domain (or the domain if not clustered). There is one of these for each cluster in the domain (or just the domain itself if not clustered). For example 192.168.16.51:7001, 192.168.16.52:7001

<origin-server-pool-name>: A domain-unique name for an origin server pool. There is one of these for each cluster in the domain (or just the domain itself if not clustered).
To come up with this name it is recommended that the base identifier for all server pools be used in conjunction with the cluster index starting with 1 (use 1 for the unclustered case); For example "origin-server-pool-1" (domain/first cluster), "origin-server-pool-2" (second cluster, if any), etc.

<http-listener-name>: A domain-unique name for an http listener. There is one of these for each cluster in the domain (or just the domain itself if not clustered).
To come up with this name it is recommended that the base identifier for all http listeners be used in conjunction with the cluster index starting with 1 (use 1 for the unclustered case); For example "http-listener-1" (domain/first cluster), "http-listener-2" (second cluster, if any), etc.

Issue the following commands to configure the domain. If clustered, these commands will configure the domain and the first cluster:

```
<otd-home>/bin/tadm create-config -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-port>
-O=<origin-servers> -t=<wls-start-listen-port> -s=<wls-domain-vip> --server-user=oracle -evQ
<wls-domain-config-name>

<otd-home>/bin/tadm set-route-prop -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-
port> --config=<wls-domain-config-name> --vs=<wls-domain-config-name> --route=default-route
always-use-keep-alive=true

<otd-home>/bin/tadm set-thread-pool-prop -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-
admin-port> --config=<wls-domain-config-name> max-threads=4096

<otd-home>/bin/tadm set-health-check-prop -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-
admin-port> --config=<wls-domain-config-name> --origin-server-pool=<origin-server-pool-name>
dynamic-server-discovery=true

<otd-home>/bin/tadm set-http-listener-prop -u=<otd-admin-user> -w=<otd-pswd-file> --
config=<wls-domain-config-name> --http-listener=<http-listener-name> ip=<wls-domain-vip>
```

Per additional cluster (within current domain) logic:

The following needs to be done for additional each cluster (if any) within the current domain. For example, if there are three clusters in the domain this should be done for the second and third (the first was handled with the domain logic above).

This part requires the following additional values, which you should have available.

<wls-domain-listen-port>: The unique port number for the current cluster

<origin-servers>: See notes for this item under the above section.

<origin-server-pool-name>: See notes for this item under the above section.

<virtual-server-name>: A domain-unique name for a virtual server. There is one of these for each cluster in the domain except the first.
To come up with this name it is recommended that the base identifier for all http listeners be used in conjunction with the cluster index starting with 1. Note that because this name is not required for the first cluster, the index used will begin with 2. For example "http-listener-2" (second cluster, if any), "http-listener-3" (third cluster, if any), etc.

Issue the following commands to configure the cluster:

```
<otd-home>/bin/tadm create-origin-server-pool -u=<otd-admin-user> -w=<otd-pswd-file> -
p=<otd-admin-port> --type=http --config=<wls-domain-config-name> -O=<origin-servers>
<origin-server-pool-name>
```

```
<otd-home>/bin/tadm set-health-check-prop -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-
admin-port> --config=<wls-domain-config-name> --origin-server-pool=<origin-server-pool-name>
dynamic-server-discovery=true
```

```
<otd-home>/bin/tadm create-http-listener -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-
admin-port> --server-name=<wls-domain-vip> --listener-port=<wls-domain-listen-port> --
default-virtual-server-name=<wls-domain-config-name> --config=<wls-domain-config-name> --
ip=<wls-domain-vip> <http-listener-name>
```

```
<otd-home>/bin/tadm create-virtual-server -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-
admin-port> --http-listener=<http-listener-name> --origin-server-pool=<origin-server-pool-
name> --config=<wls-domain-config-name> <virtual-server-name>
```

```
<otd-home>/bin/tadm set-route-prop -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-
port> --config=<wls-domain-config-name> --vs=<virtual-server-name> --route=default-route
always-use-keep-alive=true
```

```
<otd-home>/bin/tadm set-http-listener-prop -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-
admin-port> --config=<wls-domain-config-name> --http-listener=<http-listener-name> default-
virtual-server-name=<virtual-server-name>
```

Additional logic for the current domain

After all the clusters have been configured, issue the following commands to complete the configuration of the domain within the admin server.

This part requires the following additional values, which you should have available.

<log-rotation interval>: Recommended value: 86400

<public-address-for-first-admin-node>: The public IP address you assigned to the first admin node.

<public-address-for-second-admin-node>: The public IP address you assigned to the second admin node.

<public-net-prefix-len>: The prefix length associated with your public network. For example if the netmask is 255.255.252.0 then the prefix length will be 22.

Issue the following commands:

```
<OTD_HOME>/bin/tadm deploy-config -u=<otd-admin-user> -w=<otd-pswd-file> <wls-domain-config-
name>
```

```
<OTD_HOME>/bin/tadm create-event -c=<wls-domain-config-name> -C=rotate-access-log -r=<log-
rotation interval> -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-port> -evQ
```

```
<OTD_HOME>/bin/tadm create-event -c=<wls-domain-config-name> -C=rotate-log -r=<log-rotation
interval> -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-port> -evQ
```

```
<OTD_HOME>/bin/tadm create-instance -c=<wls-domain-config-name> -u=<otd-admin-user> -w=<otd-
pswd-file> -p=<otd-admin-port>-evQ <public-address-for-first-admin-node> <public-address-
for-second-admin-node>
```

```
<OTD_HOME>/bin/tadm create-failover-group -c=<wls-domain-config-name> -i=<wls-domain-vip> -
z=<public-net-prefix-len> -n=<public-address-for-first-admin-node> -b=<public-address-for-
second-admin-node> -u=<otd-admin-user> -w=<otd-pswd-file> -p=<otd-admin-port> -evQ
```

```
<OTD_HOME>/bin/tadm start-instance -c=<wls-domain-config-name> -u=<otd-admin-user> -w=<otd-
pswd-file> -p=<otd-admin-port> -evQ <public-address-for-first-admin-node> <public-address-
for-second-admin-node>
```

Final logic after all domains have been configured

After all domains have been processed, issue the following command to complete the configuration of the admin server.

```
sed -i 's/<insert script>/startAdmin/g' /usr/sbin/otd-vm-start

rm -rf <otd-pswd-file>
```

Configuring the OTD admin nodes

The following should be done for each of the two admin nodes.

OTD node configuration requires the following additional values, which you should have available.

<adminserver-internal-ip>: This is the IP address returned from the *oakcli assign ip* command when the OTD admin server vm was created.

The nodes are dependent on the OTD admin server's http server. So wait until the following URL is accessible:

```
https://<otd-adminserver-internal-ip>:<otd-admin-port>
```

Now, install the admin server SSL cert to this location:

```
/root/.tadmtruststore
```

Now issue the following commands:

```
<OTD_HOME>/bin/tadm configure-server -n -H=<public-ip> -P=<otd-admin-port> -i=<otd-inst-
home> -r=root -j=<otd-home>/jdk -u=<otd-admin-user> -w=<otd-pswd-file> -h=<otd-internal-ip-
ofadminserver> -p=<otd-admin-port> -evQ

<otd-inst-home>/admin-server/bin/startserv

sed -i "s/<insert script>/startAdminAndInstance <WlsOda_OtdConfigurationName>/g"
/usr/sbin/otd-vm-start

rm -rf <otd-pswd-file>
```

Tearing Down OTD deployments(s)

This requires the following values, which you should have available.

<trust-store-file>: The admin server trust store file you created on each admin node during initial OTD configuration.

<otd-inst-home>: The directory representing the artifacts for this version of OTD.

<otd-config-name>: The name for the OTD configuration you provided during initial OTD configuration.

First on the admin nodes issue the following command:

```
<otd-instance-home>/net-<otd-config-name>/bin/stopserv
```

Then for all three servers do the following:

```
<otd-inst-home>/admin-server/bin/stopserv

rm -rf <otd-inst-home>
```

Finally, for the admin nodes delete the admin server trust store:

```
rm -f <trust-store-file>
```

Appendix A - A simple deployment example

This section describes the steps involved in doing a simple deployment. This deployment has the following components:

1. A WLS domain with two managed servers
2. Oracle Traffic Director

Teardown of the example deployment described in this is covered in Appendix B.

This appendix (and Appendix B) covers the creation (and removal) of the VMs. The details of configuring WLS and OTD are covered in subsequent appendices as follows.

Appendix C - WLS configuration.

Appendix D - WLS teardown.

Appendix E - OTD configuration

Appendix F - OTD teardown.

Note: In this example all resources are defined to minimally support the stated use case. In a real world situation you might want to create larger artifacts (for example CPU pools, repository) to support future expansion, for example to later add a WLS domain or managed server.

The steps to deploying this installation are listed, in order, in the following subsections.

Note: For simplicity, a specific platform is assumed. The assumed environment is ODA 4 with X5-2 server hardware. Note that it is possible for users to remove IB support from their X5-2 servers. This example assumes this is not done - i.e. IB is present.

Most of the steps are the same for all platforms. Items that are different in other supported environments are flagged as such. Refer to the instructions in the main sections of this document for details for your specific environment.

Create CPU pools on each node

The commands to create the CPU pools are the same regardless of what server or ODA platform you are using.

For our simple deployment we will need the following number of virtual cpus:

2	WLS admin server
4	WLS managed server 1
4	WLS managed server 2
2	OTD admin server
4	OTD admin node 1
4	OTD admin node 2

Thus we need a total of 20 virtual cpus, or 10 per ODA node.

Issue the following command to create CPU pools (named *wlsOdaCpuPool* in this example) on each node to support the configuration:

```
create cpupool wlsOdaCpuPool -numcpu 10 -node 0

create cpupool wlsOdaCpuPool -numcpu 10 -node 1
```

Note: The above commands should execute very quickly.

You can issue the following commands to insure that the pools were created as expected:

```
oakcli show cpupool -node 0

oakcli show cpupool -node 1
```

The output of the *show* command in each case should look something like the following.

```
    Pool                            CPU List
wlsOdaCpuPool           [0, 1, 2, 3, 4, 5, 6, 7, 16, 17 ]
default-unpinned-pool   [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]
odaBaseCpuPool          [8, 9, 10, 11, 12, 13, 14, 15]
```

Using this output insure that the pool named wlsOdaCpuPool was correctly created on each node, and that it contains the correct number of virtual CPUs (in this example, 10).

Create an ODA shared repository

The commands to create a shared repository are the same regardless of what server or ODA platform you are using.

Calculate the repository size using the following values:

17GB	WLS template
17GB	WLS admin server
17GB	WLS managed server 1
17GB	WLS managed server 2
15GB	OTD template
15GB	OTD admin server
15GB	OTD admin node 1
15GB	OTD admin node 2

The required size for the VMs is:

size = 4 * 17 + 4 * 15

= 128 GB

Apply a fudge factor of 10%

size = 128 * 1.1
= 140.8

= 141 GB

Now create a shared repository (named *wlsodarepo* in this example):

```
oakcli create repo wlsodarepo -dg data -size 141
```

Give it a few seconds to complete. The execution time will increase with the repo size.

You can use the following commands to verify the repo was created correctly and is online:

```
oakcli show repo wlsodarepo -node 0
oakcli show repo wlsodarepo -node 1
```

The output should appear as follows: (for node 0):

```
Resource: wlsodarepo_0
   AutoStart        : restore
   DG               : DATA
   Device           : /dev/asm/wlsodarepo-32
   ExpectedState    : Online
   FreeSpace        : 36356.1484375M
   MountPoint       : /u01/app/sharedrepo/wlsodarepo
   Name             : wlsodarepo_0
   Node             : all
   PFreeSpace       : 63.40
```

```
    RepoType          : shared
    Size              : 141000.0M
    State             : Online
    Version           : 2
```

Using this output, verify that the repository is shared, and is the correct size, and is online (which is automatic). Note that it may take a few minutes for it to come online.

Import templates for WLS and OTD

The commands to load a template are the same regardless of what server or ODA platform you are using.

Note: As described in the main document text, you can use the base template and then configure the applications manually, or you can create custom templates with the applications already installed. This example assumes we are starting from custom templates - one with WLS installed, and one with OTD installed.

Import the templates into our shared repo as follows:

```
oakcli import vmtemplate WLS_VMT -files /OVS/staging/WLS_VMT.tar.gz -repo wlsodarepo -node 0

oakcli import vmtemplate OTD_VMT -files /OVS/staging/OTD_VMT.tar.gz -repo wlsodarepo -node 0
```

Note: The above commands take a few minutes.

Note that the example commands use the best practices template naming suggested in the main part of the document.

Note that we only needed to load the templates using one node (in this example using node 0) because the repository is shared.

Create and configure each middleware VM

We need three VMs for WLS and three for OTD.

Here we assume that our WLS domain will be named *MyDomain*.

Be aware that the commands to start a VM vary slightly with the environment.

Use these commands to create and configure the VMs:

WLS admin server

```
oakcli clone vm MyDomain_AS -vmtemplate WLS_VMT -snap

oakcli configure vm MyDomain_AS -vcpu 2 -memory 2048 -maxmemory 2048 -cpupool wlsOdaCpuPool
-prefnode 0

oakcli assign ip -vm  MyDomain_AS
```

WLS managed server (first)

```
oakcli clone vm MyDomain_MS1 -vmtemplate WLS_VMT -snap

oakcli configure vm MyDomain_MS1 -vcpu 4 -memory 4096 -maxmemory 4096 -cpupool wlsOdaCpuPool
-prefnode 1

oakcli assign ip -vm  MyDomain_MS1
```

WLS managed server (second)

```
oakcli clone vm MyDomain_MS2 -vmtemplate WLS_VMT -snap

oakcli configure vm MyDomain_MS2 -vcpu 4 -memory 4096 -maxmemory 4096 -cpupool wlsOdaCpuPool
-prefnode 0

oakcli assign ip -vm  MyDomain_MS2
```

OTD admin server

```
oakcli clone vm OTD_AS -vmtemplate OTD_VMT -snap
oakcli configure vm OTD_MS1 -vcpu 2 -memory 1024 -maxmemory 1024 -cpupool wlsOdaCpuPool -
prefnode 1

oakcli assign ip -vm  MyDomain_MS1
```

OTD Admin Node (first)

```
oakcli clone vm OTD_AN1 -vmtemplate OTD_VMT -snap

oakcli configure vm OTD_AN1 -vcpu 4 -memory 8192 -maxmemory 8192 -cpupool wlsOdaCpuPool -
prefnode 0

oakcli assign ip -vm  OTD_AN1
```

OTD Admin Node (second)

```
oakcli clone vm OTD_AN2 -vmtemplate OTD_VMT -snap

oakcli configure vm OTD_AN2 -vcpu 4 -memory 8192 -maxmemory 8192 -cpupool wlsOdaCpuPool -
prefnode 0

oakcli assign ip -vm  OTD_AN2
```

Note that the above commands use the VM naming conventions recommended in the main sections of this document.

Note also that, as recommended, the commands evenly distribute the VMs across the two ODA nodes.

After you create and configure each VM you can use the following command for each VM to verify that it worked:

```
  oakcli show vm <vm-name>
```

The output of the above command will look something like this (this example is for the WLS admin server). This has some of the expected fields highlighted.

```
Resource: MyDomain_AS
    AutoStart         : restore
    CPUPriority       : 100
    Disks             : |file:/OVS/Repositories/wlsodarepo/
                        .ACFS/snaps/WL_DefaultDomain_AS/Vir
                        tualMachines/WL_DefaultDomain_AS/Sy
                        stem.img,xvda,w||file:/OVS/Reposito
                        ries/wlsodarepo/.ACFS/snaps/WL_Defa
                        ultDomain_AS/VirtualMachines/WL_Def
                        aultDomain_AS/Weblogicserver.img,xv
                        db,w||file:/OVS/Repositories/wlsoda
                        repo/.ACFS/snaps/WL_DefaultDomain_A
                        S/VirtualMachines/WL_DefaultDomain_
                        AS/U01.img,xvdc,w|
    Domain            : XEN_PVM
    ExpectedState     : offline
    FailOver          : false
    IsSharedRepo      : true
    Keyboard          : en-us
    MaxMemory         : 2048
    MaxVcpu           : 2
    Memory            : 2048
    Mouse             : OS_DEFAULT
    Name              : WL_DefaultDomain_AS
    Networks          : ['type=netfront,bridge=priv1','type
                        =netfront,bridge=net1','type=netfro
                        nt,bridge=net2']
    NodeNumStart      : 0
```

```
        OS                : OL_5
        PrefNodeNum       : 0
        PrivateIP         : 192.168.16.51
        ProcessorCap      : 100
        RepoName          : wlsodarepo
        Source            : 0
        State             : Offline
        TemplateName      : WLS_VMT
        VDisks            : |0|
        Vcpu              : 2
        cpupool           : wlsOdaCpuPool
        vncport           : 5901
```

The VMs will be offline until you explicitly start them.

Platform notes:

1. The memory sizes used are for existing supported environments and might not be sufficient for later versions.
2. If your ODA version is less that version 12 the *oakcli clone* commands will be different. See the main sections of this document for details.
3. If your server hardware is earlier than X5-2 the *oakcli configure* commands will be different. See the main sections of this document for details.
4. If your server hardware has IB support you will need to issue additional *oakcli modify* commands to add networks. See the main sections of this document for details.

Start each middleware VM

After creating and configuring the VMs they can be started using the following commands.

Note that the commands to start a VM are the same regardless of what server or ODA platform you are using.

```
oakcli start vm MyDomain_AS

oakcli start vm MyDomain_MS1

oakcli start vm MyDomain_MS2

oakcli start vm OTD_AS

oakcli start vm OTD_AN1

oakcli start vm OTD_AN2
```

You can use the following command to monitor when the VMs come online:

```
oakcli show vm <vm-name>
```

Configure your middleware applications

See Appendix C for an example of WLS configuration
See Appendix E for an example of OTD configuration

Appendix B - Example of tearing down a simple deployment

This section tears down the deployment described in Appendix 1.
The steps to deploying this domain are listed, in order, in the following subsections.

Note that the commands to teardown a deployment are the same regardless of what server or ODA platform you are using.

Do any required WLS and/or OTD cleanup.

See Appendix D for an example of WLS cleanup.
See Appendix F for an example of OTD cleanup.

Stop all associated VMs

The following commands stop all the VMs in the deployment:

```
oakcli stop vm MyDomain_AS -force

oakcli stop vm MyDomain_MS1 -force

oakcli stop vm MyDomain_MS2 -force

oakcli stop vm OTD_AS -force

oakcli stop vm OTD_AN1 -force

oakcli stop vm OTD_AN1 -force
```

After stopping the VMs, you can use the following command to verify that each VM is in the Offline state:

```
 oakcli show vm <vm-name>
```

Release the allocated IP for each VM

Issue the following commands to free the IP allocated for each VM:

```
oakcli release ip -vm MyDomain_AS

oakcli release ip -vm MyDomain_MS1

oakcli release ip -vm MyDomain_MS2

oakcli release ip -vm OTD_AS

oakcli release ip -vm OTD_AN1

oakcli release ip -vm OTD_AN2
```

After releasing the IPs you can use the following command to verify that the IPs are released:

```
oakcli show vm
```

Delete each VM

Issue the following commands to delete the VMs:

```
oakcli delete vm MyDomain_AS -force

oakcli delete vm MyDomain_MS1 -force

oakcli delete vm MyDomain_MS2 -force

oakcli delete vm OTD_AS -force
oakcli delete vm OTD_AN1 -force

oakcli delete vm OTD_AN1 -force
```

After deleting the VMs if you can use the following command to verify that the VMs are no longer registered:

```
oakcli show vm
```

Delete the loaded templates

Issue the following commands to delete the templates:

```
oakcli delete vmtemplate WLS_VMT

oakcli delete vmtemplate OTD_VMT
```

After deleting the templates you can use the following command to verify that the templates are no longer registered:

```
oakcli show vmtemplate
```

Delete the CPU pools

Issue the following commands to remove the CPU pools:

```
oakcli delete cpupool wlsOdaCpuPool -node 0

oakcli delete cpupool wlsOdaCpuPool -node 1
```

After deleting the CPU pools you can use the following command to verify that the pools are no longer registered:

```
oakcli show cpupool -node <node-num>
```

Stop the shared repository

Issue the following commands to stop the shared repository:

```
oakcli stop repo wlsodrepo -node 0

oakcli stop repo wlsodrepo -node 1
```

After stopping the repository you can use the following command to verify the results:

```
oakcli show repo wlsodrepo
```

Delete the shared repository

Use the following command to delete the shared repository:

```
oakcli delete repo wlsodrepo
```

The above command deletes the repo from both nodes (i.e. shared).

After deleting the shared repo you can use the following command to verify that the repo is no longer registered:

```
oakcli show repo
```

Appendix C - Example of configuring WLS for a simple configuration

This section illustrates the steps one might use to configure WLS for the minimal configuration described in Appendix A (X5-2, IB one WLS domain, one WLS cluster with two managed servers).

This example assumes that you installed the WLS software into a custom WLS template and used that template to create the VM.

The following WLS specific values are assumed:

- The WLS version is 12.1.3
- The password for all user accounts is welcome1
- The temporary installation password file is /tmp/.otdpasswd
- The domain's public network is eth1
- The internal netmask is 255.255.255.0
- The internal gateway is 10.133.40.1
- The public netmask is 255.255.252.0
- The public gateway is 10.133.50.1
- The private IP assigned to the WLS admin server is 192.168.16.51
- The private IP assigned to the first WLS managed server is 192.168.16.52
- The private IP assigned to the second WLS managed server is 192.168.16.53
- The public IP for the WLS admin server is 10.133.42.51
- The public IP for the WLS first managed server is 10.133.42.52
- The public IP for the WLS second managed server is 10.133.42.63
- The listen port for WLS servers is 7001
- The network domain name (for fully qualified hosts) is my.domain.com
- The hostname for the WLS admin server is wlsadmin
- The hostname for the WLS first managed server is wlsms1
- The hostname for the WLS second managed server is wlsms2
- The DNS servers are 130.35.249.52 130.35.249.41
- The NTP server is 129.6.15.28
- The WLS_HOME is /opt/oracle/middleware/wlserver
- The WLS domains root is /u01
- The node manager home is /u01/MyDomain
- The server IDs are 1 and 2 respectively for the two managed servers

Optimize the environment on each WLS VM (this section applies to all servers)

Note the following settings are only for systems that support IB (must be at least X5-2). For non-IB systems no optimizations are recommended.

```
grep_sed_util "MLX4_XEN_FMR_LOAD" "MLX4_XEN_FMR_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "MLX4_VNIC_LOAD" "MLX4_VNIC_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "SDP_LOAD" "SDP_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "SRP_LOAD" "SRP_LOAD=no" "/etc/ofed/openib.conf"

grep_sed_util "ISER_LOAD" "ISER_LOAD=no" "/etc/ofed/openib.conf"

echo "alias ib0 mlx4_en">>/etc/modprobe.conf

echo "alias ib1 mlx4_en">>/etc/modprobe.conf

grep_sed_util "tsc" "      kernel \/vmlinuz-2.6.39-400.245.1.el5uek ro root=LABEL=\/ nohpet
nopmtimer tsc=reliable hda=noprobe hdb=noprobe ide0=noprobe numa=off pci=noaer,nocrs
console=tty console=ttyS0,115200n8 selinux=0 nohz=off crashkernel=256M@64M loglevel=7
panic=60 ipv6.disable=1" "/boot/grub/grub.conf"
```

Set up the network (this section applies to all servers)

Issue the following commands for each of the three servers.

```
 . /etc/sysconfig/network-scripts/network-functions



//configure private interface

service openibd start

chkconfig openibd on

service openibd status

configure_master 192.168.16.51 255.255.255.0
```

Note: The above commands will be different for non IB platforms. See the main sections of this document for details.

Now do the following for the servers mentioned:

For admin server

```
createNetworkConfigFile eth1 10.133.42.51 255.255.252.0 10.133.50.1

set_hostname wlsadmin.my.domain.com

export HOSTNAME=wlsadmin.my.domain.com

echo "192.168.16.51 $HOSTNAME ${HOSTNAME%%.*}" >>/etc/hosts

echo "10.133.42.51 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts"
```

```
# include managed servers

echo "192.168.16.52 wlsms1.my.domain.com wlsms1" >> /etc/hosts

echo "192.168.16.53 wlsms2.my.domain.com wlsms2" >> /etc/hosts
```

For first managed server

```
createNetworkConfigFile eth1 10.133.42.52 255.255.252.0 10.133.50.1

set_hostname wlsms1.my.domain.com

export HOSTNAME=wlsms1.my.domain.com

export ADMIN_HOSTNAME=wlsadmin.my.domain.com

echo " 10.133.42.52 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "192.168.16.52 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "192.168.16.51 $ADMIN_HOSTNAME ${ADMIN_HOSTNAME%%.*}" >> /etc/hosts
```

For second managed server

```
createNetworkConfigFile eth1 10.133.42.53 255.255.252.0 10.133.50.1

set_hostname wlsms2.my.domain.com

export HOSTNAME=wlsms2.my.domain.com

export ADMIN_HOSTNAME=wlsadmin.my.domain.com

echo " 10.133.42.53 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "192.168.16.53 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

echo "192.168.16.51 $ADMIN_HOSTNAME ${ADMIN_HOSTNAME%%.*}" >> /etc/hosts
```

Now issue the following for all three WLS servers

```
sed -i -e "s/^HOSTNAME=.*/HOSTNAME=$HOST_NAME/g" /etc/sysconfig/network

echo "nameserver " 130.35.249.52 > /etc/resolv.conf

echo "nameserver " 130.35.249.41 > /etc/resolv.conf

echo "search ${HOSTNAME#*.}" >> /etc/resolv.conf

  /sbin/service network restart
```

Set up time services (this section applies to all servers)

Issue the following command on all three servers:

```
ntpdate 129.6.15.28 2>&1
```

Create the domain (this section applies to the admin server only)

Do this on the admin server:

```
mkdir -p /u01

/opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --createDomain /u01 "/opt/oracle/middleware/wlserver" 1
```

Generate Demo Cert (this section applies to all servers)

On all three servers use the WLS java utilities to generate a demo certificate:

```
. /opt/oracle/middleware/wlserver/server/bin/setWLSEnv.sh

java utils.CertGen -keyfile privatekeyfile -keyfilepass DemoIdentityPassPhrase -certfile certfile -cn $HOSTNAME

java utils.der2pem /opt/oracle/middleware/wlserver/server/lib/CertGenCA.der

java utils.ImportPrivateKey -keystore DemoIdentity.jks -storepass DemoIdentityKeyStorePassPhrase -keyfile privatekeyfile.pem -keyfilepass DemoIdentityPassPhrase -certfile newcerts.pem -alias DemoIdentity

cp DemoIdentity.jks /opt/oracle/middleware/wlserver/server/lib
```

Set up log files (this section applies to the admin server only)

Do the following on the admin server:

```
mkdir -p /u01/MyDomain/servers/AdminServer/logs
```

Set up boot.properties (this section applies to the admin server only)

Do the following on the admin server:

```
mkdir -p /u01/MyDomain/servers/AdminServer/security

echo "username=system" >> /u01/MyDomain/servers/AdminServer/security/boot.properties

echo "password=welcome1" >> /u01/MyDomain/servers/AdminServer/security/boot.properties

chown -R oracle:oinstall /u01
```

Set up the node manager (this section applies to the managed servers only)

Do the following for both managed servers:

```
mkdir -p /u01/MyDomain

export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust -Dssl.debug=true -Dweblogic.StdoutDebugEnabled=true -Dweblogic.security.SSL.verbose=true"

/opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-functions.py --setupNodeManager /u01 /u01/MyDomain 1

export JAVA_OPTIONS="-Dverbose=true -DStartScriptEnabled=false -DCrashRecoveryEnabled=true -DListenPort=5556"
```

Note: If the WLS version were not 12.1.3 the above command would be different. See the main sections of this document for details.

```
cp /opt/oracle/middleware/wlserver/server/bin/startNodeManager.sh /u01/MyDomain
```

```
sed -i "s/NODEMGR_HOME=.*\/common\/nodemanager.*/NODEMGR_HOME=\\u01\\MyDomain/g"
$NODE_MANAGER_HOME/startNodeManager.sh
```

```
cp /opt/oracle/middleware/wlserver/server/lib/DemoIdentity.jks /u01/MyDomain/security
```

Note: The above command is only required for WLS 12.1.2 and 12.1.3.

```
chown -R oracle:oinstall /u01
```

```
su oracle -c "/u01/MyDomain/startNodeManager.sh  > /u01/MyDomain/nodemanager.out 2>&1&"
```

Start the server (this section applies to all servers)

Admin server

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust"
```

```
chown -R oracle:oinstall /u01
```

```
su - oracle -c "nohup /u01/MtDomain/bin/startWebLogic.sh >
/u01/MtDomain/servers/AdminServer/logs/AdminServer.out 2>&1&"
```

```
sed -i "s/<insert script>/startAdmin MyDomain/g" /usr/sbin/wls-vm-start
```

First managed server

```
chown -R oracle:oinstall /u01
```

```
/opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-
functions.py --startManagedServer 1 1
```

```
sed -i 's/<insert script>/startNodeManager/g' /usr/sbin/wls-vm-start
```

Second managed server

```
chown -R oracle:oinstall /u01
```

```
/opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-
functions.py --startManagedServer 2 1
```

```
sed -i 's/<insert script>/startNodeManager/g' /usr/sbin/wls-vm-start
```

Set up node manager (this section applies to the admin server only)

Do the following on the admin server:

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust"
```

```
/opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-
functions.py --modifyNodeManager welcome1 MyDomain "system" welcome1 "12.1.3" 1
```

Configure JMS distributed distributions (this section applies to the admin server only)

Do the following on all admin servers:

```
nohup /opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning
/usr/sbin/wlst-functions.py --setupDistributedDestinations welcome1 1
```

Appendix D - Example of tearing down WLS for a simple configuration

This section continues the example from the previous appendices and uses the same values.

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=DemoTrust"

/opt/oracle/middleware/wlserver/common/bin/wlst.sh -skipWLSModuleScanning /usr/sbin/wlst-
functions.py --shutdownDomain welcome1 12.1.3 1
```

Appendix E - Example of configuring OTD for a simple configuration

This section illustrates the steps one might use to configure OTD for the minimal configuration described in Appendix A (X5-2, IB one WLS domain, one WLS cluster with two managed servers).

This section assumes that you already installed the OTM software on the three OTD VMs.

The same WLS configuration and domain data (IP addresses, etc) provided in Appendix D are assumed here.

In addition the following OTD specific values are assumed:

- The password for all user accounts is “welcome1”
- The temporary installation password file is /tmp/.otdpasswd
- The internal netmask is 255.255.255.0
- The internal gateway is 10.133.40.1
- The public netmask is 255.255.252.0
- The public gateway is 10.133.50.1
- The network domain name (for fully qualified hosts) is my.domain.com
- The private IP for the OTD admin server is 192.168.16.61
- The private IP for the OTD first admin node is 192.168.16.62
- The private IP for the OTD second admin node is 192.168.16.63
- The public IP for the OTD admin server is 10.133.42.61
- The public IP for the OTD first admin node is 10.133.42.62
- The public IP for the OTD second admin node is 10.133.42.63
- The hostname for the OTD admin server is otdhost1
- The hostname for the OTD first admin node is otdhost2
- The hostname for the OTD second admin node is otdhost3
- The DNS servers are 130.35.249.52 130.35.249.41
- The NTP server is 129.6.15.28
- The OTD admin port is 8989
- Thee OTD admin user name is admin
- The OTD_HOME is /opt/oracle/trafficdirector
- The OTD instance home is /u01/OTDInstanceHome
- The OTD virtual IP for the WLS domain is 10.133.42.64
- The OTD config name for the one WLS domain is otd-config1
- The OTD http listener for the one WLS domain is http-listener-1
- The origin server pool for the one WLS domain is origin-server-pool-1
- The OTD log rotation interval is 86400

Set up the user accounts

```
echo welcome1 | root --stdin root

echo welcome1 | oracle --stdin oracle
```

Optimize the environment on each OTD VM

Do the following for all three servers

```
grep_sed_util "net.ipv4.ip_nonlocal_bind" "net.ipv4.ip_nonlocal_bind = 1"

  "/etc/sysctl.conf"

grep_sed_util "fs.file-max" "fs.file-max=6291456" "/etc/sysctl.conf"

grep_sed_util "fs.nr_open" "fs.nr_open=2097152" "/etc/sysctl.conf"

grep_sed_util "net.ipv4.tcp_fin_timeout" "net.ipv4.tcp_fin_timeout=30"

  "/etc/sysctl.conf"

   /sbin/sysctl-p
```

```
grep_sed_util "oracle hard nofile" "oracle hard nofile 2097152"
  "/etc/security/limits.conf"
grep_sed_util "oracle soft nofile" "oracle soft nofile 1024"
  "/etc/security/limits.conf"
echo "ulimit -n 131072">> ~oracle/.bash_profile
echo "ulimit -n 131072">> ~root/.bash_profile
```

Note the following settings are only for X5-2 (and possibly any later server versions that support IB):

/etc/ofed/openib.conf parameters:

```
grep_sed_util "MLX4_XEN_FMR_LOAD" "MLX4_XEN_FMR_LOAD=no" "/etc/ofed/openib.conf"
grep_sed_util "MLX4_VNIC_LOAD" "MLX4_VNIC_LOAD=no" "/etc/ofed/openib.conf"
grep_sed_util "SDP_LOAD" "SDP_LOAD=no" "/etc/ofed/openib.conf"
grep_sed_util "SRP_LOAD" "SRP_LOAD=no" "/etc/ofed/openib.conf"
grep_sed_util "ISER_LOAD" "ISER_LOAD=no" "/etc/ofed/openib.conf"
```

/etc/ofed/openib.conf parameters:

```
echo "alias ib0 mlx4_en">>/etc/modprobe.conf
echo "alias ib1 mlx4_en">>/etc/modprobe.conf
```

Set up the network

Issue the following commands for each of the three servers.

```
 . /etc/sysconfig/network-scripts/network-functions


//configure private interface
service openibd start
chkconfig openibd on
service openibd status
```

> *Note: The above commands will be different for non IB platforms. See the main sections of this document for details.*

Do the following for the servers mentioned:

Admin server

```
#Configure private infiniband network...
```

```
configure_slave "ib0"

configure_slave "ib1"

configure_master 192.168.16.61 255.255.255.0
```

Note: If the environment does not have IB support, you will instead configure eth0 as the private interface. See the main sections of this document for details.

```
createNetworkConfigFile eth1 10.133.42.61 255.255.252.0 10.133.50.1

set_hostname otdhost1.my.domain.com

export HOSTNAME=otdhost1.mydomain.domain.com

sed -i -e "s/^HOSTNAME=.*/HOSTNAME=otdhost1/g" /etc/sysconfig/network

echo "nameserver " 130.35.249.52 > /etc/resolv.conf

echo "nameserver " 130.35.249.41 > /etc/resolv.conf

echo "search ${HOSTNAME#*.}" >> /etc/resolv.conf

echo "10.133.42.61 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

/sbin/service network restart
```

First admin node

```
configure_slave "ib0"

configure_slave "ib1"

configure_master 192.168.16.62 255.255.255.0
```

Note: If the environment does not have IB support, you will instead configure eth0 as the private interface. See the main sections of this document for details.

```
createNetworkConfigFile eth1 10.133.42.62 255.255.252.0 10.133.50.1

set_hostname otdhost2.my.domain.com

export HOSTNAME=otdhost2.my.domain.com

sed -i -e "s/^HOSTNAME=.*/HOSTNAME=otdhost2/g" /etc/sysconfig/network

echo "nameserver " 130.35.249.52 > /etc/resolv.conf

echo "nameserver " 130.35.249.41 > /etc/resolv.conf

echo "search ${HOSTNAME#*.}" >> /etc/resolv.conf

echo "10.133.42.62 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

/sbin/service network restart
```

Second admin aode

```
configure_slave "ib0"

configure_slave "ib1"

configure_master 192.168.16.63 255.255.255.0
```

> *Note: If the environment does not have IB support, you will instead configure eth0 as the private interface. See main document section for details.*

```
createNetworkConfigFile eth1 10.133.42.63 255.255.252.0 10.133.50.1

set_hostname otdhost3.my.domain.com

export HOSTNAME=otdhost3.my.domain.com

sed -i -e "s/^HOSTNAME=.*/HOSTNAME=otdhost3/g" /etc/sysconfig/network

echo "nameserver " 130.35.249.52 > /etc/resolv.conf

echo "nameserver " 130.35.249.41 > /etc/resolv.conf

echo "search ${HOSTNAME#*.}" >> /etc/resolv.conf

echo "10.133.42.63 $HOSTNAME ${HOSTNAME%%.*}" >> /etc/hosts

/sbin/service network restart
```

Set up time services

```
ntpdate 129.6.15.28 2>&1
```

Create password file

```
 echo "tadm_password=welcome1" > /tmp/.otdpasswd

chmod 0600 /tmp/.otdpasswd
```

Configure the OTD Servers

Step 1- Begin the OTD admin server configuration

On the admin server:

```
/opt/oracle/trafficdirector/bin/tadm configure-server -i=/u01/OTDInstanceHome -r=root -
j=/opt/oracle/trafficdirector/jdk -u=admin -w=/tmp/.otdpasswd -p=8989 -evQ

/u01/OTDInstanceHome/admin-server/bin/startserv
```

Now wait until the following URL (admin server admin http server) is available before continuing:

```
https://192.168.16.61:8989
```

Step 2 - Do the OTD admin node configurations

On each admin node install the admin server SSL cert to this location:

```
/root/.tadmtruststore
```

Now do the following commands for the server mentioned.

First admin node

```
/opt/oracle/trafficdirector/bin/tadm configure-server -n -H=10.133.42.62 -P=8989 -
i=/u01/OTDInstanceHome -r=root -j=/opt/oracle/trafficdirector/jdk -u=admin -
w=/tmp/.otdpasswd -h=192.168.16.61 -p=8989 -evQ
```

```
/u01/OTDInstanceHome/admin-server/bin/startserv
```

```
sed -i "s/<insert script>/startAdminAndInstance otd-config/g" /usr/sbin/otd-vm-start
```

```
# password file no longer needed
```

```
rm -rf /tmp/.otdpasswd
```

Second admin node

```
/opt/oracle/trafficdirector/bin/tadm configure-server -n -H=10.133.42.63 -P=8989 -
i=/u01/OTDInstanceHome -r=root -j=/opt/oracle/trafficdirector/jdk -u=admin -w=tmp/.otdpasswd
-h=192.168.16.61 -p=8989 -evQ
```

```
/u01/OTDInstanceHome/admin-server/bin/startserv
```

```
sed -i "s/<insert script>/startAdminAndInstance otd-config/g" /usr/sbin/otd-vm-start
```

```
rm -rf /tmp/.otdpasswd
```

Step 3 - Finish the OTD admin server configuration

```
/opt/oracle/trafficdirector/bin/tadm create-config -u=admin -w=/tmp/.otdpasswd -p=8989 -
O=192.168.16.51:7001,192.168.16.52:7001 -t=5353 -s=10.133.42.64 --server-user=oracle -evQ
otd-config1
```

```
/opt/oracle/trafficdirector/bin/tadm set-route-prop -u=admin -w=/tmp/.otdpasswd -p=8989 --
config=otd-config1 --vs=otd-config1 --route=default-route always-use-keep-alive=true
```

```
/opt/oracle/trafficdirector/bin/tadm set-thread-pool-prop -u=admin -w=/tmp/.otdpasswd -
p=8989 --config=otd-config1 max-threads=4096
```

```
/opt/oracle/trafficdirector/bin/tadm set-health-check-prop -u=admin -w=/tmp/.otdpasswd -
p=8989 --config=otd-config1 --origin-server-pool=origin-server-pool-1 dynamic-server-
discovery=true
```

```
/opt/oracle/trafficdirector/bin/tadm set-http-listener-prop -u=admin -w=/tmp/.otdpasswd --
config=otd-config1 --http-listener=http-listener-1 ip=10.133.42.64
```

Note that additional commands would be needed here if the domain being configured contained multiple clusters - or if we had additional domains. Refer to the main document text for details.

```
 /opt/oracle/trafficdirector/bin/tadm deploy-config -u=admin -w=/tmp/.otdpasswd  otd-config1
```

```
/opt/oracle/trafficdirector/bin/tadm create-event -c=otd-config1 -C=rotate-access-log -
r=86400 -u=admin -w=/tmp/.otdpasswd -p=8989 -evQ
```

```
/opt/oracle/trafficdirector/bin/tadm create-event -c=otd-config1 -C=rotate-log -r=86400 -
u=admin -w=/tmp/.otdpasswd -p=8989 -evQ
```

```
/opt/oracle/trafficdirector/bin/tadm create-instance -c=otd-config1 -u=admin -
w=/tmp/.otdpasswd -p=8989 -evQ 10.133.42.62 10.133.42.63

/opt/oracle/trafficdirector/bin/tadm create-failover-group -c=otd-config1 -i=10.133.42.64 -
z=22 -n=10.133.42.62 -b= 10.133.42.63 -u=admin -w=/tmp/.otdpasswd -p=8989 -evQ

/opt/oracle/trafficdirector/bin/tadm start-instance -c=otd-config1 -u=admin -
w=/tmp/.otdpasswd -p=8989 -evQ 10.133.42.62  10.133.42.63

sed -i 's/<insert script>/startAdmin/g' /usr/sbin/otd-vm-start

# password file no longer needed

rm -rf /tmp/.otdpasswd
```

Appendix F - Example of tearing down OTD for a simple configuration

This example tears down the OTD deployment described in Appendix E.

First on the admin nodes issue the following command:

```
/u01/OTDInstanceHome/net-otd-config1/bin/stopserv
```

Then for all three servers do the following:

```
/u01/OTDInstanceHome/admin-server/bin/stopserv

rm -rf /u01/OTDInstanceHome
```

Finally, for the admin nodes only delete the admin server trust store:

```
rm -f /root/.tadmtruststore
```

Conclusion

Having followed the above configuration steps, you will have built middleware applications on an Oracle Database Appliance machine.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Hardware and Software, Engineered to Work Together

Franklin Simpson

Copyright © 2015, 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. 0815

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