ORACLE

APPLICATION EXPRESS

An Oracle White Paper
May 2014

Example Web Listener Deployment for Oracle Application Express

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Executive Overview

Oracle Application Express is a declarative, rapid web application development tool for the Oracle Database. It is a fully supported, no cost option available with all editions of the Oracle Database. Oracle Application Express enables you to create database-centric Web applications that are reliable, scalable, and secure.

Successful enterprise deployment of Oracle Application Express requires forethought around planning and installing the required software. The Example Web Listener Deployment for Oracle Application Express white paper provides a step-by-step approach to implement the suggested architecture for enterprise deployment of the web tier for Oracle Application Express.

This white paper defines the key components involved in the middle-tier deployment and includes the relevant steps and guidelines to integrate these components.

Introduction

Oracle Application Express utilizes a simple architecture where pages are dynamically generated using metadata stored within the Oracle Database. There is no code generation or file based compilation. Once fully installed and configured, the Uniform Resource Locator (URL) will be defined for both developers and end users to access Oracle Application Express. Users require only a Web browser to build, develop, and deploy Oracle Application Express applications. No additional client software is required.

The Oracle Application Express architecture requires some form of Web server to proxy requests between a client Web browser and the Oracle Application Express engine. The Web server options include:

- **Oracle REST Data Services**: Oracle REST Data Services (ORDS) is a Java based, free tool that is fully supported when used with Oracle WebLogic Server and Oracle Glassfish Server. Oracle REST Data Services was formerly known as Oracle Application Express Listener prior to Release 2.0.6.
- **Oracle HTTP Server**: The Oracle HTTP Server (Apache) with mod_plsql plugin can be placed on the same physical machine as the database, or on a separate physical machine.
- **Embedded PL/SQL Gateway**: The Embedded PL/SQL Gateway (EPG) runs in the Oracle XML DB Protocol Server within the Oracle Database and includes the core features of mod_plsql.

When designing enterprise-class or mission critical deployments, for Oracle Application Express, it must be architected to incorporate appropriate scalability and redundancy. Therefore you should consider:

- Separating the database tier and middle-tierChoosing an HTTP Server
- Selecting the appropriate middle-tier components and setting up the middle-tier architecture

- Setting up multiple Application Express instances on a single server that share the same middle-tier infrastructure

The Example Web Listener Deployment for Oracle Application Express white paper provides a template solution for these requirements. By using the detailed steps provided in this white paper, you should be able to build the suggested architecture for a successful enterprise deployment of Oracle Application Express.

Prerequisites

The solution presented in this white paper requires two different servers, for example Mid Tier and Database Tier. The operating system for these two servers in the example is Oracle Enterprise Linux 5.

Oracle Application Express requires an Oracle Database. If you are performing a single deployment of Oracle Application Express then, you should have downloaded and installed an Oracle Database. The example covered in the *Performing a Single Deployment of the Web Tier* section requires an Oracle Database installed on the Database Tier server and the SID is **orcl1**.

If you are performing multiple deployments of Oracle Application Express then, you should have multiple database installations. The example discussed in *Performing Multiple Deployments of the Web Tier* section needs two Oracle Application Express instances. Therefore, this example requires two database installations on the Database Tier server: one with the SID **orcl1** and the other one with the SID **orcl2**.

You can download Oracle Database software from the Downloads page on Oracle Technology Network. For more information on how to install Oracle Database, see *Oracle Database Installation Guide*.

The steps and screenshots provided in this white paper use the following software installed on the Mid Tier server:

- Oracle REST Data Services 2.0.6
- Oracle WebLogic Server 12c (12.1.2)
- Oracle HTTP Server 12c (12.1.2)

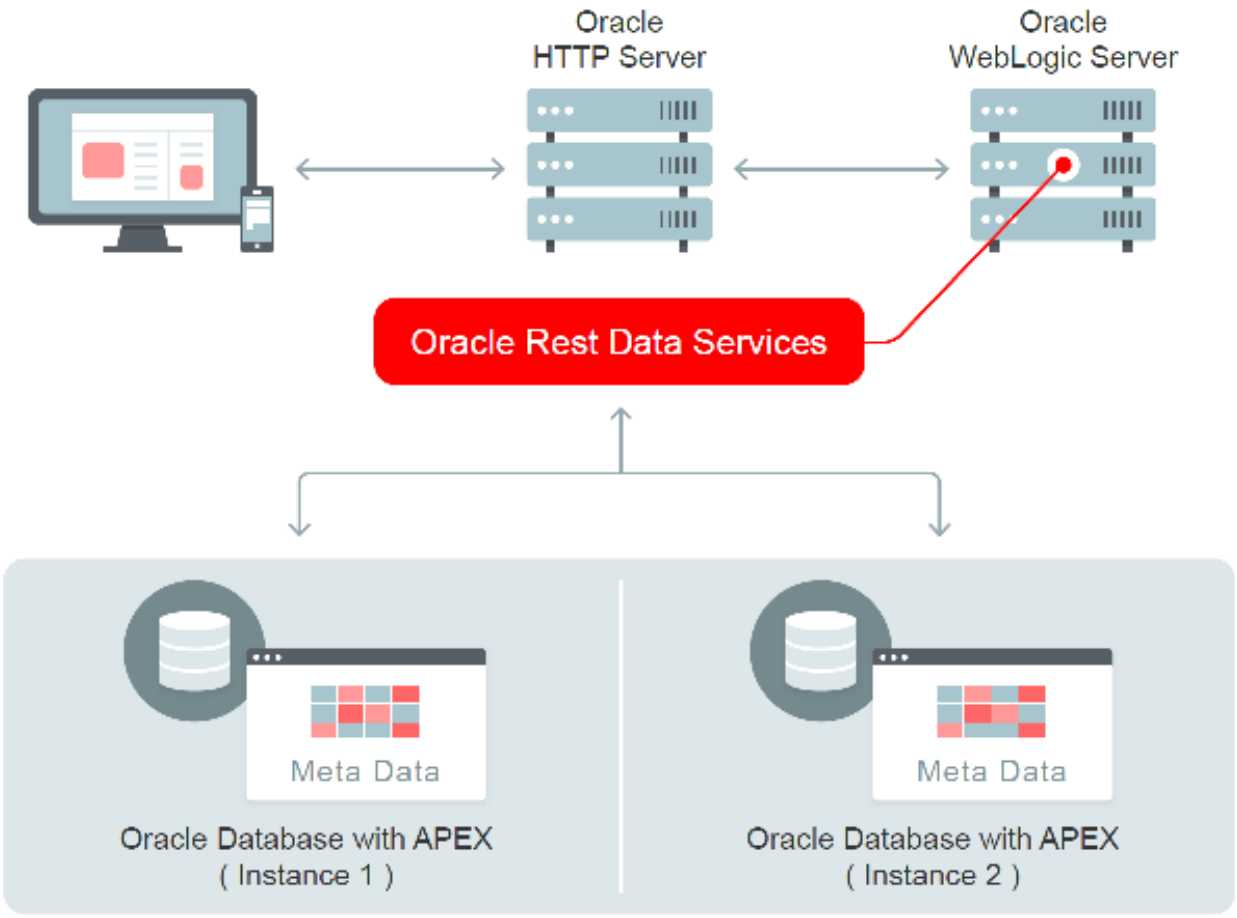
On the Database Tier server:

- Oracle Application Express 4.2.4
- Oracle Database 11g Release 2 (11.2.0.4)

The screenshots and steps might be different for later versions of the above Oracle products.
For more information on where to download the software see *Planning the Required Software* section.

Web Listener Deployment for Oracle Application Express

The example middle-tier architecture in this document is to utilize Oracle REST Data Services installed on Oracle WebLogic Server. Oracle HTTP Server acts as a front end to Oracle WebLogic Server.



Oracle REST Data Services as the Preferred Web Listener

The Oracle REST Data Services is Java EE-based and is the preferred Web listener option. The Java EE implementation offers increased functionality including a command line based configuration, enhanced security, file caching, and RESTful Web Services. Oracle REST Data Services also provides increased flexibility by supporting deployments using Oracle WebLogic Server, Oracle GlassFish Server, Apache Tomcat, and a standalone mode. The Oracle REST Data Services (Release >= 2.0) includes FOP support and is part of the reference architecture used to operate the Oracle Database Cloud Service.

Oracle REST Data Services communicates directly with the Oracle Application Express engine. It acts as a communication broker between the Web server and the Oracle Application Express objects

in the Oracle Database. More specifically, Oracle REST Data Services maps browser requests to Database stored procedure calls.

OHS as a Front-end to WebLogic

In this example architecture Oracle HTTP Server (OHS) acts a front-end solution to WebLogic and includes:

- An Apache HTTP Server to handle incoming requests and route them to the appropriate processing utility.
- Apache HTTP Modules (mods), to implement and extend the basic functionality of Oracle HTTP Server.
  Many of the standard Apache modules are included with Oracle HTTP Server. Oracle also includes several modules that are specific to Oracle Fusion Middleware to support integration between Oracle HTTP Server and other Oracle Fusion Middleware components.

The mod_wl_ohs module, which is included in Oracle HTTP Server, routes requests to Oracle WebLogic Server. This module also provides load balancing functionality with the Oracle WebLogic Server. Additionally, OHS can be used to manage and serve the static content (images, CSS files, etc.) of an Oracle Application Express instance.

Planning the Required Software

Before performing the tasks in this paper, you should have an Oracle Database installed on the Database Tier server. For more information, see the *Prerequisites* section.

To build the architecture discussed in this white paper, you need the following software:

- Oracle Application Express
- Oracle REST Data Services
- Oracle WebLogic Server
- Oracle HTTP Server
- Java Development Kit (JDK)

Note:

- Oracle Application Express is a no-cost feature of the Oracle Database which is fully supported as part of your Oracle Database maintenance agreement.
- If installed on the same physical machine as the database, then the Oracle HTTP Server is included as part of the restricted-use license included with the Oracle Database license. However, in this example architecture the Oracle HTTP Server will be running on a separate server. Therefore, there will be licensing costs associated with Oracle HTTP Server.
- There are licensing costs associated with Oracle WebLogic Server Enterprise Edition.
- Oracle WebLogic Server 12.1.2 installer requires that a certified Java Development Kit already exists on your system. For more information, see the appropriate certification document for 12c (12.1.2) on the Oracle Fusion Middleware Supported System Configurations page.

For more information, navigate to the respective product's page on Oracle Technology Network: http://www.oracle.com/technetwork/index.html.

You can download the suggested software from the following locations:

- Oracle Application Express
- Oracle REST Data Services
- Oracle WebLogic Server
- Oracle HTTP Server
- Java Development Kit (JDK)

Performing the Example Web Listener Deployment

In a production environment, the database and middle-tiers should be deployed on different servers. In this document Oracle WebLogic Server, Oracle HTTP Server, and Oracle REST Data Services are installed on the Mid Tier server. The Database Tier server includes Oracle Database and Oracle Application Express.

The task of setting up the example architecture discussed in this document involves the following steps:

1. Setting up the basic environment
2. Installing the required software -
   On the Mid Tier server:
   a) Installing and configuring Oracle WebLogic Server
   b) Installing and configuring Oracle HTTP Server
   c) Installing and Configuring Oracle REST Data Services
   d) Installing Oracle REST Data Services deployment in Oracle WebLogic Server

   On the Database Tier server:
   a) Installing and Configuring Oracle Application Express
3. Performing other Oracle Application Express tasks
4. Configuring Oracle HTTP Server as a front end to Oracle WebLogic Server
5. Testing the environment

Setting up the Basic Environment

In this section, you create Linux operating system users and directory structures on the Mid Tier server.

Note: You do not have to follow the same naming conventions for directories and users as discussed in this document.

On the Mid Tier server, perform the following steps as the root user:

1. Create a directory structure to store the installation and configuration files, called `/d1/home`
2. Create an operating system users under the oinstall (Oracle Inventory Group) and dba groups, called `webtier`. Create this user by executing the following command as root:

```
/usr/sbin/useradd -g oinstall -G dba webtier -d /d1/home/webtier
```

Performing a Single Deployment of the Web Tier

In this section, you learn how to integrate all of the components required for a single deployment of the web tier for Oracle Application Express. For information on how to extend this architecture for multiple deployments, see the *Performing Multiple Deployments of the web tier* section.

Installing and Configuring Oracle WebLogic Server

Oracle WebLogic Server is a scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. The WebLogic Server infrastructure supports the deployment of many types of distributed applications. Before performing the steps in this section, download Oracle WebLogic Server 12c binary installer to your Mid Tier server.

Installing Oracle WebLogic Server

Perform the following steps to install Oracle WebLogic Server 12c on the Mid Tier server.

1. First, install the required JDK by logging in as the root user.

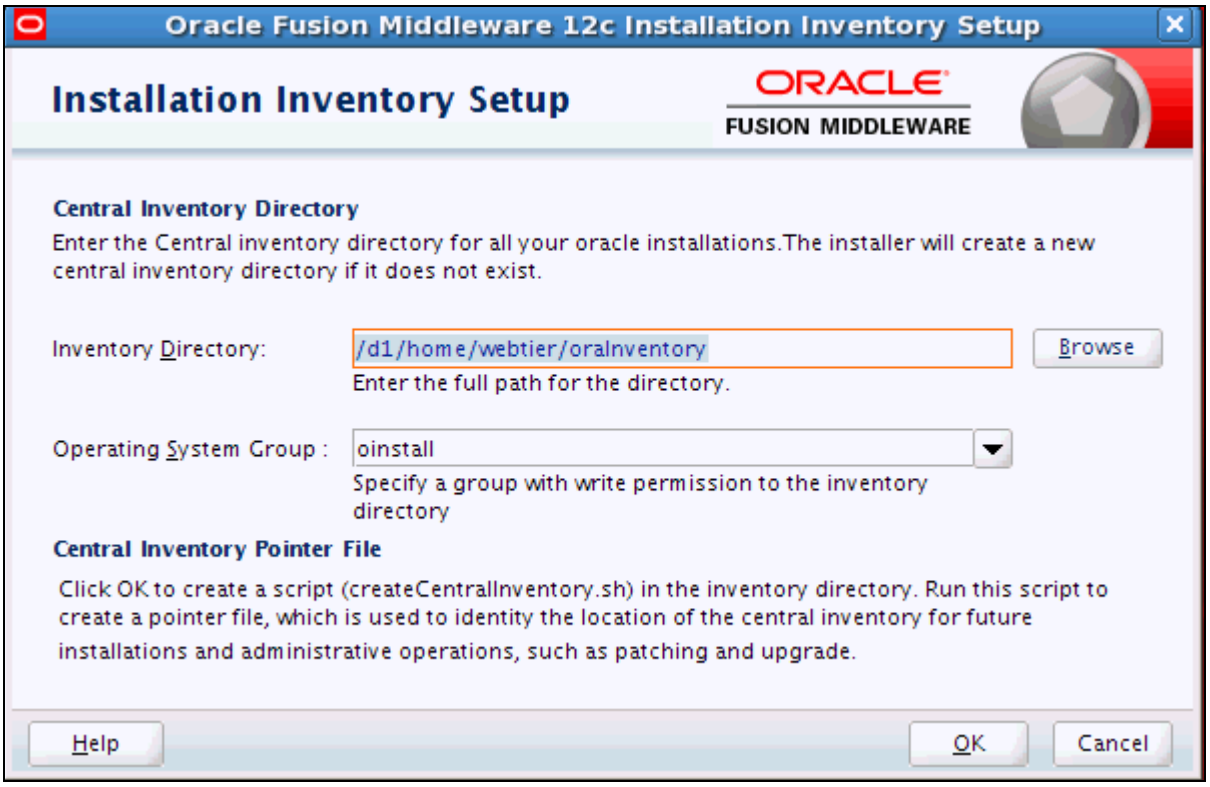
```
bash-3.2# rpm -ivh jdk-7u51-linux-i586.rpm
Preparing...                ########################################### [100%]
   1:jdk                    ########################################### [100%]
```

2. Open a Terminal window and log in as the operating system user. Navigate to the location where you downloaded the installation program. For example, in this document, login as `webtier` and navigate to the `/scratch` directory. Then, launch the installation program by invoking **`java -jar`** from the JDK directory on your system as shown in the example below:

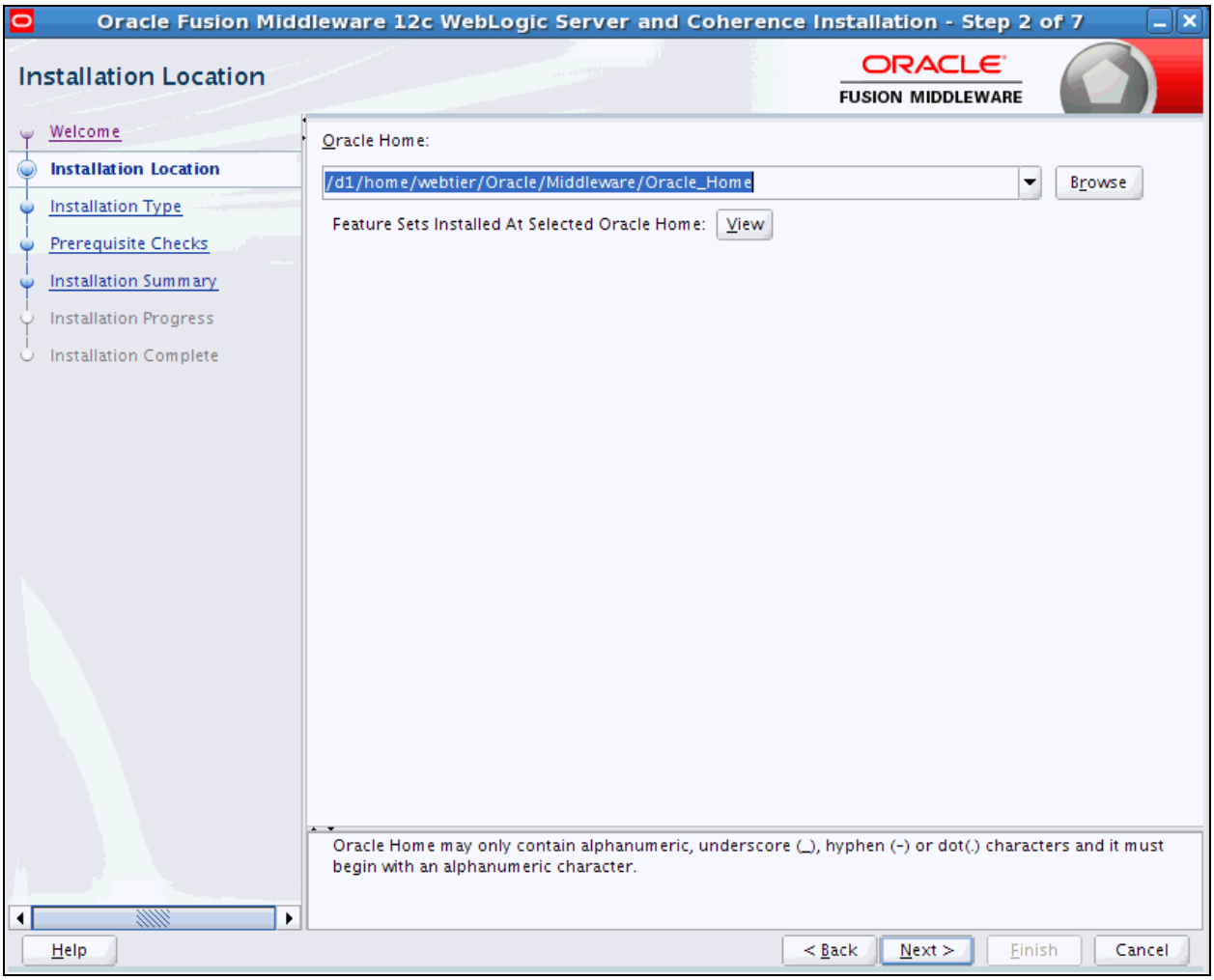
   ```
   /usr/java/jdk1.7.0_51/bin/java -jar wls_121200.jar
   ```

   The installation program appears and you are now ready to begin the installation.

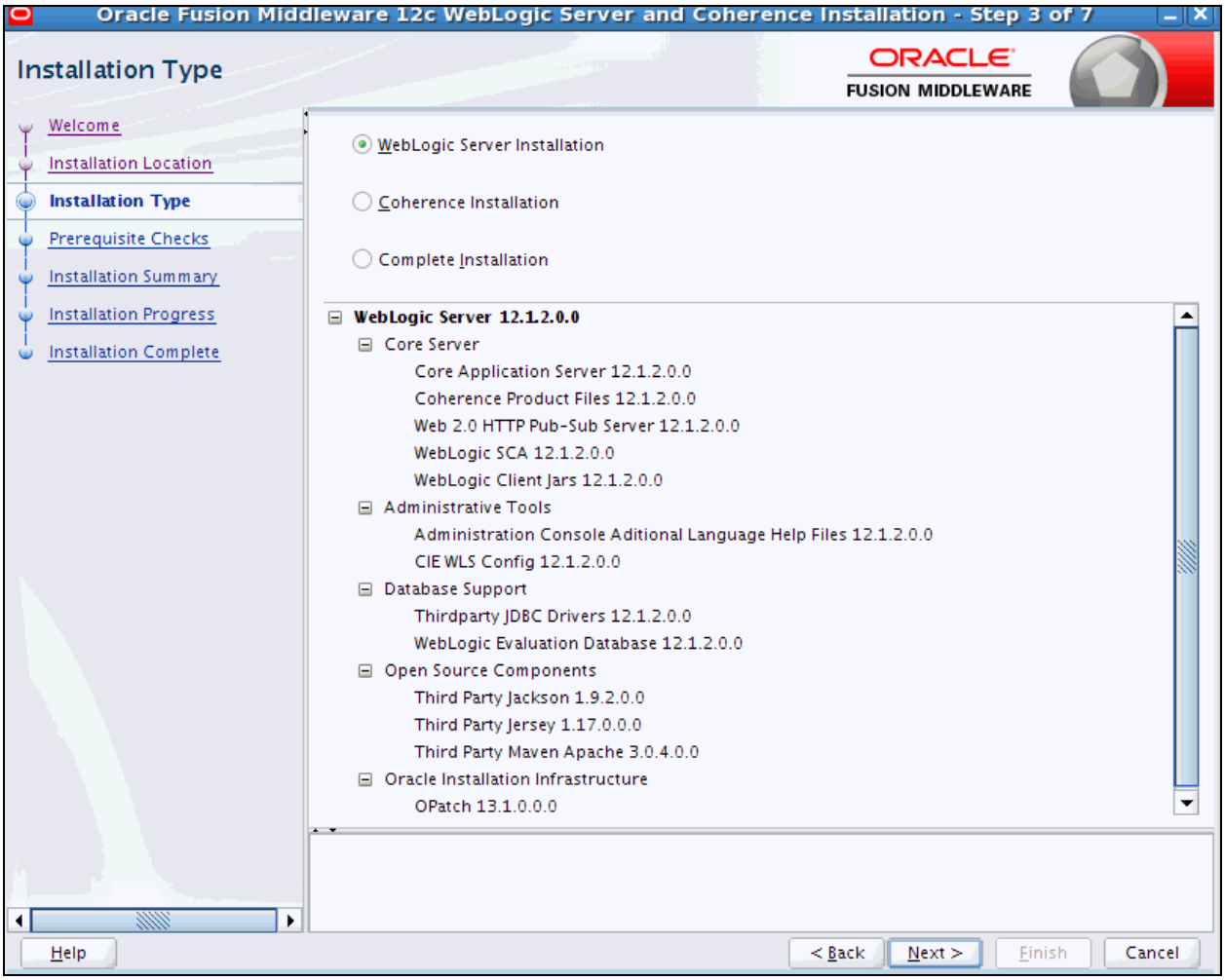
3. The installer will create a new central inventory directory if it does not already exist. Specify the directory where you want to create the central inventory. Click **OK**.



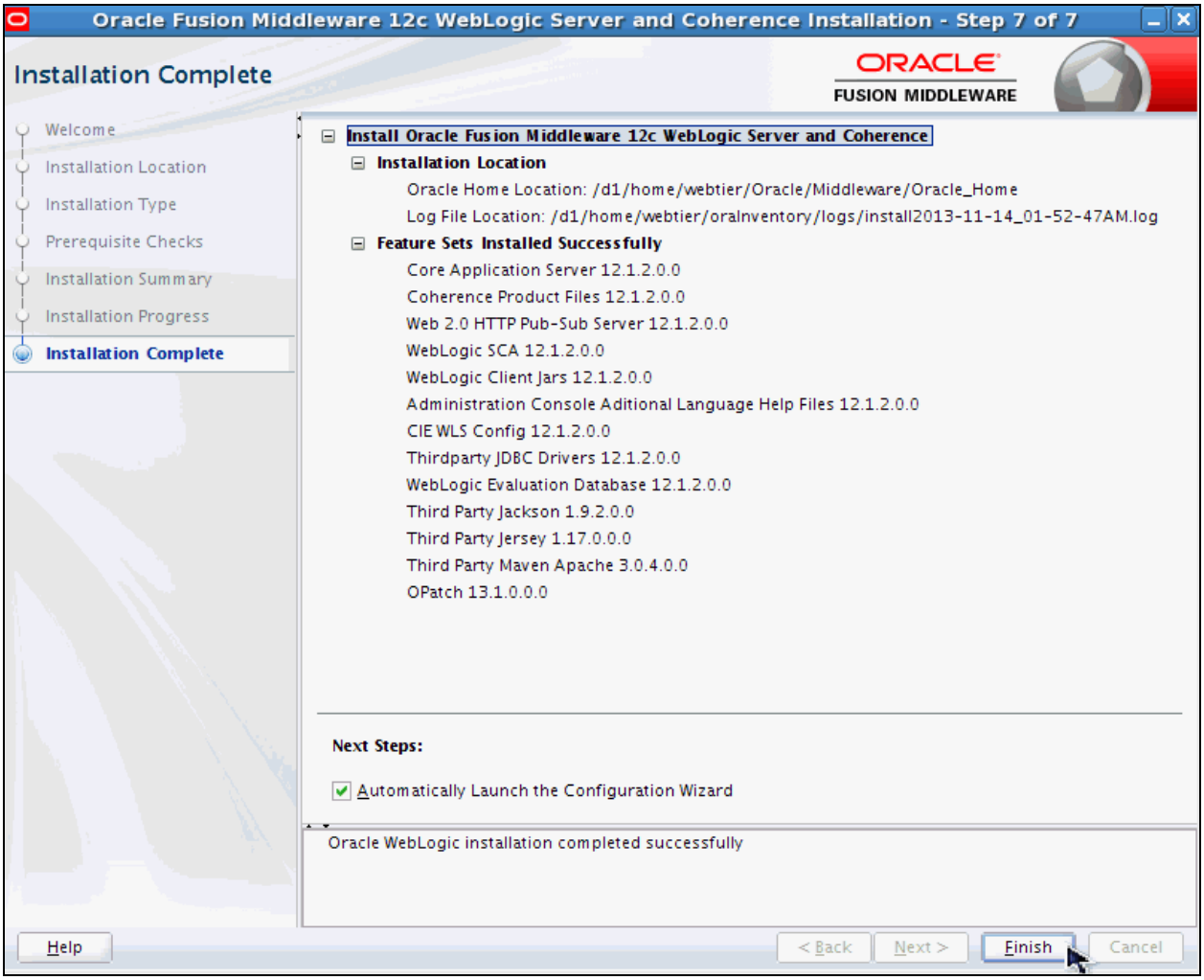
4. On the Welcome page, click **Next**. On the Installation Location page, enter the name of the Oracle Home directory. In this document, it is: `/d1/home/webtier/Oracle/Middleware/Oracle_Home`. Then click **Next**.



5. You need to create the standard installation topology for WebLogic Server and Coherence. Therefore, select **WebLogic Server Installation** for Installation Type and click **Next**.



6. The Prerequisite Checks screen verifies that your system meets the minimum requirements. Click **Next**.
7. On the Installation Summary page, verify the installation options you selected. If you want to save these options to a response file, click **Save Response File** and enter the location and name of the file. Click **Install**. Then, click **Next**.
8. On the Installation Complete page, make sure to select the **Automatically Launch the Configuration Wizard** radio button, and click **Finish**. The Configuration Wizard appears and you will use this wizard to create your WebLogic domain.

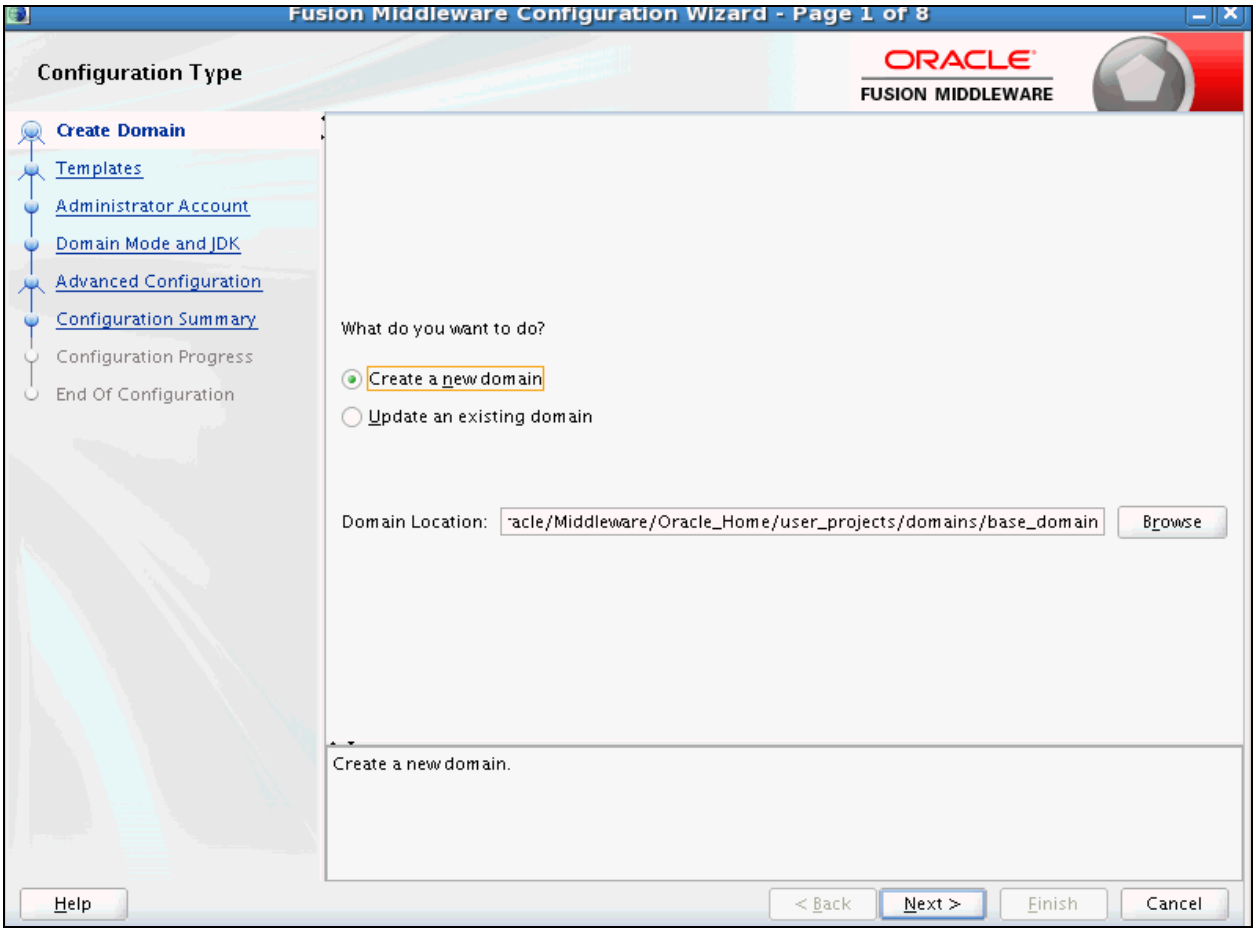


Creating a Domain

A domain is the basic administration unit for Oracle WebLogic Server. It consists of one or more Oracle WebLogic Server instances and logically related resources and services that are managed collectively. A domain must contain one administration server. It may contain any number of managed servers.

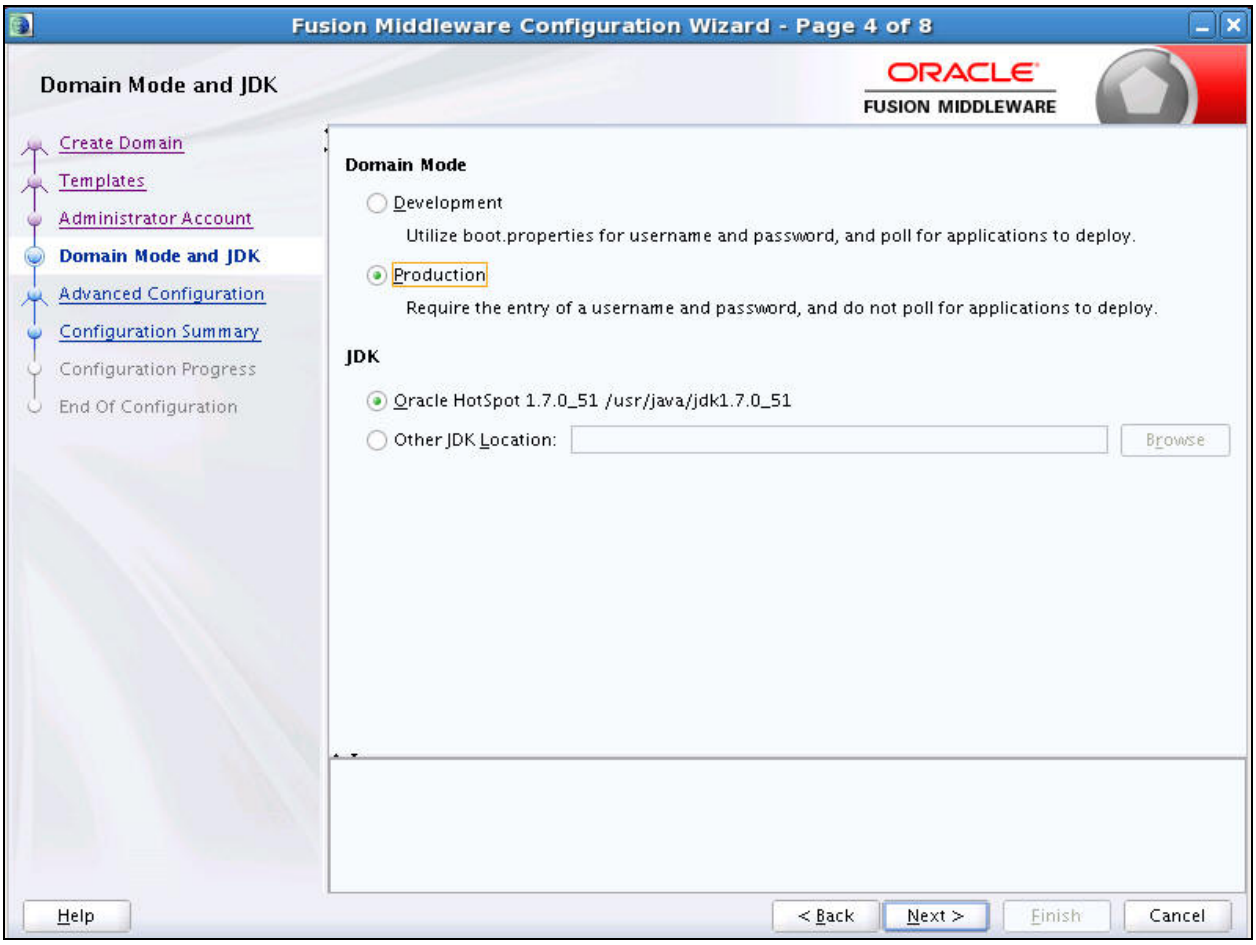
In our example, we need a managed server pointing to a specific Oracle REST Data Services deployment in the WebLogic server. Perform the following steps:

1. On the Configuration Type page, select **Create a new domain** radio button and click **Next**. In this document, the domain name is **`base_domain`**. You are free to choose a different name and location for the domain.

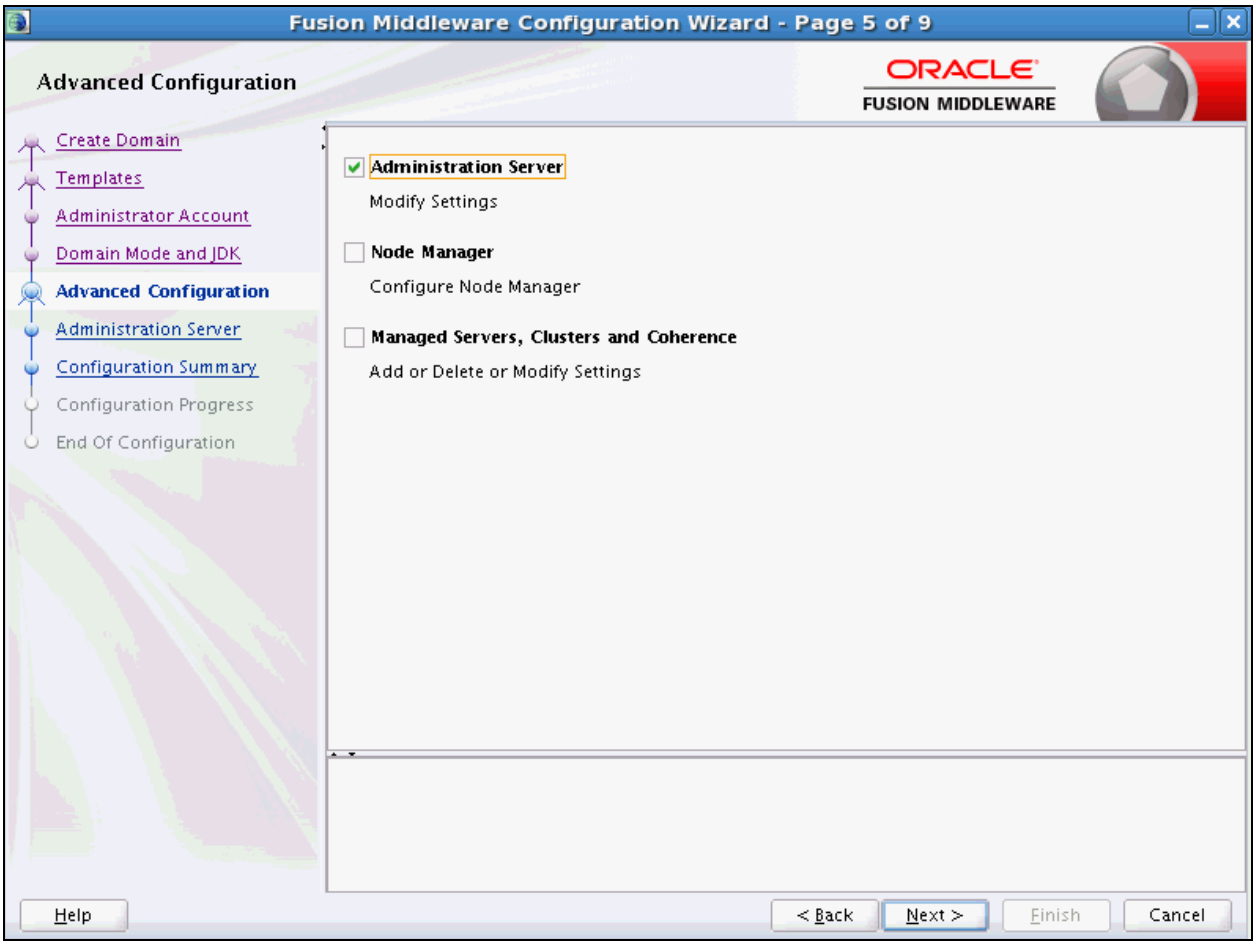


2. Make sure that the **Create Domain Using Product Templates** radio button is selected and click **Next**.

3. Specify the user name and password for the default WebLogic Administrator account for the domain. This account is used to boot and connect to the domain's Administration server. Click **Next**.
   Note: Do not forget this username and password. These are required to do anything with the domain, from starting servers to changing the domain configuration.
4. Select **Production** for Domain Mode and **Oracle HotSpot JDK** for JDK. Click **Next**.



5. Select **Administration Server** check box and click **Next**. In this document, you will configure the managed server and the machine using the browser-based WebLogic Server Administration Console.



6. Select the **Enable SSL** checkbox, leave the default values for Server Name, Listen address, and Listen Port. Click **Next**. Then Click **Create**.
   Note: In this document, the default value for Listen Port, 7001 is used. If any other program on your machine is already using 7001, you can always enter a different port number on this screen.
7. A WebLogic Server domain has now been created. Click **Next**.
8. Note the domain location and Admin Server URL. Click **Finish**.
9. Navigate to the newly created domain directory. For example, in this example, navigate to `/d1/home/webtier/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain`. Now, to start the WebLogic Server Administration Console, enter **`./startWebLogic.sh`**
10. To access the WebLogic Server administration console, open a web browser and enter the URL: http://<hostname>:7001/console. On the Welcome screen, log in using the Username and Password entered to start the administration server. The home page of the administration console is displayed.
    Note: If you used a different port number for the administration server, use that port instead.

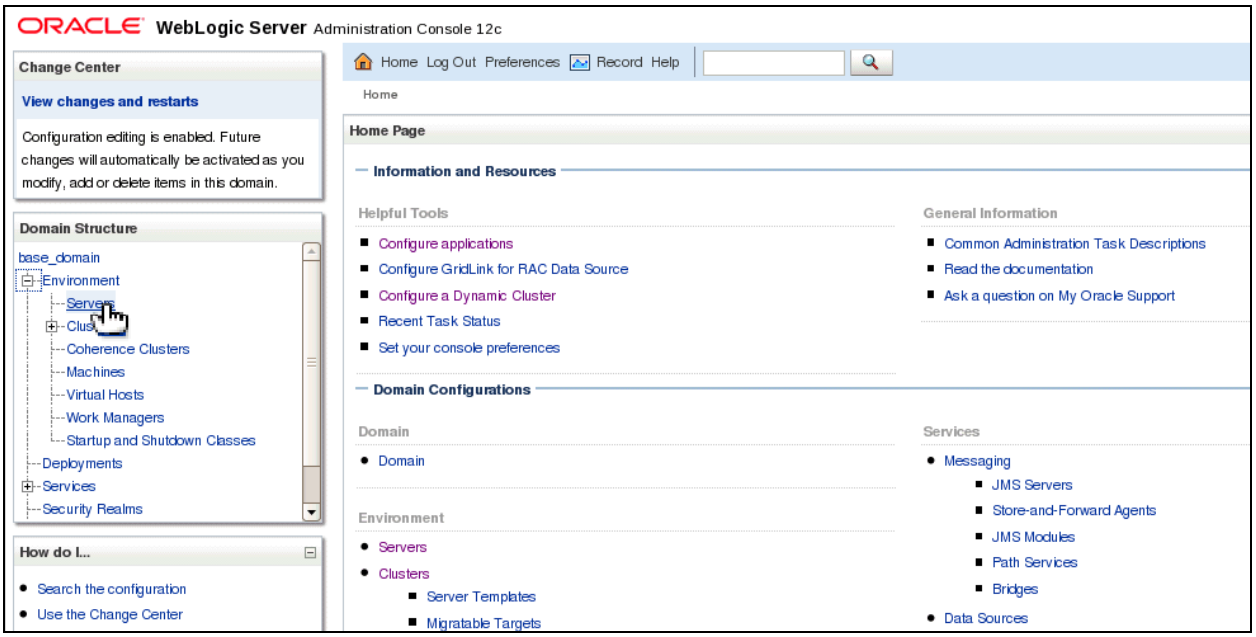
Creating and Configuring Managed Servers

You configure a domain to include additional WebLogic Server instances called Managed Servers. You deploy Web applications, EJBs, and other resources onto the Managed Servers and use the Administration Server for configuration and management purposes only. When a Managed Server starts up, it connects to the domain's Administration Server to obtain configuration and deployment settings.

In this section, you create a managed server under the domain that you created in the previous section. In our example, you will create the **http8080** managed server under **base_domain**.

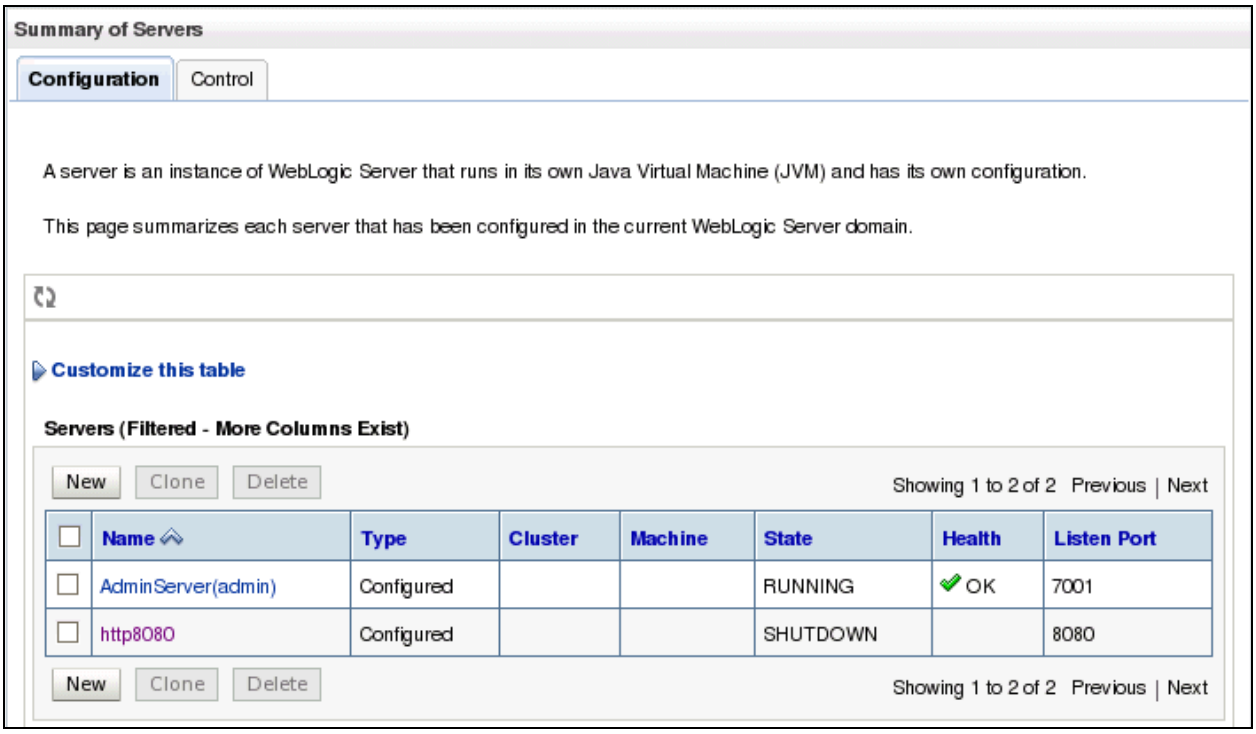
Perform the following steps:

1. Log in to the WebLogic Server Administration Console. Under Domain Structure > <your_domain_name> >Environment, click **Servers**.



2. You see that the AdminServer (admin) is listed in the table. Click **New**.
3. Under Server Properties, enter a name for Server Name, **http8080**, and port number for Server Listen Port, **8080**. This is a standalone server. Accept the defaults and click **Next**.
4. Click **Finish**.

The Summary of Servers page should look like:

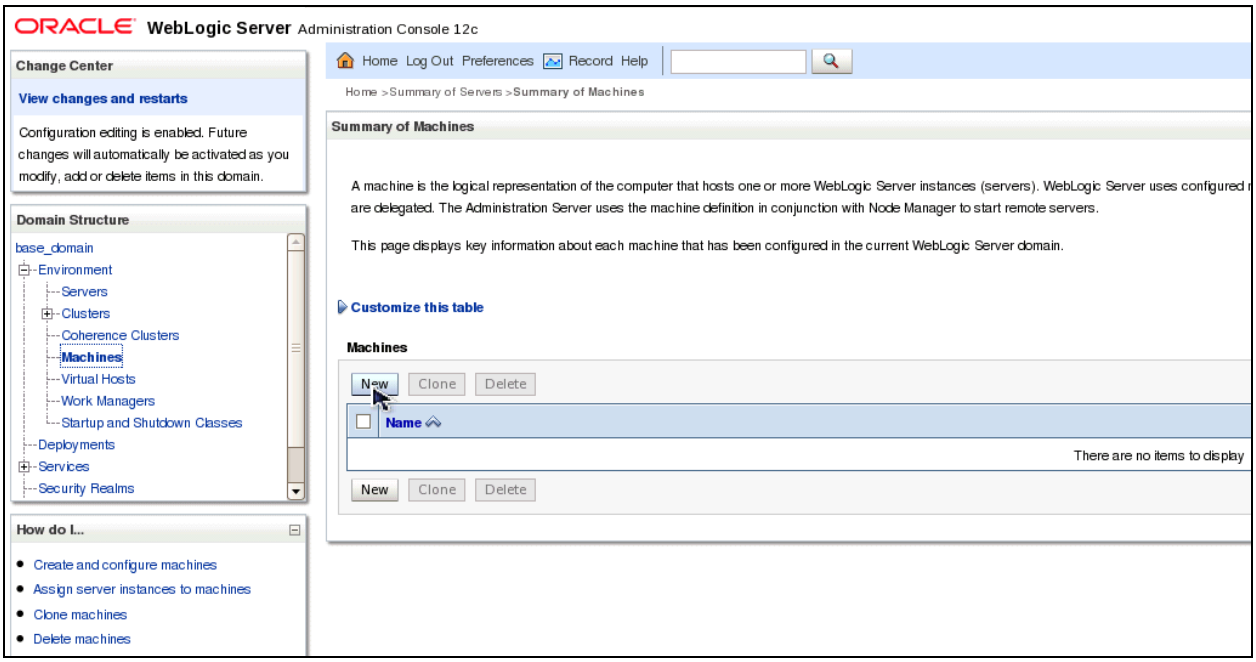


Creating and Configuring a Machine

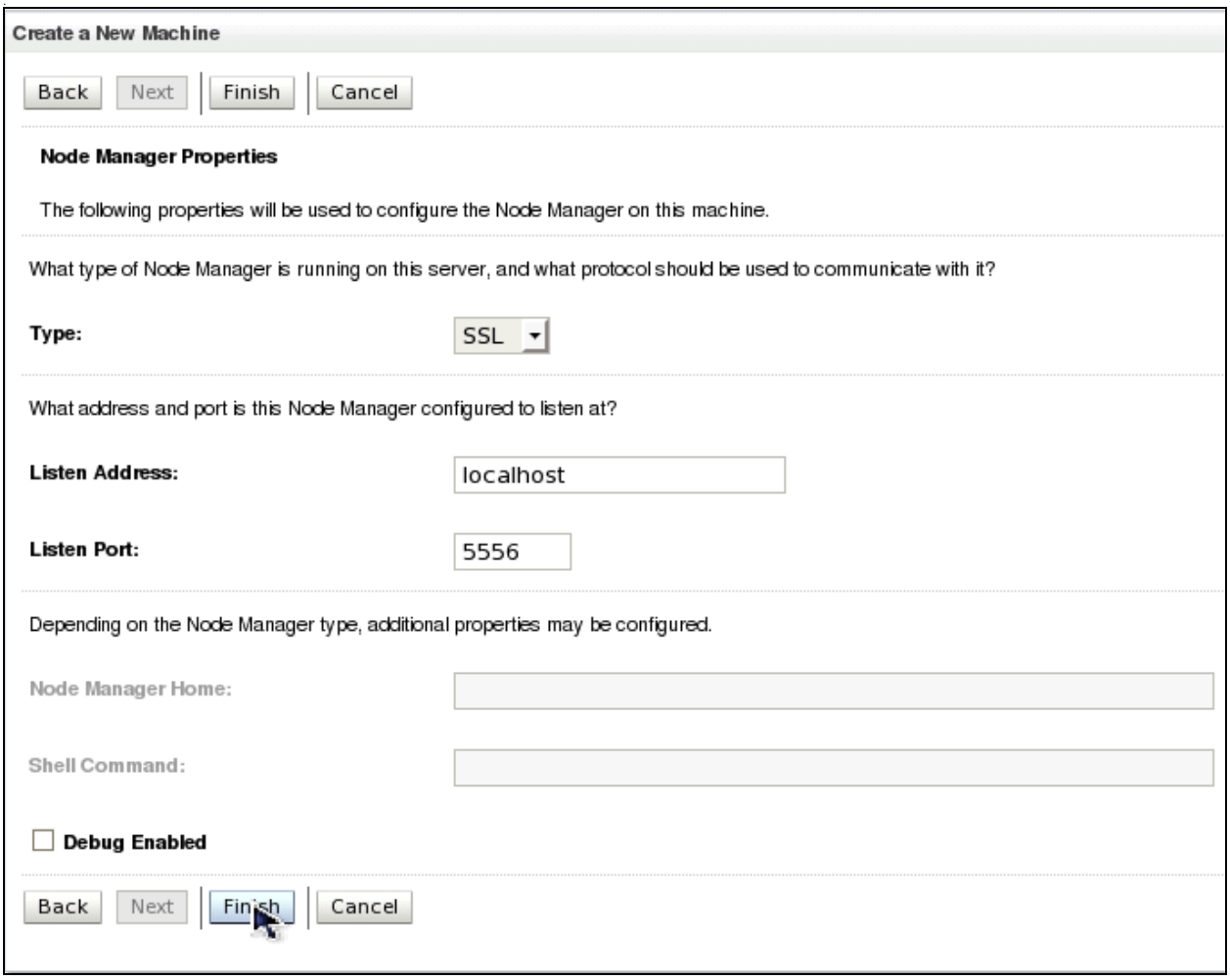
In a WebLogic Server domain, machines can be defined to represent physical computers that host one or more Oracle WebLogic Server instances. Machine definitions help WebLogic Server choose backup servers for session replication.

In this document, we use **localhost** for the machine and then associate the managed server with this machine. Perform the following steps:

1. Under Domain Structure > <your_domain_name> >Environment, click **Machines**. Then, click **New**.



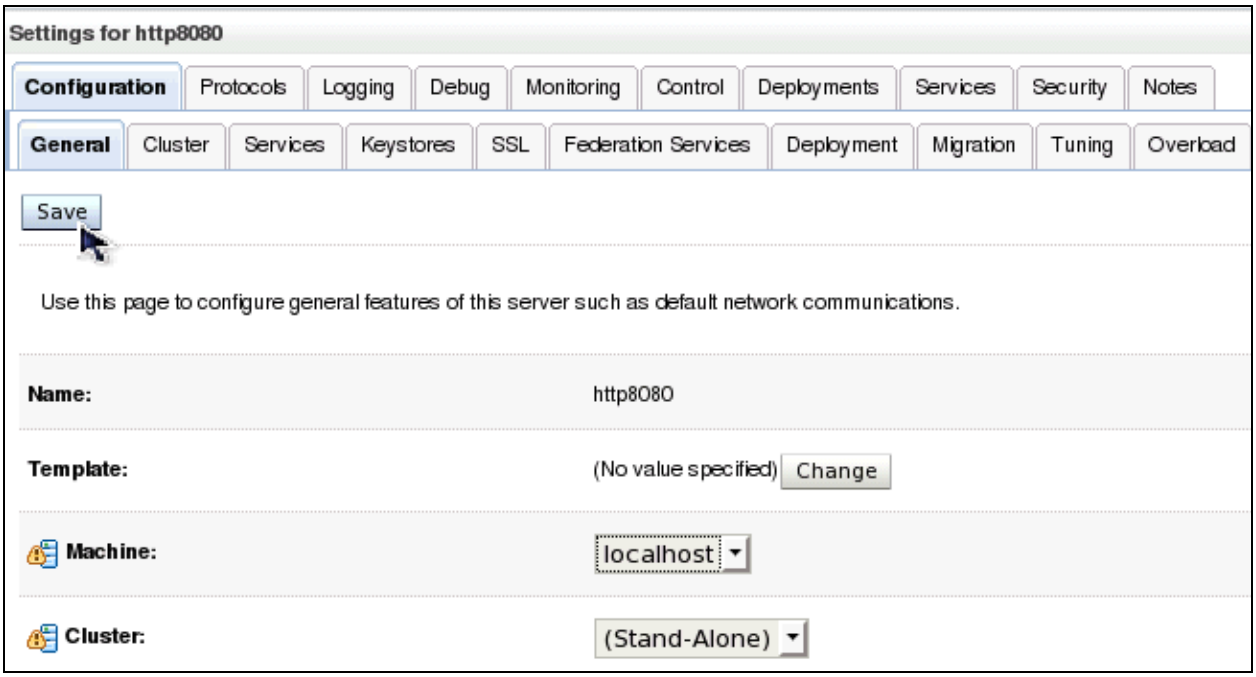
2. Under Machine Identity, enter a machine name for Name, **localhost**.
3. Select **Unix** from the Machine OS select list and click **Next**.
4. Review the following details and click **Finish**.
    - Type: **SSL**
    - Listen Address: **localhost**
    - Listen Port: **5556**



Now, you need to configure the machine that you just created. To add the managed server to the newly created machine, perform the following steps:

1. Under Domain Structure > <your_domain_name> >Environment, click **Servers**.
2. Click the managed server **http8080** in the table.

3. Under Machine, select the machine name **localhost**. Click **Save**.



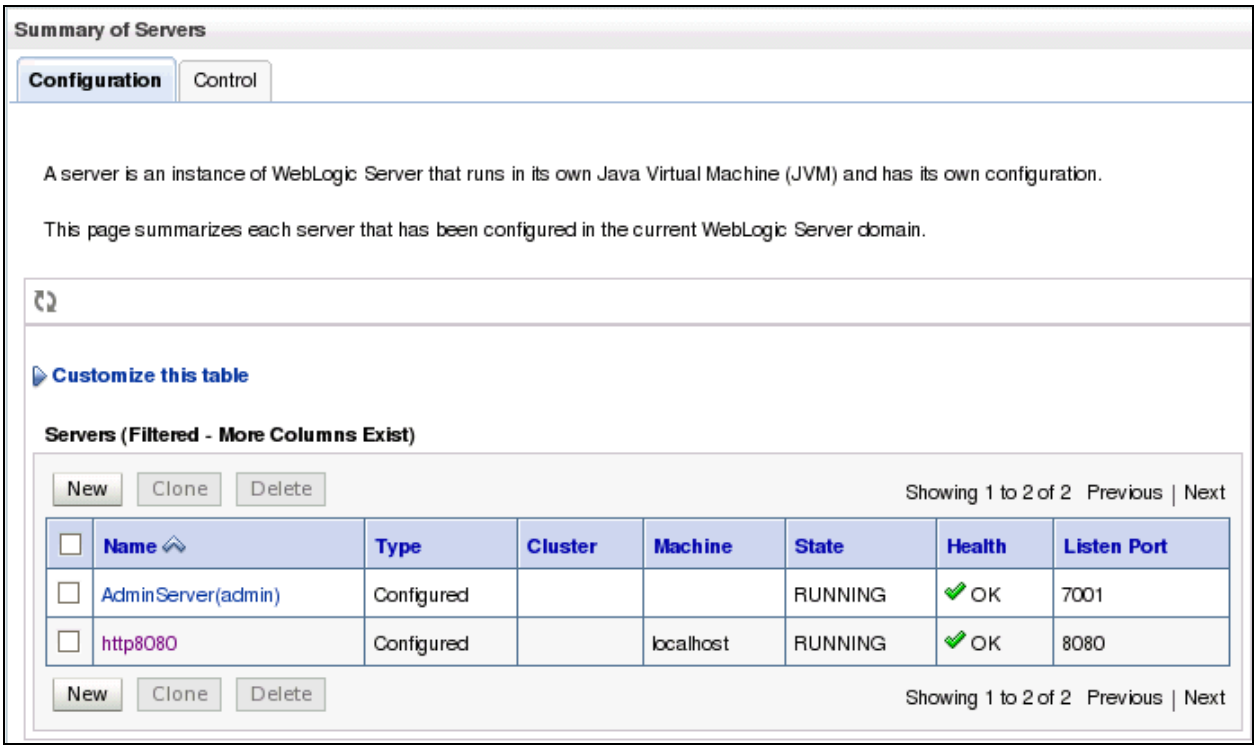
Starting Node Manager and the Servers

Node Manager is a WebLogic Server utility that enables you to start, shut down, and restart the administration server and managed servers from a remote location. Although Node Manager is not required, it is recommended if your WebLogic Server environment hosts applications with high availability requirements.

A Node Manager process is not associated with a specific WebLogic domain, but with a particular machine. You can use the same Node Manager process to control server instances from any WebLogic Server domain, as long as those server instances reside on the same machine as the Node Manager process.

In this section, you start the Node Manager using the command-line and then start the servers by using the WebLogic Server administration console. Perform the following steps:

1. On machine A, navigate to **<Domain_Home>/bin** and run the following command: **`./startNodeManager.sh`**
2. Access the WebLogic Server administration console again (http://<hostname>:7001/console). In the Domain Structure section, expand **Environment** and click **Servers**. Then click the **Control** tab.
3. Select the check boxes for all of the managed servers and then click the **Start** button.
4. Click the refresh icon to periodically refresh the Servers table. After a few minutes, verify that each server's State is "Running."



Installing and Configuring Oracle HTTP Server

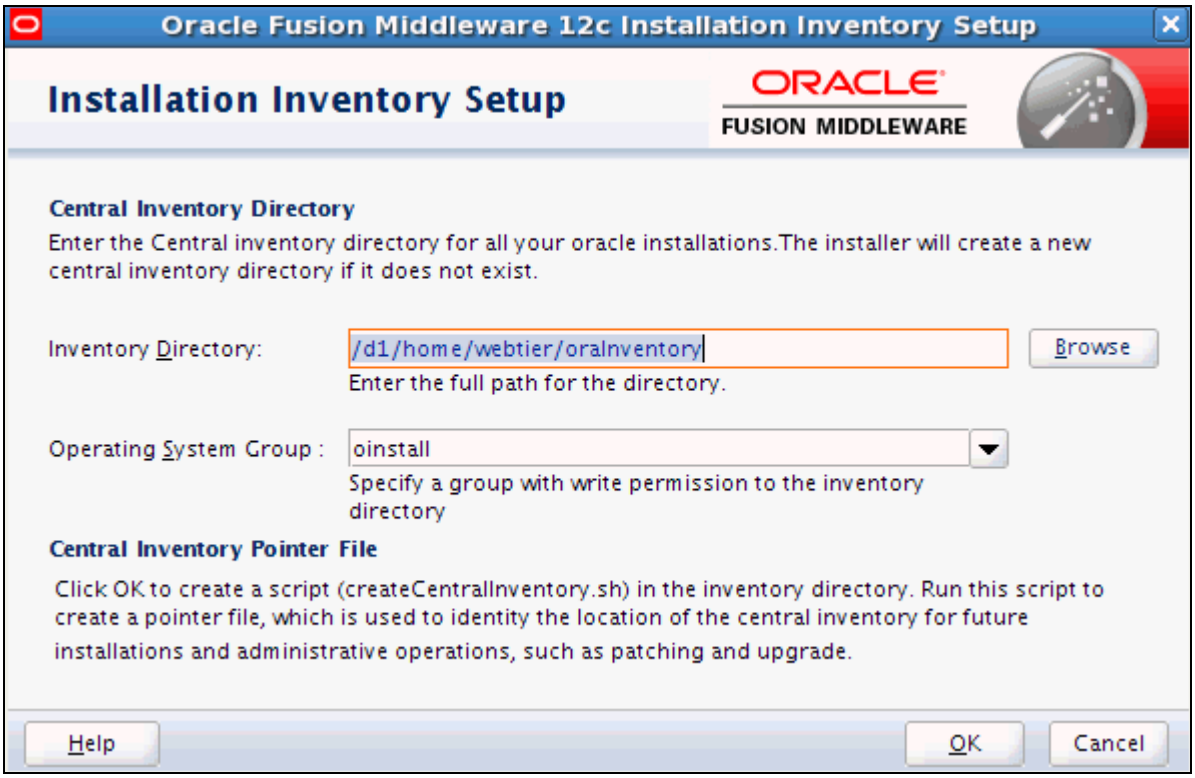
Oracle HTTP Server is the Web server component for Oracle Fusion Middleware. It provides a HTTP listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the web.

Before performing the steps in this section, you should have downloaded Oracle HTTP Server 12c to your Linux machine A and unzipped the file.

Installing Oracle HTTP Server

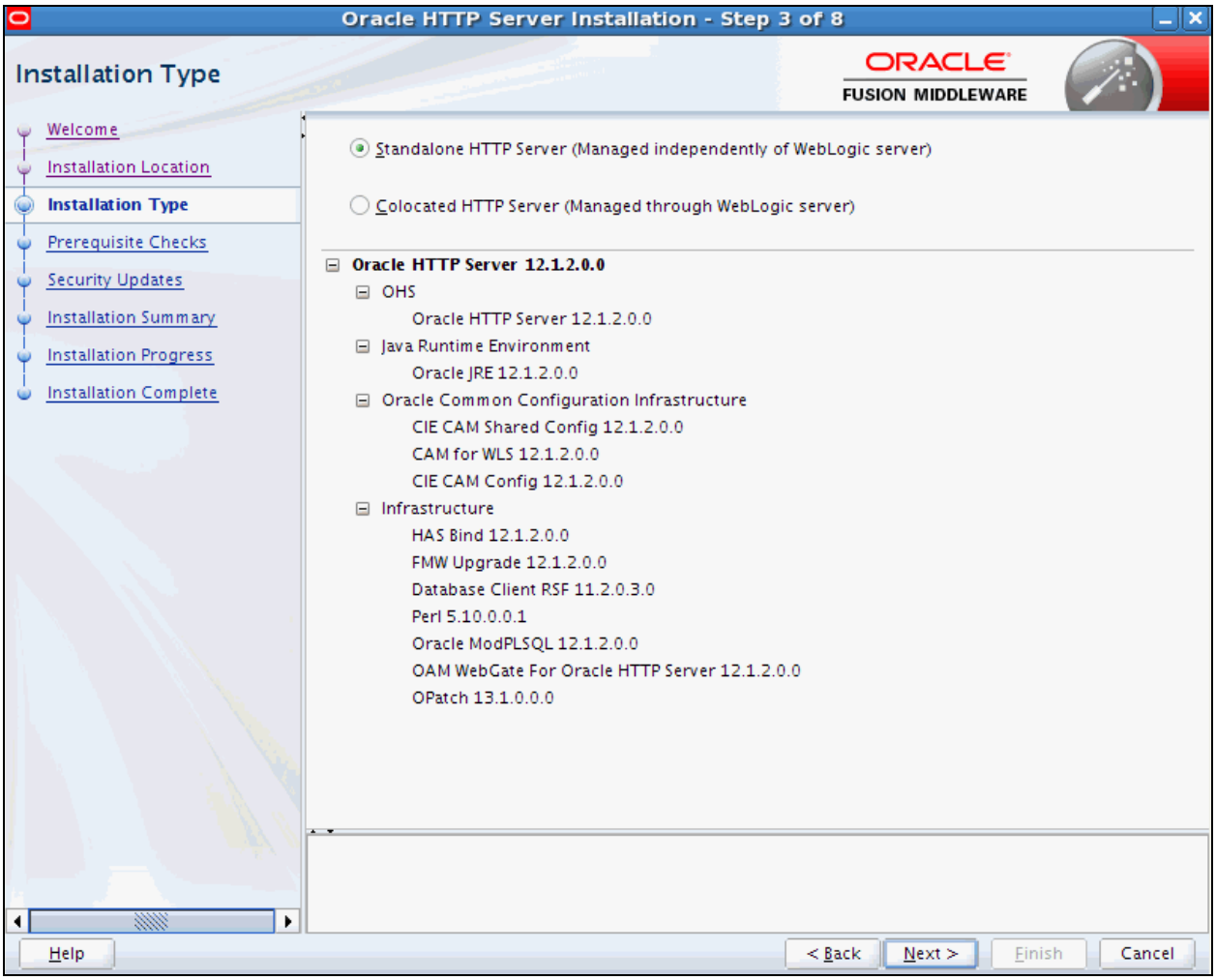
On machine A, perform the following steps:

1. In a Terminal window, navigate to the directory where you have unzipped the installation program zip file. Then, launch the installation program by executing the following command:
   `./ohs_121200_linux64.bin`
2. You already have the central inventory directory created. Click **OK**.

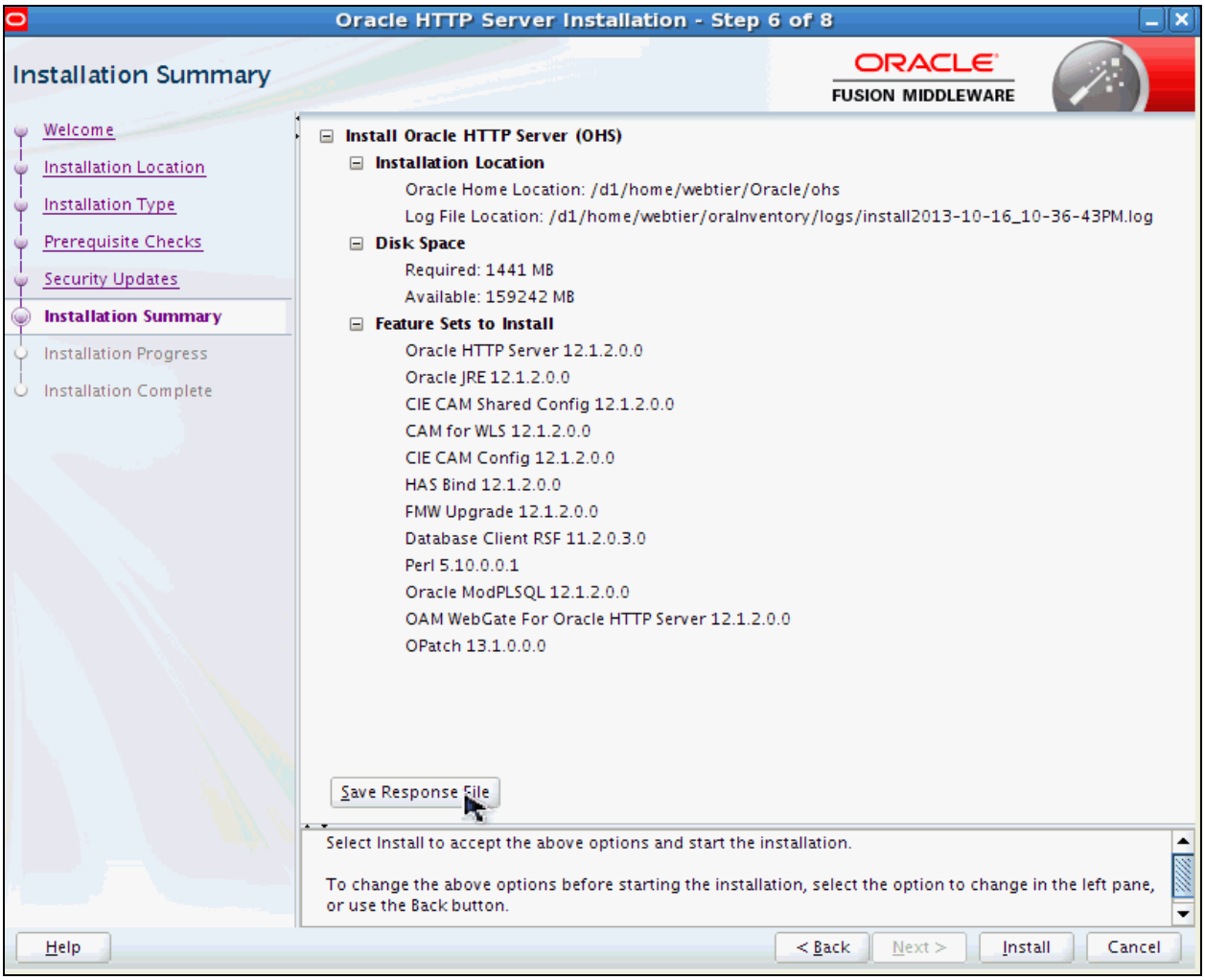


3. On the Welcome page, click **Next**.
4. In this document, you are using the standard installation topology for Oracle HTTP Server in a standalone domain. Therefore, you can specify an Oracle home directory of your choice. Specify a new Oracle home as **`/d1/home/webtier/Oracle/ohs`**.

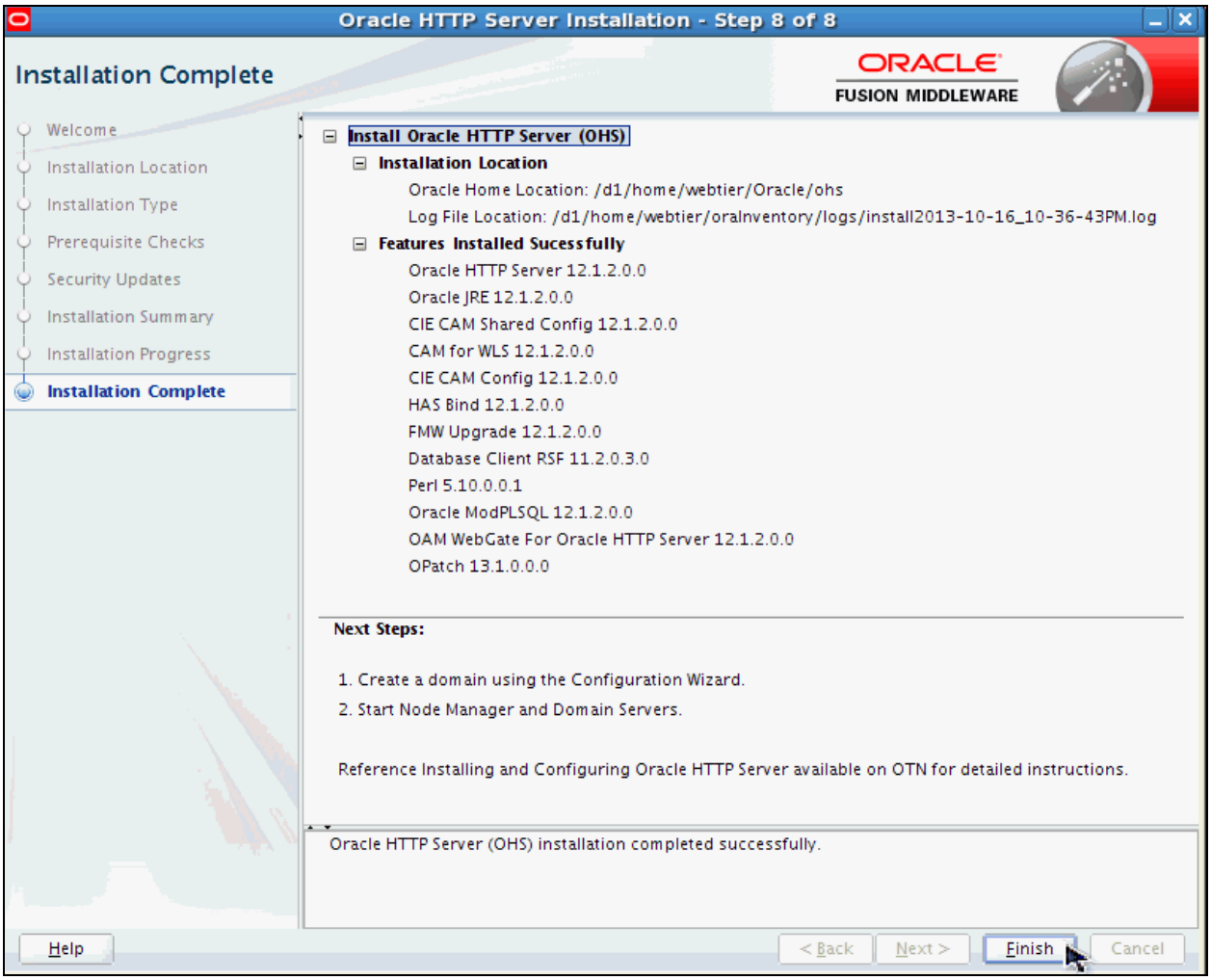
5. Select Standalone **HTTP Server (Managed independently of WebLogic server)** for Installation Type, and click **Next**.



6. Once the prerequisite checks are done, click **Next**.
7. On the Specify Security Updates page, you can choose to receive updates and enter an email address. Alternatively, you can just deselect the option and click **Next**.
8. Review the Installation Summary, save the response file and click **Install**.



9. Once you see the Installation Progress is 100%, click **Next**.
10. Review the Installation Complete page and click **Finish**.



Creating a Domain

In this step, you configure Oracle HTTP Server in a standalone domain. To start the Oracle Fusion Middleware configuration wizard, perform the following steps:

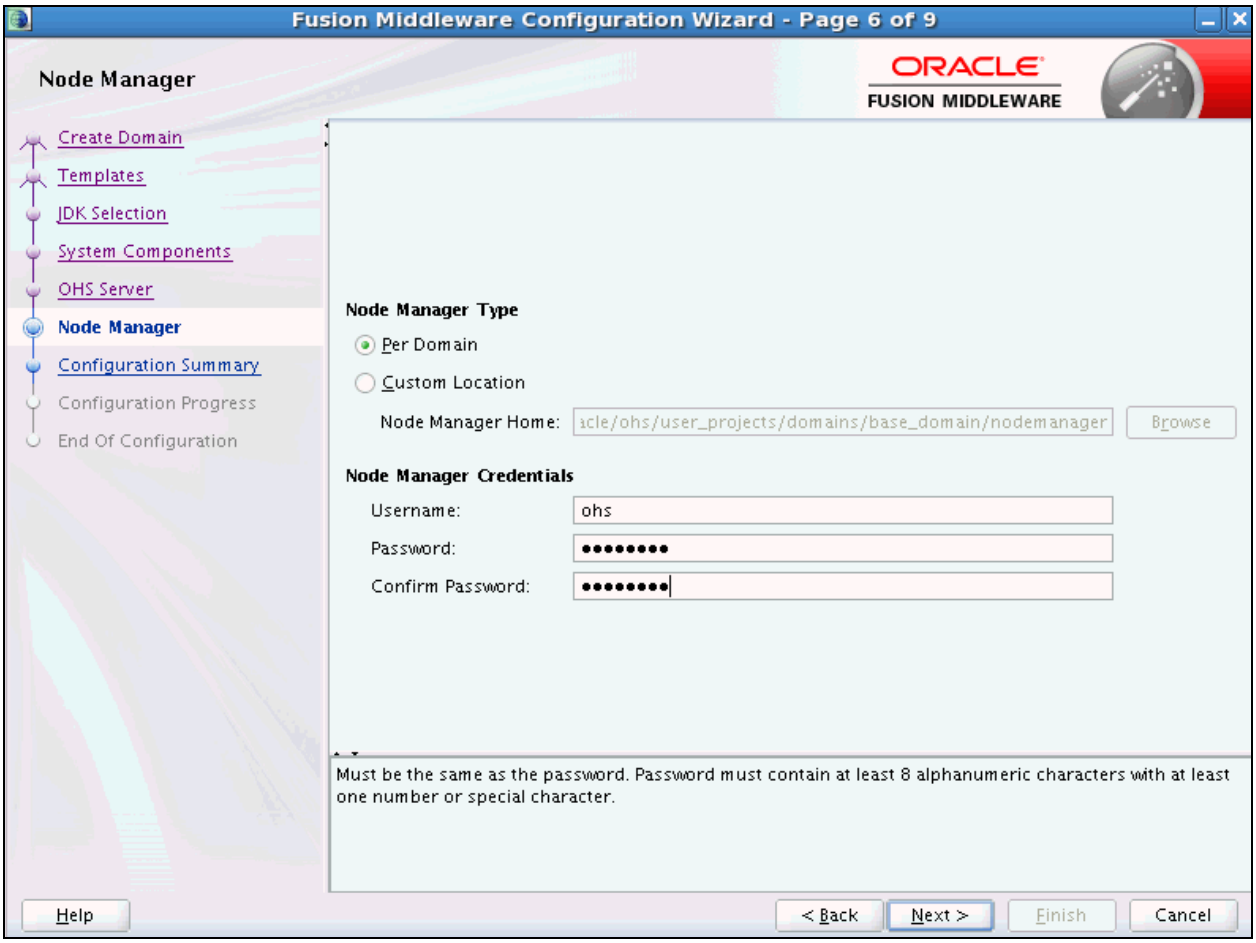
1. Navigate to the **ORACLE_HOME/oracle_common/common/bin** directory. In this example, navigate to `/d1/home/webtier/Oracle/ohs/oracle_common/common/bin`.
2. Then, run **`./config.sh`**

Note: The configuration wizard does not perform automatic port allocation. Therefore, make sure to verify that the ports you are assigning to your components are available.

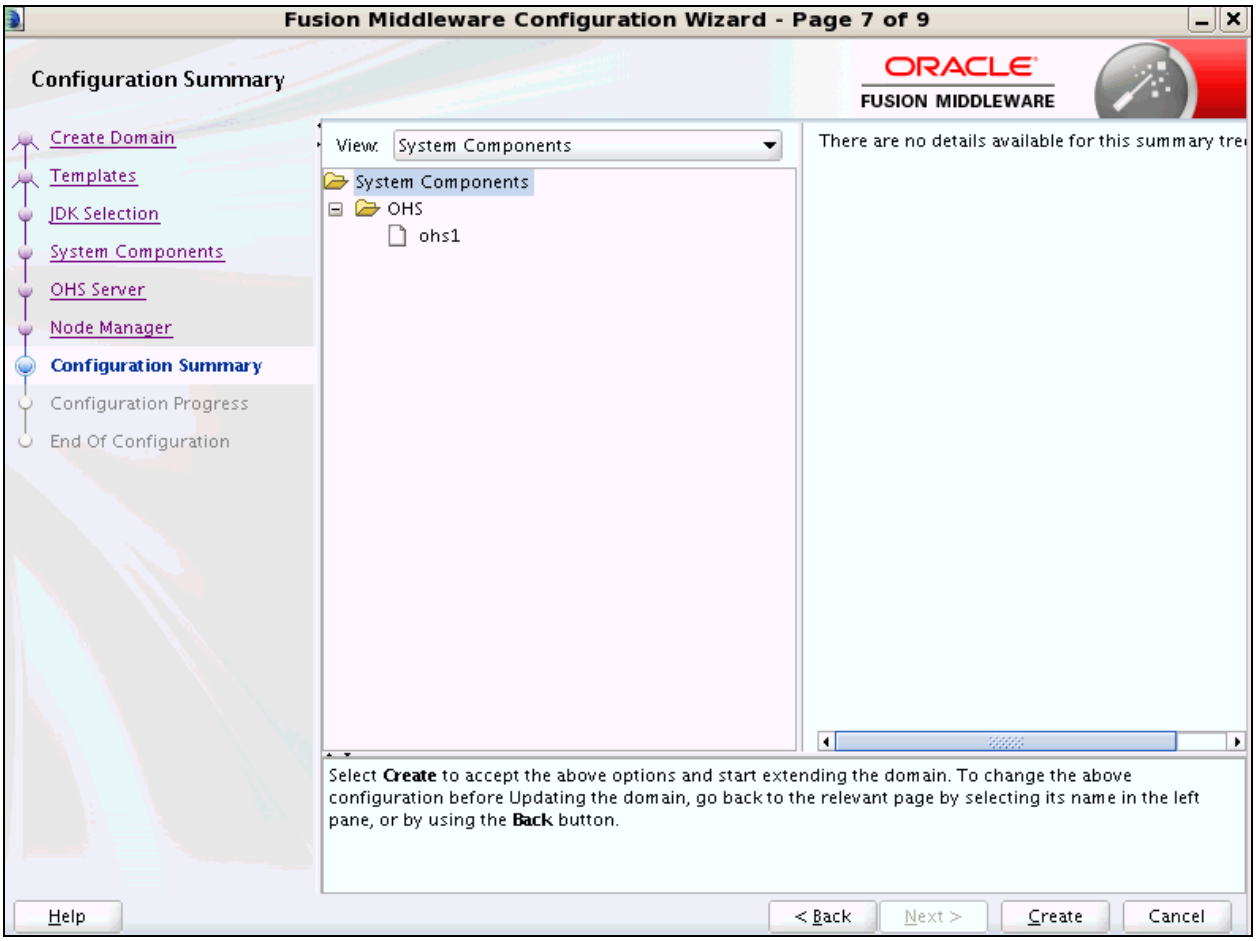
Now perform the following steps:

1. Select **Create a new domain**. Your domain home can reside anywhere on your system. Click **Next**.
2. On the Templates page, select **Oracle HTTP Server (Standalone) – 12.1.2.0 [ohs]**. Click **Next**.
3. Select **Oracle HotSpot JDK** for JDK and click **Next**.
4. On the System Components page, accept the defaults and click **Next**.

5. Note that the ohs1 system component is populated on the OHS Server page. Make sure that the default ports are not used by other services. Accept the defaults and click **Next**.
6. Select Per Domain as the Node Manager type and specify the Node Manager credentials. Click **Next**.



7. On the Configuration Summary page, click **Create**.



8. Once you see the Configuration Progress is 100%, click **Next**.
9. Note the domain location and click Finish.

Starting the Instances

In this step, you will start the Oracle HTTP Server. First, you may need to set the required environment variables. Perform the following steps:

1. Navigate to **DOMAIN_HOME/nodemanager** and then update the nodemanager.properties file. You need to set the ListenPort value as **5557**.

```
DomainsFile=/d1/home/webtier/Oracle/ohs/user_projects/domains/base_domain/nodema
nager/nodemanager.domains
LogLimit=0
PropertiesVersion=12.1.2
AuthenticationEnabled=true
NodeManagerHome=/d1/home/webtier/Oracle/ohs/user_projects/domains/base_domain/no
demanager
JavaHome=/d1/home/webtier/Oracle/ohs/oracle_common/jdk
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
ListenAddress=localhost
NativeVersionEnabled=true
ListenPort=5557
LogToStderr=true
SecureListener=true
```

2. To start your per-domain Node Manager, go to the **DOMAIN_HOME/bin** directory. DOMAIN_HOME is the complete path to the location in which you have configured Oracle HTTP Server. Start the Node Manager, by entering the following command:

   `./startNodeManager.sh`

3. Now, navigate to the **DOMAIN_HOME/config** directory and then update the config.xml to include the listen port tag and value as:

   `<listen-port>5557</listen-port>`

```
  <name>base_domain</name>
  <domain-version>12.1.2.0</domain-version>
  <security-configuration>
    <name>base_domain</name>
    <node-manager-username>ohs</node-manager-username>
    <node-manager-password-encrypted>{AES}GjcRrxlLGa72lZ7GwBLACwUBYBH1uyRj5NrB7G
ieZ2A=</node-manager-password-encrypted>
  </security-configuration>
  <server>
    <name>AdminServer</name>
    <listen-address/>
  </server>
  <configuration-version>12.1.2.0</configuration-version>
  <machine>
    <name>localmachine</name>
    <node-manager>
      <name>localmachine</name>
      <listen-address>localhost</listen-address>

<listen-port>5557</listen-port>
      <password-encrypted>{AES}tBayDJb3X8Eu0OQ3kd6SD+CAfZ8j/yMaFnG6NU1zzd8=</pas
sword-encrypted>
    </node-manager>
```

4. To start the Oracle HTTP Server instance, go to **DOMAIN_HOME/bin** directory. Then, run the following command:

   ```
   ./startComponent.sh ohs1
   ```

Installing and configuring Oracle Application Express

Now, on the Database Tier server, you install and configure Oracle Application Express. You should have already installed an Oracle Database with the SID **orcl1**. For more information, see the *Prerequisites* section. The Oracle Application Express instance in the example is **apexdev**.

First, unzip the downloaded Application Express installation zip file. Perform the following steps:

1. Log in as the operating system user and change your working directory to apex.
   ```
   cd apex
   ```

   Note: You may need to set environment variables to run SQL*Plus.

2. Start SQL*Plus and connect to the Oracle Database where Oracle Application Express is installed as SYS specifying the SYSDBA role. For example:
   ```
   $ sqlplus /nolog
   SQL> CONNECT SYS as SYSDBA
   ```
   Enter password: SYS_password

3. Run the following command to install Application Express full development environment:
   ```
   @apexins.sql  SYSAUX  SYSAUX  TEMP /i/
   ```

4. Start SQL*Plus and connect to the Database as SYS specifying the SYSDBA role.
   ```
   $ sqlplus /nolog
   SQL> CONNECT SYS as SYSDBA
   ```
   Enter password: SYS_password

5. Run apxchpwd.sql to change the password of the ADMIN account. For example:
   ```
   @apxchpwd.sql
   ```
   When prompted, enter a password for the ADMIN account.

6. Now unlock the APEX_PUBLIC_USER account and change the password:
   ```
   ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
   ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password;
   ```

For more information on the installation, see Oracle Application Express Installation Guide, Downloading from OTN and Configuring Oracle REST Data Services section.

Installing and Configuring Oracle REST Data Services

On the Mid Tier server, you need to install and configure Oracle REST Data Services. Once the Oracle REST Data Services is configured, you will then deploy it to the WebLogic Server.

Perform the following steps:

1. Configure Oracle REST Data Services
2. Create a directory for images and copy the images from /apex directory

Configuring Oracle REST Data Services

Starting with Oracle REST Data Services Release 2.0.6, the name of the product WAR file has changed from **`apex.war`** to **`ords.war`**. Therefore, the context path that WebLogic and Tomcat will assign has changed from `/apex` to `/ords`. If you want to maintain the existing context root of /apex, then you should rename the ords.war file to apex.war before the deployment. The context root is part of the URL of the Oracle Application Express instance (http://host:port/apex/).

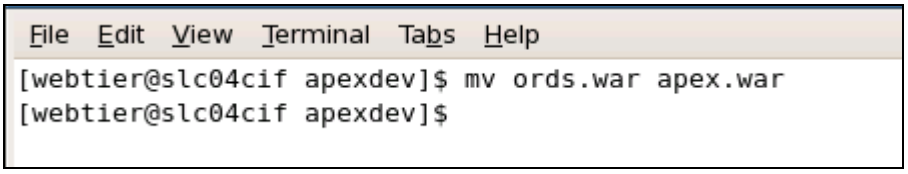
In this section, you use the command-line interface to configure Oracle REST Data Services. The example in this document uses the `/apex` context path. Therefore, you need to rename the `ords.war` file to `apex.war`.
First, create a directory structure on the Mid Tier server for the `apex.war` file and the images folder:

- First, create two directories as: wkdir and deploy
- Then, create two directories as: wkdir/apexdev and deploy/apexdev

Now, perform the following steps:

1. Unzip the downloaded **ords.version.number.zip** file under /deploy/apexdev. You can now see an ords.war file under /deploy/apexdev.

2. Change your active directory to the directory where you unzipped Oracle REST Data Services, namely to /deploy/apexdev.
3. Now, rename the **ords.war** file to **apex.war**.



4. Next, you configure the database connection details appropriate to where Application Express was installed. Execute the following command:
   `java -jar apex.war`

```
Enter the location to store configuration data:/d1/home/webtier/wkdir/apexdev
Apr 3, 2014 5:19:52 AM oracle.dbtools.common.config.cmds.ConfigDir execute
INFO: Set config.dir to /d1/home/webtier/wkdir/apexdev in: /d1/home/webtier/depl
oy/apexdev/apex.war
Apr 3, 2014 5:19:54 AM oracle.dbtools.common.config.file.ConfigurationFolder log
ConfigFolder
INFO: Using configuration folder: /d1/home/webtier/wkdir/apexdev/apex
Enter the name of the database server [localhost]: <hostname>
Enter the database listen port [1521]:
Enter 1 to specify the database service name, or 2 to specify the database SID [
1]:2
Enter the database SID [xe]:orcl1
Enter the database user name [APEX_PUBLIC_USER]:
Enter the database password for APEX_PUBLIC_USER:
Confirm password:
Enter 1 to enter passwords for the RESTful Services database users (APEX_LISTENE
R,APEX_REST_PUBLIC_USER), 2 to use the same password as used for APEX_PUBLIC_USE
R or, 3 to skip this step [1]:2
Apr 3, 2014 5:22:47 AM oracle.dbtools.common.config.file.ConfigurationFiles upda
te
INFO: Updated configurations: defaults, apex, apex_al, apex_rt
Enter 1 if you wish to start in standalone mode or 2 to exit [1]:2
```

You are prompted to enter the following data:

a. The location to store configuration data. There is no default value. You should enter the directory you created earlier, namely **`/d1/home/webtier/wkdir/apexdev`**.
b. The name of the database server. The default value is: localhost. In this document, the database is hosted on the Database Tier server. Enter the <hostname> of the Database Tier server.
c. The database listener port. The default value is **1521**. If the port number of the Oracle Database that you have installed is different, then make sure to enter this value instead of 1521.
d. The database service name or the database SID. In this step, enter **2** to specify the database SID. Enter the appropriate value you specified when installing the Oracle Database, which should be **orcl1** if you followed our earlier recommendations.
e. The database user name. Enter the default value, **APEX_PUBLIC_USER**.
f. The database user password of the user APEX_PUBLIC_USER.
   Note: Make sure the APEX_PUBLIC_USER account or whichever user account you specified in step e is unlocked and matches the password supplied in this step. If the password doesn't match then you see an error (503-Service Unavailable) when trying to access Application Express.
g. RESTful Services database users' passwords. Enter **2** to ensure the APEX_LISTENER and APEX_REST_PUBLIC_USER accounts have the same password as APEX_PUBLIC_USER.
h. Start standalone? You do not want to start in standalone mode so enter **2** to exit.

Copying the Images folder from the Application Express Installation Directory

You need to have a copy of the images folder on the server where the middle-tier is installed. This images folder includes all of the Oracle Application Express images available from the installation directory. Perform the following steps:

1) From the Mid Tier server, connect to the Database Tier server, and navigate to the Application Express installation directory of the apexdev instance.
2) Copy the images folder from apexdev installation on the Database Tier server, and place it under `/d1/home/webtier/wkdir/apexdev` on the Mid Tier server.

The Mid Tier server, `/d1/home/webtier/wkdir/apexdev` directory contents would now look like:

```
[webtier@slc04cif apexdev]$ ls
apex  images
[webtier@slc04cif apexdev]$
```

You see that `apexdev` directory includes an `images` folder (copied from the corresponding Application Express installation directory on the Database Tier server) and `apex` folder (Oracle REST Data Services configuration folder that was created earlier in this section).

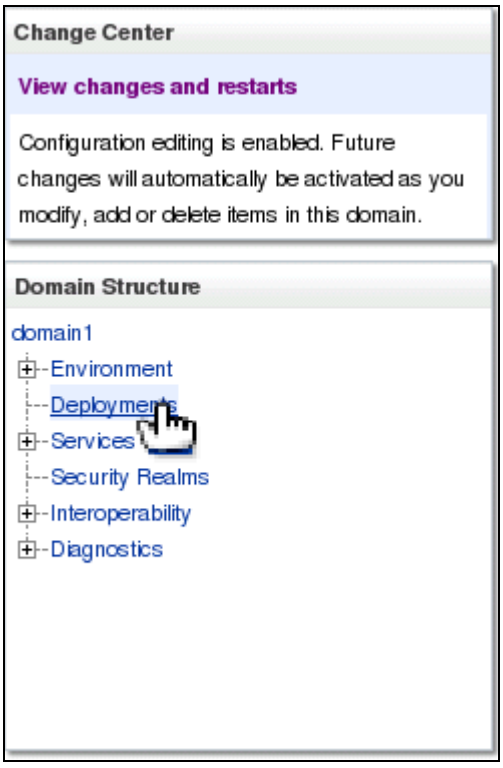
Note:

- As you have manually copied the Application Express images directory and placed it on the server, you do not need to create a web archive for Oracle Application Express images in this setup.
- You can include this images folder in a directory of your choice and do not need to necessarily use the same directory as mentioned in this section. However, make sure you specify the correct path of this images folder when you enter the reference in httpd.conf file. See *Configuring Oracle HTTP Server as a Front End to WebLogic* section.

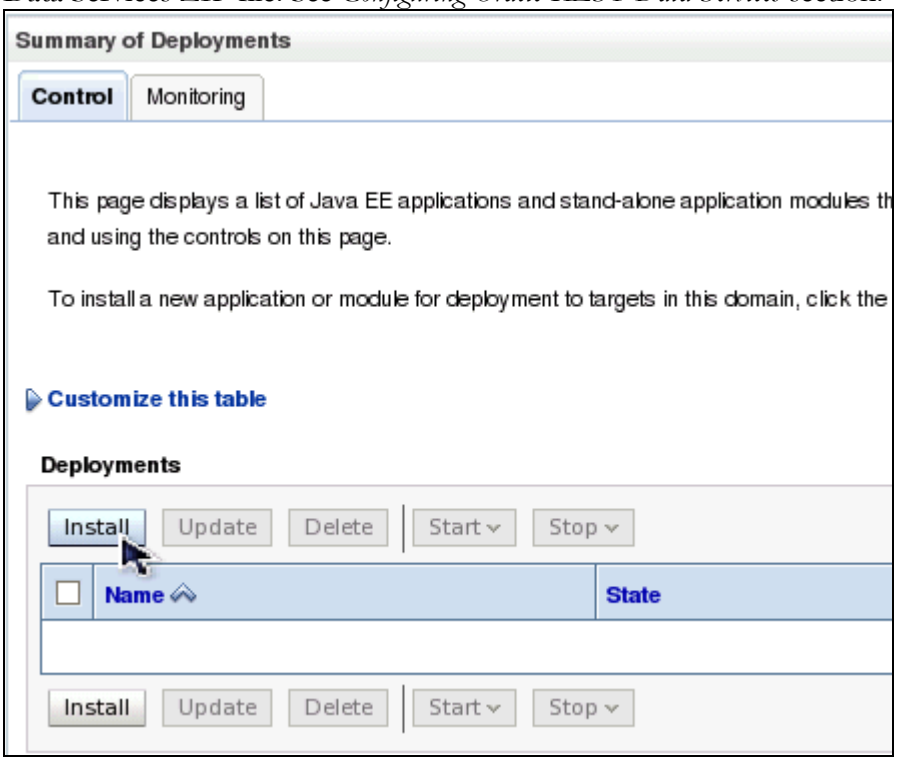
Deploying Oracle REST Data Services to Oracle WebLogic Server

In this section, you install Oracle REST Data Services deployment in Oracle WebLogic Server. The name of the deployment in the example is **apexdev** and the deployment target is the **http8080** managed server.
Perform the following steps:

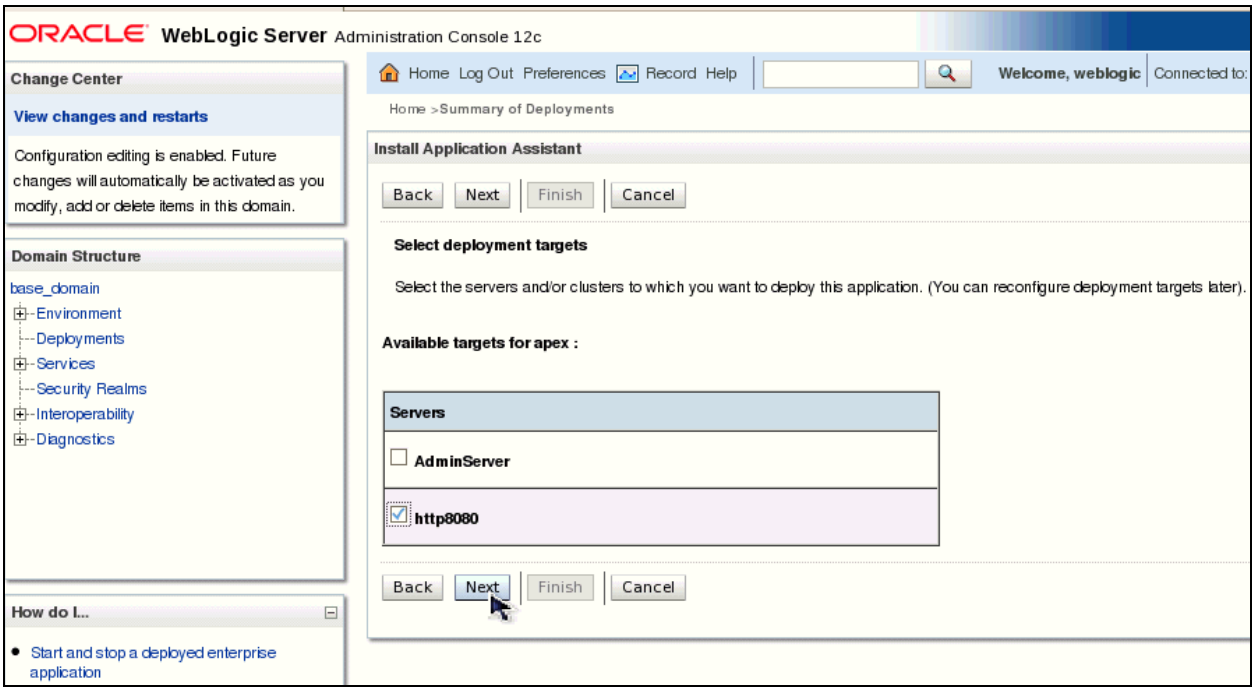
1. Launch the WebLogic Administration Console by typing the following URL in your web browser:
   `http://<hostname>:<port>/console`
   In general the port is 7001. Enter your WebLogic Administrator username and password.
2. Go to the WebLogic Server Home Page. Below Domain Structure, select **Deployments**. The Summary of Deployments displays.



3. Click **Install**. Specify the location of the apex.war file and click **Next**. Select the **`/d1/home/webtier/deploy/apexdev/apex.war`** file.
   Note: The apex.war file is located in the directory where you unzipped the Oracle REST Data Services ZIP file. See *Configuring Oracle REST Data Services* section.



4. Install Application Assistant appears. Select **Install this deployment as an application** and click **Next**.
5. Select the servers to which you want to deploy the application or module and click **Next**. In this example, select the **http8080** managed server.



6. Under Optional Settings:

    - Enter name for the deployment. In this example, enter apexdev.
    - Select **DD Only: Use only roles and policies that are defined in the deployment descriptors** radio button.

    Then, click **Next**.
7. Select **No, I will review the configuration later** radio button and click **Finish**.

Configuring WebLogic to Handle HTTP Basic Challenges Correctly

By default, WebLogic will attempt to intercept all HTTP Basic Authentication challenges. This default behavior needs to be disabled for Oracle REST Data Services to function correctly. Consult your WebLogic documentation for the location of the WebLogic configuration file named **config.xml**. In this document, the config.xml is located on the Mid Tier server, under: `/d1/home/webtier/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain/config/config.xml`.

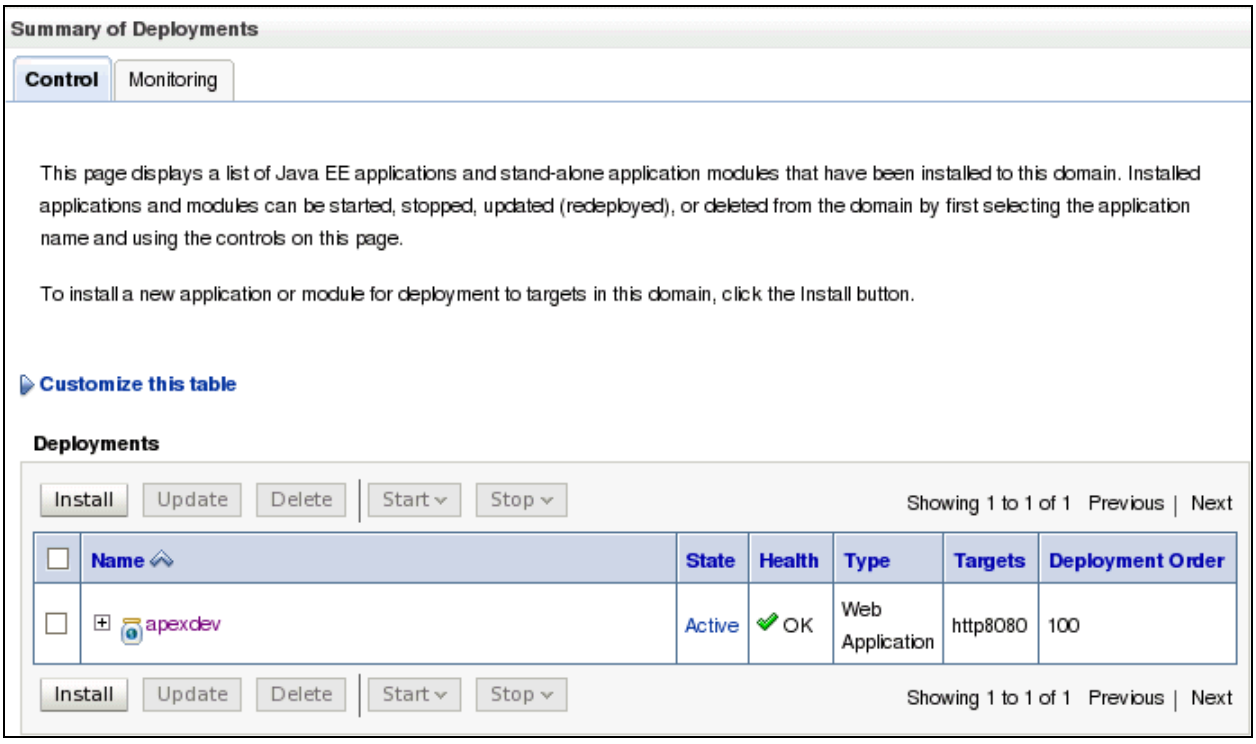
Add the **`<enforce-valid-basic-auth-credentials>`** element to config.xml within the `<security-configuration>` element. The edited file should look like the following:

```
    <default-realm>myrealm</default-realm>
    <credential-encrypted>{AES}PCYCi5iLpqkls6nIkWkgvuQePt1m+l0ZT7+1wzF4b1qxYMawaM0JxC
    <node-manager-username>admin</node-manager-username>
    <node-manager-password-encrypted>{AES}BG+Lc0sWf3fZ0ipTR4vrkkjxfY14slj9sCqvpVjVIfg
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
  </security-configuration>
  <server>
    <name>AdminServer</name>
    <listen-address></listen-address>
```

Save the updated config.xml file, and restart WebLogic if it is running.

Verifying the State of Deployments

In the Summary of Deployments, select the **Control** tab and verify that the deployment' State is Active. If it is not active, select the check box adjacent to the deployment name, **apexdev**, click **Start** and select **Servicing all requests** to make it active.



Performing Other Oracle Application Express tasks

In this section, you perform certain other tasks in Oracle Application Express as required by this architecture. Perform the following tasks for the Application Express installation (apexdev) on the Database Tier server.

- **Disabling Oracle XML DB Protocol Server**: In this setup, you are using Oracle HTTP Server. Therefore, you need to disable the existing Oracle XML DB HTTP Server. Change your working directory to the Application Express installation directory, and Log in to

SQL*Plus as SYS specifying the SYSDBA role. Then, execute the following command to disable Oracle XML DB Protocol server:

```
EXEC DBMS_XDB.SETHTTPPORT(0);
```

- **Configuring RESTful Services**: This step is optional and only needs to be completed if you wish to use RESTful Services with the Oracle REST Data Services on the instance. Change your working directory to the Application Express directory where you unzipped the installation software. Log in to SQL*Plus as SYS specifying the SYSDBA role. Then, run the following command:

  ```
  @apex_rest_config.sql
  ```

  When prompted, enter a password for the APEX_LISTENER and APEX_REST_PUBLIC_USER accounts.

- **Enabling Network Services in Oracle Database 11g**: By default, the ability to interact with network services is disabled in Oracle Database 11g Release 1 or later. Therefore, if you are running Oracle Application Express with Oracle Database 11g Release 1 or later, you must use the DBMS_NETWORK_ACL_ADMIN package to grant connect privileges to any host for the APEX_040200 database user.

For more information on Configuring RESTful Services and Enabling Network Services in Oracle Database 11*g*, see *Oracle Application Express Installation Guide*.

Configuring Oracle HTTP Server as a Front End to WebLogic

By default, routing between Oracle HTTP Server and Oracle WebLogic Server is not configured. In order to allow requests to be routed from Oracle HTTP Server to Oracle WebLogic Server, you must manually edit and make some changes to the OHS configuration files.

The main configuration file is **`httpd.conf`** that includes information about listening ports, server names, virtual hosts, proxy configurations, and the like. This file also configures Secure Sockets Layer (SSL) support by defining information such as certificates and other HTTPS configuration directives. This file is available at the following location: **`DOMAIN_HOME/config/fmwconfig/components/OHS/instances/componentName`**. For this implementation the file is under `/d1/home/webtier/Oracle/ohs/user_projects/domains/base_domain/config/fmwconfig/components/OHS/ohs1`.

The mod_wl_ohs module allows requests to be proxied from an Oracle HTTP Server to Oracle WebLogic Server. The mod_wl_ohs module is installed and loaded out-of-the-box with Oracle HTTP Server, but it is not configured by default. The mod_wl_ohs.conf file is available under the location: **`DOMAIN_HOME/config/fmwconfig/components/OHS/instances/componentName`**. For this implementation the file is under `/d1/home/webtier/Oracle/ohs/user_projects/domains/base_domain/config/fmwconfig/components/OHS/ohs1`.

In this section, you will modify the **mod_wl_ohs.conf** file on the Oracle HTTP Server to reference the Oracle WebLogic server where the Oracle REST Data Services is configured. After editing and

saving the changes in `mod_wl_ohs.conf` file, you modify the `httpd.conf` file to enter values for ports, include host information, and provide Oracle Application Express references.

Perform the following steps:

1) On the Mid Tier server, navigate to the location of **`mod_wl_ohs.conf`**. Then, modify this file to enter the host and port of the WebLogic Server and specify location details. Your `mod_wl_ohs.conf` should include contents like the screenshot below:

```
# NOTE : This is a template to configure mod_weblogic.

LoadModule weblogic_module   "${PRODUCT_HOME}/modules/mod_wl_ohs.so"

# This empty block is needed to save mod_wl related configuration from EM to thi
s file when changes are made at the Base Virtual Host Level
<IfModule weblogic_module>
#      WebLogicHost <WEBLOGIC_HOST>
#      WebLogicPort <WEBLOGIC_PORT>
#      MatchExpression *.jsp
</IfModule>

# <Location /weblogic>
#      SetHandler weblogic-handler
#      PathTrim /weblogic
#      ErrorPage  http:/WEBLOGIC_HOME:WEBLOGIC_PORT/
#  </Location>
<IfModule weblogic_module>

<Location /apex/>
      SetHandler weblogic-handler
      WebLogicHost 127.0.0.1
      WeblogicPort 8080
      Debug ON
      WLLogFile ${ORACLE_INSTANCE}/diagnostics/logs/${COMPONENT_TYPE}/${COMPONEN
T_NAME}/wlsdebug-8080.log
      DebugConfigInfo ON
      KeepAliveEnabled on
      KeepAliveSecs 10
    </Location>
</IfModule>
```

Save the changes to this file.

2) Now, navigate to the location of the `httpd.conf` file and modify the file to enter values for OHS listen port, User, Server name, and Virtual Host Configuration.

The key sections of the httpd.conf file include:

a) **OHS Listen Port**: Enter values for the OHS Listen Port. Use **7777** as the port for the apexdev instance.

```
#
# Listen: Allows you to bind Apache to specific IP addresses
and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below
to
# prevent Apache from glomming onto all bound IP addresses
(0.0.0.0)
#
# Listen 12.34.56.78:80

# OHS Listen Port
Listen 7777
```

b) **User**: Enter the operating system user to run the httpd.conf, in this case **webtier**.

```
#
# User/Group: The name (or #number) of the user/group to run
httpd as.
#   . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#   . On HPUX you may not be able to use shared memory as
nobody, and the
#     suggested workaround is to create a user www and use that
user.
#  NOTE that some kernels refuse to setgid(Group) or semctl
(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group #-1 on these systems!
#
User webtier
#Group GROUP_TEMPLATE
</IfModule>
```

c) **Server Name**: Enter the IP address or the host name of the machine where the middle-tier is installed.

```
# ServerName gives the name and port that the server uses to
identify itself.
# This can often be determined automatically, but we recommend
you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP
address here.
#
ServerName <hostname>
```

d) **Virtual Host Configuration**: Include the virtual host configuration details such as port, and location of the Application Express images directory. The screenshot below shows the configuration details with the port 7777.

```
include "moduleconf/*.conf"

NameVirtualHost *:7777

<VirtualHost *:7777>
    <Directory  /d1/home/webtier/wkdir/>
       AllowOverride None
       Order allow,deny
       Allow from all
    </Directory>

    Alias /i/ /d1/home/webtier/wkdir/apexdev/images/

    ExpiresActive On
    ExpiresDefault A0
    <filesMatch "\.(jpg|jpeg|ico|gif|png|swf|js|css)$">
        ExpiresDefault A300
        Header append Cache-Control "public"
    </filesMatch>
</VirtualHost>
```

e) **mod_deflate**: You can enable mod_deflate that allows for compressing files and delivering them to the browsers.

```
LoadModule deflate_module
"${ORACLE_HOME}/ohs/modules/mod_deflate.so"
#HTTP Compression
<IfModule mod_deflate.c>
  SetOutputFilter DEFLATE
  SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-
vary
  SetEnvIfNoCase Request_URI \.(?:exe|t?gz|zip|bz2|sit|rar)$ no-
gzip dont-vary
  SetEnvIfNoCase Request_URI \.(?:pdf|doc?x|ppt?x|xls?x)$ no-
gzip dont-vary
</IfModule>
```

Once you complete editing and saving the httpd.conf file, make sure you restart the OHS. For information on how to start OHS using command line, see *Starting Oracle HTTP Server* section.

Note: Instead of modifying both mod_wl_ohs.conf and httpd.conf files, you can simply edit the httpd.conf file and provide all of the configuration details. However, if you are using FMW control, it understands the changes only within mod_wl_ohs file. Therefore, using both the mod_wl_ohs and httpd.conf (as described in this document) is the recommended option. A further benefit is that you

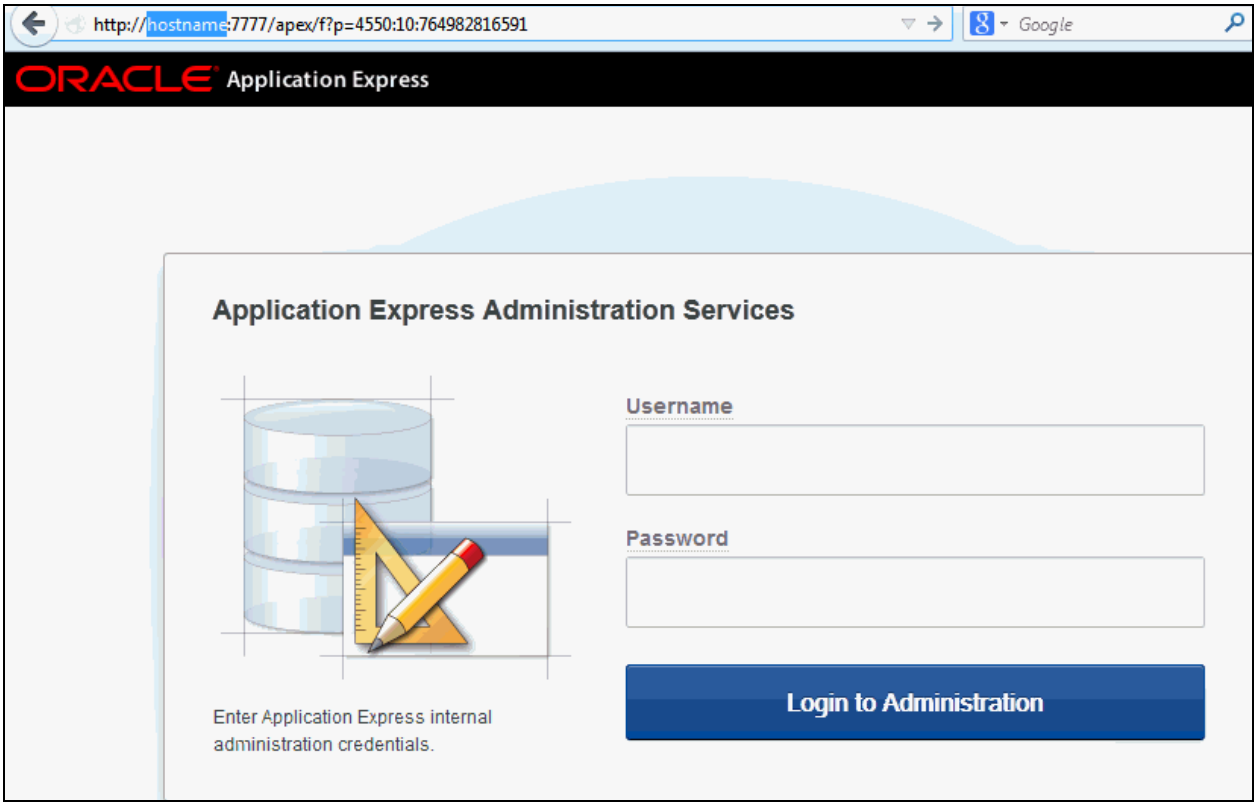
can simply comment out references to mod_wl_ohs.conf within the httpd.conf file, if required for easy troubleshooting.

Testing the environment

To test the environment, first make sure to start the WebLogic Server and Oracle HTTP Server.

Note: You can automate the startup process so that the servers keep running in the background.

Then, open a browser and enter the URL http://<hostname>:7777/apex/apex_admin. You can now see Oracle Application Express Administration Services log in page that looks like:



Performing Multiple Deployments of the Web Tier

You have learned how to perform a single deployment in the *Performing a Single Deployment of the Web Tier* section. Another typical scenario might be to set up multiple Oracle Application Express instances. For example, you would like to have three different instances for development, testing, and production. You can use either of the following two approaches to perform multiple deployments:

- **Configure Oracle REST Data Services to use URL mapping:** Oracle REST Data Services supports the ability to connect to more than one database. It provides a number of

different strategies for routing requests to the appropriate database. All of these strategies rely on examining the request URL and choosing the database based on some kind of match against the URL. For more information on URL mapping, see *Oracle REST Data Services Installation and Configuration Guide*.

- **Extend the single deployment architecture for multiple deployments of the web tier:** You need to repeat all of the following steps for each new additional deployment that you would like to make:
    1. Install a new Oracle Database
    2. Install and configure Oracle Application Express
    3. Add a new managed server in the WebLogic domain and then, associate it with the machine.
    4. Configure a new deployment of Oracle REST Data Services. Make sure to copy the images folder from the Application Express installation directory and place it on the server where the Oracle REST Data Services is installed.
    5. Deploy Oracle REST Data Services to Oracle WebLogic Server
    6. Update the Oracle HTTP Server configuration files: **mod_wl_ohs.conf** and **httpd.conf**

This document covers how to extend the single deployment architecture for multiple deployments of the web tier. In this example, you set up the Oracle Application Express instance **apexstage**. On the Database Tier server, you should have already installed an Oracle Database with the SID **orcl2**. For more information, see the *Prerequisites* section.

On the Database Tier Server:

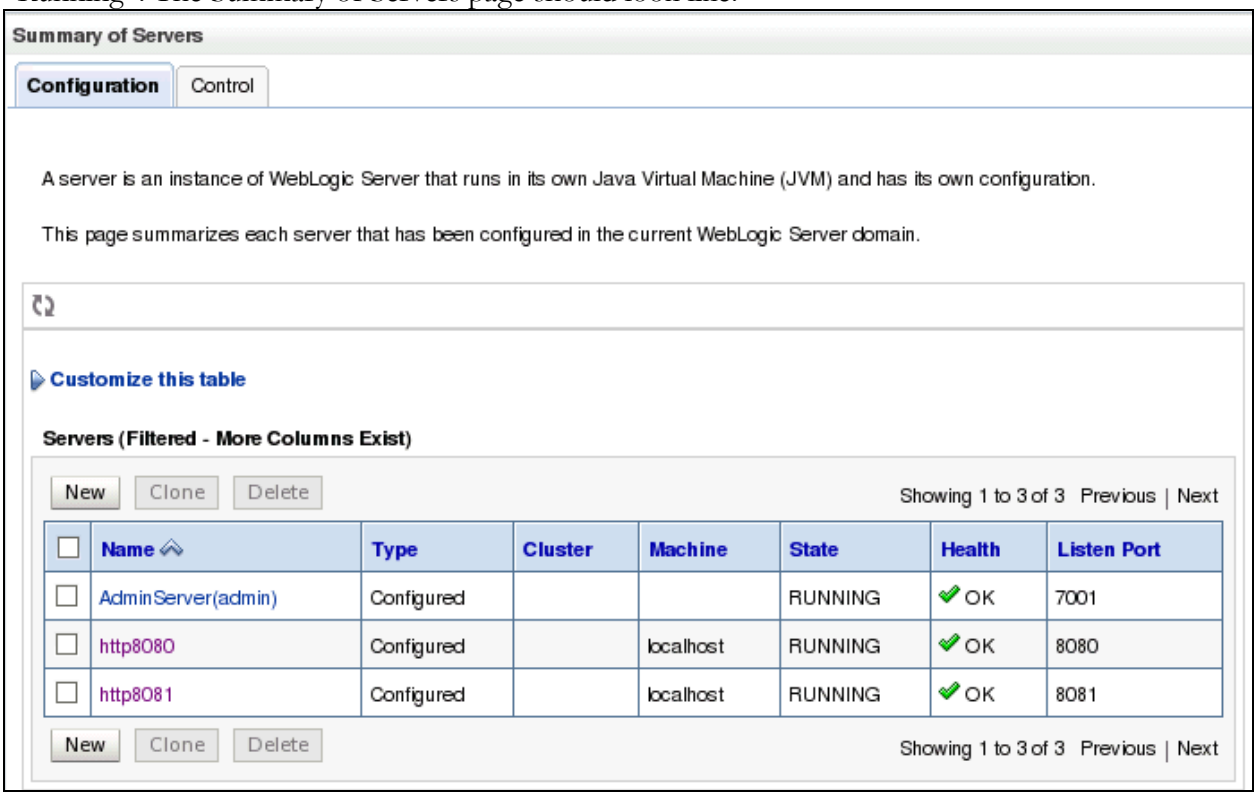
1. Unzip the downloaded Application Express installation zip file. Then, repeat all of the steps in the *Installing and Configuring Oracle Application Express* section.
2. Perform all of the tasks in the *Performing Other Oracle Application Express Tasks* section.

Configuring the WebLogic Domain

In this section, you create a new managed server in the WebLogic domain. Perform the following steps:

1. Log in to the WebLogic Server Administration Console. Under Domain Structure > <your_domain_name> >Environment, click **Servers**.
2. Click **New**. Under Server Properties, enter **http8081** for Server Name and **8081** for Server Listen Port. Click **Next** and then click **Finish**.
3. From the Servers table, click **http8081**.
4. For machine, select **localhost** and then click **Save**.
5. Now, navigate to Servers and click the **Control** tab.
6. Select the checkbox for the http8081 server and click **Start**. You can see that a request has been sent to the Node Manager to start the selected server.

7. Click the **Configuration** tab and refresh the page to verify that each server's State is "Running". The Summary of Servers page should look like:



Configuring a new Deployment of Oracle REST Data Services

On the Mid Tier server, perform the following steps:

1. Create two directories as: `wkdir/apexstage` and `deploy/apexstage`
2. Unzip the downloaded **ords.version.number.zip** under `/deploy/apexstage`.
3. Change your active directory to `/deploy/apexstage`. Rename the **`ords.war`** file to **`apex.war`**.
4. Next, you configure the database connection details appropriate to where Oracle Application Express was installed. Execute the following command:
   `java -jar apex.war`

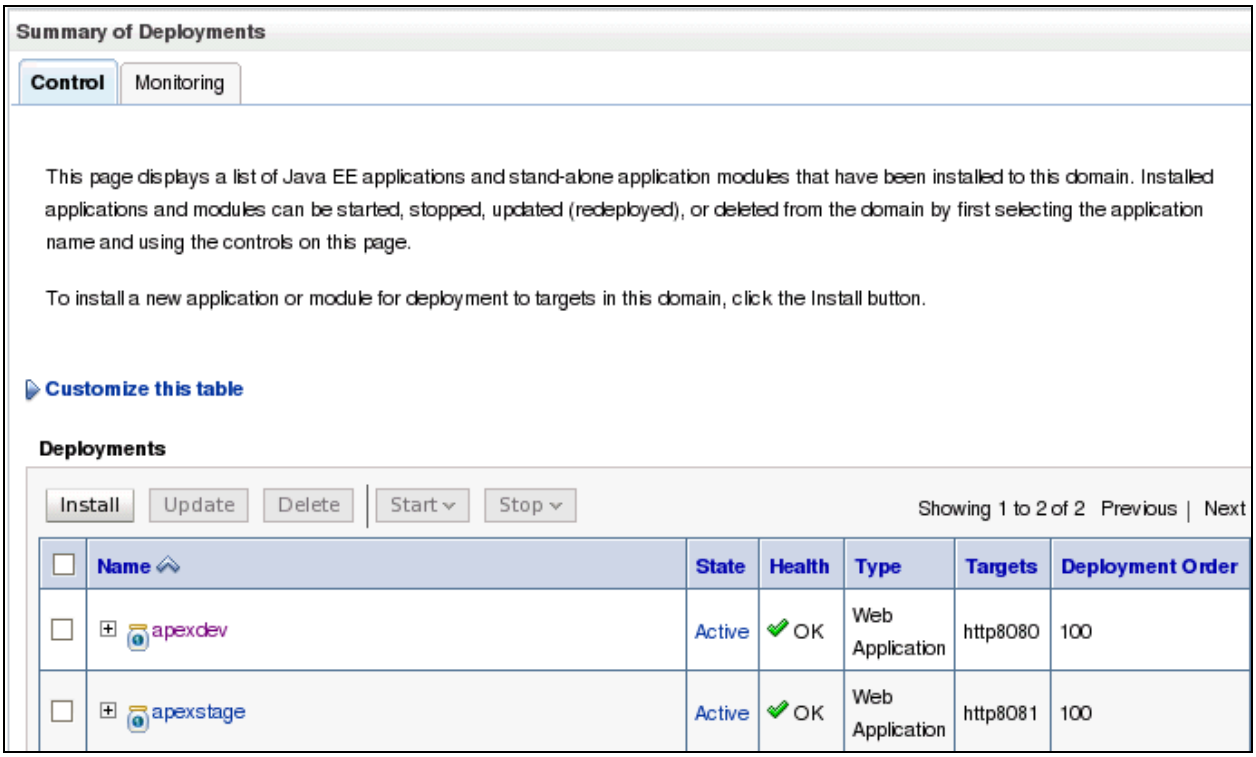
```
Enter the location to store configuration data:/d1/home/webtier/wkdir/apexstage
Apr 4, 2014 1:00:02 AM oracle.dbtools.common.config.cmds.ConfigDir execute
INFO: Set config.dir to /d1/home/webtier/wkdir/apexstage in: /d1/home/webtier/de
ploy/apexstage/apex.war
Apr 4, 2014 1:00:05 AM oracle.dbtools.common.config.file.ConfigurationFolder log
ConfigFolder
INFO: Using configuration folder: /d1/home/webtier/wkdir/apexstage/apex
Enter the name of the database server [localhost]: <hostname>
Enter the database listen port [1521]:1522
Enter 1 to specify the database service name, or 2 to specify the database SID [
1]:2
Enter the database SID [xe]:orcl2
Enter the database user name [APEX_PUBLIC_USER]:
Enter the database password for APEX_PUBLIC_USER:
Confirm password:
Enter 1 to enter passwords for the RESTful Services database users (APEX_LISTENE
R,APEX_REST_PUBLIC_USER), 2 to use the same password as used for APEX_PUBLIC_USE
R or, 3 to skip this step [1]:2
Apr 4, 2014 1:01:13 AM oracle.dbtools.common.config.file.ConfigurationFiles upda
te
INFO: Updated configurations: defaults, apex, apex_al, apex_rt
Enter 1 if you wish to start in standalone mode or 2 to exit [1]:2
```

Enter the following data:

- Configuration folder: **`/d1/home/webtier/wkdir/apexstage`**
- Name of the database server: <hostname> of the Database Tier server
- Listener port: **1522** (or whatever appropriate per your database installation)
- Database SID: **orcl2** (or whatever appropriate per your database installation)
- Database User and Password: APEX_PUBLIC_USER/<Password> specified during Oracle Application Express installation for this instance

5. You need to copy the images from the Application Express installation directory. From the Mid Tier server, connect to the Database Tier server and then navigate to the Application Express installation directory of the apexstage instance. Copy the images folder and place it under `/d1/home/webtier/wkdir/apexstage` on the Mid Tier server.
6. Now, perform all of the steps as mentioned in the *Deploying Oracle REST Data Services to Oracle WebLogic Server* section. Enter the following details in the wizard steps:
   - apex.war location: Select the **`/d1/home/webtier/deploy/apexstage/apex.war`**
   - Available targets for apex: **http8081** managed server
   - Name for the deployment: **apexstage**

The Summary of Deployments page looks like:



Modifying the Oracle HTTP Server Configuration Files

In this section, you update the two configuration files `mod_wl_ohs.conf` and `httpd.conf`. Perform the following steps on the Mid Tier server:

1. Update `mod_wl_ohs.conf` with the WebLogic port and location details. For example, your mod_wl_ohs.conf should look like:

```
<IfModule weblogic_module>

<Location /apex/>
      SetHandler weblogic-handler
      WebLogicHost 127.0.0.1
      WeblogicPort 8080
      Debug ON
      WLLogFile ${ORACLE_INSTANCE}/diagnostics/logs/
${COMPONENT_TYPE}/${COMPONENT_NAME}/wlsdebug-8080.log
      DebugConfigInfo ON
      KeepAliveEnabled on
      KeepAliveSecs 10
    </Location>


<Location /apex/>
      SetHandler weblogic-handler
      WebLogicHost 127.0.0.1
      WeblogicPort 8081
      Debug ON
      WLLogFile ${ORACLE_INSTANCE}/diagnostics/logs/
${COMPONENT_TYPE}/${COMPONENT_NAME}/wlsdebug-8081.log
      DebugConfigInfo ON
      KeepAliveEnabled on
      KeepAliveSecs 10
    </Location>


</IfModule>
```

2. Update `httpd.conf` with the OHS Listen Port. For example, in this document, enter **7778** as the port for the apexstage instance.

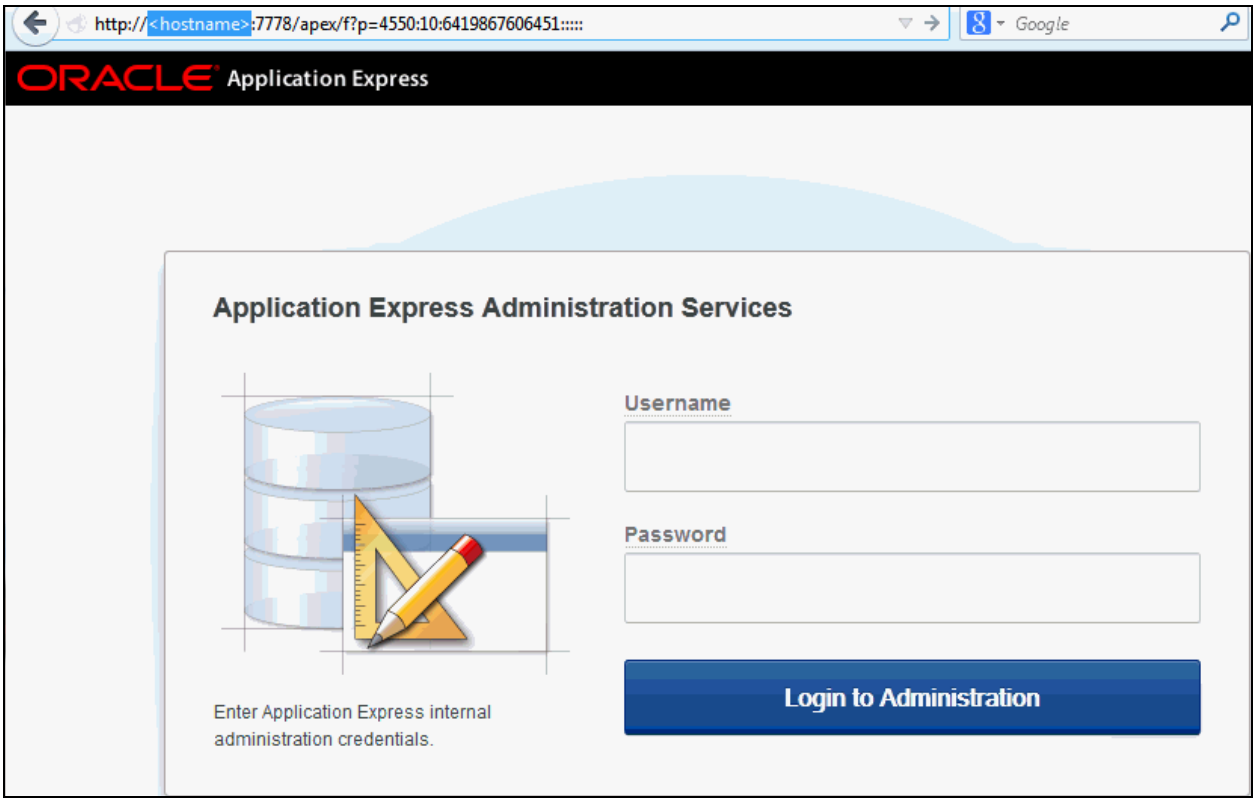
```
#
# Listen: Allows you to bind Apache to specific IP addresses
and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below
to
# prevent Apache from glomming onto all bound IP addresses
(0.0.0.0)
#
# Listen 12.34.56.78:80

# OHS Listen Port
Listen 7777
Listen 7778
```

3. Update `httpd.conf` to include the virtual host configuration details such as port and location of the Application Express images directory. Your httpd.conf should look like:

```
include "moduleconf/*.conf"

NameVirtualHost *:7777
NameVirtualHost *:7778

<VirtualHost *:7777>
    <Directory  /d1/home/webtier/wkdir/>
       AllowOverride None
       Order allow,deny
       Allow from all
    </Directory>

    Alias /i/ /d1/home/webtier/wkdir/apexdev/images/

    ExpiresActive On
    ExpiresDefault A0
    <filesMatch "\.(jpg|jpeg|ico|gif|png|swf|js|css)$">
        ExpiresDefault A300
        Header append Cache-Control "public"
    </filesMatch>
</VirtualHost>


<VirtualHost *:7778>
    <Directory  /d1/home/webtier/wkdir/>
       AllowOverride None
       Order allow,deny
       Allow from all
    </Directory>

    Alias /i/ /d1/home/webtier/wkdir/apexstage/images/

    ExpiresActive On
    ExpiresDefault A0
    <filesMatch "\.(jpg|jpeg|ico|gif|png|swf|js|css)$">
        ExpiresDefault A300
        Header append Cache-Control "public"
    </filesMatch>
</VirtualHost>
```

Testing the Environment

Open a browser and enter the URL http://<hostname>:7778/apex/apex_admin . You can now see Oracle Application Express Administration Services log in page that looks like:



Summary

The most common scenario of setting up an Oracle Application Express production environment is to separate the database tier and middle-tier. The focus of this white paper is how to set up this architecture so that the middle-tier is deployed on one host and multiple Application Express instances are installed on another host. This document provides guidelines and instructions on how to set up each of the components involved in a middle-tier deployment for Oracle Application Express.

Example Web Listener Deployment for Oracle Application Express

May 2014

Authors: Chaitanya Koratamaddi

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

Hardware and Software, Engineered to Work Together