Oracle® Fusion Middleware

Using Oracle Data Integrator on Oracle Cloud Marketplace

12.2.1.4.0
F22173-10
July 2023

Oracle Fusion Middleware Using Oracle Data Integrator on Oracle Cloud Marketplace, 12.2.1.4.0

F22173-10

Primary Author: Oracle Corporation

Contents

Preface

This book describes how to use Oracle Data Integrator on Oracle Cloud Marketplace.

This preface contains the following topics:

Audience

This document helps you to use Oracle Data Integrator on Oracle Cloud Marketplace.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in *Oracle Data Integrator Library*:

- *Release Notes for Oracle Data Integrator*
- *Understanding Oracle Data Integrator*
- *Developing Integration Projects with Oracle Data Integrator*
- *Installing and Configuring Oracle Data Integrator*
- *Upgrading Oracle Data Integrator*
- *Integrating Big Data with Oracle Data Integrator Guide*
- *Application Adapters Guide for Oracle Data Integrator*
- *Developing Knowledge Modules with Oracle Data Integrator*
- *Connectivity and Knowledge Modules Guide for Oracle Data Integrator Developer's Guide*

- *Oracle Data Integrator Tools Reference*
- *Data Services Java API Reference for Oracle Data Integrator*
- *Open Tools Java API Reference for Oracle Data Integrator*
- *Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator*
- *Java API Reference for Oracle Data Integrator*
- *Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator*
- *Oracle Data Integrator 12c Online Help*, which is available in ODI Studio through the JDeveloper Help Center when you press **F1** or from the main menu by selecting **Help**, and then **Search** or **Table of Contents**.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1
Using ODI Studio

This chapter helps you to configure and use ODI Studio on Oracle Cloud Marketplace.

For detailed information on the terminologies associated with ODI Studio and their respective functions, see Terminology Information.

It contains the following sections:

- Accessing ODI Studio
- Using Autonomous Databases in ODI
- Working with ODI Instance

1.1 Accessing ODI Studio

To access ODI studio through VNC, do the following:

1. Install a VNC viewer on your local computer.

> **Note:**
>
> In order to access the ODI instance using VNC, perform the following:
>
> a. Log in to the provisioned ODI instance on Oracle Cloud Marketplace using SSH as opc user:
> `ssh -i <private_key> opc@<<IP Address>`
>
> b. Execute the following firewall commands to open the VNC ports:
>
> ```
> sudo firewall-cmd --permanent --new-service=odissh
> sudo firewall-cmd --permanent --service=odissh --set-
> description="ODI SSH
> server"
> sudo firewall-cmd --permanent --service=odissh --add-
> port=5901-5905/tcp
> sudo firewall-cmd --permanent --add-service=odissh
> sudo firewall-cmd --reload
> ```

2. Use SSH to connect to the compute instance running the Oracle Data Integrator Image, as described in Connecting to ODI Compute Instance .
3. On your local computer, connect to your instance and create a ssh tunnel for port 5901 (for display number 1):

```
$ ssh -L 5901:localhost:5901 -i id_rsa oracle@<IP Address>
```

4. On your local computer, for the VNC to work, add an Ingress rule as follows:

   ```
   No 0.0.0.0/15 TCP All 5901 TCP traffic for ports: 5901
   ```

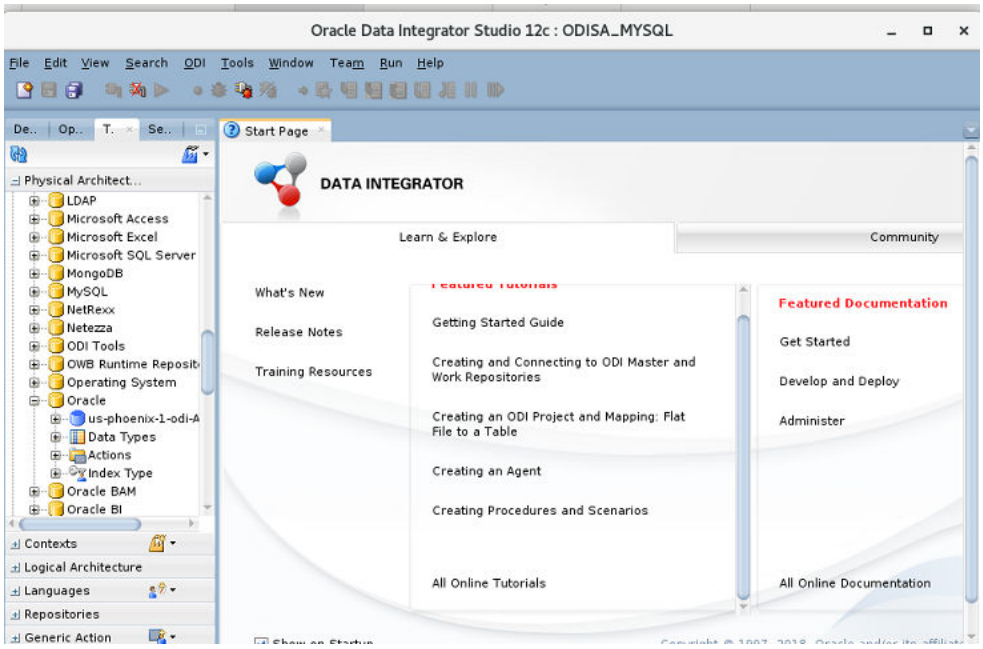
5. On your local computer, start a VNC viewer and establish a VNC connection to `localhost:1`.
6. Enter the VNC password that you had provided during the stack creation.
7. For connecting multiple users, after the vncpasswd utility exits, start the VNC server by typing `vncserver`. This will start a VNC server with display number 1 for the oracle user, and the VNC server starts automatically if your instance is rebooted. For example `vncserver@:2` or `vncserver@:3`.

To launch the ODI instance,

1. From the Applications menu, navigate to Programming -> ODI Studio
   or

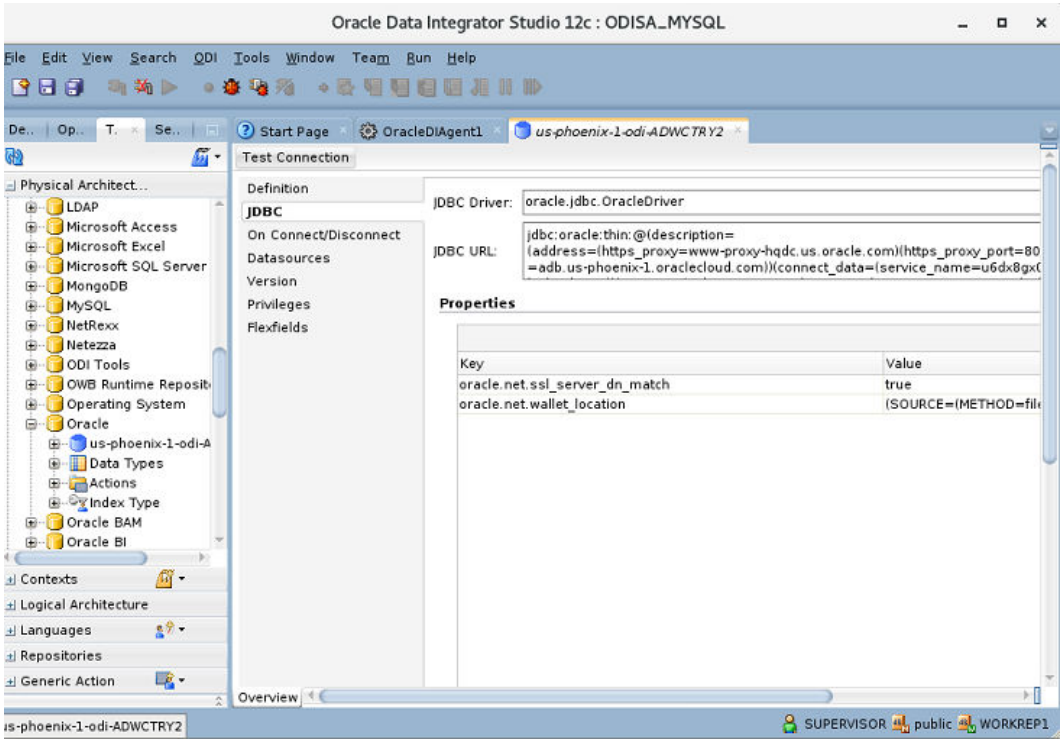
   Double click the short icon for ODI Studio present in your Desktop
   or

   Navigate to the location `$MW_HOME/oracle/odi/studio/bin/odi` in the VNC.
2. Connect to the repository with already populated login credentials. The Login Name value varies based on the selected repository. For ADB repostiory, the Login Name is `ODI_ADW_REPO` and for MySQL Embedded repository it is `ODISA_MYSQL`.
3. Post successful configuration, check if the newly created data server is available in the `Topology navigator -> Technologies -> Oracle`.



   For more details on MySQL services, refer to Troubleshooting ODI on OCI

   .
4. In ODI studio, navigate to `Topology -> Physical Architecture -> Agent -> OracleDIAgent1` and click Test, to check if the Standalone Agent is working.
5. Depending on your network, you may need to provide proxy details for the database server JDBC connection.



6. Click Test connection, to check if the created ADB Dataserver is working.
7. Depending on your network, you can setup a proxy for ODI. In ODI Studio, navigate to Tools, Preferences, Web Browser and Proxy, to setup a proxy for your network. Proxy may be required for accessing certain hosts, for example - Oracle Object Storage.

> **Note:**
>
> Depending on your OCI network configurations, you may or may not require access through proxy-hosts. While you are connecting through proxy, make sure that the proxy address/port or the source dataserver is allowed through OCI VCN configurations.

> **Note:**
>
> If you are using a BI Cloud Connector Dataserver, you may need to add the BI Cloud Connector host to the Proxy Exclusion field.

1.2 Using Autonomous Databases in ODI

The newly created ODI repository will be pre-populated with Oracle Data Servers representing all accessible Autonomous Databases based on defined policies. If you aim to use any of these as a part of your ODI transformations, then you have to add the username and password to the Data Server properties in the Topology tab in ODI Studio.

If, at a later date, more Autonomous Databases become available to you, you can use the "Discover ADB's" feature available in Create New Dataserver on Oracle Technology of ODI Studio, to quickly setup the additional instances that were not available at the time when the

instance was created. When you select the required ADB instance from displayed instance list, the wallet gets auto downloaded and once you provide the dataserver name, credentials and then select connection details/service profiles and save, the new Oracle Dataserver for the selected ADB instance is created.

To create an Autonomous (ADB) Dataserver in ODI repository,

- Connecting to the Pre-created ADB Dataservers in ODI Repository
- Using Dataserver Setup in ODI Studio

Connecting to the Pre-created ADB Dataservers in ODI Repository

Connect to the readily available or pre-created ADB dataservers in ODI studio. You have to add actual username and password by connecting to the dataserver, do a test connection and continue with your data integration project in ODI studio.

> **Note:**
>
> You need to provide the username and password for the created instance as prepopulated login credentials may not work.

Using Dataserver Setup in ODI Studio

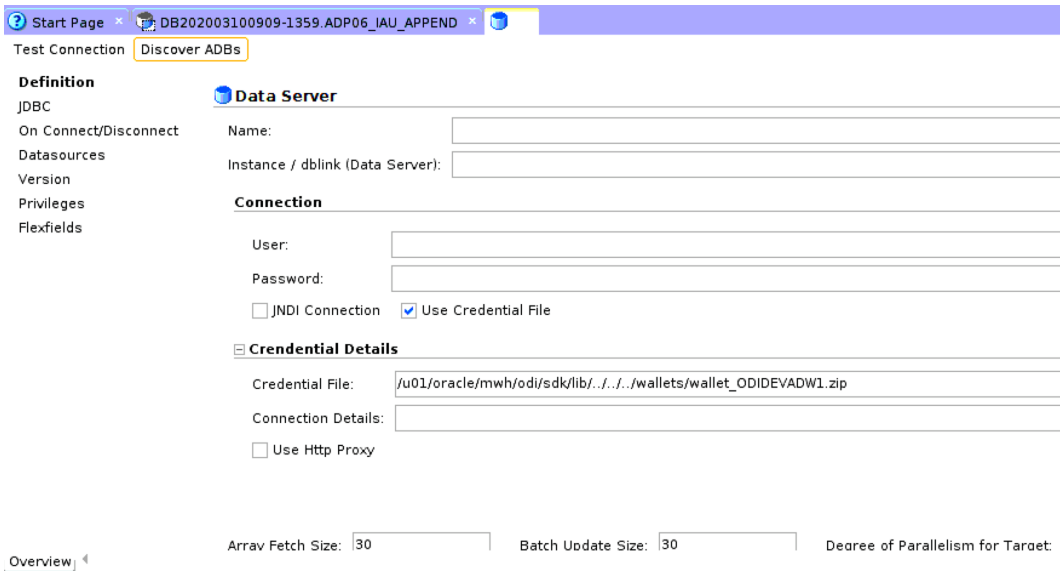
You can create additional ADB Dataservers using the Oracle technologies Dataserver setup in ODI Studio.

Navigate to the Topology navigator, expand the Technologies node in the Physical Architecture navigation tree and under Oracle technology, select any pre-created ADB Dataserver.

1. In the Definition tab, click Discover ADBs. The list of available ADB instances are displayed.
2. Select the required ADB instance from the Discover Autonomous Databases drop-down list.
   Upon selection, **Use Credential File** checkbox is auto-selected in the connection node.

   In the Credential Details node, **Credential File** text box is auto-populated with the respective mapped credential file.

Figure 1-1 Discover ADBs



3. In the Data Server node,
   - Name: Enter the name of the newly created data server.
   - Instance/dblink(Data Server): TNS Alias used for this Oracle instance. It will be used to identify the Oracle instance when using database links and SQL*Loader.
4. In the Connection node,
   - User/Password: Oracle user (with its password), having select privileges on the source schemas, select/insert privileges on the target schemas and select/insert/object creation privileges on the work schemas that will be indicated in the Oracle physical schemas created under this data server.
   - JNDI Connection: Select this check-box to configure the JNDI connection settings. Navigate to the JNDI tab, and fill in the required fields.
5. In the Credentials Details node,
   - Connection Details - Click the Connection Details drop down arrow to choose the required connection URL from the list of available connection URLs retrieved from tnsnames.ora.
6. Click Test Connection.

Upon successful test connection, the new Dataserver gets created in the ODI repository.

1.3 Working with ODI Instance

This chapter guides you to connect and work with the ODI instance.

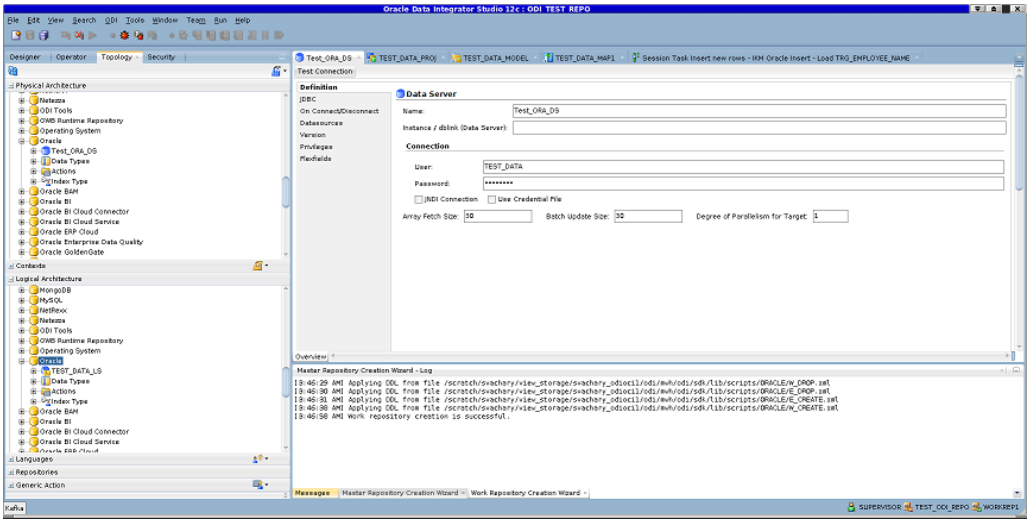
It contains the following sections:

- Configuring Data Sources and Targets
- Reverse Engineering Data Models
- Creating Mappings
- Monitoring ODI Executions

1.3.1 Configuring Data Sources and Targets

The physical components that store and expose structured data in Oracle Data Integrator (ODI) are defined as data servers. Each data server is always linked to a single technology. It stores information according to a specific technical logic, which is declared in the physical schemas attached to it.

For example -

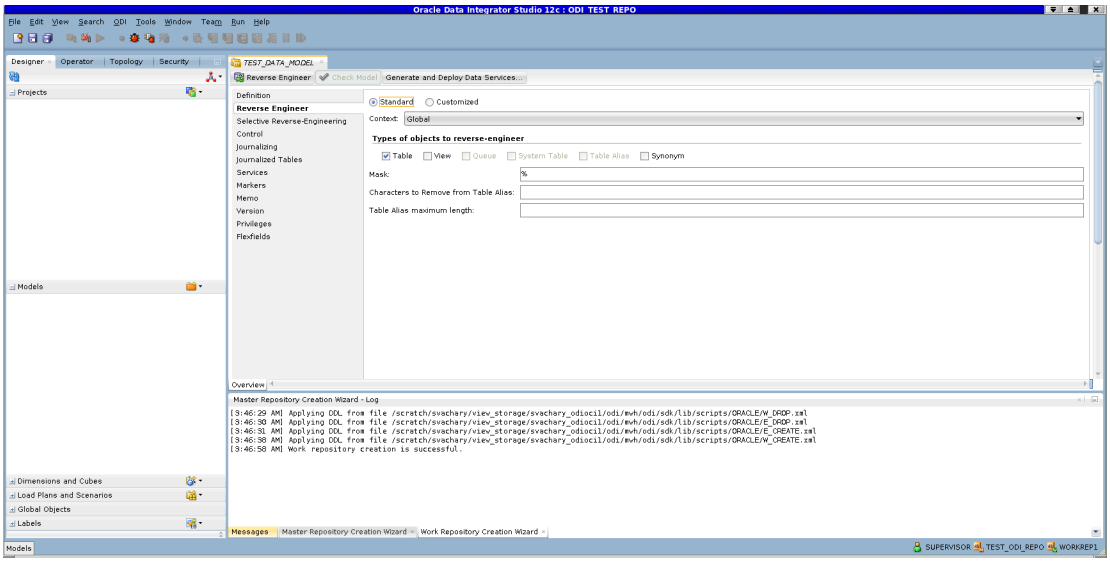


For more information, refer to Overview of Oracle Data Integrator Topology chapter in *Developing Integration Projects with Oracle Data Integrator* guide.

1.3.2 Reverse Engineering Data Models

To automatically populate datastores into the model, you reverse-engineer the model. A standard reverse-engineering uses the capacities of the JDBC driver used to connect the data server to retrieve the model metadata. A customized reverse-engineering uses a reverse-engineering Knowledge Module (RKM), to retrieve metadata for a specific type of technology and create the corresponding datastore definition in the data model.

For example -

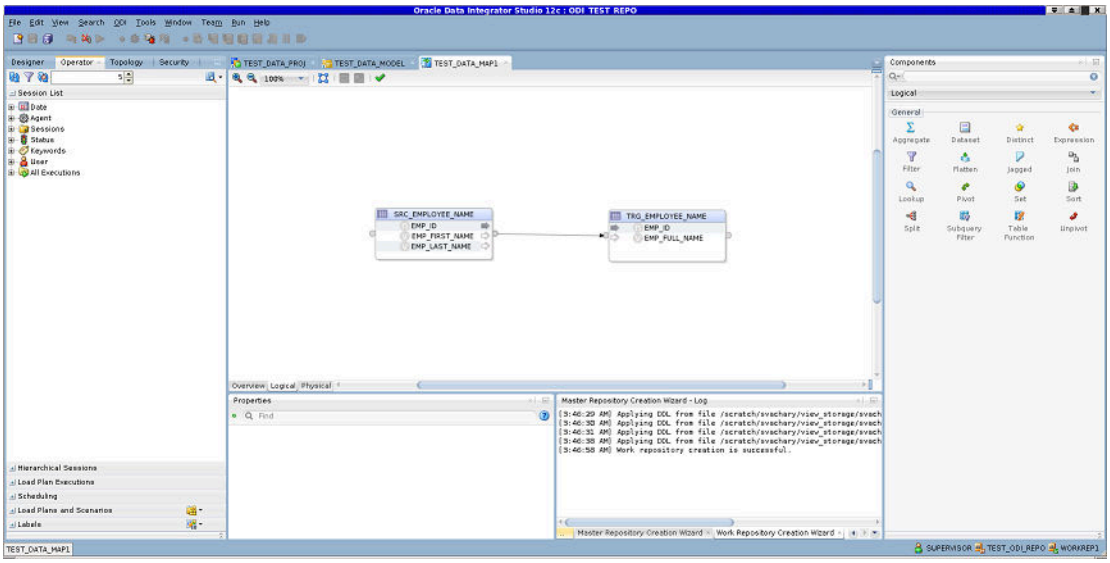


For more information, refer to Creating and Reverse-Engineering a Model chapter in *Developing Integration Projects with Oracle Data Integrator* guide.

1.3.3 Creating Mappings

Mappings in Oracle Data Integrator (ODI) are the logical and physical organization of your data sources, targets, and the transformations through which the data flows from source to target. Mappings are made up of several parts, datastores, datasets, re-usable mappings, connectors, knowledge modules, variables, sequences, user functions, and other components. Optionally, you can specify a staging schema. You create and manage mappings using the mapping editor, which opens whenever you open a mapping. Mappings are organized in folders under individual projects, found under Projects in the Designer Navigator.

For example -

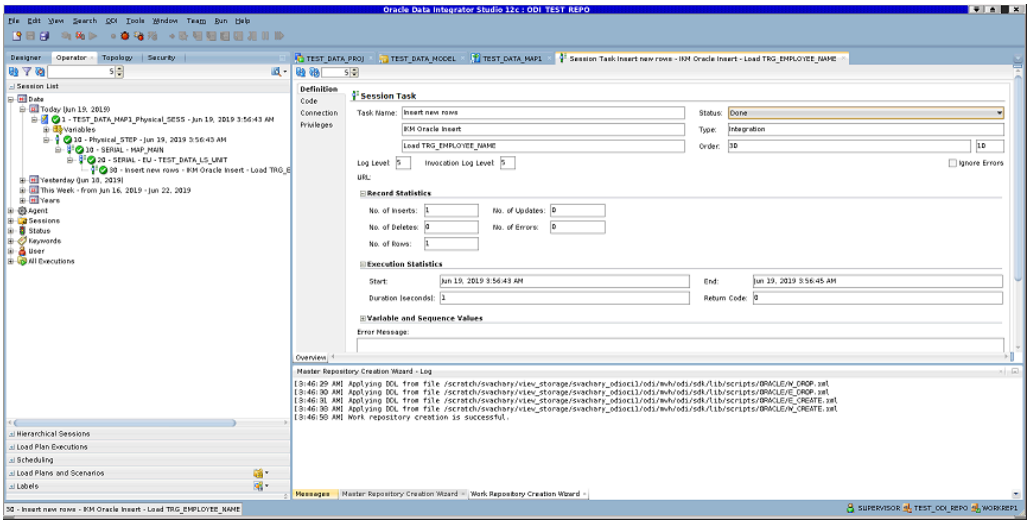


For more information, refer to Creating and Using Mappings in *Developing Integration Projects with Oracle Data Integrator* guide.

1.3.4 Monitoring ODI Executions

Monitoring your development executions consists of viewing the execution results and managing the development executions when the executions are successful or in error. Through Operator Navigator, you can view your execution results and manage your development executions in the sessions, as well as the Scenarios and Load Plans in production. Operator navigator stores this information in a work repository, while using the topology defined in the master repository.

For example -



For more information, refer to Monitoring Integration Processes chapter in *Administering Oracle Data Integrator* guide.

2

Managing ODI Setup

This chapter helps you to manage the ODI setup that you have provisioned on Oracle Cloud Marketplace.

It contains the following sections:

2.1 Working with ODI Linux Services

The following table lists all the available services in ODI marketplace installation for applicable technology and stack deployment:

Name of the Linux Service	Database Technology	Type of Stack Deployment	Supported Release Versions	Funtions
`agentodi.service`	`MYSQL,ADB`	ODI Studio	Supported in release(s) prior to 12.2.1.4.200618 version of ODI Marketplace.	You can start, stop and check the status of the service.
`mysqlodi.service`	`MYSQL`	ODI Studio	Supported in release(s) prior to 12.2.1.4.200618 version of ODI Marketplace.	You can start, stop and check the status of the service.
`manageodiapps.service`	`MYSQL,ADB`	ODI Studio	Supported only from ODI Web V12.2.1.4.200618 and later versions of ODI Marketplace.	You can only check the status of the service. Use the python commands listed below, to start, stop and restart the ODI Agent.

2.2 Changing Repository in Oracle Data Transforms Administrator

Follow the below procedure to change repository in Oracle Data Transforms (if not already created):

1. Launch ODI Studio.
2. Select the option **Connect to Repository**.
3. Create new login using '+' icon and provide the connection details.
4. Click **Test** and then click **OK**, if test connection is successful.
5. Click **OK** on the Oracle Data Integrator Login dialog.

2.3 Switching Repositories of the ODI App Server

You can switch to any existing ADB or DBCS repository from existing ODI VM Instance.

> **Note:**
>
> You can switch between repositories only when the repositories in stack mode and repo mode match.

In ODI App Server, you can switch repository in the following technologies:

- Switching from ADB to ADB
- Switching from MYSQL to ADB or DBCS

> **Note:**
>
> But the reverse (switching back to MYSQL from ADB or DBCS) is not supported.

- Switching from DBCS to DBCS
- Switching from DBCS or ADB

> **Note:**
>
> Stop the server before running any configuration using the following command:
>
> ```
> python manageOdiApps.py shutdown
> ```
>
> For more information, refer to Managing ODI App Server.

Switching Between ADB Repositories

If you already have a ADB repository in which you have your transformation project developed and wish to continue with your development in the same repository, follow the below procedure to switch from the new ADB repository (that you just created) to your existing ADB repository:

1. Create `odi-setup.properties` file in the location `$MW_HOME/odi/common/scripts` and if the file already exists, clear the existing content of the file and then add the following properties:

```
dbTech=ADB
rcuCreationMode=false
odiSchemaPassword=<valid password>
odiSchemaUser=<odi schema username>
odiSupervisorPassword=<odi SUPERVISOR password>
walletZipLoc=<path_to_zipped_wallet>
workRepoName=<WORK REPO NAME>
adwInstancePassword= <adw Instance password>
```

> **Note:**
>
> - `workRepoName=<WORK REPO NAME>` is an optional property but you may have to configure this property if your default work repository name is not `WORKREP`.
> - `adwInstancePassword= <adw Instance password>` is an optional property but configure this property only when you have used OPTACH for applying a patch on your ODI instance and wish to run Upgrade Assistant (UA) using the configuration script `odiMPConfiguration.py`.

2. Create `repository.properties` file in the location `$MW_HOME/odi/common/scripts` and if the file already exists, clear the existing content of the file and then add the following properties:

```
masterReposDriver=oracle.jdbc.OracleDriver
masterReposUser=<odi schema username>
workReposName=<WORK REPO NAME>
```

3. Navigate to the location `$MW_HOME/odi/common/scripts` directory and execute the following python scripts in the given order:

```
python odiMPConfiguration.py
python manageOdiApps.py start
```

> **Note:**
>
> Stop the server before running any configuration. For more information on this, refer to Managing ODI App Server.

Switching Between DBCS Repositories

If you already have a DBCS repository in which you have your transformation project developed and wish to continue with your development in the same repository, follow the below procedure to switch from the new DBCS repository (that you just created) to your existing DBCS repository:

1. Create `odi-setup.properties` file in the location `$MW_HOME/odi/common/scripts` and if the file already exists, clear the existing content of the file and then add the following properties:

   ```
   dbTech=DBCS
   dbHost=<IP Address of the DBCS Instance>
   dbPort=<port of DBCS Instance>
   dbServiceName=<Service Name of DBCS Instance>
   odiSchemaUser=<odi schema username>
   odiSchemaPassword=<valid password>
   odiSupervisorPassword=<odi SUPERVISOR password>
   workRepoName=<WORK REPO NAME>
   ```

2. Create `repository.properties` file in the location `$MW_HOME/odi/common/scripts` and if the file already exists, clear the existing content of the file and then add the following properties:

   ```
   masterReposDriver=oracle.jdbc.OracleDriver
   masterReposUser=<odi schema username>
   workReposName=<WORK REPO NAME>
   ```

3. Navigate to the location `$MW_HOME/odi/common/scripts` directory and execute the following python scripts in the given order:

   ```
   python odiMPConfiguration.py
   python manageOdiApps.py start
   ```

2.4 Managing ODI App Server

The following commands help you to manage ODI App server associated with your provisioned ODI instance on Oracle Cloud Marketplace.

Application available in ODI Studio are:

```
APPODIAGENT
```

You can use ODI App Server to manage all the ODI applications deployed in ODI App Server.

Navigate to the location `$MW_HOME/odi/common/scripts` to run the following commands:

- Use the following command to check the status of the service (as the oracle user):

  ```
  systemctl status manageodiapps.service
  ```

> **Note:**
>
> You cannot use this command to start or stop the service.

- Use the following command to start the service:

```
python manageOdiApps.py start
```

- Use the following command to shutdown the service:

```
python manageOdiApps.py shutdown
```

- Use the following command to restart the service:

```
python manageOdiApps.py restart
```

> **Note:**
>
> When you execute any of the above `python manageOdiApps.py` commands, the terminal holds the session to run the jetty sever. Open a new terminal, if you wish to perform any other operations.

- Use the following command to start all the applications associated with the service:

```
python manageOdiApps.py start -apps=<allowed values>

allowed values: all or APPODIAGENT with combination separated by ","
```

- Use the following command to stop all the applications associated with the service:

```
python manageOdiApps.py stop -apps=<allowed values>

allowed values: all or APPODIAGENT with combination separated by ","
```

> **Note:**
>
> When you execute the command `python manageOdiApps.py`, two log files `odiagent.log` and `odi_adp_rest_txt.log` are created. For details on the location of the files, refer to Log Files Location.

- Use the following command to get the status of all applications associated with the service:

```
python manageOdiApps.py status
```

- If you have provisioned this stack prior to 12.2.1.4.200618 release version of ODI Marketplace or if you have provisioned this stack for ODI Studio, follow the below procedure to manage your ODI Agent lifecycle:

- To stop the ODI Agent:

```
python stopAgent.py
```

- To start the ODI Agent:

```
python startAgent.py $MW_HOME
```

2.5 Managing ODI Credential

If you have either updated the odi schema password on the database or the SUPERVISOR password in ODI repository, you can use the `manageCredentials.py` script to update or manage ODI credentials required to start the ODI App Server successfully.
Navigate to the location `$MW_HOME/odi/common/scripts` to run the following commands:

S.No.	Key Name
1	`odiSchemaPassword`
2	`odiSupervisorPassword`

Use the following command to set the credential key in the Credential Store:

```
python manageCredentials.py set <Key Name>=<value>
```

Enclose the password string with single quotes so that the Linux shell treats the string as an exact value and does not parse the contents. For example:

```
python manageCredentials.py set odiSchemaPassword='pas$word'
```

Use the following command to get the credential key value stored in the Credential Store:

```
python manageCredentials.py read <key Name>
```

2.6 Configuring Proxy Settings

Depending on your network, you can setup a proxy for ODI. Proxy may be required for accessing certain hosts, for example - Oracle Object Storage.

> **Note:**
>
> Depending on your OCI network configurations, you may or may not require access through proxy-hosts. While you are connecting through proxy, make sure that the proxy address/port or the source dataserver is allowed through OCI VCN configurations.

You can set proxy:

- In ODI Studio or ODI Studio Administrator
- For ODI Agent
- In ODI App server

To set proxy in ODI Studio and Oracle Data Transforms Administrator, navigate to Tools, Preferences, Web Browser and Proxy, to setup a proxy for your network.

Follow the below procedure to set proxy for ODI Agent if, you have provisioned this stack prior to 12.2.1.4.200618 release version of ODI Marketplace:

> **Note:**
>
> For backward compatibility, use the scripts `startAgent.py` and `stopAgent.py` to manage ODI Agent Lifecycle.

1. From the location `$MW_HOME/oracle/odi/common/scripts`, locate and edit the file `startAgent.py` and add the following lines after the property after `-Drepo.props=`

   ```
   -Xms1024m -Xmx4048 -cp


   -Dhttp.proxyHost=www-proxy-xxx.com -Dhttp.proxyPort=80
   -Dhttps.proxyHost=www-proxy-xxx.com -Dhttps.proxyPort=80 -cp
   ```

   For example, after adding the above lines, your file should be like this:

   ```
   subprocess.call('nohup java
   -Drepo.props=odi-setup.properties
    -Xms1024m -Xmx4048 -cp
   -Dhttp.proxyHost=www-proxy-xxx.com -Dhttp.proxyPort=80
   -Dhttps.proxyHost=www-proxy-xxx.com -Dhttps.proxyPort=80 -cp
   $AGENTCLASSPATH oracle.odi.OdiStandaloneAgentStarter'+'
   '+oracledIagentPath+"
   &", shell=True)
   ```

2. Save the file and use the following command to start the agent:

   ```
   python startAgent.py $MW_HOME
   ```

   > **Note:**
   >
   > Ensure you do not add any extra lines or space or tab on the file `startAgent.py`. Just add `-D` option within the line content. It is a python script and it requires proper line indentation to work.

3. Test the standalone agent from ODI studio to see if the agent has started successfully. Then execute the packages/mappings using the standalone agent.

> **Note:**
>
> If you are using a BI Cloud Connector Dataserver, you may need to add the BI Cloud Connector host to the Proxy Exclusion field.

Follow the below procedure to set proxy in ODI App Server:

1. Open the script file `manageOdiApps.py`.
2. Find the below lines in the file:

```
JETTY_SERVER_COMMAND_STR = 'java -DAPP_LOGS='+APP_LOGS+' -
Dconfig.template.file=../../apps/webapps.template.yaml -
Dapps.config=../../apps/webapps.yaml -Drepo.props=odi-
setup.properties -Drestrepo.props=repository.properties -
Djetty.enabled=true -Dagent.logging.config=../logging/agent-logging-
config.xml -cp $CLASSPATH
oracle.odi.setup.util.ODIMPJettyServerAppsManager
```

3. After the above lines, add the below line before `-cp`:

```
-Dhttp.proxyHost=<proxyhost> -Dhttp.proxyPort=<proxy port> -
Dhttps.proxyHost=<proxyhost> -Dhttps.proxyPort=<proxy port>
```

4. Save the file.
5. Restart the ODI App server.

2.7 Configuring Email Delivery Service

Oracle Cloud Marketplace Email Delivery is an email sending service that provides a fast and reliable managed solution for sending high-volume emails that need to reach your recipients' inbox.

Email Delivery provides the tools necessary to send application-generated email for mission-critical communications such as receipts, fraud detection alerts, multi-factor identity verification, and password resets. You can set up the Email Delivery service within the Console. To begin sending email with Email Delivery, complete the following steps:

- Generate SMTP credentials for a user
- Set up permissions
- Create an approved sender
- Configure SPF on the approved sender domain
- Configure the SMTP connection
- Begin sending email

> **Note:**
>
> Before configuring the Email Delivery service, make sure to have permissions to Generate SMTP credentials and create Email Approved Senders. Also, the Email Approved Sender must be in a group that has IAM policy permissions to send outgoing emails. For more details, refer to Generate SMTP Credentials for a User section of OCI documentation.

Generating a SMTP Credential

Simple Mail Transfer Protocol (SMTP) credentials are necessary to send email through Email Delivery. Each user is limited to a maximum of two SMTP credentials. If more than two are required, SMTP credentials must be generated on other existing users or more users must be created.

- To generate SMTP credentials for a user, login to Oracle Cloud Infrastructure and navigate to **Email Delivery → Manage Credentials** and select the option **Generate SMTP Credentials**. It allows you to generate the SMTP user name and password details. Copy the generated password for your future reference. Click **Close**.

Setting Up Permissions

An email approved sender must be in a group that has IAM policy permissions to send emails. The approved sender must be in a compartment with permissions to manage approved senders. You have to create a policy to manage approved senders in the entire tenant, if the approved senders exist in root compartment.

Add the following policy statement to enable `odi_group` to manage approved senders:

```
Allow dynamic-group odi_group to use approved-senders in compartment odi
```

For more information about policies and policy syntax, see Policy Basics.

Creating your Email Approved Sender

You must set up an approved sender for all “From:” addresses sending email via Oracle Cloud Infrastructure or the email will be rejected. An approved sender is associated with a compartment and only exists in the region where the approved sender was configured.

> **Note:**
>
> Approved senders should not be created in the root compartment.

Creating approved senders in a compartment other than the root allows the policy to be specific to that compartment.

- To create your Email Approved Sender, login to Oracle Cloud Infrastructure and navigate to **Email Delivery → Email Approved Senders** and select the option **Create Approved Senders**.

> **Note:**
>
> Configure this option for the user already created on the instance.

For example, `opc@oracle-odi-inst-3mnc.localdomain`, where `oracle-odi-inst-3mnc` is the hostname.

Configuring SPF on the Approved Sender Domain

Configure SPF, if necessary. The Approved Senders section within the Console provides validation of an SPF record for each of your approved senders. SPF is required for subdomains of oraclegovcloud.com and recommended in other cases.

Refer to Configure SPF for detailed steps on configuring SPF.

Configuring the SMTP connection

For securing your email connections, get SSL/TLS CA details from OCI email SMTP hosts

1. Log in to the instance using ssh as opc user and sudo su and create a directory `nss-config-dr` and then run `certutil` to manage keys and certificate in both NSS databases.

```
[root@localhost ~]# mkdir /etc/certs
[root@localhost ~]# cd /etc/certs
[root@localhost certs]# certutil -N -d /etc/certs/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.
Enter new password:
Re-enter password:
[root@localhost certs]# ls
cert8.db key3.db secmod.db
[root@localhost certs]#
```

2. To get SMTP domain CA details, run `openssl s_client` to smtp host.

> **Note:**
>
> - If it is on ashburn: `openssl s_client -showcerts -connect smtp.us-ashburn-1.oraclecloud.com:587 -starttls smtp > /etc/certs/mycerts-ashburn`
> - If it is on phoenix : `openssl s_client -showcerts -connect smtp.us-phoenix-1.oraclecloud.com:587 -starttls smtp > /etc/certs/mycerts-phoenix`

For example:

```
[root@localhost certs]# openssl s_client -showcerts -connect smtp.us-
phoenix-1.oraclecloud.com:587 -starttls smtp > /etc/certs/mycerts-phoenix
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert
Global Root CA
verify return:1
depth=1 C = US, O = DigiCert Inc, CN = DigiCert SHA2 Secure Server CA
verify return:1
depth=0 C = US, ST = California, L = Redwood City, O = Oracle
Corporation, OU = Oracle DYN-DEV US, CN = smtp.us-
phoenix-1.oraclecloud.com
verify return:1
250 Ok
[root@localhost certs]#
```

3. Execute `cat` on `mycerts-phoenix` or `ashburn` and copy each certificate including the --BEGIN CERTIFICATE-- and --END CERTIFICATE-- and paste it to their respective files. For example -

```
ocismtp-phoenix1.pem ocismtp-phoenix2.pem ocismtp-phoenix3.pem
[root@localhost certs]# ls -la | grep -i ocism
-rw-r--r--. 1 root root 2443 Jan 31 18:00 ocismtp-phoenix1.pem
-rw-r--r--. 1 root root 1648 Jan 31 18:01 ocismtp-phoenix2.pem
-rw-r--r--. 1 root root 1338 Jan 31 18:01 ocismtp-phoenix3.pem
[root@localhost certs]#

[root@localhost certs]# cat ocismtp-phoenix1.pem
-----BEGIN CERTIFICATE-----
MIIG3jCCBcagAwIBAgIQDD6TwDfguDbn1CI1U46l0zANBgkqhkiG9w0BAQsFADBN
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMScwJQYDVQQDEx5E
aWdpQ2VydCBTSEEyIFNlY3VyZSBTZXJ2ZXIgQ0EwHhcNMTgxMjA3MDAwMDAwWhcN
MjEwMTA1MTIwMDAwWjCBnjELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3Ju
aWExFTATBgNVBAcTDFJlZHdvb2QgQ2l0eTEbMBkGA1UEChMST3JhY2xlIENvcnBv
cmF0aW9uMRowGAYDVQQLExFPcmFjbGUgRFlOLURFViBVUzEqMCgGA1UEAxMhc210
cC51cy1waG9lbml4LTEub3JhY2xlY2xvdWQuY29tMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEA2ZUXc/xjwwlmsaSKxy2e0Y3K9UeWs/MQSBkQALC0+Pi9
tIdS7BLmYtpTjGmUpwiNzG9pMYHpWjQlQFkxNpqd6JwegpgdEG/8SnbrhH9kRsRg
MG8kRNZiJYsDrpwLnjE74gNIjVldqbcMHmBinfKbfFAcPzp5sqOFw3hfSz8TU45A
7UHfbWmF3HiLF+Ozhnr0cUdiVb79HVYH4fm15V4uwewj/ZvALmK000jdOaeOgOna
vrx30WSqfkoqOpferIrW4a6wsrj82vaAjuxgBU3rbuaJb2KFYYes3SeUoFkFAZp7
URMy3DZD7MmgmWIXnjGu75xqF4Ul/uEF6cjnYeuDpwIDAQABo4IDZjCCA2IwHwYD
VR0jBBgwFoAUD4BhHIIxYdUvKOeNRji0LOHG2eIwHQYDVR0OBBYEFCN96Xt5uS1q
xt2ZgTWONBD4VHfdMCwGA1UdEQQlMCOCIXNtdHAudXMtcGhvZW5peC0xLm9yYWNs
ZWNsb3VkLmNvbTAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEG
CCsGAQUFBwMCMGsGA1UdHwRkMGIwL6AtoCuGKWh0dHA6Ly9jcmwzLmRpZ2ljZXJ0
LmNvbS9zc2NhLXNoYTItZzYuY3JsMC+gLaArhilodHRwOi8vY3JsNC5kaWdpY2Vy
dC5jb20vc3NjYS1zaGEyLWc2LmNybDBMBgNVHSAERTBDMDcGCWCGSAGG/WwBATAq
MCgGCCsGAQUFBwIBFhxodHRwczovL3d3dy5kaWdpY2VydC5jb20vQ1BTMAgGBmeB
DAECAjB8BggrBgEFBQcBAQRwMG4wJAYIKwYBBQUHMAGGGGh0dHA6Ly9vY3NwLmRp
Z2ljZXJ0LmNvbTBGBggrBgEFBQcwAoY6aHR0cDovL2NhY2VydHMuZGlnaWNlcnQu
Y29tL0RpZ2lDZXJ0U0hBMlNlY3VyZVNlcnZlckNBLmNydDAJBgNVHRMEAjAAMIIB
fQYKKwYBBAHWeQIEAgSCAW0EggFpAWcAdgDuS723dc5guuFCaR+r4Z5mow9+X7By
2IMAxHuJeqj9ywAAAWeJXuinAAAEAwBHMEUCIQDqeInMySXAN1UDIJOLG3v/ViBJ
xsY3lK2JY/zwebUaugIgepOPAwKQdVrnY7CMCzWGGGqJbLgkFWIRMGK0FUJ8+RsA
```

```
dQCHdb/nWXz4jEOZX73zbv9WjUdWNv9KtWDBtOr/XqCDDwAAAWeJXumPAAAEAwBG
MEQCIA1jRQ0797YV7BLzCANvicAsYk2QdGjCuZ4YxxRgTIs+AiBRztTbnjiT9WGE
HIRVEJa/Bx7eSlcu7J2gpEZruOWrFwB2ALvZ37wfinG1k5Qjl6qSe0c4V5UKq1Lo
GpCWZDaOHtGFAAABZ4le6LcAAAQDAEcwRQIgMk9G/KNM9xR3GR9q/2vEB85skPlL
EgDFVpKBQxQN2f8CIQD2Cn54OAL8HkDDYglLpAjTnzaSUJeP2h07NG90xS5VOjAN
BgkqhkiG9w0BAQsFAAOCAQEAP8q05wiAKVkvv+Y6l0aPclFiW5/yZmnQeGNE85kx
CmQgbdeGcNUgQ9PjDaBMhHMErVasq1E//oYjuRuF4bFO9QYYMn2QOuz1p61s+60/
IDNCP8xJuBAJ61Gu0mAw7mm44Z+jfD1LMdg/xyZwlH9wFZID9lgVdqpvhlLiYRNy
zBtKfgLhzu2B08T4a/V3w2SaDyhPIED2ry+HV+9B7CnzpmLrSqRFw7kk9ihm9Iwq
YlyJV3qzO1tIykRALDvYAT50yd+d9ZfTcEQvSrMLoM6N0HJezdTnf67UqwYFF5jT
KhyG/2LIAn4XGK0AyS8ieCmmEnW1Hku2ykCo4Ls0gdcYOA==
-----END CERTIFICATE-----
[root@localhost certs]#

[root@localhost certs]# cat ocismtp-phoenix2.pem
-----BEGIN CERTIFICATE-----
MIIElDCCA3ygAwIBAgIQAf2j627KdciIQ4tyS8+8kTANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0xMzAzMDgxMjAwMDBaFw0yMzAzMDgxMjAwMDBaME0xCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxJzAlBgNVBAMTHkRpZ2lDZXJ0IFNIQTIg
U2VjdXJlIFNlcnZlciBDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
ANyuWJBNwcQwFZA1W248ghX1LFy949v/cUP6ZCWA1O4Yok3wZtAKc24RmDYXZK83
nf36QYSvx6+M/hpzTc8zl5CilodTgyu5pnVILR1WN3vaMTIa16yrBvSqXUu3R0bd
KpPDkC55gIDvEwRqFDu1m5K+wgdlTvza/P96rtxcflUxDOg5B6TXvi/TC2rSsd9f
/ld0Uzs1gN2ujkSYs58O09rg1/RrKatEp0tYhG2SS4HD2nOLEpdIkARFdRrdNzGX
kujNVA075ME/OV4uuPNcfhCOhkEAjUVmR7ChZc6gqikJTvOX6+guqw9ypzAO+sf0
/RR3w6RbKFfCs/mC/bdFWJsCAwEAAaOCAVowggFWMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYYY
aHR0cDovL29jc3AuZGlnaWNlcnQuY29tMHsGA1UdHwR0MHIwN6A1oDOGMWh0dHA6
Ly9jcmwzLmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEdsb2JhbFJvb3RDQS5jcmwwN6A1
oDOGMWh0dHA6Ly9jcmw0LmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEdsb2JhbFJvb3RD
QS5jcmwwPQYDVR0gBDYwNDAyBgRVHSAAMCowKAYIKwYBBQUHAgEWHGh0dHBzOi8v
d3d3LmRpZ2ljZXJ0LmNvbS9DUFMwHQYDVR0OBBYEFA+AYRyCMWHVLyjnjUY4tCzh
xtniMB8GA1UdIwQYMBaAFAPeUDVW0Uy7ZvCj4hsbw5eyPdFVMA0GCSqGSIb3DQEB
CwUAA4IBAQAjPt9L0jFCpbZ+QlwaRMxp0Wi0XUvgBCFsS+JtzLHgl4+mUwnNqipl
5TlPHoOlblyYoiQm5vuh7ZPHLgLGTUq/sELfeNqzqPlt/yGFUzZgTHbO7Djc1lGA
8MXW5dRNJ2Srm8c+cftIl7gzbckTB+6WohsYFfZcTEDts8Ls/3HB40f/1LkAtDdC
2iDJ6m6K7hQGrn2iWZiIqBtvLfTyyRRfJs8sjX7tN8Cp1Tm5gr8ZDOo0rwAhaPit
c+LJMto4JQtV05od8GiG7S5BNO98pVAdvzr508EIDObtHopYJeS4d60tbvVS3bR0
j6tJLp07kzQoH3jOlOrHvdPJbRzeXDLz
-----END CERTIFICATE-----

[root@localhost certs]# cat ocismtp-phoenix3.pem
-----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIQCDvgVpBCRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0wNjExMTAwMDAwMDBaFw0zMTExMTAwMDAwMDBaMGExCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IENBMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUKKPC3eQyaKl7hLOllsB
CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTtxRuLWZscFs3YnFo97
nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt
43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P
```

```
T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4
gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95QNVbR
TLtm8KPiGxvDl7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDl7I90VUw
DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr
hMAtudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg
06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5BjfIttep3Sp+dWOIrWcBAI+0tKIJF
PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls
YSEY1QSteDwsOoBrp+uvFRTp2InBuThs4pFsiv9kuXclVzDAGySj4dzp30d8tbQk
CAUw7C29C79Fv1C5qfPrmAESrciIxpg0X40KPMbp1ZWVbd4=
-----END CERTIFICATE-----
[root@localhost certs]#
```

4. Import to the location `nss-config-dr /etc/certs` by using following commands:

```
[root@localhost certs]# certutil -A -n "DigiCert SHA2 Secure Server CA" -
t "TC,," -d /etc/certs -i /etc/certs/ocismtp-phoenix1.pem
[root@localhost certs]#
[root@localhost certs]# certutil -A -n "DigiCert SHA2 Secure Server CA
smtp " -t "TC,," -d /etc/certs -i /etc/certs/ocismtp-phoenix2.pem
[root@localhost certs]#
[root@localhost certs]# certutil -A -n "DigiCert SHA2 Secure Server CA
smtp2 " -t "TC,," -d /etc/certs -i /etc/certs/ocismtp-phoenix3.pem
```

5. To check whether the imports are done correctly, execute the command `certutil -L -d /etc/certs`

```
[root@localhost certs]# certutil -L -d /etc/certs

Certificate Nickname Trust Attributes
SSL,S/MIME,JAR/XPIDigiCert SHA2 Secure Server CA CT,,

DigiCert SHA2 Secure Server CA smtp CT,,
DigiCert SHA2 Secure Server CA smtp2 CT,,
```

Configuring PostFix for Relaying Host with Authentication

- Make sure the latese version of Postfix is installed along with `cyrus-sasl-*` packages.

```
[root@localhost ~]# rpm -qa | grep -i postfix
postfix-2.6.6-8.el6.x86_64
[root@localhost ~]# yum install postfix
Loaded plugins: security, ulninfo
Setting up Install Process
Package 2:postfix-2.6.6-8.el6.x86_64 already installed and latest version
Nothing to do
[root@localhost ~]#
[root@localhost ~]#yum install -y cyrus-sasl-*
```

> **Note:**
>
> All the available SASL mechanisms can be installed on the system by pulling in the relevant `cyrus-sasl-*` packages.

- Add the following config directives in the file `/etc/postfix/main.cf`:

```
#OCI SMTP Relay Host:
#relayhost = <Replace with your OCI SMTP server>
relayhost = smtp.us-phoenix-1.oraclecloud.com:587
#SASL Authentication settings:
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options =
#TSL Settings:
smtp_tls_loglevel = 2
smtp_use_tls = yes
smtpd_tls_security_level = may
smtp_tls_CApath = /etc/certs
```

- Create the file `/etc/postfix/sasl_passwd` to store the credentials created in the Generating a SMTP Credential step and make sure permissions are set to 600.

```
#vi /etc/postfix/sasl_passwd
relay_host:587 username:password
Example:
[root@localhost postfix]# cat /etc/postfix/sasl_passwd
smtp.us-phoenix-1.oraclecloud.com:587
ocid1.user.oc1..aaaaaaaajjcwynf4ebqp32wdpdy6h4lpeknqiyld7s35t2psfmmf
w3y4iosq@ocid1.tenancy.oc1..aaaaaaaavcpbui4wu2ttfnipykravgudbooie2eu
cf3odrsltgwj236epvha.fa.com:pP)QB&[YIz2ehe>7}fj_
[root@localhost postfix]#


[root@localhost postfix]# chmod 600 /etc/postfix/sasl_passwd
[root@localhost postfix]#
```

- Create `sasl_passwd.db` that Postfix can read:

```
[root@localhost postfix]# postmap /etc/postfix/sasl_passwd
[root@localhost postfix]#
[root@localhost postfix]# ls -l | grep -i passwd
-rw-------. 1 root root 224 Jan 31 18:17 sasl_passwd
-rw-------. 1 root root 12288 Jan 31 18:21 sasl_passwd.db
[root@localhost postfix]#
```

Starting Postfix

```
[root@localhost postfix]# chkconfig postfix on
[root@localhost postfix]# service postfix start
[root@localhost postfix]# service postfix status
master (pid 12162) is running...
```

```
[root@localhost postfix]#

If you are running Oracle Linux 7 run
#systemctl start --now postfix
```

Configuring Firewall Ports

Add these ports to firewall list of the smtp client machines (VM from where we have to send emails)

```
sudo firewall-cmd --zone=public --permanent --add-port=25/tcp
sudo firewall-cmd --zone=public --permanent --add-port=587/tcp
sudo firewall-cmd --reload
```

Beginning to Send Email

- Send Email

  ```
  approval is : user@<instancename.localdomain> e.g. opc@oracle-odi-
  inst-31up.localdomain
  In this case, login as user and test it with mailx
  [user@localhost ~]$ echo "test" | mailx -v -s "OCI Test Message [mailx]"
  user@oracle.com
  Mail Delivery Status Report will be mailed to <user>.
  [user@localhost ~]
  ```

- Verify `/var/log/maillog` for any error messages:

  ```
  Jan 31 18:24:36 localhost postfix/pickup[13812]: ECF9BA00B4: uid=501
  from=<user>
  Jan 31 18:24:36 localhost postfix/cleanup[14692]: ECF9BA00B4: message-
  id=<20190131182436.ECF9BA00B4@localhost.sub12182009561.cnvmau.oraclevcn.co
  m>
  Jan 31 18:24:36 localhost postfix/qmgr[12172]: ECF9BA00B4:
  from=<user@localhost.sub12182009561.cnvmau.oraclevcn.com>, size=549,
  nrcpt=1 (queue active)
  Jan 31 18:24:36 localhost postfix/smtp[14694]: initializing the client-
  side TLS engine
  Jan 31 18:24:37 localhost postfix/smtp[14694]: setting up TLS connection
  to smtp.us-phoenix-1.oraclecloud.com[Public IP]:587
  Jan 31 18:24:37 localhost postfix/smtp[14694]: smtp.us-
  phoenix-1.oraclecloud.com[Public IP]:587: TLS cipher list
  "ALL:+RC4:@STRENGTH"
  Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:before/connect
  initialization
  Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv2/v3 write
  client hello A
  Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3 read
  server hello A
  Jan 31 18:24:37 localhost postfix/smtp[14694]: smtp.us-
  phoenix-1.oraclecloud.com [Public IP]:587: certificate verification
  depth=2 verify=1 subject=/C=US/O=DigiCert Inc/OU=www.digicert.com/
  CN=DigiCert Global Root CA
  Jan 31 18:24:37 localhost postfix/smtp[14694]: smtp.us-
  ```

```
phoenix-1.oraclecloud.com[Public IP]:587: certificate verification
depth=1 verify=1 subject=/C=US/O=DigiCert Inc/CN=DigiCert SHA2
Secure Server CA
Jan 31 18:24:37 localhost postfix/smtp[14694]: smtp.us-
phoenix-1.oraclecloud.com [Public IP]:587: certificate verification
depth=0 verify=1 subject=/C=US/ST=California/L=Redwood City/
O=Oracle Corporation/OU=Oracle DYN-DEV US/CN=smtp.us-
phoenix-1.oraclecloud.com
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
read server certificate A
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
read server key exchange A
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
read server done A
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
write client key exchange A
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
write change cipher spec A
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
write finished A
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
flush data
Jan 31 18:24:37 localhost postfix/smtp[14694]: SSL_connect:SSLv3
read finished A
Jan 31 18:24:37 localhost postfix/smtp[14694]: Trusted TLS
connection established to smtp.us-phoenix-1.oraclecloud.com[public
ip]:587: TLSv1.2 with cipher DHE-RSA-AES256-SHA256 (256/256 bits)
Jan 31 18:24:38 localhost postfix/smtp[14694]: ECF9BA00B4:
to=<user@oracle.com>, relay=smtp.us-
phoenix-1.oraclecloud.com[public ip]:587, delay=1.6,
delays=0.02/0.03/0.57/1, dsn=2.0.0, status=sent (250 Ok)
Jan 31 18:24:38 localhost postfix/cleanup[14692]: 94136A00B8:
message-
id=<20190131182438.94136A00B8@localhost.sub12182009561.cnvmau.oracle
vcn.com>
Jan 31 18:24:38 localhost postfix/bounce[14696]: ECF9BA00B4: sender
delivery status notification: 94136A00
```

- The email has been delivered correctly:

```
-------- Forwarded Message --------
Subject: OCI Test Message [mailx]
Date: Thu, 31 Jan 2019 18:24:36 +0000
From: user@localhost.sub12182009561.cnvmau.oraclevcn.com
To: user@oracle.com
```

3

Configuring ODI Marketplace Repositories on DBCS Instance

This chapter helps you to create a new DBCS instance and set up ODI master and work repositories for the newly created DBCS instance on Oracle Cloud Marketplace.

Perform the following steps to create a DBCS instance and set up ODI master and work repositories for the created DBCS instance:

Creating a DBCS instance on Oracle Cloud Marketplace

Follow the below steps to create a DBCS instance on Oracle Cloud Marketplace:

1. From the OCI console, navigate to the left menu, click Database and select Bare Metal, VM and Exadata.
2. Select the required compartment and click **Create DB System**. DB System Information screen appears. It allows you to configure all the required details for creating a DB instance, such as compartment, Database name, username and password. All the default values are auto-populated.
3. In the **Add SSH keys** section, select the option **Paste SSh key** and provide the genereated SSH key in the SSH key text box.
4. In the **Specify the network information** section, click the **Virtual Cloud Network in ODI** drop-down arrow to select the required virtul cloud network.
5. Provide the required details in **Hostname Prefix**, **Host domain name** and **Host domain URL** text boxes. Click **Next**.
6. **Create Administrator Credentials** screen appears allowing you to configure the password for the default admin user.

> **Note:**
>
> Admin password that you provide should be 9 to 30 characters in length, contain at least two alphabets in upper case, two alphabets in lower case, two special characters (which includes _, # or -) and two numeric characters.

7. After configuring all the details, click **Create DB system**.

A new DBCS instance is created.

Setting up ODI on DBCS

For setting up ODI master and work repositories on a DBCS instance, you need to install ODI schemas using RCU. After installing the schemas you can set up ODI on DBCS by providing the DBaaS connection credentials.

For more information on creating schemas using RCU, refer to Creating Master and Work Repositories.

> **Note:**
>
> If you wish to use the newly created DBCS repository for your default ODI Marketplace Agent, refer to Switching Repositories of the ODI App Server.

Configuring the Domain for a Standalone Agent

Create and configure a standalone domain for a standalone agent using the Configuration Wizard. For more information on configuring ODI Domain `($ODI_DOMAIN_HOME)` with Standalone/JEE Agent template, refer to Configuring the Domain for a Standalone Agent chapter of Installing and Configuring Oracle Data Integrator guide.

After creating the domain based on the type of agent deployed, start the agent (standalone agent) or Admin Server and Managed server (JEE agent) from the location `$ODI_DOMAIN_HOME/bin`.

> **Note:**
>
> For more details on creating and configuring a JEE agent, refer to Configuring the Domain for a Java EE Agent chapter of *Installing and Configuring Oracle Data Integrator* guide.

4

Configuring Oracle Enterprise Manager for Oracle Fusion Middleware on Oracle Cloud Marketplace

This chapter helps you to access Oracle Enterprise Manager Fusion Middleware Control Console on Oracle Cloud Marketplace, and display WebCenter-related pages from where you can perform all necessary configuration, monitoring, and management tasks.

Prerequisites

Ensure to configure the following prerequisites before configuring Oracle Enterprise Manager for Oracle Fusion Middleware on Oracle Cloud Marketplace:

1. STB and other required schema as created by Repository Creation Utility (RCU).

> **Note:**
>
> Repository Creation Utility (RCU) schemas are created by default in Oracle Cloud Marketplace and the default STB schema name is in the format `<schema_name>_STB`. Check with support team to know the exact schema name. You can also find the schema name (for example - `ADW11_STB`) in `odi-setup.properties` file along with the following properties:
>
> - `sMasterDBDriver`
> - `sMasterDBUsername`
> - `odiSupervisorUser`
> - `sMasterDBUrl`

2. Physical agent in ODI studio with agent name as `OracleDIAgent` pointing to `host name` and `port` where you plan to configure `OracleDIAgent`.

> **Note:**
>
> This is a Java EE Agent and when you create it in ODI studio it works only after completing all the configurations in the configuration wizard and starting admin and manager servers.

Configuring Oracle Enterprise Manager

Following steps help you to configure Oracle Enterprise Manager for Oracle Fusion Middleware on Oracle Cloud Marketplace:

1. Navigate to `/u01/oracle/mwh/oracle_common/common/bin` and execute the file `config.sh` using the command `/u01/oracle/mwh/oracle_common/common/bin>./config.sh`.
   Oracle WebLogic Server Configuration wizard appears.
2. From the **Create Domain** screen, select **Create a new domain** option and click **Browse** to select the desired domain location. The selected location appears in the **Domain Location** text box. Click **Next**.
3. From the **Templates** screen, select **Create Domain Using Product Templates** option. **Available Templates** appear. Select Oracle Data Integrator - Console, Agent and Enterprise Manager Plugin. All the dependent templates are selected automatically. Click **Next**.
4. From the **Application Location** screen, click **Browse** to select the desired location for the application. The selected location appears in **Application Location** text box. Click **Next**.
5. From the **Administrator Account** screen, provide the admin server login credentials in the following fields:
   - Name
   - Password
   - Confirm Password and click **Next**.
6. From the **Domain Mode and JDK** screen, for **JDK**, select **Oracle HotSpot 1.8.0_191/u01/oracle/mwh/jdk1.8.0_191** option and click **Next**.
7. From the **Database Configuration Type** screen, for **Specify AutoConfiguration Options Using** parameter, select **RCU Data** option and fill-in the required details for the following fields:
   - Connection URL
   - Schema Owner
   - Schema Password

   Click **Connection Properties**. The **Connection Properties** screen displays the following details:

```
oracle.net.authentication_service TCPS
oracle.net.ssl_server_dn_match false
javax.net.ssl.trustStore /u01/oracle/mwh/wallet/cwallet.sso (Path
of the wallet location)
javax.net.ssl.keyStoreType SSO
javax.net.ssl.keyStore /u01/oracle/mwh/wallet/cwallet.sso(Path of
the wallet location)
javax.net.ssl.trustStoreType SSO
```

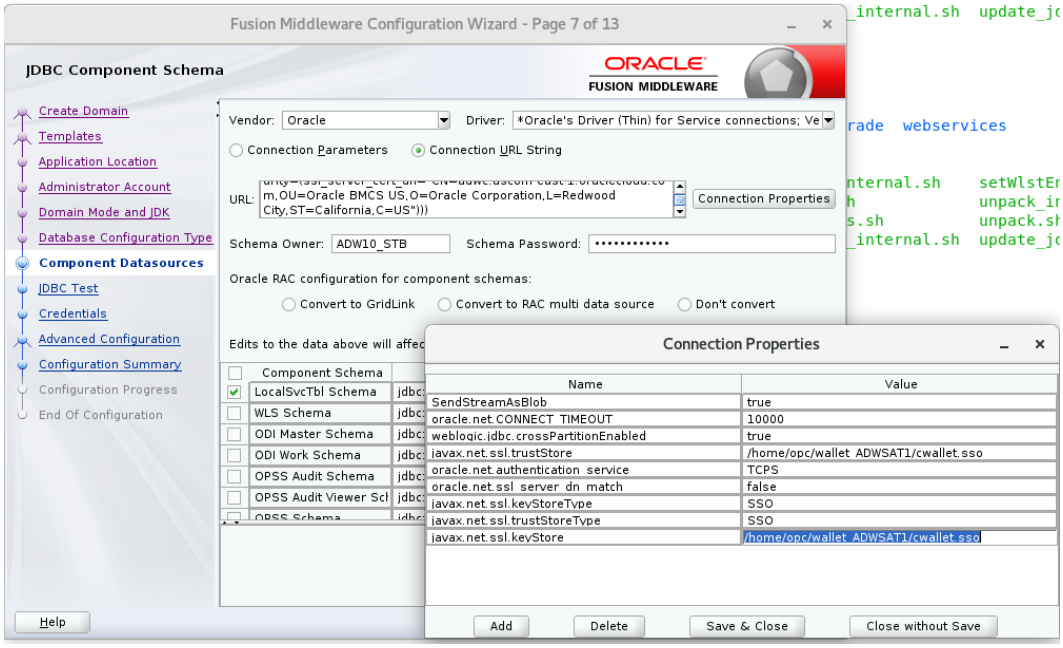
> **Note:**
>
> Make sure to provide the path for truststore and keystore parameters same as the path of your wallet file.

Click **Save & Close**. Click **RCU Configuration** and Click **Next**.

8. Provide the essential details in **Component Datasources** screen.

> **Note:**
>
> Make sure to change the connection string, connection properties for each component schema, as they are not carried forward from the previous screens.

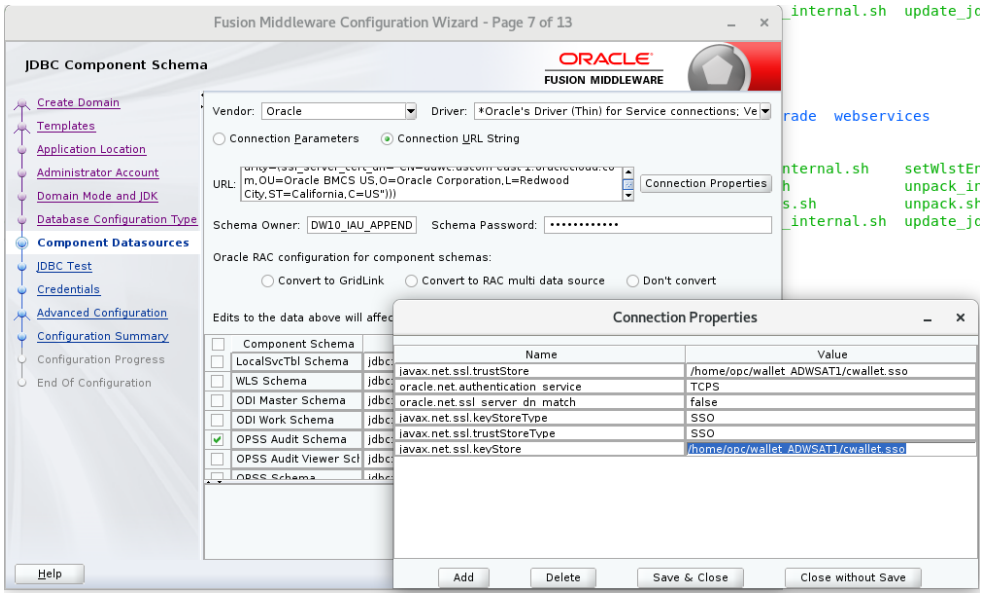
Figure 4-1 Sample for Component Datasources



This image is a sample with changes for component `LocalSvcTblSchema` and you have to make changes for the following:

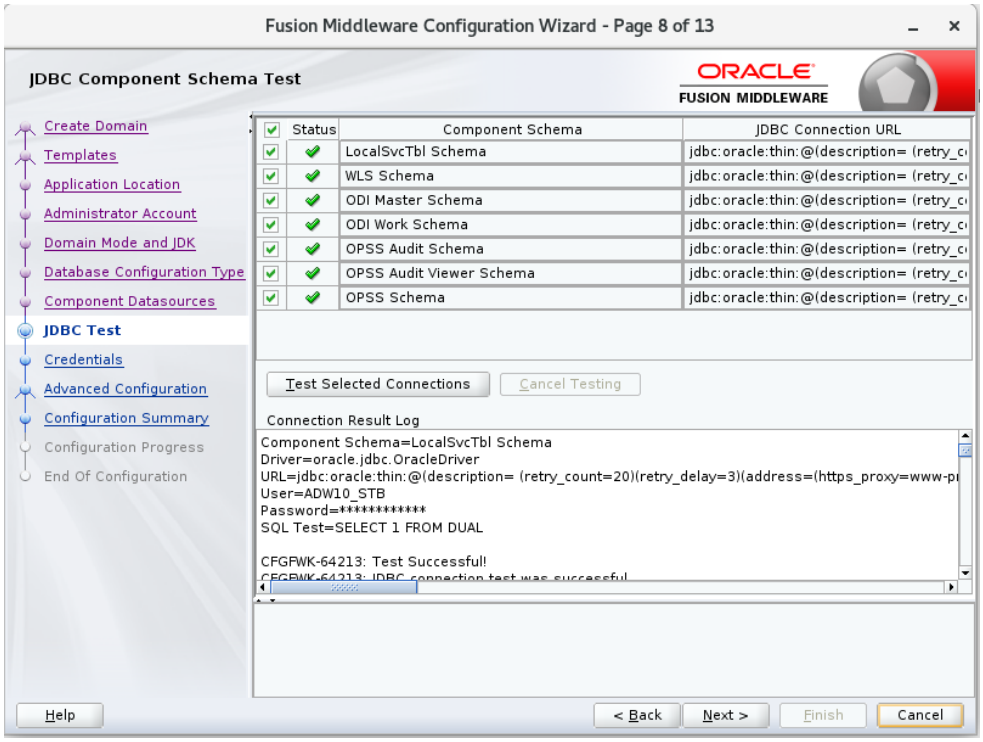
- WLS Schema
- ODI Master Schema
- ODI Work Schema
- OPSS Audit Schema
- OPSS Audit Viewer Schema
- OPSS Schema

Figure 4-2 Sample for OPSS Audit Schema



9. After providing all the above details, **JDBC Test** screen appears. Review all the details and click **Next**.

Figure 4-3 Sample JDB Test Configuration



10. From the **Credentials** screen, enter the Supervisor password for your ODI instance and click **Next**.
11. From the **Advanced Configuration** screen, select the check boxes for the following servers/services:

- Administration Server
- Node Manager
- Topology
- Deployments and Services and click **Next**.

12. The **Administration Server** screen displays the port number in which the admin server listens. It is auto-populated by default. Click **Next**.
13. From the **Node Manager** screen, provide the node manager login credentials in **Username**, **Password** and **Confirm Password** fields and click **Next**. Node manager login credentials can be same as Admin server login credentials, which means you can use the same username and password for logging into both Node manager and Admin server.
14. From the **Managed Servers** screen, provide the server name and port details in **Server Name** and **Listen Port** columns respectively.
15. From the **Clusters** screen, click **Add**, to create a new cluster. Click **Next**.
16. The **Server Templates** screen displays all the available server templates. Click **Add**, to add new server templates, if required, else, click **Next**. Assign the required server template to the newly created Cluster.
17. The **Coherence Cluster** screen displays the available cluster name and port number. Click **Next**.
18. In the **Machines** screen, click **Add**, to add a machine and provide the name, port and address of the Node Manager. After adding a machine, add `Admin server` and `ODI_server` to the created machine. Click **Next**.
19. The **Virtual Targets** screen displays the details of configured virtual targets, if any. It includes details such as name, target, host names, URI prefix, explicit port and port offset. Click **Next**.
20. The **Partitions** screen displays the details of all the available partitions. Click **Next**.
21. The **Deployments** screen displays a list of all the **Deployments** and **Deployment Targets**. Click **Next**.
22. The **Service Targets** screen displays a list of all the **Services** and **Deployment Targets**. Click **Next**.
23. The **Configuration Summary** screen displays a summary of all the performed configurations. Click **Create**. **Configuration Process** screen appears displaying the status of the process and configurations.
24. Upon completion, **End of Configuration** screen appears, displaying the **Configuration Succeeded** message along with the configured **Domain Location** and **Admin Server URL** details. Click **Finish**.

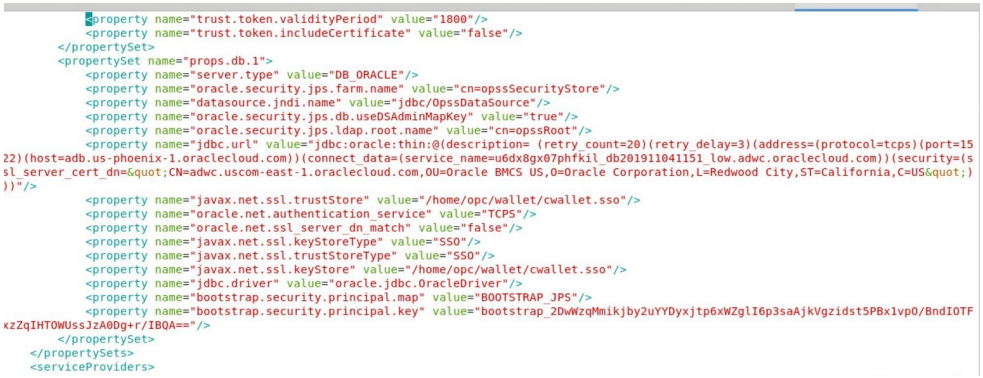
Other Configurations

1. Locate `jps-config.xml` file and configure the following properties in your `jps-config.xml` file. By default, you can find this file in the following location - `~/oracle/user_projects/domains/base_domain/config/fmwconfig` and this location is based on the created base domain.

```
<property name="javax.net.ssl.trustStore" value="/u01/oracle/mwh/
wallet_ADWSAT1/cwallet.sso"/>
            <property name="oracle.net.authentication_service"
```

```
value="TCPS"/>
            <property name="oracle.net.ssl_server_dn_match"
value="false"/>
            <property name="javax.net.ssl.keyStoreType"
value="SSO"/>
            <property name="javax.net.ssl.trustStoreType"
value="SSO"/>
            <property name="javax.net.ssl.keyStore" value="/u01/
oracle/mwh/wallet_ADWSAT1/cwallet.sso"/>
```

Figure 4-4 JPS Configuration File



2. In the command prompt, navigate to Node manager properties file `/u01/oracle/mwh/user_projects/domains/base_domain/nodemanager/nodemanger.properties` and edit the property `SecureListener= false` and make sure the listener port is matching with the configured port. Save the file and navigate to the domain creation path and start the node manager using the following command :

   ```
   /u01/oracle/mwh/user_projects/domMains/base_domain/bin>./
   startNodeManager.sh
   ```

3. In the command prompt, navigate to the domain creation path (configured in the above steps) and perform the following:

   - Start the Admin server using the following command:

     ```
     /u01/oracle/mwh/user_projects/domains/base_domain/bin> ./
     startWebLogic.sh
     ```

   - Start the Managed server using the following command:

     ```
     /u01/oracle/mwh/user_projects/domains/base_domain/bin> ./
     startManagedWebLogic.sh ODI_server1 http://localhost:7001
     ```

     The command `/startManagedWebLogic.sh` helps you to start the managed server which is required to start the ODI Console and Oracle DIAgent.

After performing all the above configurations, navigate to a web browser and access this URL `http://<<hostname>>:<<port>>/em`.

For example,

```
http://localhost:7001/em/
```

Provide your login credentials and click **Sign In**, to access the newly configured Oracle Enterprise Manager Fusion Middleware Control Console on Oracle Cloud Marketplace.

After logging in, you can check the health of the servers. From the Oracle Enterprise Manager Fusion Middleware Control Console you can start and stop the servers and agents. You can also check if, the console is working by logging into the console `http://localhost:managedserverport/odiconsole`.

5

Configuring the Domain for Collocated/Java EE Agent on Oracle Cloud Marketplace

This chapter helps you to create and configure a Oracle Data Integrator (ODI) domain for the Collocated/Java EE agent on Oracle Cloud Marketplace.

Configuring Collocated/Java EE agent on Oracle Cloud Marketplace includes the following high level steps:

- Navigate to the location (`ORACLE_HOME/oracle_common/common/bin`) and execute the script `config.sh` to launch the WebLogic Server Configuration Wizard.
- Select the ODI Collocated / JEE agent template and deploy.
- After succesful deployment, start the Collocated / JEE Agent, Weblogic Server and Managed Server.

> **Note:**
>
> The following steps are applicable only for Oracle Cloud Marketplace instances using Autonomous Database (ADW or ATP) for the ODI Repository. Collocated/ J2EE Agents are not supported using the Embedded MySQL Repository.

Prerequisites

Ensure to configure the following prerequisites before configuring Collocated/Java EE agent on Oracle Cloud Marketplace:

1. STB and other required schema as created by Repository Creation Utility (RCU).

   > **Note:**
   >
   > Repository Creation Utility (RCU) schemas are created by default in Oracle Cloud Marketplace and the default STB schema name is in the format `<schema_prefix>_STB`. Check with support team to know the exact schema name. You can also find the schema name (for example - `ADW11_STB`) in `odi-setup.properties` file along with the following properties:
   >
   > - `sMasterDBDriver`
   > - `sMasterDBUsername`
   > - `odiSupervisorUser`
   > - `sMasterDBUrl`

2. Physical agent in ODI studio with agent name as `OracleDIAgent` pointing to `host name` and `port` where you plan to configure `OracleDIAgent`.

> **Note:**
>
> This is a Java EE Agent and when you create it in ODI studio it works only after completing all the configurations in the configuration wizard and starting admin and manager servers.

Configuring the Domain for Collocated/Java EE Agent

Following steps help you to configure the domain for the Collocated/Java EE agent on Oracle Cloud Marketplace:

1. Navigate to `/u01/oracle/mwh/oracle_common/common/bin` and execute the file `config.sh` using the command `/u01/oracle/mwh/oracle_common/common/bin>./config.sh`.
   Oracle WebLogic Server Configuration wizard appears.
2. From the **Create Domain** screen, select **Create a new domain** option and click **Browse** to select the desired domain location. The selected location appears in the **Domain Location** text box. Click **Next**.
3. From the **Templates** screen, select **Create Domain Using Product Templates** option. **Available Templates** appear. Select Oracle Data Integrator - Console, Agent and J2EE Plugin. All the dependent templates are selected automatically. Click **Next**.
4. From the **Application Location** screen, click **Browse** to select the desired location for the application. The selected location appears in **Application Location** text box. Click **Next**.
5. From the **Administrator Account** screen, provide the admin server login credentials in the following fields:
   - Name
   - Password
   - Confirm Password and click **Next**.
6. From the **Domain Mode and JDK** screen, for **JDK**, select **Oracle HotSpot 1.8.0_191/u01/oracle/jdk1.8.0_191** option and click **Next**.
7. From the **Database Configuration Type** screen, for **Specify AutoConfiguration Options Using** parameter, select **RCU Data** option and fill-in the required details for the following fields:
   - Connection URL
   - Schema Owner
   - Schema Password

   Click **Connection Properties**. The **Connection Properties** screen displays the following details:

   ```
   oracle.net.authentication_service TCPS
   oracle.net.ssl_server_dn_match false
   <Path to wallet location for ODI repository>)
   javax.net.ssl.keyStoreType SSO
   ```

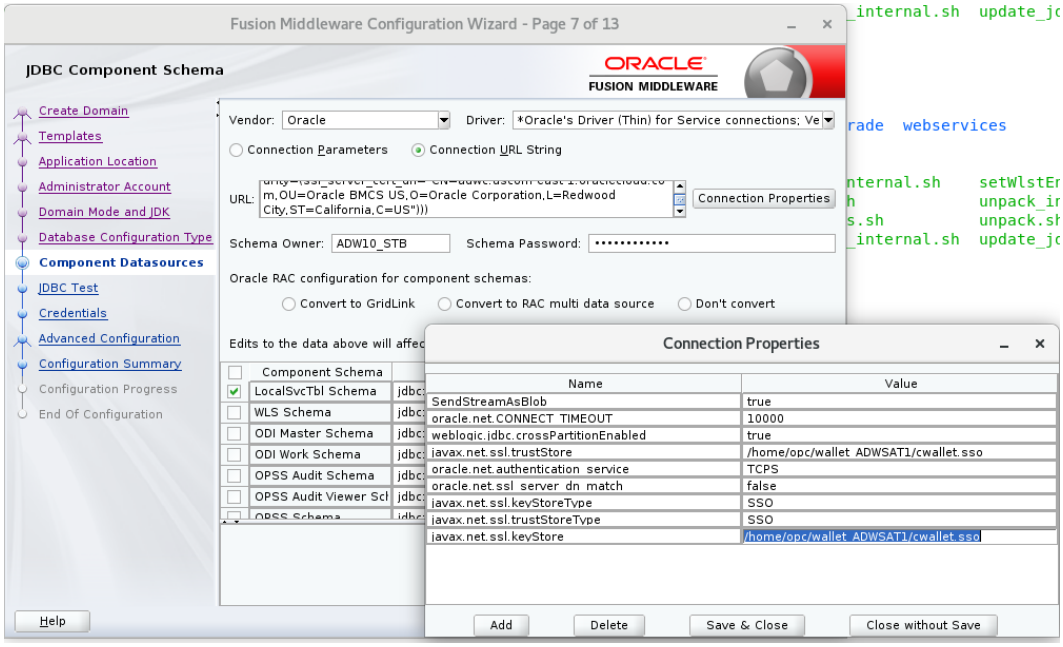
> **Note:**
>
> Make sure to provide the path for truststore and keystore parameters same as the path of your wallet file.

Click **Save & Close**. Click **RCU Configuration** and Click **Next**.

8. Provide the essential details in **Component Datasources** screen.

> **Note:**
>
> Make sure to change the connection string, connection properties for each component schema, as they are not carried forward from the previous screens.

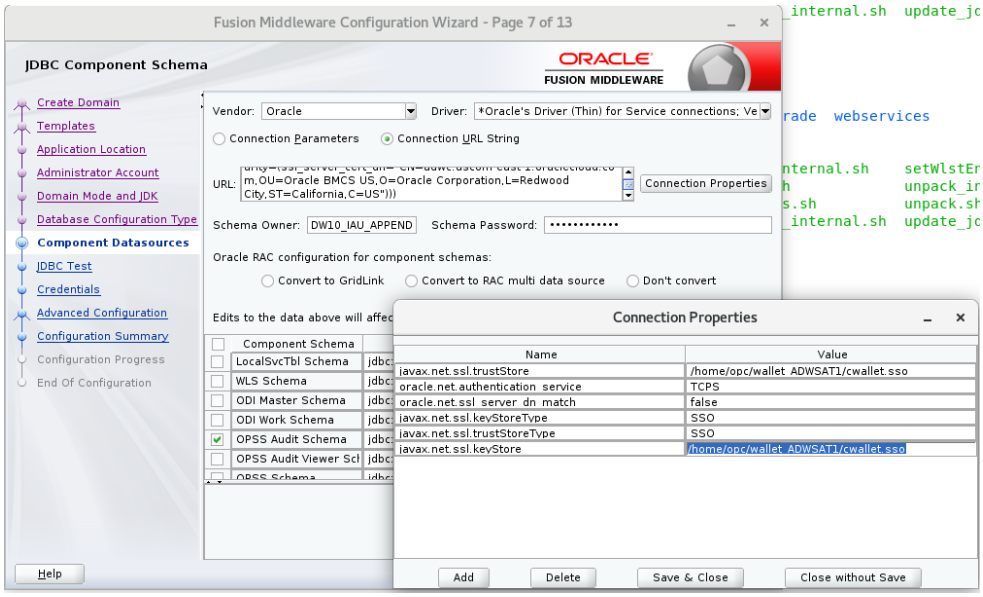
Figure 5-1 Sample for Component Datasources



This image is a sample with changes for component `LocalSvcTblSchema` and you have to make changes for the following:

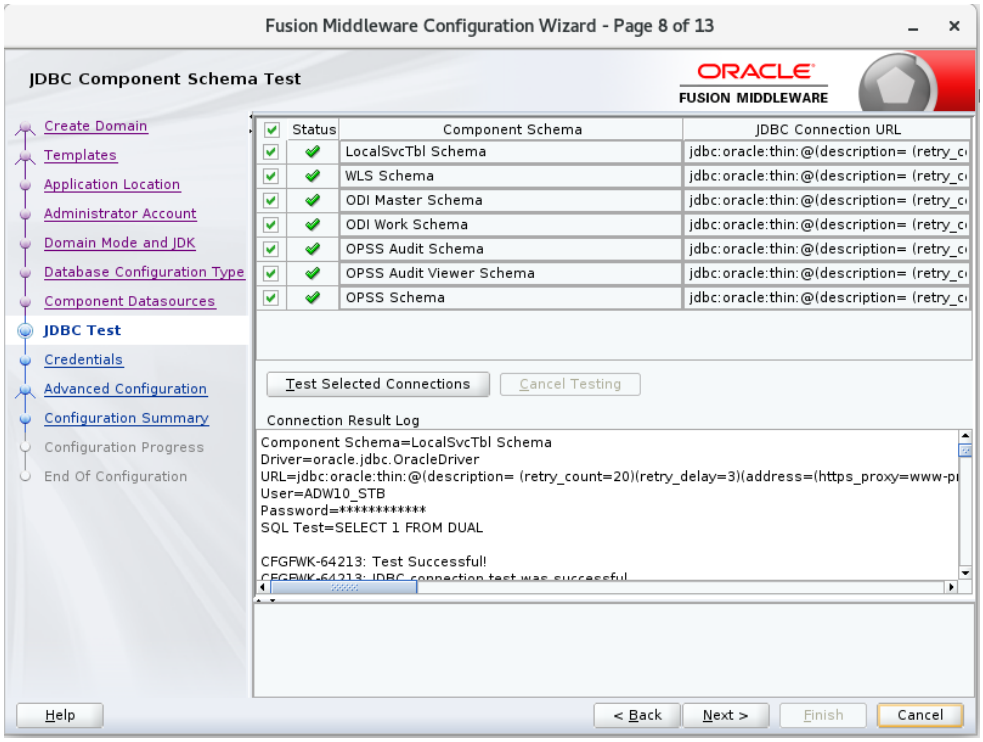
- WLS Schema
- ODI Master Schema
- ODI Work Schema
- OPSS Audit Schema
- OPSS Audit Viewer Schema
- OPSS Schema

Figure 5-2 Sample for OPSS Audit Schema



9. After providing all the above details, **JDBC Test** screen appears. Review all the details and click **Next**.

Figure 5-3 Sample JDB Test Configuration



10. From the **Credentials** screen, enter the Supervisor password for your ODI instance and click **Next**.
11. From the **Advanced Configuration** screen, select the check boxes for the following servers/services:

- Administration Server
- Node Manager
- System Components
- Topology
- Deployments and Services and click **Next**.

12. The **Administration Server** screen displays the port number in which the admin server listens. It is auto-populated by default. Click **Next**.
13. From the **Node Manager** screen, provide the node manager login credentials in **Username**, **Password** and **Confirm Password** fields and click **Next**. Node manager login credentials can be same as Admin server login credentials, which means you can use the same username and password for logging into both Node manager and Admin server.
14. From the **Managed Servers** screen, provide the server name and port details in **Server Name** and **Listen Port** columns respectively.
15. From the **Clusters** screen, click **Add**, to create a new cluster. Click **Next**.
16. The **Server Templates** screen displays all the available server templates. Click **Add**, to add new server templates, if required, else, click **Next**. Assign the required server template to the newly created Cluster.
17. The **Coherence Cluster** screen displays the available cluster name and port number. Click **Next**.
18. In the **Machines** screen, click **Add**, to add a machine and provide the name, port and address of the Node Manager. After adding a machine, add `Admin server` and `ODI_server` to the created machine. Click **Next**.
19. The **Virtual Targets** screen displays the details of configured virtual targets, if any. It includes details such as name, target, host names, URI prefix, explicit port and port offset. Click **Next**.
20. The **Partitions** screen displays the details of all the available partitions. Click **Next**.
21. The **Deployments** screen displays a list of all the **Deployments** and **Deployment Targets**. Click **Next**.
22. The **Service Targets** screen displays a list of all the **Services** and **Deployment Targets**. Click **Next**.
23. The **Configuration Summary** screen displays a summary of all the performed configurations. Click **Create**. **Configuration Process** screen appears displaying the status of the process and configurations.
24. Upon completion, **End of Configuration** screen appears, displaying the **Configuration Succeeded** message along with the configured **Domain Location** and **Admin Server URL** details. Click **Finish**.

Other Configurations

1. Locate `jps-config.xml` file and configure the following properties in your `jps-config.xml` file. By default, you can find this file in the following location - `~/oracle/user_projects/domains/base_domain/config/fmwconfig` and this location is based on the created base domain.

```
<property name="javax.net.ssl.trustStore" value="/u01/oracle/mwh/
wallet_ADWSAT1/cwallet.sso"/>
```

```
            <property name="oracle.net.authentication_service"
value="TCPS"/>
            <property name="oracle.net.ssl_server_dn_match"
value="false"/>
            <property name="javax.net.ssl.keyStoreType"
value="SSO"/>
            <property name="javax.net.ssl.trustStoreType"
value="SSO"/>
            <property name="javax.net.ssl.keyStore" value="/u01/
oracle/mwh/wallet_ADWSAT1/cwallet.sso"/>
```

Figure 5-4 JPS Configuration File

2. In the command prompt, navigate to Node manager properties file `/u01/oracle/mwh/user_projects/domains/base_domain/nodemanager/nodemanger.properties` and edit the property `SecureListener= false` and make sure the listener port is matching with the configured port. Save the file and navigate to the domain creation path and start the node manager using the following command :

   ```
   /u01/oracle/mwh/user_projects/domMains/base_domain/bin>./
   startNodeManager.sh
   ```

3. In the command prompt, navigate to the domain creation path (configured in the above steps) and perform the following:

   - Start the Admin server using the following command:

     ```
     /u01/oracle/mwh/user_projects/domains/base_domain/bin> ./
     startWebLogic.sh
     ```

   - Start the Managed server using the following command:

     ```
     /u01/oracle/mwh/user_projects/domains/base_domain/bin> ./
     startManagedWebLogic.sh ODI_server1 http://localhost:7001
     ```

     The command `/startManagedWebLogic.sh` helps you to start the managed server which is required to start the ODI Console and Oracle DIAgent.

Provide your login credentials and click **Sign In**, to access the newly configured domain for the Collocated/Java EE agent domain for the Collocated/Java EE agent on Oracle Cloud Marketplace.

After logging in, you can check the health of the servers. From the OCI Console you can start and stop the servers and agents. You can also check if, the console is working by logging into the console `http://localhost:managedserverport/odiconsole`.

6

Configuring High Availability for ODI on Oracle Cloud Marketplace

This chapter helps you to configure High Availability (HA) topology for Oracle Data Integrator on Oracle Cloud Marketplace. The sections in this chapter outline the concepts and steps that are important for designing high availability deployment.

It contains the following sections:

6.1 Prerequisites for setting up 2-Node Cluster for High Availability

Go through the following prerequisites before setting up 2-Node Cluster for High Availability. Make sure you have the following before setting up 2-Node Cluster for High Availability:

1. OCI Virtual Cloud Network (VCN) setup that supports communication with all the compute instances created in its subnet.
   - All the communication channels are through private IPs.
   - External communication established outside the subnet are through public IPs.
   - Ably configured for Ingress/Egress. For more information, see Enabling Incoming Ports and Services.
2. Autonomous Transaction Processing instance having the following configuration so as to leverage auto scaling and access to the `DBMS_CLOUD` package:
   - Workload type: **Transaction Processing**
   - Deployment type: **Shared Infrastructure**
   - Network Access: **Allow secure access from everywhere**
3. ADB or DBaaS instance created in the same subnet and VCN as described in Step 1.
4. ODI compute instance 1 created in the same subnet and VCN as described in Step 1.
5. ODI compute instance 2 created in the same subnet and VCN as described in Step 1 but with a different availability domain.
6. Firewall configurations in all the compute instances that are part of the cluster. For more information, see Firewall Rules.

6.2 Creating and configuring the ODI Domain

This section contains the following topics:

6.2.1 Creating a domain on Node 1

Follow the below steps to create the domain on Node 1:

1. Navigate to the `cd /u01/oracle/mwh/oracle_common/common/bin` directory.
2. Execute `config.sh` to start the Configuration Wizard.
3. In the first screen, click **Next**.
4. In the Configuration Type screen, specify "`/u01/oracle/mwh/user_projects/domains/odi_domain`" in the Domain Location field. Click **Next** to continue.
5. In the Templates screen, select **Oracle Data Integrator – Agent [odi]** (which will auto-select **Oracle Data Integrator – Agent Libraries [odi]** and **Oracle Data Integrator SDK Shared Libraries Template [odi]**), **Oracle Data Integrator – Console [odi]** and **Oracle Enterprise Manager Plugin for ODI [em]** (which will auto-select **Oracle Enterprise Manager [em]**) from the Available Templates field. Click **Next** to continue.
6. In the Application Location screen, click **Next**.
7. In the Administrator Account screen, provide the following information and click **Next**:
   a. **Name**: Specify `weblogic`.
   b. **Password**: Enter the password that you would like to assign to the weblogic user.
8. In the Domain Mode and JDK screen, select **Production** and click **Next**.
9. In the Database Configuration Type screen, provide the following information:
   a. **URL**: If your repository is in DBCS/Exa, specify "`jdbc:oracle:thin:@<HOST>:<PORT>/<SERVICE_NAME>`" as the URL. If your repository is in ADW/ATP, specify "`jdbc:oracle:thin:@(description=(retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)(host=<HOST>))(connect_data=(service_name=<SERVICE_NAME>))(security=(ssl_server_cert_dn="CN=adwc.uscom-east-1.oraclecloud.com,OU=Oracle BMCS US,O=Oracle Corporation,L=Redwood City,ST=California,C=US")))`" as the URL.

b. **Schema Owner**: Specify `<ODI_PREFIX>_STB`.

c. **Schema Password**: Enter the schema owner password.

10. If your repository is in DBCS/Exa, skip this step; if your repository is in ADW/ATP, click **Connection Properties** and verify that you have the following properties in place (in addition to what is already there):

```
oracle.net.authentication_service TCPS
oracle.net.ssl_server_dn_match false
javax.net.ssl.trustStore <PATH>/cwallet.sso
javax.net.ssl.keyStoreType SSO
javax.net.ssl.keyStore <PATH>/cwallet.sso
javax.net.ssl.trustStoreType SSO
```

where `<PATH>` is the path to the wallet directory

11. Click **Get RCU Configuration**. If the connection is successful, the Next button will be activated. Click **Next**.

12. In the JDBC Component Schema screen, if your ODI repository is in DBCS/Exa, skip to step 13. If it is in ADW/ATP, select **Local SvcTbl Schema**. Verify that all the connection information (that is, URL and connection properties) is correct.

13. Deselect **Local SvcTbl Schema** and select the following schemas:

- WLS Schema
- ODI Master Schema
- ODI Work Schema
- OPSS Audit Schema
- OPSS Audit Viewer
- OPSS Schema

Verify that all the connection information (that is, URL and connection properties) is correct.

14. When the connection information to all schemas has been verified, click **Next**.

15. In the JDBC Component Schema Test screen, verification of all schemas will happen. As they pass the verification, a green check mark will appear. When all the schemas show a check mark in the Status column, click **Next**.

16. In the Credentials screen, provide the following information and click **Next**:

a. **Username**: Specify `SUPERVISOR`.

b. **Password**: Specify the password for the `SUPERVISOR`.

17. In the Advanced Configuration screen, select **Administration Server**, **Node Manager**, **Topology** and **Deployments and Services**. Click **Next**.

18. In the Administration Server screen, provide the following information and click **Next**:

a. **Listen Address**: Specify the private IP address of Node 1.

b. **Listen Port**: Specify the port you want (7001/ 13001) for it.

19. In the Node Manager screen, provide the following information and click **Next**:

a. **Node Manager Type**: Select **Per Domain Default Location**.

b. **Username**: Specify `weblogic`.

c. **Password**: Enter the password for the `weblogic` user.

d. **Confirm Password**: Enter the password for the `weblogic` user.

20. In the Managed Servers screen, add the following information and click **Next**:

    a. Specify `ODI_server1` as the Server Name and the ODI Node 1 Private IP Address as the Listen Address. Assign the Listen Port (`8001/15101`) and make sure that the Server Groups is `(ODI-MGD …)`.

    b. Specify `ODI_server2` as the Server Name and the ODI Node 2 Private IP Address as the Listen Address. Assign the Listen Port (`8001/15101`) and make sure that the Server Groups is `(ODI-MGD …)`.

21. In the Clusters screen, click **Next**.

22. In the Server Templates screen, assign the port (`7100/13100`) for both templates in the Listen Port field. Click **Next**.

23. In the Dynamic Servers screen, click **Next**.

24. In the Assign Servers to Clusters screen, assign both servers to "`ODI_Cluster1`" and click **Next**.

25. In the Coherence Clusters screen, specify the port (`7574/13574`) in the Cluster Listen Port field. Click **Next**.

26. In the Machines screen, select the **Unix Machine** tab, then click **Add** to provide the following information:

    a. Specify `odi-node1` in the Name field and the private IP Address of ODI Node 1 in the Node Manager Listen Address field. Assign the port for the Node Manager (`5556/9556`).

    b. Specify `odi-node2` in the Name field and the private IP Address of ODI Node 2 in the Node Manager Listen Address field. Assign the port for the Node Manager (`5557/9557`). Click **Next**.

27. In the Assign Servers to Machines screen, make sure that “`AdminServer`” and “`ODI_server1`” are assigned to “`odi-node1`” and “`ODI_server2`” is assigned to “`odi-node2`”. Click **Next**.

28. In the Virtual Targets screen, click **Next**.

29. In the Partitions screen, click **Next**.

30. In the Deployments Targeting screen, click **Next**.

31. In the Services Targeting screen, click **Next**.

32. In the Configuration Summary screen, click **Create**.

33. In the Configuration Progress screen, a green check will appear as the tasks progress. This indicates that the task finished successfully. Once everything finishes, click **Next**.

34. In the End of Configuration screen, make a note of the first link as that is your domain path. On clicking the second link, a web browser will appear. Disregard the error message and copy the URL. Click **Finish**.

6.2.2 Setting up the Administration Server on Node 1

Follow the below steps to set up the administration server on Node 1:

1. If your repository is in DBCS/Exa, skip to step 4. If your repository is in ADW/ATP, navigate to the `/u01/oracle/mwh/user_projects/domains/odi_domain/config/fmwconfig` directory.
2. Make a backup copy of the `jps-config.xml` file as follows:

   ```
   cp jps-config.xml jps-config.xml.ORIG
   ```

3. Edit the file and add the following entries just after the “jdbc url” entry:

   ```
   <property name="javax.net.ssl.trustStore" value="/home/opc/.odi/oracledi/
   userlib/<PATH>/cwallet.sso"/> <property
   name="oracle.net.authentication_service" value="TCPS"/> <property
   name="oracle.net.ssl_server_dn_match" value="false"/> <property
   name="javax.net.ssl.keyStoreType" value="SSO"/> <property
   name="javax.net.ssl.trustStoreType" value="SSO"/> <property
   name="javax.net.ssl.keyStore" value="/home/opc/.odi/oracledi/userlib/<PATH>/
   cwallet.sso"/>
   ```

4. Navigate to the `/u01/oracle/mwh/user_projects/domains/odi_domain` directory.
5. Start the administration server using the following command:

   ```
   ./startWebLogic.sh
   ```

6. Enter the weblogic admin user and password. Wait for the server to start.
7. Using a web browser, log into Weblogic.
8. Click **Lock & Edit** available at the top left corner of the screen.
9. Using the left side panel, navigate to `Environments > Machines`.
10. Drill down on Node 1 (`odi-node1`).
11. On the right side of the screen, select **Configuration** and **Node Manager**.
12. In the Type field, select **Plain**.
13. Click **Save**.
14. Repeat steps 9 – 13 to do the same for Node 2 (`odi-node2`). That is, perform the following steps for Node 2:
    a. Using the left side panel, navigate to `Environments > Machines`.
    b. Drill down on Node 2 (`odi-node2`).
    c. On the right side of the screen, select **Configuration** and **Node Manager**.
    d. In the Type field, select **Plain**.
    e. Click **Save**.
15. Using the left side panel, navigate to `Environments > Servers`.
16. Drill down on Node 1 (`ODI_server1`).
17. On the right side of the screen, select **Configuration** and **SSL**.
18. Click **Advanced** available at the bottom of the screen.

19. In the Hostname Verification field, select **None**.
20. Click **Save**.
21. On the right side of the screen, select **Configuration** and **Server Start**.
22. In the Arguments field, provide the following information:

```
-Dtangosol.coherence.localport=8095
-Doracle.odi.coherence.wka1=<PRIVATE_IP_ADDRESS_NODE1>
-Doracle.odi.coherence.wka1.port=8095
-Doracle.odi.coherence.wka2=<PRIVATE_IP_ADDRESS_NODE2>
-Doracle.odi.coherence.wka2.port=8096
```

23. At the bottom of the screen, provide the following information:
    a. **User Name**: Specify `weblogic`.
    b. **Password**: Enter the password for the weblogic user.
    c. **Confirm Password**: Enter the password for the weblogic user.
24. Click **Save**.
25. Repeat steps 15 – 24 to do the same for Node 2. That is, perform the following steps for Node 2:
    a. Using the left side panel, navigate to `Environments > Servers`.
    b. Drill down on Node 2 (`ODI_server2`).
    c. On the right side of the screen, select **Configuration** and **SSL**.
    d. Click **Advanced** available at the bottom of the screen.
    e. In the Hostname Verification field, select **None**. Click **Save**.
    f. On the right side of the screen, select **Configuration** and **Server Start**.
    g. In the Arguments field, provide the following information:

```
-Dtangosol.coherence.localport=8096
-Doracle.odi.coherence.wka1=<PRIVATE_IP_ADDRESS_NODE1>
-Doracle.odi.coherence.wka1.port=8095
-Doracle.odi.coherence.wka2=<PRIVATE_IP_ADDRESS_NODE2>
-Doracle.odi.coherence.wka2.port=8096
```

    h. At the bottom of the screen, specify `weblogic` in the User Name field. Provide the password for the weblogic user in the Password and Confirm Password fields.
    i. Click **Save**.
26. Click **Activate Changes** available at the top left corner of your screen.
27. Using the left side panel, navigate to `Environments > Servers`; on the right side of the screen, select **Control** and then select **AdminServer** from the Server field.
28. Click **Shutdown** and select **Force shutdown now** from the dropdown menu.
29. To exit the session, click **Yes**. Leave the web browser open.
30. Start a new terminal and navigate to the `/u01/oracle/mwh/user_projects/domains/odi_domain/servers/AdminServer` directory.

31. Create a new directory as follows:

    ```
    mkdir security
    ```

32. Move into that directory using the following command:

    ```
    cd security
    ```

33. Edit a new file called “`boot.properties`” under `Admin/security` using the following command:

    ```
    vi boot.properties
    ```

34. Add the following content and save the file:

    ```
    username=weblogic
    password=<WEBLOGIC_PASSWORD>
    ```

35. Disable the secure listener. Navigate to the following directory:

    ```
    cd /u01/oracle/mwh/user_projects/domains/odi_domain/nodemanager
    ```

36. Edit the "`nodemanager.properties`" file using the following command:

    ```
    vi nodemanager.properties
    ```

37. Look for the entry “`Secure Listener=true`” and change it to “`Secure Listener=false`”. Save the file.

38. If ODI was provisioned before stack release 12.2.1.4.200618, skip to step 17. If it was provisioned with stack release 12.2.1.4.200618 or later, execute the following steps:

    a. Navigate to the following directory:

       ```
       cd /u01/oracle/mwh/wlserver/server/bin
       ```

    b. Copy the following file:

       ```
       cp startNodeManager.sh startNodeManger.sh.OLD
       ```

    c. Edit the following file:

       ```
       vi startNodeManager.sh
       ```

    d. Change line 52 from `WL_HOME="/home/opc/oracle/wlserver"` to `WL_HOME="${MW_HOME}/wlserver"` .

39. Return to your first terminal.

40. Start the Weblogic server again and send the process to the backend using the following command:

    ```
    nohup ./startWebLogic.sh &
    ```

41. Monitor the startup process using the following command:

    ```
    tail -100 nohup.out
    ```

42. Once the server is up and running, start the Node Manager. Move to the `bin` directory using the following command:

    ```
    cd bin
    ```

43. Execute “startNodeManager” and send the process to the background using the following command:

    ```
    nohup ./startNodeManager.sh &
    ```

44. Monitor the startup process using the following command:

    ```
    tail -100 nohup.out
    ```

45. Verify the Node Manager. Go back to the web browser and log into the WebLogic Server.
46. Using the left side panel, navigate to `Environments > Machines`.
47. Drill down on Node 1 (`odi-node1`).
48. On the right side of the screen, select **Monitoring** and then **Node Manager Status**. You can view the current status information for the Node Manager instance configured for the machine.

6.2.3 Setting up the ODI agent on Node 1

Follow the below steps to set up the ODI agent on Node 1:

1. Start ODI Studio.
2. Navigate to the **Topology** tab.
3. Define a new agent as follows:
    a. **Name**: Specify `OracleDIAgent` as the agent name.
    b. **Host**: Enter the private IP address of the ODI node (Node1).
    c. **Port**: Specify the port defined for the managed server "`ODI_server1`" (15101 / 8001).
4. Save the configuration.
5. Define the corresponding logical agent and save it.
6. In the web browser, using the left side panel of the Weblogic Server, navigate to `Environments > Servers`.
7. On the right side of the screen, select **Control** and then select **ODI_server1**.
8. Click **Start**.
9. In the Server Life Cycle Assistant screen, click **Yes**.
10. Keep refreshing the page until you see a "RUNNING" status for `ODI_server1`.
11. Verify the deployment. Navigate to the Deployments screen using the hierarchy pane on the left side of your screen.
12. Test the agent. Using the left side panel of the WebLogic Server, navigate to `Environments > Servers`.
13. Drill down on Server 1 (`ODI_server1`).
14. On the right side of the screen, select **Deployments**.
15. Drill down on the agent (`oraclediagent`).
16. On the right side of the screen, select **Testing**.
17. Expand the `oraclediagent` node.
18. Click the URL available in the Test Point field.
19. Save it for future reference.
20. Go back to ODI Studio.

21. Click **Test**. You should receive a “Successful test” message.
22. Log out from ODI Studio.

6.2.4 Packing the domain on Node 1

Follow the below steps to pack the domain on Node 1:

1. Shut down `ODI_server1` and go back to your browser.
2. Using the left side panel of the Weblogic Server, navigate to `Environments > Servers`.
3. On the right side of the screen, select **Control** and then select **ODI_server1**.
4. Click **Shutdown** and then select **Force shutdown now** from the drop-down list.
5. Click **Yes** in the Server Life Cycle Assistant screen.
6. Shut down the Node Manager. In the terminal screen, navigate to the following directory:

   ```
   cd /u01/oracle/mwh/user_projects/domains/odi_domain/bin
   ```

7. Stop the Node Manager. Execute the following command:

   ```
   ./stopNodeManager.sh.
   ```

8. Navigate to the following directory:

   ```
   cd /u01/oracle/mwh/oracle_common/common/bin
   ```

9. Pack the domain information. Execute the following command:

   ```
   ./pack.sh -domain=/u01/oracle/mwh/user_projects/domains/odi_domain -
   template=odiclusterdomain.jar -template_name=odiclusterdomain -managed=true
   ```

6.2.5 Unpacking the domain on Node 2

Follow the below steps to unpack the domain on Node 2:

1. Copy the file from Node 1 to Node 2. The method shown here uses the "scp" command. If you prefer to use a different method (for example, a third party tool like WinSCP or Firezilla), skip this step. From node 1, execute the following command:

   ```
   scp -i <PRIVATE_KEY_FILE_PATH> odiclusterdomain.jar opc@<IP_ADDRESS_NODE2>:/
   <PATH>
   ```

2. In Node 2, navigate to the following directory:

   ```
   cd /u01/oracle/mwh/oracle_common/common/bin
   ```

3. Unpack the domain. Execute the following command:

   ```
   ./unpack.sh -domain=/u01/oracle/mwh/user_projects/domains/odi_domain -
   template=<PATH>/odiclusterdomain.jar
   ```

4. If ODI was provisioned with stack release 12.2.1.4.200618 or later, execute the following steps:
   a. Navigate to the following directory:

      ```
      cd /u01/oracle/mwh/wlserver/server/bin
      ```

   b. Copy the following file:

      ```
      cp startNodeManager.sh startNodeManger.sh.OLD
      ```

c. Edit the following file:

```
vi startNodeManager.sh
```

d. Change line 52 from `WL_HOME="/home/opc/oracle/wlserver"` to `WL_HOME="${MW_HOME}/wlserver"`.

6.2.6 Setting up the Managed Server and Node Manager on Node 2

Follow the below steps to set up the managed server and node manager on Node 2:

1. Navigate to the following directory:

   ```
   cd /u01/oracle/mwh/user_projects/domains/odi_domain/nodemanager
   ```

2. Edit the `nodemanager.properties` file. Execute the following command:

   ```
   vi nodemanager.properties
   ```

3. Verify that the value for the “ListenAddress” entry shows the Node 2 private IP address, and the value for the "ListenPort” entry shows the Node 2 node manager listen port (`5557/9557`).

4. Restart Node Manager in both nodes. Navigate to the following directory:

   ```
   cd /u01/oracle/mwh/user_projects/domains/odi_domain/bin
   ```

5. Execute “startNodeManager” and send the process to the background using the following command:

   ```
   nohup ./startNodeManager.sh &
   ```

6. Monitor the startup process using the following command:

   ```
   tail -100 nohup.out
   ```

7. Verify both the Node Managers. In the web browser, log into the WebLogic Server.

8. Using the left side panel, navigate to `Environments > Machines`.

9. Drill down on Node 1 (`odi-node1`).

10. On the right side of the screen, select **Monitoring** and then **Node Manager Status**. You can view the current status information for the Node Manager instance configured for the machine.

11. Repeat steps 7 to 10 but now with Node 2 (`odi-node2`).

12. Start both the ODI servers.

13. Using the left side panel of the WebLogic server, navigate to `Environments > Servers`.

14. On the right side of the screen, select **Control** and then select **ODI_server1**.

15. Click **Start**.

16. Click **Yes** in the Server Life Cycle Assistant screen.

17. Keep refreshing the page until you see a “RUNNING” status for `ODI_server1`.

18. Repeat steps 12 to 14 but now with Node 2 (`ODI-server2`).

19. In Node 2, start ODI Studio.

20. Click **No** in the Confirm Import Preferences screen.

21. From the welcome screen, click **Connect to Repository**.

22. If you want to create a wallet to store ODI passwords, select **Store passwords in secure wallet** in the New Wallet Password screen and provide the password. Otherwise, select **Store passwords without secure wallet**. Click **OK**.
23. Click the "+" sign in the Oracle Data Integrator Login screen.
24. In the Repository Connection Information screen, provide the following information:
    a. **Login Name**: This will be the name of the connection to repository. Assign any name you deem appropriate. We recommend using the same name you used in Node 1.
    b. **User**: Specify `SUPERVISOR`.
    c. **Password**: Specify the password for the `SUPERVISOR`.
    d. **User**: Specify the owner of the repository, which is `DEV_ODI_REPO` in this case.
    e. **Password**: Specify the password for `DEV_ODI_REPO`.
    f. **Driver List**: Select **Oracle JDBC Driver**.
    g. **URL**: Specify the JDBC database URL as follows:

    ```
    jdbc:oracle:thin:@<SERVER>:<PORT>/<SERVICE>
    ```
25. Click **Test**. You will receive a confirmation of the connection. Click **OK** in the Information screen.
26. Select **Work Repository** and then click on the magnifying glass on the far right.
27. In the Select Repository screen, select **WORKREP** and click **OK**.
28. If everything is fine, a successful connection message will be sent. Click **OK** in the Information screen.
29. Click **OK** in the Repository Connection Information screen.
30. Click **OK** again to log into ODI Studio.
31. Navigate to the **Topology** tab.
32. Open `OracleDIAgent`.
33. Change the Host field to point to the Node 2 private IP Address.
34. Save the configuration.
35. Click **Test**. You should receive a success message.

6.3 Configuring the Load Balancer

Load balancer created can either be private or public. For more information on load balancers, refer to Overview of Load Balancing. Persistence should not be enabled on the load balancer.

Follow the below steps to create and configure the load balancer:

1. Open the navigation menu. Under the **Core Infrastructure** group, go to **Networking** and click **Load Balancers**.
2. Choose a **Compartment** you have permission to work in under **Scope**, and then click **Create Load Balancer**.
3. Specify the attributes of the load balancer as follows:
    a. **Load Balancer Name**: Specify a name for your load balancer.

   b. **Choose Visibility Type**: Select **Private**.
   c. **Choose the Maximum Total Bandwidth**: Select **Medium**.
   d. **Choose Networking**: Select the same VCN and subnet where your ODI nodes are located.
4. Click **Next** located at the left bottom corner of the screen.
5. Click **Add Backends** to select resources from a list of available Compute instances.
6. Select the two ODI nodes and click **Add Selected Backends** located at the bottom of the screen.
7. Change the port of both nodes to point to the ODI Managed Servers “ODI_server1” and “ODI_server2” (8001 / 15101).
8. In the Specify Health Check Policy section, provide the following information:
   a. **Port**: Provide the ODI Managed Servers port (8001/15101).
   b. **URL Path (URI)**: Specify `/oracledіagent/`.
9. Click **Next** located at the left bottom corner of the screen.
10. In the Configure Listener section, provide the following information:
    a. **Specify the type of traffic your listener handles**: Select **HTTP**.
    b. **Specify the port your listener monitors for ingress traffic**: Specify **80**.
11. Click **Show Advanced Options** located at the bottom of the screen to access additional options.
12. Specify **3600** in the Specify the Maximum Timeout in Seconds field.
13. Click **Submit** located at the left bottom corner of the screen. This starts the provisioning process for the load balancer.
14. Once the Load Balancer is provisioned, it will be shown as “Active” in the screen. Make a note of the private IP Address.
15. At this time, start ODI Studio (in any node) and navigate to the Topology tab.
16. Modify the agent “OracleDIAgent” as follows:
    a. **Host**: Specify the Load Balancer Private IP Address.
    b. **Port**: Specify **80**.
17. Click **Test**. The ODI Information dialog box should say, "Agent Test Successful".

6.3.1 Updating the Load Balancer Health Check

Follow the below steps to change the health check information for the load balancer:

1. Drill down in the Load Balancer name.
2. Click the **Backends** hyperlink available on the left panel.
3. Drill down in the backend set name.
4. Click the **Backends** hyperlink available on the left panel.
5. Click **Update Health Check** located at the top of the screen.
6. Modify the check information and click **Save Changes** located at the bottom of the screen. You will receive a message about the information being accepted.

7. Test the ODI agent multiple times.

6.4 Enabling Incoming Ports and Services

For establishing communication between the instances, you need to ensure that the underlying Security List (associated with VCN), has all the IP protocols enabled.

The following is an example of security list that enables communication within the instances participating in High Availability (HA) cluster:

> **Note:**
>
> All the instances participating either directly or indirectly should be following the below ingress and egress rules.

Table 6-1 Ingress and Egress Rules Table

Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Comments
No	0.0.0.0/0	TCP	All	22	Nil	TCP traffic for ports: 22 SSH Remote Login Protocol	For SSH communication, we need to open port # 22.
No	0.0.0.0/0	ICMP	Nil	Nil	3,4	ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set	ICMP is a supporting protocol and at the minimum, ingress rules should allow for type 3, 4 and 8. For more information on ICMP protocols, refer to the IANA list.

Table 6-1 (Cont.) Ingress and Egress Rules Table

Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Comments
No	10.0.0.0/16	ICMP	Nil	Nil	3	ICMP traffic for: 3 Destination Unreachable	ICMP is a supporting protocol and at the minimum, ingress rules should allow for type 3, 4 and 8. For more information on ICMP protocols, refer to the IANA list.
No	0.0.0.0/0	ICMP	Nil	Nil	8	ICMP traffic for: 8 Echo	ICMP is a supporting protocol and at the minimum, ingress rules should allow for type 3, 4 and 8. For more information on ICMP protocols, refer to theIANA list.
No	0.0.0.0/0	TCP	All	1521	Nil	TCP traffic for ports: 1521	Port 1521 is for database traffic.
No	0.0.0.0/0	TCP	All	443	Nil	TCP traffic for ports: 443 HTTPS	Port 443 is the SSL traffic.
No	0.0.0.0/0	TCP	All	7001	Nil	TCP traffic for ports: 7001	WLS Admin Server communication port.

Table 6-1 (Cont.) Ingress and Egress Rules Table

Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Comments
No	0.0.0.0/0	TCP	All	8001	Nil	TCP traffic for ports : 8001	Managed Server communication port for all nodes.
No	0.0.0.0/0	TCP	All	5556	Nil	TCP traffic for ports : 5556	Node Manager Port in Node1.
No	0.0.0.0/0	TCP	All	5557	Nil	TCP traffic for ports : 5557	Node Manager Port in Node2.
No	0.0.0.0/0	UDP	All	7574	Nil	UDP traffic for ports : 7574	WLS Cluster port
No	0.0.0.0/0	TCP	All	8095	Nil	TCP traffic for ports : 8095	Oracle Coherence port on Node1.
No	0.0.0.0/0	TCP	All	8096	Nil	TCP traffic for ports : 8096	Oracle Coherence port on Node2.
No	10.0.17.0/24	TCP	All	8001	Nil	TCP traffic for ports : 8001	Oracle ODI Agents both nodes
No	0.0.0.0/0	TCP	All	7	Nil	TCP traffic for ports : 7 ECHO	Coherence TCP Ring/IP Monitor death detection feature.
No	10.0.17.0/24	TCP	All	80	Nil	TCP traffic for ports : 80	OCI Load balancer

6.5 Firewall Rules

Even after setting the ingress and egress rules, in some cases the instances may not allow the incoming traffic. This is because of the firewall associated with the instance. Ensure to enable all the communication ports by configuring the firewall.

The following is an example of firewall commands for the ingress/egress ports (that has enabled port communication) :

Firewall command to enable port communication between the nodes (run on both machines):

```
sudo firewall-cmd --permanent --new-service=odiwls
sudo firewall-cmd --permanent --service=odiwls --set-description="ODI
WLS server"
sudo firewall-cmd --permanent --service=odiwls --add-port=7001/tcp
sudo firewall-cmd --permanent --add-service=odiwls
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --new-service=odimanagedwls
sudo firewall-cmd --permanent --service=odimanagedwls --set-
description="ODI WLS Managed Server"
sudo firewall-cmd --permanent --service=odimanagedwls --add-
port=8001/tcp
sudo firewall-cmd --permanent --add-service=odimanagedwls
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --new-service=odiwlsnodemgr1
sudo firewall-cmd --permanent --service=odiwlsnodemgr --set-
description="ODI WLS Node Manager1"
sudo firewall-cmd --permanent --service=odiwlsnodemgr --add-
port=5556/tcp
sudo firewall-cmd --permanent --add-service=odiwlsnodemgr1
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --new-service=odiwlsnodemgr2
sudo firewall-cmd --permanent --service=odiwlsnodemgr --set-
description="ODI WLS Node Manager2"
sudo firewall-cmd --permanent --service=odiwlsnodemgr --add-
port=5557/tcp
sudo firewall-cmd --permanent --add-service=odiwlsnodemgr2
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --new-service=odiwlscluster
sudo firewall-cmd --permanent --service=odiwlscluster --set-
description="ODI WLS cluster"
sudo firewall-cmd --permanent --service=odiwlscluster --add-
port=7574/udp
sudo firewall-cmd --permanent --add-service=odiwlscluster
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --new-service=odiwlscoherencewk1
sudo firewall-cmd --permanent --service=odiwlscoherencewk1 --set-
description="ODI WLS coherence WKA1"
sudo firewall-cmd --permanent --service=odiwlscoherencewk1 --add-
port=8095/tcp
sudo firewall-cmd --permanent --add-service=odiwlscoherencewk1
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --new-service=odiwlscoherencewk2
sudo firewall-cmd --permanent --service=odiwlscoherencewk2 --set-
description="ODI WLS coherence WKA2"
sudo firewall-cmd --permanent --service=odiwlscoherencewk2 --add-
```

```
port=8096/tcp
sudo firewall-cmd --permanent --add-service=odiwlscoherencewk2
sudo firewall-cmd --reload
```

TCP Ring port 32783 – Coherence Cluster

If you run a firewall, you need to configure it to enable the specified addresses and ports. Firewalls are not typically set up between cluster members. If a solution requires the use of a firewall, then ensure the following:

- The cluster port (7574 by default) is open for both UDP and TCP for both multicast and unicast configurations.
- TCP port 7 is open for the Coherence TCP Ring/IP Monitor death detection feature.
- The unicast port range is open for both UDP and TCP traffic. Ensure that the unicast listen port range is explicitly set rather than relying upon a system assigned ephemeral port.

Cluster member unicast ports are automatically assigned from the operating system's available ephemeral port range. This ensures that Coherence cannot accidentally cause port conflicts with other applications. However, if a firewall is required between cluster members (an atypical configuration), then the port must be manually configured.

You can specify the unicast port using the -D arguments as shown below:

```
-Dcoherence.localport=9000 -Dcoherence.localport.adjust=9200
```

The `coherence.localhost`, `coherence.localport`, and `coherence.localport.adjust` system properties are used to specify the unicast port and automatic port adjustment settings instead of using the operational override file. The `coherence.localport.adjust` value is the upper limit to auto adjust the local ports. In the above example, the port range values used are 9000 and 9200. You can use any other port range.

You need to add the following firewall rule on both the nodes:

```
sudo firewall-cmd --permanent --new-service=odicoherencecluster
sudo firewall-cmd --permanent --service=odicoherencecluster --set-
description="ODI Coherence Cluster TCP Ring"
sudo firewall-cmd --permanent --service=odicoherencecluster --add-
port=32783/tcp
sudo firewall-cmd --permanent --add-service=odicoherencecluster
sudo firewall-cmd --reload
```

TCP Port 7 – Coherence Death Detect

You need to add the following firewall rule on both the nodes:

```
sudo firewall-cmd --permanent --new-service=odicoherencedeathdetect
sudo firewall-cmd --permanent --service=odicoherencedeathdetect --set-
description="ODI Coherence Cluster TCP Ring"
sudo firewall-cmd --permanent --service=odicoherencedeathdetect --add-
port=7/tcp
sudo firewall-cmd --permanent --add-service=odicoherencedeathdetect
sudo firewall-cmd –reload

sudo firewall-cmd --list-all
```

Load Balancer HTTP Traffic to both nodes

You need to add the following firewall service on both the nodes:

```
sudo firewall-cmd --permanent --zone=public --add-service=http
sudo firewall-cmd --reload
```

7
Troubleshooting ODI on OCI

This chapter describes about various services associated with ODI on OCI and ways to troubleshoot them when you encounter issues while using them.

> **Note:**
>
> If you are facing issues connecting to ADB dataserver and MySQL repository after long hours of inactivity, try reconnecting to ODI repository to overcome this problem.

How Tos

> **Note:**
>
> The following commands are supported in release(s) prior to 12.2.1.4.200618 version of ODI Marketplace. For latest release, refer to Managing ODI Credential .

- How to get the password for a MYSQL based ODI repository?

```
$MW_HOME/odi/common/scripts/getPassword.sh
```

- How to update ODI Repository and schema credentials for ADB technology?

```
cd $MW_HOME/odi/common/scripts
Create a new file updateCredentials.sh and add below contents
echo "Updating credentials in wallet..."
WCLASSPATH=../../sdk/lib/odi-core.jar:../../sdk/lib/commons-
lang-2.2.jar:../../../oracle_common/modules/oracle.jps/*:../../../
oracle_common/modules/oracle.igf/identitydirectory.jar:../../../
oracle_common/modules/oracle.idm/identitystore.jar:../../../oracle_common/
modules/oracle.osdt/osdt_cert.jar:../../../oracle_common/modules/
oracle.osdt/osdt_core.jar:../../../oracle_common/modules/oracle.osdt/
osdt_xmlsec.jar:../../../oracle_common/modules/oracle.pki/
oraclepki.jar:../../../wlserver/modules/
com.oracle.weblogic.security.jar:../../../wlserver/modules/
com.oracle.weblogic.security.subject.jar
export WCLASSPATH
java -cp $WCLASSPATH oracle.odi.setup.util.ODIWalletSetupUtil odi ../../
common/scripts/jps-config-jse.xml <supervisorPassword> <schemaPassword>
Save the file and stop the ODI agent.
Run the script ./updateCredentials.sh.
Restart the ODI agent.
```

> **Note:**
>
> In the above commands, change the values for `<supervisorPassword>` `<schemaPassword>` fields with your respective ODI supervisor password and ADB Repository schema password.

8
Known Issues and Workarounds

This chapter details known issues in this release, and their workarounds.

Mismatch in Server Certificate DN with connection details having tls for ATP/ADW-D:

When you navigate to Dataserver configured with ATP/ADW-D and select Connection details with TLS, Test connection fails due to server certificate issue. It happens when you force JDBC driver to match DN (Server's Distinguished Name) by setting the property `oracle.net.ssl_server_dn_match=true` and DN mentioned in JDBC URL does not match with the DN mentioned in server certificate.

As a workaround, mention the correct DN in JDBC URL or set `oracle.net.ssl_server_dn_match=false` in the JDBC tab of ATP/ADW data server.

For example, the expected DN in the JDBC URL is:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=servername)
(PORT=
2484))(CONNECT_DATA=(SERVICE_NAME=servicename))
(SECURITY=(SSL_SERVER_CERT_DN=\
"CN=server_test,C=US\")))
```

Issue with ODI Studio when rebooting the ODI instance

When starting ODI studio after rebooting an ODI instance, an exception is observed. As a workaround, start the ODI studio with `-clean` option to clear the cache.