

Oracle® FMW

Getting Started with Oracle Data Integrator on Oracle Cloud Marketplace



12.2.1.4.0

F33188-05

July 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle FMW Getting Started with Oracle Data Integrator on Oracle Cloud Marketplace, 12.2.1.4.0

F33188-05

Copyright © 2020, 2022, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1 Getting Started with Oracle Cloud Marketplace

1.1	Prerequisites	1-2
1.2	Selecting Your Product	1-5
1.2.1	Terminology Information	1-6
1.3	Launching Your Oracle Data Integrator Instance	1-7
1.4	Connecting to ODI Compute Instance	1-11
1.5	Accessing the ODI Compute Instance	1-12
1.6	Using Autonomous Databases in ODI	1-14
1.6.1	Connecting to the Pre-created ADB Dataservers in ODI Repository	1-14
1.6.2	Using Dataserver Setup in ODI Studio	1-14
1.6.3	Manually Registering the Created ODI Instance to Autonomous Database	1-16
1.7	Installation Locations	1-16
1.8	Log Files Location	1-17
1.9	Patching	1-17

2 Migrating from Oracle Data Integrator Cloud Services to Oracle Data Integrator Marketplace

2.1	Exporting the ODI-CS Repository	2-2
2.2	Provisioning DBaaS	2-2
2.3	Provisioning ATP	2-2
2.4	Provisioning ODI Marketplace	2-2
2.5	Installing the required ODI version in ODI Marketplace node	2-2
2.6	Creating an ODI Repository in DBaaS	2-3
2.7	Importing the ODI Repository into the DBaaS node	2-4
2.8	Verifying access to the DBaaS ODI Repository in ODI Studio	2-4
2.9	Upgrading the Repository	2-5
2.10	Exporting the Migrated Repository	2-6
2.11	Importing the Migrated Repository into ATP	2-6
2.12	Verifying access to the Migrated Repository in ODI Marketplace Studio	2-7

A Creating a Standalone Collocated Agent

A.1	Installing the Agent	A-1
A.2	Defining the Agent in ODI	A-2
A.3	Starting the Agent	A-3

B Importing the ODI Repository dump

List of Figures

1-1 Discover ADBs

1-15

1

Getting Started with Oracle Cloud Marketplace

Oracle Data Integrator on Oracle Cloud Marketplace is a product offering that enables customers to quickly set up and run Oracle Data Integrator (ODI) on Oracle Cloud. It provides a fully unified solution for building, deploying, and managing complex data warehouses or as part of data-centric architectures in a SOA or business intelligence environment. In addition, it combines all the elements of data integration - data movement, data synchronization, data quality, data management, and data services - to ensure that information is timely, accurate, and consistent across complex systems.

This chapter contains the following sections:

- [Prerequisites](#)
- [Selecting Your Product](#)
- [Launching Your Oracle Data Integrator Instance](#)
- [Connecting to ODI Compute Instance](#)
- [Accessing the ODI Compute Instance](#)
- [Using Autonomous Databases in ODI](#)
- [Installation Locations](#)
- [Log Files Location](#)
- [Patching](#)



Note:

Refer to [Launching Your First Linux Instance](#) documentation before creating the ODI instance.

- [Prerequisites](#)
- [Selecting Your Product](#)
- [Launching Your Oracle Data Integrator Instance](#)
- [Connecting to ODI Compute Instance](#)
- [Accessing the ODI Compute Instance](#)
- [Using Autonomous Databases in ODI](#)
- [Installation Locations](#)
- [Log Files Location](#)
- [Patching](#)

1.1 Prerequisites

Make sure you have the following prerequisites before using Oracle Data Integrator on Oracle Cloud Marketplace:

- [Oracle Cloud Account](#)
- Have access to assigned Oracle Cloud Tenant
- Compute node resources within Oracle Cloud Tenant

Go through the following prerequisites carefully before creating the ODI instance:

- Refer to [Creating Dynamic Group Policies](#) for compartment, before provisioning ODI and ADB Instances.
- Read thoroughly the [password policies](#) provided in the Stack UI and create passwords accordingly during provisioning.
- Wait till the completion of ODI instance creation, check for its status and then proceed working on it.
For more information, refer to the usage instructions in [Launching Your Oracle Data Integrator Instance](#).

Supported Browsers

Oracle Cloud Infrastructure supports the latest desktop versions of Google Chrome, Microsoft Edge, Internet Explorer 11, Safari, Firefox, and Firefox ESR. Note that mobile browsers as well as private browsing mode is not supported for Firefox, Internet Explorer, or Edge.

Creating an SSH/RSA Key

To work with the Oracle Cloud Infrastructure once the Oracle Data Integrator Compute Node is built, you have to provide a SSH Public Key to allow you to login to the node.

In order to build your SSH keys, perform the following steps:

1. In a terminal window, generate the SSH key using the following command:

```
$ ssh-keygen Generating public/private rsa key pair.
```

2. Enter the path to store this file. By default, this gets saved in your home directory under a hidden folder called `.ssh`. Change this default location, if required.

```
Enter file in which to save the key (/Users/johndoe/.ssh/id_rsa):  
<Return>
```

3. Enter a passphrase using your key.

```
Enter passphrase (empty for no passphrase): <passphrase>
```

4. Re-enter the passphrase to confirm it.

```
Enter same passphrase again: <passphrase>
```

5. Check the results.

The key fingerprint (a colon separated series of two digit hexadecimal values) is displayed. Check if the path to the key is correct. In the above example, the path is /Users/johndoe/.ssh/id_rsa.pub. You have now created a public and private key pair.

Creating Dynamic Group and Policies

During ODI Instance provisioning, data servers for all accessible Autonomous Databases (ADB) are automatically created, as long as the dynamic group and policies are created before deploying ODI on Marketplace and are set as mentioned below. Dynamic Group and Policies are required when using the ODI Repository on an Autonomous Database.

- Create a dynamic group to include matching rules for instances in a specified compartment. For example,

```
ALL {instance.compartment.id =  
'ocid1.compartment.oc1..aaaaaaaabgr34tpuanpvq6xfb667nsmy2jz45zj6dexojhxds  
v4mjayem3cq'}
```

For more information, refer to [Create Dynamic Groups and Policy](#).

- Navigate to Identity -> Policies -> Create Policy to create policy statements as specified below :

If you set policy at:

- ODI compartment level, then all ADW/ATP instances from the compartment where ODI instance is created are listed.
For example - To List ADW/ATP instances only from the ODI instance compartment, you have to setup the following policy:

```
Allow dynamic-group odi_group to inspect autonomous-database-family  
in compartment odi  
Allow dynamic-group odi_group to read autonomous-database-family in  
compartment odi  
Allow dynamic-group odi_group to inspect compartments in compartment  
odi
```

- ODI tenant level, then all ADW/ATP instances from all the compartments of the tenancy are listed.
For example - To list ADW/ATP instances from all the compartments of tenancy, you have to setup the following policy:

```
Allow dynamic-group odi_group to inspect autonomous-database-family  
in tenancy  
Allow dynamic-group odi_group to read autonomous-database-family in  
tenancy  
Allow dynamic-group odi_group to inspect compartments in tenancy
```

- To configure email delivery service for specified groups on Oracle Cloud Marketplace: An email approved sender must be in a group that has IAM policy permissions to send emails. An 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
```

Creating an Oracle Cloud Infrastructure Policy in an Identity Domain

Create a policy to grant permissions to users in a domain group to work with Oracle Integration instances within a specified tenancy or compartment.

Note:

This topic applies only to cloud accounts that use identity domains. See [About Setting Up Users and Groups](#).

1. Open the navigation menu and click **Identity & Security**. Under **Identity**, click **Policies**.
2. Click **Create Policy**.
3. In the Create Policy window, enter a name (for example, `IntegrationGroupPolicy`) and a description.
4. In the **Policy Builder**, select **Show manual editor** and enter the required policy statements.

Syntax:

- `allow group domain-name/group_name to verb resource-type in compartment compartment-name`
- `allow group domain-name/group_name to verb resource-type in tenancy`

Example: `allow group admin/oci-integration-admins to manage integration-instance in compartment OICCompartment`

Note:

If you omit the domain name, the default domain is assumed.

This policy statement allows the `oci-integration-admins` group in the `admin` domain to manage instance `integration-instance` in compartment `OICCompartment`.

You can create separate groups for different permissions, such as a group with read permission only.

Want to learn more about policies? See [How Policies Work](#) and [Policy Reference](#), or click **Help** in the window.

- When defining policy statements, you can specify either verbs (as used in these steps) or permissions (typically used by power users).

- The `read` and `manage` verbs are most applicable to Oracle Integration. The `manage` verb has the most permissions (`create`, `delete`, `edit`, `move`, and `view`).

Verb	Access
<code>read</code>	Includes permission to view Oracle Integration instances and their details.
<code>manage</code>	Includes all permissions for Oracle Integration instances.

5. If desired, you can add a policy to allow members of the group to view message metrics, as described in [Viewing Message Metrics](#).

For example:

```
allow group oci-integration-admins to read metrics in compartment
OICPMCompartment
```

6. If you intend to use custom endpoints, add one or more additional policy statements. Otherwise, skip this step.

Add policies that specify the compartment in which vaults and secrets reside and allow the admin group to manage secrets in it. See [Configure a Custom Endpoint for an Instance](#).

Note that you should specify the resource to return in *resource-type*, as described in [Details for the Vault Service](#). Also note that Oracle Integration requires the `read` verb only but `manage` is recommended if the same group will also be administering the secrets (uploading/lifecycle operations).

Examples::

- ```
allow group admin/oci-integration-admins to manage secrets in compartment
SecretsCompartment
```
- ```
allow group admin/oci-integration-admins to manage vaults in compartment
SecretsCompartment
```

7. Click **Create**.

The policy statements are validated and syntax errors are displayed.

1.2 Selecting Your Product

To search for the product Oracle Data Integrator on Oracle Cloud Marketplace, enter the product name in the search text box and click **Go**.

From the listings related to Oracle Data Integrator, choose either one of the following.

- **Data Integrator: Classic:** Full ODI functionality for loading data to Oracle Database Cloud Services.
- **Data Integrator: Classic BYOL:** Full ODI functionality for loading data to other targets.

You are navigated to the respective product listing page which has all the basic usage information along with the product documentation.

 **Note:**

The information described in this guide is applicable to Data Integrator: Classic and Data Integrator: Classic BYOL. Oracle Cloud Marketplace also displays listings related to Data Integrator: Web Edition. For instructions about getting started with Data Integrator: Web Edition, see [Getting Started with Oracle Data Transforms](#).

1. From your respective product listing page, select **Get App**.
2. Select OCI Region or Login using your Single Sign-On credentials.
 - OCI Region – Select the desired region and click **Create Stack**.
3. Provide the OCI tenant details.
4. Sign-in to the Identity provider.
5. On the **Oracle Data Integrator** page, provide the following information:
 - a. Type - It is Stack by default.
 - b. Compartment - Specifies the compartment where the compute node will be built. It is generally the location that you have access to build the compute node.
 - c. Terms of Use - This check-box is selected by default. Oracle recommends to review the licenses before proceeding.
 - d. Launch Stack - It launches the stack in the OCI environment. After selecting all the required information, click **Launch Stack**.

 **Note:**

Provisioning your instance on Oracle Cloud Marketplace greatly depends on your product selection. You cannot use the same VM instance for multiple product offerings. The procedure listed below are product specific and may differ based on the product version.

- [Terminology Information](#)

1.2.1 Terminology Information

The Terminology Information used in Oracle Data Integrator Studio are:

Oracle Data Integrator Studio	Functions
Data Server	Represents the physical object where data is stored. Examples include Database Instances, File Servers, Cloud Application Instances. The properties contain all information required to connect and access data.

Oracle Data Integrator Studio	Functions
Technology	Examples include Oracle, IBM DB2, Oracle Netsuite, Oracle Object Storage. For the list of supported technologies, see Setting Up the Connections .
Physical Schema	Examples include Database Schema, File server Folder, Object Storage Bucket.
Data Store	A Tabular representation of a data structure. Examples include Database Tables, Files.
Reverse Engineering	The process of obtaining the metadata for a filtered set of objects in a Schema. Examples include Table Definitions from an Oracle Schema, VOs from a Cloud Applications Offering.
Model	The container for the imported objects. The properties define the rules for importing (how to filter and so on).
Project	Container for the Transformation Design components (Mappings, Packages, Jobs). This doesn't include Data Servers, Schemas, Data Stores and Folders (which are shared between Projects)
Mapping	The Transformation Design. Defines how data flows from one Data Store to another, and how it is transformed.
Package	Defines the sequence in which Mappings will be executed, together with what happens of failure conditions.
Job / Session	The execution of a Package, Mapping or Reverse Engineer.
Schedule	The rules for when a session executes.
Resource	Something you can schedule (a mapping or package).

1.3 Launching Your Oracle Data Integrator Instance

Follow the below procedure to launch your ODI instance:

Once you click **Launch Stack**, you are navigated to the **Create Stack** page.

1. Fill in the required Stack information:

- Name - Name of the Stack. It has a default name and provides a datetime stamp. You can edit this detail, if required.
- Description - Description that you provide while creating the Stack.
- Create In Compartment – It defaults to the compartment you have selected on the Oracle Data Integrator for Oracle page.
- Tags (optional) – Tags are a convenient way to assign a tracking mechanism but are not mandatory. You can assign a tag of your choice for easy tracking. You have to assign a tag for some environments for cost analysis purposes.

- Click Next.
- 2. Fill in the required details to configure variables. This information is required to build the compute node with Oracle Data Integrator.
 - General Settings -
 - a. Networking Options - Denotes the networking options in which the Oracle Data Integrator compute instance and all the associated network resources have to be created. Select between two available options:
 - New networking components will be created - Select this option to have a new network (VCN) and subnet created and configured alongside this new ODI instance. Minimal inputs are required when you select this option.
 - Existing networking components will be used - Select this option to use the existing network components and subnets.
 - b. ODI Repository Location - Select between the following options for creating a repository:
 - Create a new ODI Repository in an Autonomous Database
 - Connect to an existing ODI Repository in an Autonomous Database
 - Create an Embedded ODI Repository

 **Note:**

Oracle Data Integrator Repositories that you create during provisioning are mode specific. You cannot use a repository created in Oracle Data Transforms mode with a stack in ODI Studio mode and you cannot access a repository created in ODI Studio mode with a stack in Oracle Data Transforms mode.

- c. Resource Display Name Prefix (optional) - It is the prefix added to the display name of all the compute and network resources generated . Display name is used to identify all new OCI resources. If you do not specify any prefix, then it is automatically generated.
- d. Target Compartment – Denotes the target compartment for all the provisioned resources .
- Network Configuration -
 - New networking components will be created -
 - a. Virtual Network CIDR (optional) - A single, contiguous Virtual Cloud Network (VCN) CIDR block for the VCN. For example: 172.16.0.0/16. You cannot change this value later.

 **Note:**

This option is available for both new and existing modes of network creation.

- b. Create or Use a Public ODI Subnet - It creates all the ODI resources of the newly created instance in a public subnet. This option is

selected by default. If you clear this checkbox, all the ODI resources are created in a private subnet.

- Existing networking components will be used -
 - a. Virtual Network CIDR (optional) - A single, contiguous Virtual Cloud Network (VCN) CIDR block for the VCN. For example: 172.16.0.0/16. You cannot change this value later.
 - b. VCN Compartment - Specifies the VCN compartment where the compute node will be built. It is generally the location that you have access to build the compute node. From the Virtual cloud network compartment drop-down list, select the compartment where your VCN has been setup.
 - c. VCN - A VCN is a software-defined network that you set up in the Oracle Cloud Infrastructure data centers in a particular region. Select the desired VCN for the newly created instance. From the Virtual cloud network drop-down list, choose your VCN.
 - d. Subnet Compartment - Denotes the subnet compartment to use for the newly created instance if you are not creating a new network. From the Subnet Compartment drop-down list, select the compartment containing the subnet that you wish to use.
 - e. Subnet (optional) - Existing subnet to use for the newly created instance if you are not creating a new network. From the Subnet drop-down list, select the subnet you wish to use (you need not change this normally from the default one, when you select the Subnet compartment).

 **Note:**

While creating a subnet, select the option **Use DNS Hostnames in this Subnet**. If you do not select this option you cannot edit the subnet information and as a result of this, stack creation fails. To rectify this, you have to create a different subnet again by enabling this option.

- f. Assign Public Address - This option indicates if the newly created VM should have a public IP address. This option is set to True, by default. If you set this to False, no public ip address will be assigned preventing public access to the compute node.

 **Note:**

If you are using a private IP address to access the compute node, you have to setup an IPsec VPN or FastConnect connection. Refer to [OCI documentation](#) for more details.

- ODI Instance Settings -
 - a. ODI Node Shape - Shape of new compute instance. Supported shapes are VM.Standard 2.4, VM.Standard 2.8, VM.Standard 2.16 and VM.Standard 2.24. From the Node Shape drop-down list, select the required shape.
 - b. SSH Public Key - Public Key for allowing SSH access as the 'oracle' user. Refer to [Prerequisites](#) section, for more details on creating the SSH Public Key.

- c. ODI Availability Domain - It specifies the availability domain for the newly created Oracle Data Integrator Instance. From the **Select an Availability Domain for your Instance** drop-down, choose an availability domain. If you are using regional subnet in Network Configurations, you can use any of the availability domains from the ODI Availability Domain list.
- d. ODI VNC Password - Provide a new password that you will use to connect to the VNC Server on the newly created ODI instance.
- New ODI Metadata Repository Settings -
 - Autonomous Database Instance - Select the created Autonomous Database instance from this drop down arrow.
 - Autonomous Database Password - Provide the password credentials of the administrator user of the Autonomous Database.
 - Supervisor Password - Provide the ODI Supervisor password.

 **Note:**

ODI Supervisor password should be 6-12 characters in length and no special characters except \$, # and _ are allowed.

- Schema Prefix - Specifies the prefix used for the RCU Schemas, used to identify them. For existing repository, provide the prefix configured during repository creation.

 **Note:**

A valid prefix should contain only alpha-numeric characters. It should not start with a number and should not contain any special characters. A maximum of 12 characters are allowed.

- Schema Password - Specifies the password used for accessing the RCU Schemas. For existing repository, provide the password configured during repository creation.

 **Note:**

Schema password that you provide should :

- * contain only alpha-numeric characters,
- * be a minimum of 12 characters in length,
- * contain at least one alphabet in upper case,
- * cannot start with a number
- * no special characters except \$, # and _ are allowed.

- Load Balancer IP Address -

 **Note:**

This parameter is not available for MYSQL based ODI repository.

If you have a loadbalancer configured for your instance(s), provide the IP address of the loadbalancer in this text box. If there is no loadbalancer, you can leave this field blank and the public IP address is used to register with Autonomous Database.

3. Click Next.
4. On the Review page, review the information you provided and then click Create.
5. When you get a message `Provisioning Completed/Successful` from the OCI console, the instance is created. However please note the rest of the ODI configuration happens in the background and takes approximately 20 minutes to complete. If you attempt to access ODI prior to this, provisioning will be under progress and you may not view default configurations in ODI Studio. For useful information refer to the job log file.
6. After approximately 20 minutes log into the VNC viewer using your `<IP address>:1` and check the status of the configuration. To check the status of the configuration, open a terminal window and execute the following command: `tail -f $MW_HOME/logs/odiConfigure.log`. Upon successful completion, you will get the message `ODI Apps started Successfully..` Wait for a few minutes for ODI apps to start up successfully.

 **Note:**

This step is applicable only for Advanced administrative users.

7. Now the newly created ODI Instance is ready for use. You can now launch ODI Studio to start working on your data integration projects.
8. To check the logs, navigate to `$MW_HOME/logs/odiConfigure.log`.

1.4 Connecting to ODI Compute Instance

 **Note:**

This section is applicable only for Advanced administrative users.

You can connect to an ODI compute instance by using a Secure Shell (SSH) connection. Most Linux distributions include an SSH client by default. For Windows, you can download a free SSH client called PuTTY from <http://www.putty.org>.

1. To connect to your ODI compute instance from linux,
 - Log in to your instance using SSH.
 - Use the following command to set the file permissions so that only you can read the file:

```
$ chmod 400 <private_key>
```

where `<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

- Use the following SSH command to access the instance.

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

where

- `<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.
- `<username>` is the default name for the instance. The default user name is `oracle`.
- `<public-ip-address>` is your instance IP address that you retrieved from the Console.

2. To connect to your ODI compute instance from windows,

- Open `putty.exe`.
- In the Category pane, select Window, and then select Translation.
- In the Remote character set drop-down list, select UTF-8. The default locale setting on Linux-based instances is UTF-8, and this configures PuTTY to use the same locale.
- In the Category pane, select Session and enter the following:
 - a. Host Name (or IP address): `<username>@<public-ip-address>` , where `<username>` is the default name for the instance. For Oracle Linux and CentOS images, the default user name is `oracle`. For the Ubuntu image, the default name is `ubuntu` and `<public-ip-address>` is your instance public IP address that you retrieved from the console.
 - b. Port: 22
 - c. Connection type: SSH
- In the Category pane, expand Connection, expand SSH, and then click Auth.
- Click Browse, and then select your private key.
- Click Open to start the session.

If this is your first time connecting to the compute instance, you might see a message that the server's host key is not cached in the registry. Click **Yes** to continue the connection.

1.5 Accessing the ODI Compute Instance

To access ODI apps through VNC, do the following:

1. Log in to the provisioned Oracle Data Integrator instance on Oracle Cloud Marketplace using SSH as `opc` user:

```
ssh -i <private_key> opc@<IP Address>
```

2. 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
SSHserver"
sudo firewall-cmd --permanent --service=odissh --add-port=5901-5905/tcp
sudo firewall-cmd --permanent --add-service=odissh
sudo firewall-cmd -reload
```

3. Install a VNC viewer on your local computer.
4. Use SSH to connect to the compute instance running the Oracle Data Integrator Image, as described in [Connecting to ODI Compute Instance](#).
5. 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>
```

6. 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
```

7. On your local computer, start a VNC viewer and establish a VNC connection to localhost:1.
8. Enter the VNC password that you had provided during the stack creation.
9. 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 start developing your data flows or data mappings/transformations, launch studio from the available options.

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 repository, the Login Name is `ODI_ADW_REPO` and for MySQL Embedded repository it is `ODISA_MYSQL`.
3. Post successful configuration, click Topology navigator -> Technologies -> Oracle to check if the newly created data server is available for use.
4. In ODI studio, navigate to Topology -> Physical Architecture -> Agent -> OracleDIAgent1 and click Test, to check if the Standalone Agent is working.

 **Note:**

For more details on services, refer to Managing ODI App Server.

5. Click Test connection, to check if the created ADB Data Server is working.

1.6 Using Autonomous Databases in ODI

The newly created repository for Oracle Data Integrator 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 Oracle Data Integrator transformations, then you have to add the username and password to the Data Server properties in the Topology tab in Oracle Data Integrator 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 Data Server on Oracle Technology of Oracle Data Integrator 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 Data Server name, credentials and then select connection details/service profiles and save, the new Oracle Dataserver for the selected ADB instance is created.

Follow the below procedure to create an Autonomous (ADB) Data Server in Oracle Data Integrator repository:

- [Connecting to the Pre-created ADB Dataservers in ODI Repository](#)
- [Using Dataserver Setup in ODI Studio](#)
- [Manually Registering the Created ODI Instance to Autonomous Database](#)
- [Connecting to the Pre-created ADB Dataservers in ODI Repository](#)
- [Using Dataserver Setup in ODI Studio](#)
- [Manually Registering the Created ODI Instance to Autonomous Database](#)

1.6.1 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.

1.6.2 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.

1.6.3 Manually Registering the Created ODI Instance to Autonomous Database

If you wish to manually register the ODI instance to Autonomous Database after creating your instance,

- Connect to the compute instance running the Oracle Data Integrator Image, as described in [Connecting to ODI Compute Instance](#) .
- Navigate to your ODI studio desktop and double click the **Register with ADP** shortcut icon.
A terminal window appears allowing you to enter the following details of the Autonomous Database instance which you had already created in [Launching Your Oracle Data Integrator Instance](#) step:
 - Admin username - User name of the Autonomous Database instance.
 - Admin password - Password of the Autonomous Database instance.
 - URL Scheme - Press Enter to select the default value - http.
 - IP Address - IP address of the computer which you wish to register to Autonomous Database.

 **Note:**

You are prompted to provide all the above details repeatedly until you provide a valid IP address.

After configuring the above details, the created instance is successfully registered to the Autonomous Database. After successful registration, you can launch Oracle Data Transforms directly from the Autonomous Database, Database Actions page.

 **Note:**

In some cases, you may encounter an error ("401 Not Authorised") when launching Oracle Data Transforms from the Database Actions page. To resolve this issue, you will need to restart the Jetty server.

1.7 Installation Locations

Please note, the following installation locations are used by this image. You may need this information if you want to change any aspects of the installation:

Table 1-1 Installation Locations

Area	Location on Server
MW_HOME	/u01/oracle/mwh
ODI_HOME	MW_HOME+"/odi"
OPatch Home	\$MW_HOME/OPatch
MySQL Home	\$MW_HOME/../mysql_home

1.8 Log Files Location

The following log files help you to keep a track record on all the events happening in your ODI instance on Oracle Cloud Marketplace:

Log Files	Location
ODI Image Creation Logs	/u01/oracle/logs/odi_install_config.log
ODI Configuration Logs	/u01/oracle/logs/odiConfigure.log
ODI Agent Logs	\$MW_HOME/app_logs/odiagent.log
ODI Rest Logs	\$MW_HOME/app_logs/odi_adp_rest_txt.log
ODI Studio Logs	\$MW_HOME/odi/log/studio.log

1.9 Patching

The ODI image on the Oracle Cloud Marketplace contains an Enterprise installation of ODI.

Patching is manual using OPatch.

You can upgrade your existing ODI image to the latest version available on the Oracle Marketplace directly, by either reusing your existing repository database or by creating a new repository database.

Note:

Before upgradation backup your work (db export), create a new instance and then restore (db import) it, to proceed with your upgradation process.

Patching an Existing ODI Marketplace Instance

To manually patch an existing ODI marketplace instance,

1. Log in to your instance using SSH.

```
$ ssh -i <private_key> opc @<public-ip-address>
```

2. Use the following command to change the user access to `root`

```
$ sudo su
```

3. Navigate to the location `/etc/security` and locate the file `limits.conf`, to modify the `ulimit` parameter using the following commands:

```
vi /etc/security/limits.conf
```

Add the following lines before the end of the file:

```
oracle soft nofile 8192
oracle hard nofile 8192
oracle soft nproc 4096
oracle hard nproc 8192
oracle soft core unlimited
oracle hard core unlimited
opc soft nofile 8192
opc hard nofile 13072
```

4. Save changes and exit the file.
5. Use the following command to change the user access to `oracle`

```
sudo su - oracle
```

6. Run `OPatch` to apply the patch.

```
$ opatch apply
```

To upgrade an existing repository,

Prerequisites

Navigate to the location `cd /u01/oracle/mwh/odi/common/scripts` and get the schema password and supervisor password using the following commands:

```
python manageCredentials.py read odiSchemaPassword
python manageCredentials.py read odiSupervisorPassword
```

For more details on running the Upgrade Assistant, refer to [Upgrading Product Schemas Using the Upgrade Assistant](#).

To upgrade your ADW Instance

Without SSL:

Follow the below procedure for running Upgrade Assistant on your ADW instance without SSL:

- Export `TNS_ADMIN=<wallet_extracted_path>`, for example -

```
/u01/oracle/mwh/wallets/wallet_DB20200310xxxx
```

- Run Upgrade Assistant for the repository using the following commands:

```
$cd mwh/oracle_common/upgrade/bin
$./ua
```

 **Note:**

For Database connect string, enter a short URL during upgrade process. For example - db20200310xxxx_low. You can get this short URL from the file `tnsnames.ora` available in the wallet.

With SSL:

Follow the below procedure for running Upgrade Assistant on your ADW instance with SSL:

- Export `TNS_ADMIN=<wallet_extracted_path>`, for example -

```
/u01/oracle/mwh/wallets/wallet_DB20200310xxxx
```

- Export `UA_PROPERTIES`, for example -

```
export UA_PROPERTIES="-Dua.SSL_DB_CONNECTIONS.enabled=true"
```

- Run Upgrade Assistant for the repository using the following commands:

```
$cd mwh/oracle_common/upgrade/bin
$./ua
```

 **Note:**

For Database connect string, enter a short URL during upgrade process. For example -

```
(description= (retry_count=20) (retry_delay=3)
(address=(protocol=tcps) (port=<port_number>) (host=<host_name>))
(connect_data=(service_name=<service_name>))
(security=<ssl_certificate>))
```

You can get this short URL from the file `tnsnames.ora` available in the wallet.

- Use `TNS_Connect` as Database connect string after running Upgrade Assistant.
- Click **SSL Settings** and modify the following parameters:
 - From the **TrustStore Type** drop-down menu, select **SSO**.
 - Beside to the **TrustStore Location** text box, click **Browse** to search and select the location of the `cwallet.sso` wallet file.
 - From the **KeyStore Type** drop-down menu, select **SSO**.
 - Beside to the **KeyStore Location** text box, click **Browse** to search and select the location of the `cwallet.sso` wallet file.

- Leave **TrustStore Password** and **KeyStore Password** fields blank.
- Click **Close**.

**Note:**

Your existing repository is upgraded automatically when you create a new ODI Marketplace instance.

To upgrade your MySQL Instance

Follow the below procedure for running Upgrade Assistant on your MySQL instance:

- Run Upgrade Assistant for the repository using the following commands:

```
$cd mwh/oracle_common/upgrade/bin  
$./ua
```

- Configure Database connect string parameter as follows:

```
Database URL - //localhost:3307/sys  
DBA user - public
```

**Note:**

Public password configured above is same as schema password.

2

Migrating from Oracle Data Integrator Cloud Services to Oracle Data Integrator Marketplace

This chapter describes how to migrate the Oracle Data Integrator repository of Oracle Data Integrator Cloud Service (ODI-CS) to Oracle Data Integrator on Marketplace at the database level, using Data Pump.

This chapter contains the following sections:

- [Exporting the ODI-CS Repository](#)
- [Provisioning DBaaS](#)
- [Provisioning ATP](#)
- [Provisioning ODI Marketplace](#)
- [Installing the required ODI version in ODI Marketplace node](#)
- [Creating an ODI Repository in DBaaS](#)
- [Importing the ODI Repository into the DBaaS node](#)
- [Verifying access to the DBaaS ODI Repository in ODI Studio](#)
- [Upgrading the Repository](#)
- [Exporting the Migrated Repository](#)
- [Importing the Migrated Repository into ATP](#)
- [Verifying access to the Migrated Repository in ODI Marketplace Studio](#)



Note:

Migration using data pump should only be done when the repository is very large. For smaller repositories, a standard export/import is enough. For more information on exporting/importing repositories, refer Repository-Level Export/Import section of *Developing Integration Projects with Oracle Data Integrator*.

- [Exporting the ODI-CS Repository](#)
- [Provisioning DBaaS](#)
- [Provisioning ATP](#)
- [Provisioning ODI Marketplace](#)
- [Installing the required ODI version in ODI Marketplace node](#)
- [Creating an ODI Repository in DBaaS](#)
- [Importing the ODI Repository into the DBaaS node](#)

- [Verifying access to the DBaaS ODI Repository in ODI Studio](#)
- [Upgrading the Repository](#)
- [Exporting the Migrated Repository](#)
- [Importing the Migrated Repository into ATP](#)
- [Verifying access to the Migrated Repository in ODI Marketplace Studio](#)

2.1 Exporting the ODI-CS Repository

You can export the ODI-CS repository using SQL Developer (Data Pump Wizard) or with the following OS-line command:

```
expdp SYSTEM/<PASSWORD>@<TNS_NAME> schemas=<SCHEMA_NAME> directory=
DATA_PUMP_DIR_PDB dumpfile=<DUMP_FILE_NAME > logfile=<LOG_FILE_NAME>
```

2.2 Provisioning DBaaS

Follow the provisioning Wizard for DBaaS System and Database as described in the below section:

<https://docs.cloud.oracle.com/en-us/iaas/Content/Database/Tasks/creatingDBsystem.htm>

The DBaaS version should be the same as the DBCS version used in ODI-CS.

2.3 Provisioning ATP

Refer to the tutorial, [Getting Started and Provisioning Autonomous Transaction Processing](#) for detailed steps on how to provision ATP.

2.4 Provisioning ODI Marketplace

For detailed information on provisioning ODI Marketplace, see [Launching Your Oracle Data Integrator Instance](#).

After provisioning, the standalone agent is available by default. It is possible to switch the standalone agent (that comes with ODI Marketplace) to point to the new repository. However, if a standalone collocated/weblogic agent is required, it has to be installed separately. For more information on setting up a standalone collocated agent, see [Creating a Standalone Collocated Agent](#).

2.5 Installing the required ODI version in ODI Marketplace node

The ODI version installed in the ODI Marketplace node must be the same as that used in ODI-CS.

Follow the below steps to install the required ODI version in the ODI Marketplace node:

1. Use SSH to connect to the ODI Marketplace node.

2. Create a new directory and download the required ODI version into this directory.
3. Unzip the files and run the ODI installer.
4. In the Welcome screen, click **Next**.
5. In the Auto Updates screen, select **Skip Auto Updates**. Click **Next**.
6. In the Installation Location screen, specify the Oracle Home using the directory you created for ODI. Click **Next**.
7. In the Installation Type screen, leave the default selected “Standalone Installation” as it is and click **Next**.
8. In the Prerequisite Checks screen, click **Next** when the prerequisite check is 100% complete.
9. In the Security Updates screen, uncheck the security updates option and click **Next**.
10. Click **Yes** in the warning dialog box. This is a temporary installation, so we do NOT need to connect to support.
11. In the Installation Summary screen, verify the installation information and click **Install**.
12. In the Installation Progress screen, you can see the progress of the process. This might take several minutes. Once all tasks are successfully finished, click **Finish**.

2.6 Creating an ODI Repository in DBaaS

You can create an ODI repository in DBaaS using the repository creation utility (rcu). The repository version must be the same as that used in ODI-CS.

Follow the below steps to create an ODI repository in DBaaS with rcu:

1. Navigate to the `$MW_HOME/oracle_common/bin` directory.
2. Execute rcu to create a new (empty) repository.
3. In the Welcome screen, click **Next**.
4. In the Create Repository screen, accept the default values and click **Next**.
5. In the Database Connection Details screen, provide your DBaaS connection information. Note that a user with DBA or SYSDBA privilege is required (for example, SYS).
6. Once all the pre-requisites are verified, click **OK**.
7. In the Select Components screen, leave all defaults and select **Oracle Data Integrator**. Click **Next**.
8. Once all the pre-requisites are verified, click **OK**.
9. In the Schema Passwords screen, enter a password and verify it. Please make a note of this password as you will need it later in the process. Click **Next**.
10. In the Custom Variables screen, enter a Supervisor password and Work Repository password.

The Supervisor password should be 6 - 12 characters in length and can contain special characters “#” or “_”. The Work Repository password should be 0 - 10 characters in length and can contain special characters “#” or “_”. Please make a note of these passwords as you will need them later in the process.
11. In the Map Tablespaces screen, click **Next**.
12. In the confirmation window, click **OK**.

13. Once tablespaces are created, click **OK**.
14. In the Summary screen, click **Create**.
15. A screen that will allow you to follow the progress of the different tasks will appear. Once all tasks are finished, a new screen will appear. The new screen will show the status of all tasks. Click **Close**.

2.7 Importing the ODI Repository into the DBaaS node

Follow the below steps to import the ODI repository into the DBaaS node:

1. Use SSH to connect to the DBaaS node into which you will import the ODI repository.
2. Execute the following `impdp` command:

```
impdp SYSTEM/<PASSWORD>@<DBaaS_system> schemas=DEV_ODI_REPO  
directory=DATA_PUMP_DIR_PDB dumpfile=<DUMP_FILE_NAME>  
logfile=<LOG_FILE_NAME> TABLE_EXISTS_ACTION=REPLACE
```

Please ensure that the `TABLE_EXISTS_ACTION` option is included in the command.

2.8 Verifying access to the DBaaS ODI Repository in ODI Studio

Follow the below steps in order to verify access to the DBaaS ODI repository in ODI Studio :

1. Navigate to the `$MW_HOME/odi/studio` directory.
2. Execute `odi.sh`.
3. In the Confirm Import Preferences screen, click **No**.
4. Click **Connect to Repository** available on the left panel.
5. On the Oracle Data Integrator Login screen, click the plus sign (+) icon to add the repository. The Repository Connection Information screen appears.
6. Specify the Oracle Data Integrator connection details as follows:
 - a. **Login Name:** Specify a custom login name. You can provide a name that will remind you of the purpose of this repository.
 - b. **User:** Specify `SUPERVISOR`.
 - c. **Password:** The password of your original ODI repository.
7. Specify the Database Connection (Master Repository) details as follows:
 - a. **User:** The schema owner (in the newly created repository).
 - b. **Password:** The password of the schema owner.
 - c. **Driver List:** Select **Oracle JDBC Driver** from the drop-down list.
 - d. **Driver Name:** This is automatically populated to 'oracle.jdbc.OracleDriver'.
 - e. **URL:** Specify the connection URL. The connection URL should be in the following format: `jdbc:oracle:thin:@<host>:<port>/ServiceName`.
8. Click **Test** to test the connection to the repository.

9. On receiving a successful connection message, click **OK**.
10. Click **OK** in the Repository Connection Information screen.
11. In the Oracle Data Integrator Login screen, click **OK**.
12. Navigate to the **Topology** tab on the left panel.
13. Expand the **Work Repositories** node and double-click **WORKREP**. You will receive an error message. Click **OK**.
14. To change the connection information to the DBaaS repository, click on the plug icon in the top left corner of the screen.
15. Provide the database user (DBaaS) for the repositories and its password. Then click on the **JDBC** link on the left pane.
16. Provide the connection URL to the DBaaS server and click **Test Connection**.
17. Click **Yes** in the confirmation window.
18. Click **Test** to test the connection to the repository. Please ensure that you do not change the agent specified here.
19. On receiving a successful connection message, click **OK**.

2.9 Upgrading the Repository

Follow the below steps to upgrade the repository:

1. Navigate to the `$MW_HOME/oracle_common/upgrade/bin` directory in the ODI Marketplace node.
2. Execute the upgrade tool.
3. In the Welcome screen, click **Next**.
4. In the Selected Schemas screen, select **Individually Selected Schemas**. Click **Next**.
5. In the Available Components screen, select **Oracle Data Integrator**. Click **Next**.
6. In the Prerequisites screen, select all the check boxes and click **Next**.
7. In the ODI Schema screen, provide the connection information to the DBaaS repository and click **Connect**. Please note that a user with DBA privileges is required in your DBaaS system.

If the connection is successful, the schema user will appear in the Schema User Name field. In this case, the schema user is `DEV_ODI_REPO`.
8. Provide the password for the schema user and click **Next**.
9. In the ODI Options screen, leave all the defaults and click **Next**.
10. In the ODI Supervisor screen, provide the Supervisor user and its password. Click **Next**.
11. The system will then examine the repository and determine if it is ready for upgrade. Once the examination is complete, click **Next**.
12. Verify the summary and click **Upgrade** in the Upgrade Summary screen.
13. You can view the status of the upgrade process in the Upgrade Progress screen. Once this is finished, click **Next**.
14. Click **Close** in the Upgrade Success screen.

2.10 Exporting the Migrated Repository

You can export the migrated repository using SQL Developer (Data Pump Wizard) or with the following OS-line command:

```
expdp SYSTEM/<PASSWORD>@<TNS_NAME> schemas=<SCHEMA_NAME> directory=
DATA_PUMP_DIR_PDB dumpfile=<DUMP_FILE_NAME > logfile=<LOG_FILE_NAME>
```

Please note that you need to use `DATA_PUMP_DIR_PDB` in the command, and not `DATA_PUMP_DIR`.

2.11 Importing the Migrated Repository into ATP

To import the migrated repository into ATP, do the following:

1. Upload the exported upgrade repository to Object Storage, so that it can be accessed from ATP.
2. Create credentials in your ATP system so that you can access your Object Storage:

```
BEGIN
  DBMS_CLOUD.PUT_OBJECT(credential_name => 'ODI_MIG',
    object_uri => 'https://objectstorage.us-
    phoenix-1.oraclecloud.com/n/<NAMESPACE>/b/<BUCKET_NAME>/o/<
    DUMP_FILE_NAME>',
    directory_name => 'DATA_PUMP_DIR',
    file_name => '<DUMP_FILE_NAME>');
END;
/
```

3. You can import the file using SQL Developer (Data Pump Wizard) or with the following OS-line command:

```
impdp ADMIN/<PASSWORD>@<TNS_NAME> directory=data_pump_dir
credential=ODI_MIG dumpfile=https://objectstorage.us-
phoenix-1.oraclecloud.com/n/orasematdpltinfgmt01/b/MAGU_Bucket/o/
MIG_ODI_REPO.dmp parallel=16 partition_options=merge
table_exists_action=REPLACE transform=segment_attributes:n
transform=dwcs_cvt_iots:y transform=constraint_use_default_index:y
exclude=index,cluster,indextype,materialized_view,materialized_view_lo
g,materialized_zonemap,db_link
REMAP_SCHEMA=<SOURCE_SCHEMA>:<TARGET_SCHEMA>
```

Note:

This procedure is only applicable for ATP-D.

2.12 Verifying access to the Migrated Repository in ODI Marketplace Studio

Follow the below steps to verify access to the migrated repository in ODI Marketplace Studio:

1. Once the repository has been successfully imported, start ODI Studio in your new ODI Marketplace node.
2. Click **Connect to Repository** available on the left panel. The Oracle Data Integrator Login screen appears.
3. To modify the repository connection information, click on the pencil icon. The Repository Connection Information screen appears.
4. Modify the repository connection information as follows:
 - a. **Password** (for `SUPERVISOR`): Change the Supervisor password to the one you had in your original ODI repository in the Oracle Data Integrator Cloud Service.
 - b. **Use Credential File**: Select the check box.
 - c. **Credential File**: Browse and select the wallet file in `/home/opc` directory.
 - d. **Connection Details**: Select your low connection.
 - e. **Work Repository**: Select **Master Repository Only**.
5. Click **Test Connection**.
6. On receiving a successful connection message, click **OK**.
7. Click **OK** in the Repository Connection Information screen.
8. In the Oracle Data Integrator Login screen, click **OK**.
9. Navigate to the **Topology** tab on the left panel.
10. Open the **Repositories** panel.
11. Expand the **Work Repositories** node and double-click **WORKREP**.
12. To change the connection information to the ADB repository, click the **Connection Information** button on the top left corner of the screen.
13. Provide the database user (ADB) for the repositories and its password. Then click the **JDBC** link on the left pane.
14. Provide the connection URL to the ADB server in the JDBC URL field and click **Test Connection**.
15. In the confirmation window, click **Yes**.
16. Click **Test**. Ensure that you do not change the Physical Agent specified here.
17. On receiving a successful connection message, click **OK**.
18. Disconnect from the repository by clicking on the unplug icon.
19. Click **Connect to Repository** available on the left panel. The Oracle Data Integrator Login screen appears.
20. To repoint the work repository, click on the pencil icon. The Repository Connection Information screen appears.

21. Select the **Work Repository** radio button at the bottom of the screen. Then click on the magnifying glass on the right side.
22. Select the work repository that is displayed in the Select Repository screen and click **OK**.
23. Click **OK** in the Repository Connection Information screen. You are now in your repository in ODI Marketplace.

A

Creating a Standalone Collocated Agent

This appendix describes how to create a standalone collocated agent.

Setting up a standalone collocated agent involves the following steps:

- [Installing the Agent](#)
- [Defining the Agent in ODI](#)
- [Starting the Agent](#)
- [Installing the Agent](#)
- [Defining the Agent in ODI](#)
- [Starting the Agent](#)

A.1 Installing the Agent

Follow the below steps to install the standalone collocated agent:

1. Verify that the existing standalone agent is down by running the following command:

```
sudo systemctl status -l agentodi.service
```

If the service is still active (up and running), use the following command to stop it:

```
sudo systemctl stop agentodi.service
```

2. Navigate to the following directory:

```
$MW_HOME/oracle_common/common/bin
```

3. Start the Configuration Wizard by running the following command:

```
./config.sh
```

4. In the Create Domain screen, select **Create a new domain** and specify `$MW_HOME/user_projects/domains/odisac_domain` in the Domain Location field. Click **Next**.
5. In the Templates screen, select **Oracle Data Integrator – Standalone Collocated Agent [odi]**. This will automatically select other options. Click **Next**.
6. In the Administrator Account screen, provide the following information and click **Next**:
 - a. **Name**: By default, **weblogic** is specified.
 - b. **Password**: Specify a password for the weblogic user.
 - c. **Confirm Password**: Retype the password.
7. In the Domain Mode and JDK screen, select **Production** in the Domain Mode field and verify that you are pointing to the right JDK path `"/u01/oracle/jdk1.8.0_191"`. Click **Next**.
8. In the Database Configuration Type screen, provide the following information:
 - a. **Host Name**: Name of the server where the repository database is stored.
 - b. **DBMS/Service**: Service name of the database where the repository is stored.

- c. **Port:** Port where the database where the repository is stored is running (1521).
- d. **Schema Owner:** Login owner of all ODI schemas. Leave the default as DEV_STB.
- e. **Schema password:** Password for ODI schemas provided in previous steps.
9. Click **Get RCU Configuration**. This will verify the connectivity to the database.
10. Once connectivity has been successfully verified, click **Next**.
11. In the Component Datasources screen, click **Next**.
12. In the JDBC Component Schema Test screen, you can see the progress of the status verifications in all components. Once all have green check marks, click **Next**.
13. In the Advanced Configuration screen, select **Node Manager** and **System Components**. Click **Next**.
14. In the Node Manager screen, provide the following information and click **Next**:
 - a. **Username:** Specify **weblogic**.
 - b. **Password:** Provide the password for the weblogic user.
 - c. **Confirm Password:** Confirm the password.
15. In the System Components screen, accept the defaults and click **Next**.
16. In the ODI Server Configuration screen, provide the Supervisor user password. Leave all other default values and click **Next**.
17. In the Machines screen, select the **Unix Machine** tab. Click **Add**. Leave all defaults and click **Next**.
18. In the Assign System Components to Machines screen, leave all default values and click **Next**.
19. In the Configuration Summary screen, you can review all your choices. Click **Create**.
20. As the tasks progress, a green check will appear. This means that the tasks finished successfully. Once it finishes, click **Next**.
21. In the End of Configuration screen, click **Finish**.

A.2 Defining the Agent in ODI

Follow the below procedure to define the agent:

1. Log into ODI Studio.
2. On the **Topology** tab, expand **Physical Architecture**. Right-click **Agents** and select **New Agent**.
3. On the **Agent** panel, on the **Definition** tab, in **Name**, enter the name for the new agent. Save the configuration.
4. On the **Topology** tab, expand **Logical Architecture**. Right-click **Agents** and select **New Logical Agent**.
5. On the **Logical ODI Agent** panel, on the **Definition** tab, in **Name**, enter the name for the new agent.
6. Select **Global** as the context. Save the configuration.

A.3 Starting the Agent

Follow the below procedure to start the standalone collocated agent:

1. Navigate to the following directory:

```
$MW_HOME/user_projects/domains/odisac_domain/bin
```

2. Start the agent by executing the following command:

```
nohup ./agent.sh -NAME=OracleDIAgent1 &
```

3. Execute the `tail nohup.out` command.
4. Start ODI Studio by running `odi.sh` available in the `$MW_HOME/odi/studio/` directory.
5. In the **Topology** tab, open the required agent and click **Test**.
6. On receiving a successful test message, click **OK**.

B

Importing the ODI Repository dump

This appendix describes how to import the ODI repository dump.



Note:

Please contact your database administrator to perform the following procedure for importing the ODI repository.

Follow the below steps to import the ODI repository dump:

1. Start SQL Developer.
2. Define a connection as `SYS` to the DBaaS system that has the new ODI repository and connect. Please ensure that you connect to the pluggable database and NOT to the container database.
3. Create the `DATA_PUMP_DIR_PDB` directory, and point it to a directory in the DBaaS system into which you will import your ODI-CS repository dump file.
4. Define a connection as `SYSTEM` to the DBaaS system that has the new ODI repository and connect. Using this connection, open the **DBA** panel.
5. Expand the **Data Pump** node, right-click **Import Jobs** and select **Data Pump Import Wizard**. This opens the Import Wizard.
6. In the Type screen, provide the following information:
 - a. **Data or DDL**: Select **Data Only** from the drop-down list.
 - b. **Type of import**: Select **Schemas**.
 - c. **Credentials or Directories**: Specify `DATA_PUMP_DIR_PDB`.
 - d. **File Names or URI**: Enter the name of the dump file.
7. Click **Next**.
8. In the Filter screen, select **ODI_REPO** from the Available source schemas window and click the right arrow to move it to the Selected source schemas window. Click **Next**.
9. In the **Re-Map Schemas** section of the Remapping screen, provide the mapping from the `ODI_REPO` schema to import to the schema in your DBaaS. Click **Next**. If all defaults were taken during repository installation, the name will be `DEV_ODI_REPO`.
10. In the Options screen, fill in the details according to the way you performed the export. Click **Next**.
11. In the Schedule screen, leave all the defaults. Click **Next**.
12. In the Summary screen, click **Finish** to start the import process.



Note:

This procedure is only applicable for ATP-D.