Practice: Create and Manage Autonomous Database

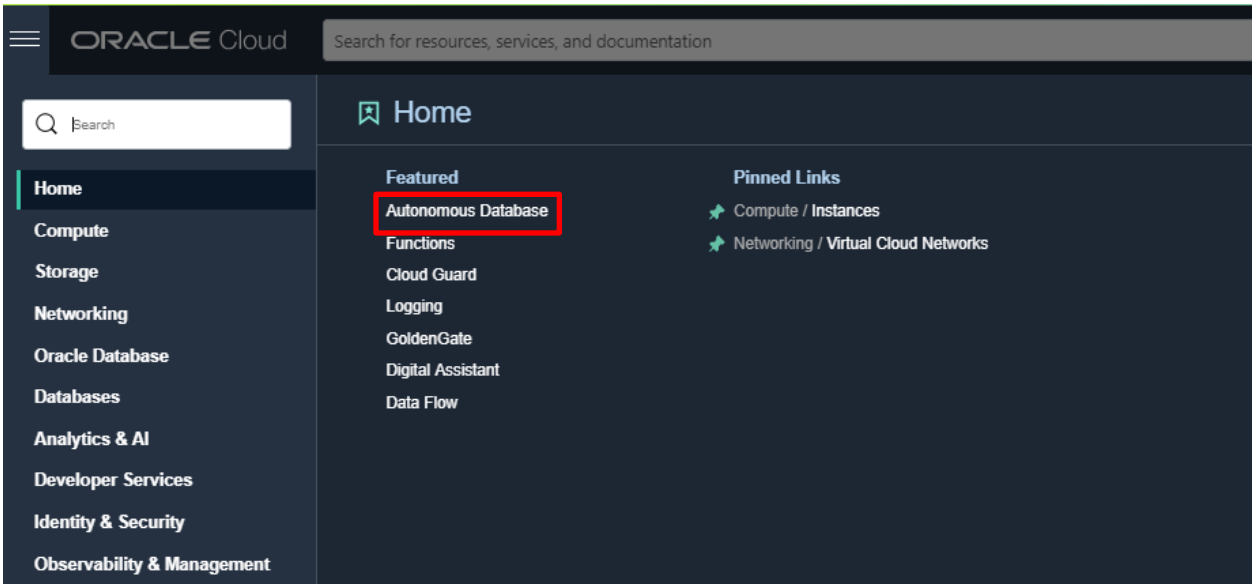
Try this hands-on lab with the Oracle Cloud 30 days Free Trial account or your own tenancy. If you do not have a free account, click here to get one.

Overview

In this practice, you create an Autonomous Database.

Tasks

1. Log in to your Oracle Cloud Free Tier Account.
2. At this point, you should be logged in to **Oracle Cloud Infrastructure (OCI) Dashboard**, also called as the OCI web console.
3. Click **Menu** ☰ on the top-left corner and explore the available services option. You will use this navigation path throughout the practice.
4. Expand the options available under **Navigation Menu**.



5. Click **Autonomous Database.**
6. Select the ***Compartment.***
7. Click **Create Autonomous Database** to start creating your ATP instance.
8. Select or fill in the following values in the Create Autonomous Database wizard:
   a. Compartment: **Ensure the Compartment assigned to you is selected**
   b. Display Name: ***MYATP***
   c. Database Name: ***MYORCL***
   d. Workload Type: **Transaction Processing**

e. Deployment Type: **Shared Infrastructure**

f. Ensure the "*Show only Always Free configuration options*" option is **turned on**.

g. Database Version: **Select the latest Version available**

h. CPU Core Count: **1**

i. Storage (TB): **0.02**

j. Password: *Set the password for your Autonomous Database **ADMIN** user*

The password must meet the strong password complexity criteria based on Oracle Cloud security standards. Example for an Admin password: "***Wwelcome#123***".

Important: Do not use the special characters double quote (**"**), '**@**' and '**!**' as part of your **admin** user password; it is known to cause issues while working with SQL*Plus.

k. Confirm password by reentering the same admin password.

l. Network Access: **Secure access from everywhere**

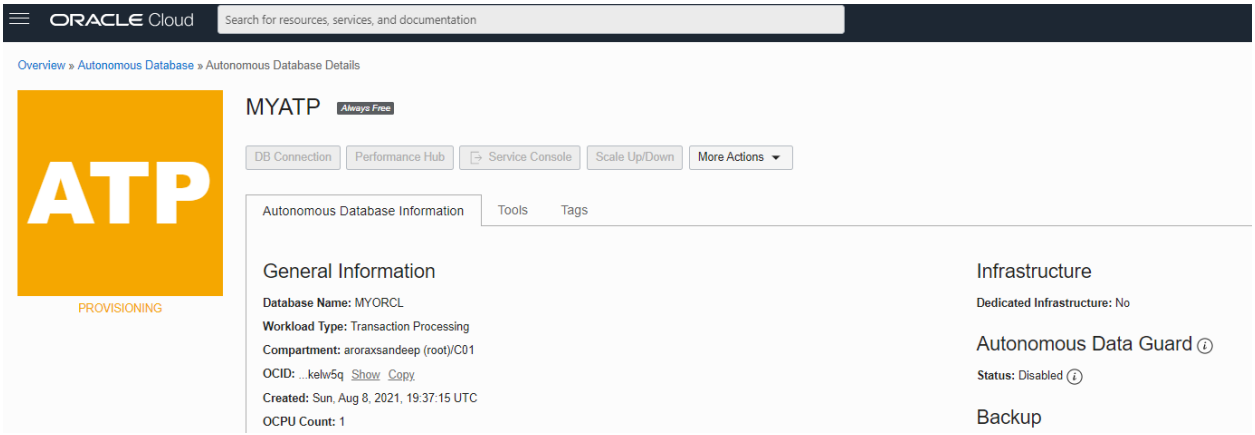
m. License Type: **License Included**

n. Advance Options: Ignore this section (accept the default)

9. After you have filled the details, click **Create Autonomous Database.**

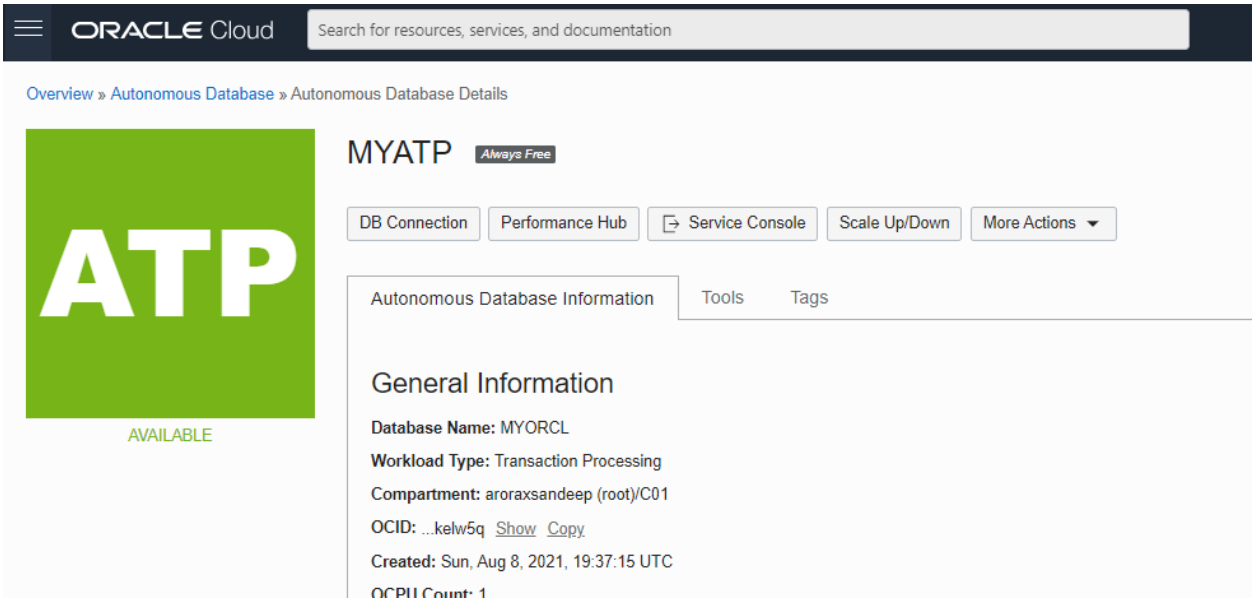
Note

- Initially, the status of the service instance will read "Provisioning."
- Generally, provisioning takes anywhere between 5 and 10 minutes to complete depending on resource availability. **Allow sufficient time for this to complete**.



10. Refresh the Instances page after a while to see if your instance is created.

11. Check if the status for your instance is "**Available**," indicating your instance is ready to use.



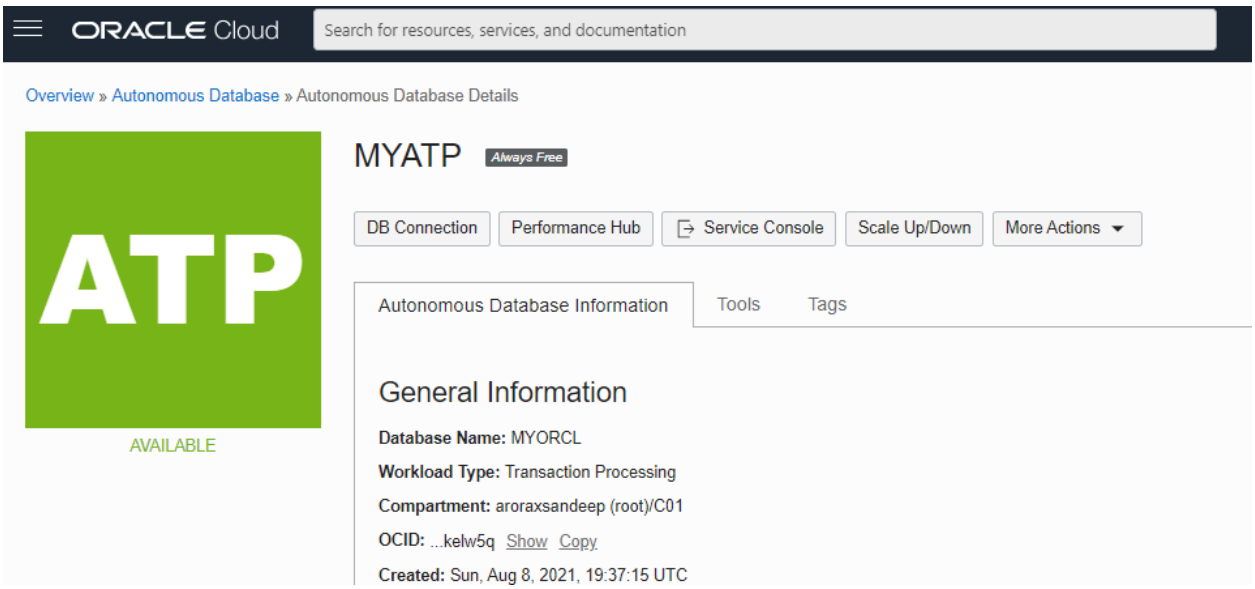
Practice: Connect to Autonomous Database Using SQL Developer

Overview

In this practice, you connect to the Autonomous Database instance using SQL Developer.

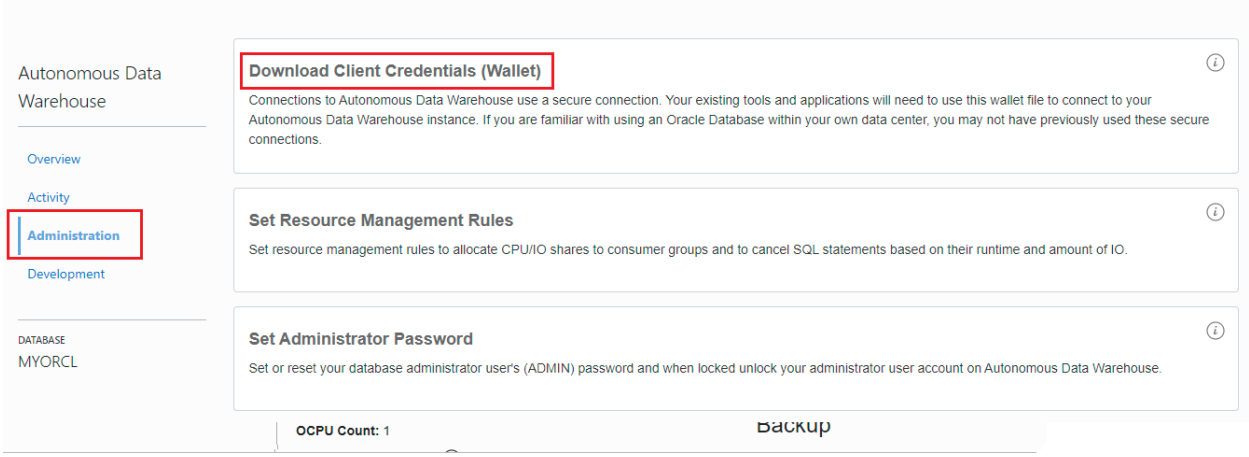
Tasks

1. Log in to your Oracle Cloud Free Tier Account.
2. In the menu, click **Autonomous Database**.
3. On the Instances page, find your ATP instance (e.g., **MYATP**) and click **Service Console** in the Actions menu.



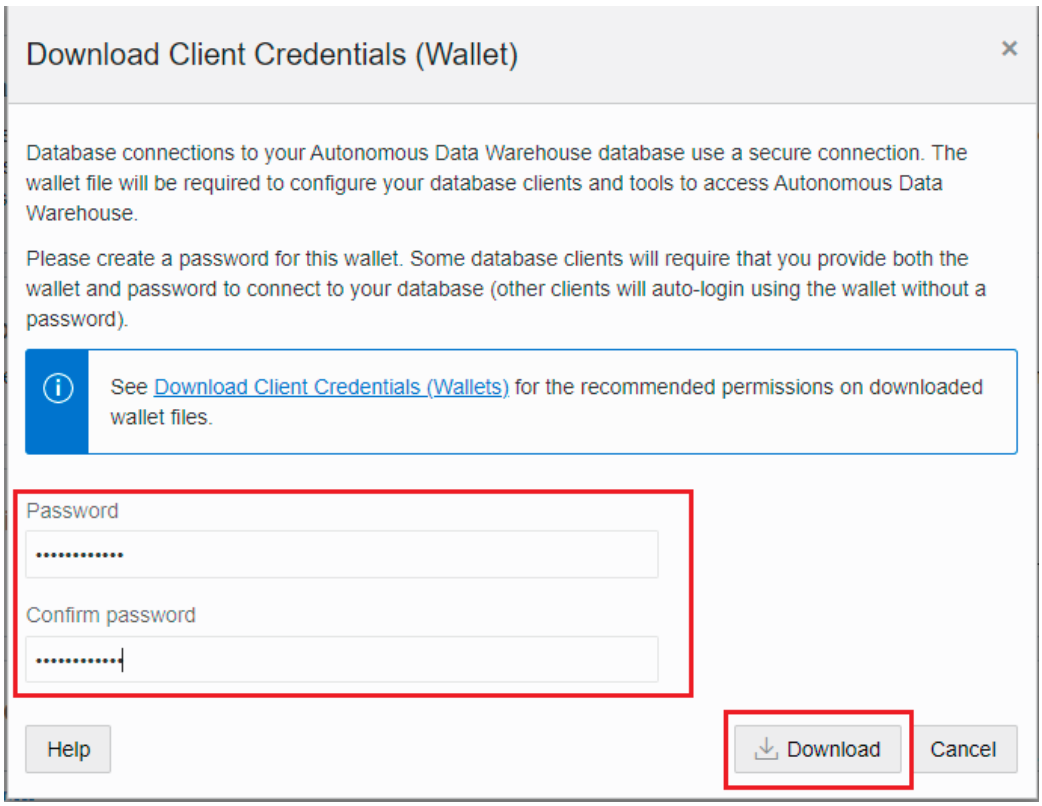
4. This will open a new browser tab for the service console. If prompted, **sign in** to the service console as **admin** user.

5. Click the Administration tab and click Download Client Credentials (Wallet) to download the wallet.



Note: Specify a password of your choice for the wallet. You will need this password when connecting to the database later. Note that this password is separate from the **admin password**. You will need this password as your **keystore password** when connecting JDBC thin applications.

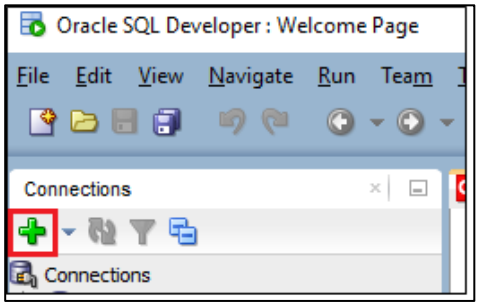
6. Click **Download** to download the wallet file to your client machine.



7. Connect to the database using SQL Developer installed on your local system.

   Start SQL Developer and create a connection for your database using the default administrator account, **ADMIN**, by following these steps:

   a. Click the **Create Connection** icon in the Connections toolbox on the top left of the SQL Developer home page.



b. Enter the connection details as follows:
   - Connection Name**: admin_low**
   - Username**: admin**
   - Password**: *The password you specified during instance creation***
   - Connection Type: **Cloud Wallet**
   - Role: **default**
   - Configuration File: **Full path for the Client Credentials wallet file** you downloaded earlier
   - Keystore Password: *The password you specified when downloading the wallet from the ATP service console*.
     Note: Required only if the SQL Developer version is less than 18.2
   - Service: Pick **myatp_low** for this list (select the name corresponding to your instance). There are three preconfigured database services for each database.
   - *Accept default for other fields if any*

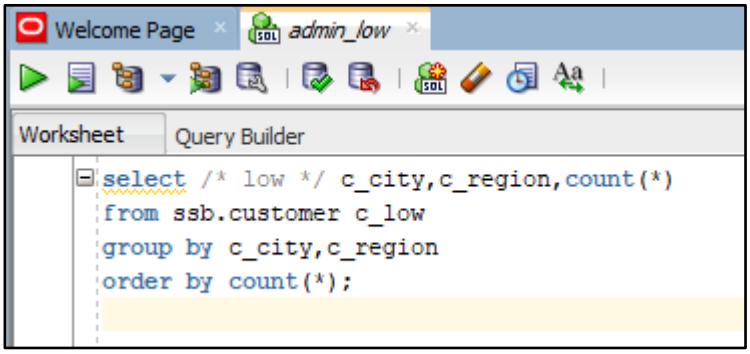
c. Test your connection by clicking the **Test** button. If it succeeds, save your connection information by clicking **Save**.

d. Connect to your database by clicking the **Connect** button.

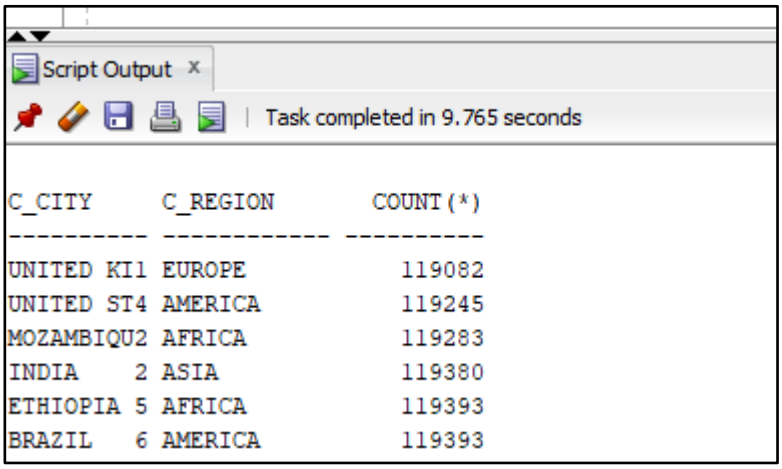
e. In SQL Developer, run this query with the **admin_low** connection you created just now and note the response time.

```
select /* low */ c_city,c_region,count(*)
from ssb.customer c_low
group by c_city,c_region
order by count(*);
```

f. Make sure you click the **Run Script** button in SQL Developer so that all the rows are displayed on the screen.



g. Note the response time.



Note: In this example, the query with the **LOW** database service finished in around **9.7 seconds**.

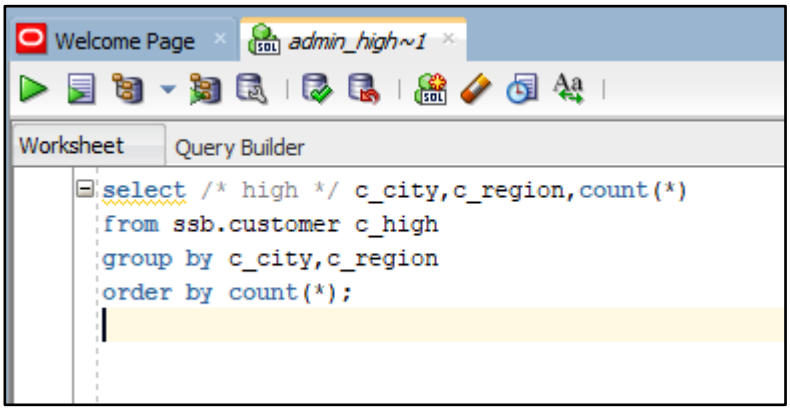
8. Create another connection named **admin_high** using the same information as in **step 2.** This time pick **myatp_high** as the service name.

   a. In SQL Developer, run this query with the **admin_high** connection you created just now and note the response time.

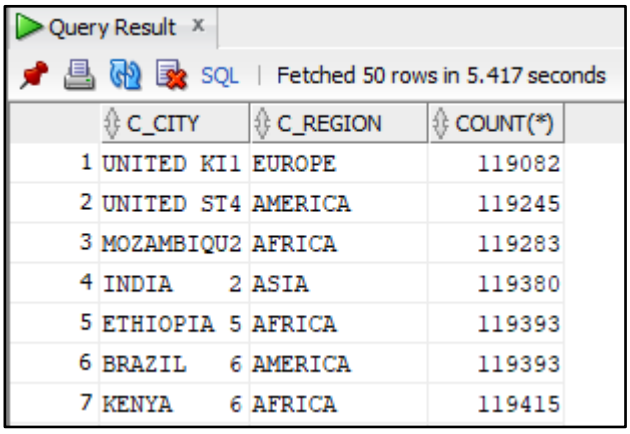
```
select /* high */ c_city,c_region,count(*)
from ssb.customer c_high
group by c_city,c_region
order by count(*);
```

   Note: These two queries are basically the same except for the comment after the select keyword and the table alias.

   b. Make sure you click the **Run Script** button in SQL Developer so that all the rows are displayed on the screen.



c. Note the response time.



	C_CITY	C_REGION	COUNT(*)
1	UNITED KI1	EUROPE	119082
2	UNITED ST4	AMERICA	119245
3	MOZAMBIQU2	AFRICA	119283
4	INDIA 2	ASIA	119380
5	ETHIOPIA 5	AFRICA	119393
6	BRAZIL 6	AMERICA	119393
7	KENYA 6	AFRICA	119415

Note: In the following example, the query with the **HIGH** database service finished in around **5.4 seconds**. A query running in the HIGH database service can use more resources and run faster compared to a query running in the LOW database service.

Practice: Load a Local File into Autonomous Database

Overview

In this practice, you load data from a local file on your machine into ADB.

Tasks

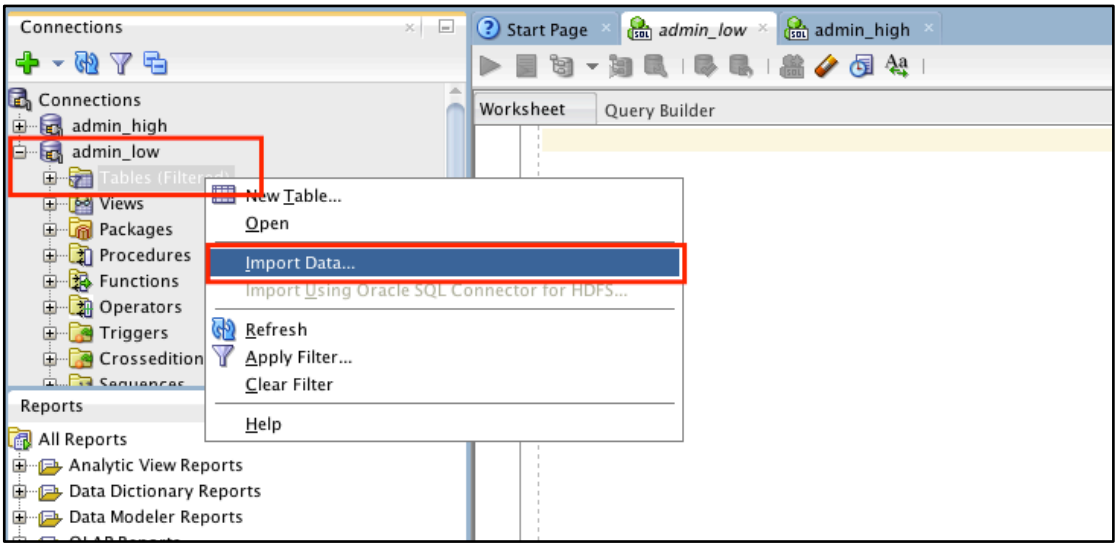
1. Download and save the **channels.csv** lab file on your local machine. This file is available for download. It is recommended that you use this file to ensure you do not have any formatting issues.

   As an alternate option, you can create a new CSV (Comma-Separated Values) file on your local machine and copy and paste the following data into the file. Finally, ensure you save the file as a CSV file with the name **channels.csv** on **your local machine**.

   Note: You can save this file in any location on your local machine.

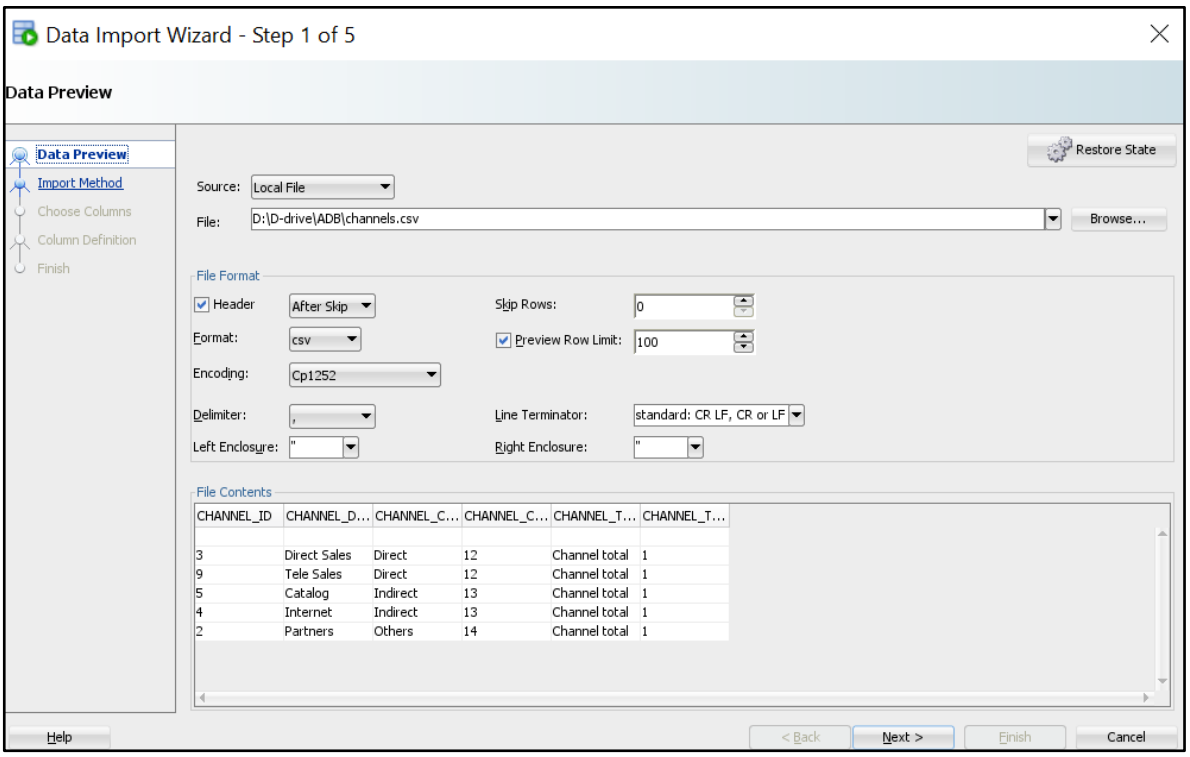
CHANNEL_ID	CHANNEL_DESC	CHANNEL_CLASS	CHANNEL_CLASS_ID	CHANNEL_TOTAL	CHANNEL_TOTAL_ID
3	Direct Sales	Direct	12	Channel total	1
9	Tele Sales	Direct	12	Channel total	1
5	Catalog	Indirect	13	Channel total	1
4	Internet	Indirect	13	Channel total	1
2	Partners	Others	14	Channel total	1

2. Go to SQL Developer and expand your `admin_low` connection for the ADW instance.
3. Right-click **Tables** and then click **Import Data.**

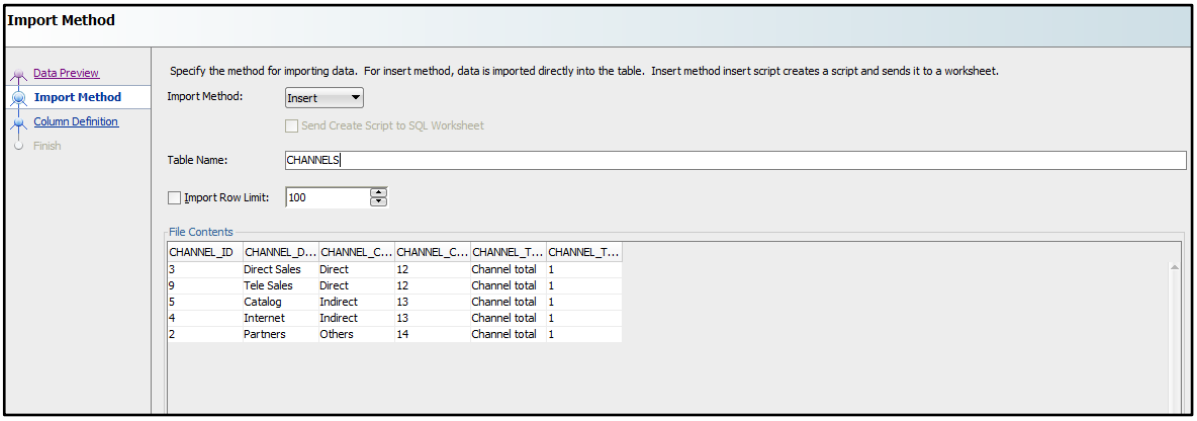


4. This will open the data import wizard. Click **Browse** and locate the **channels.csv** file you created.

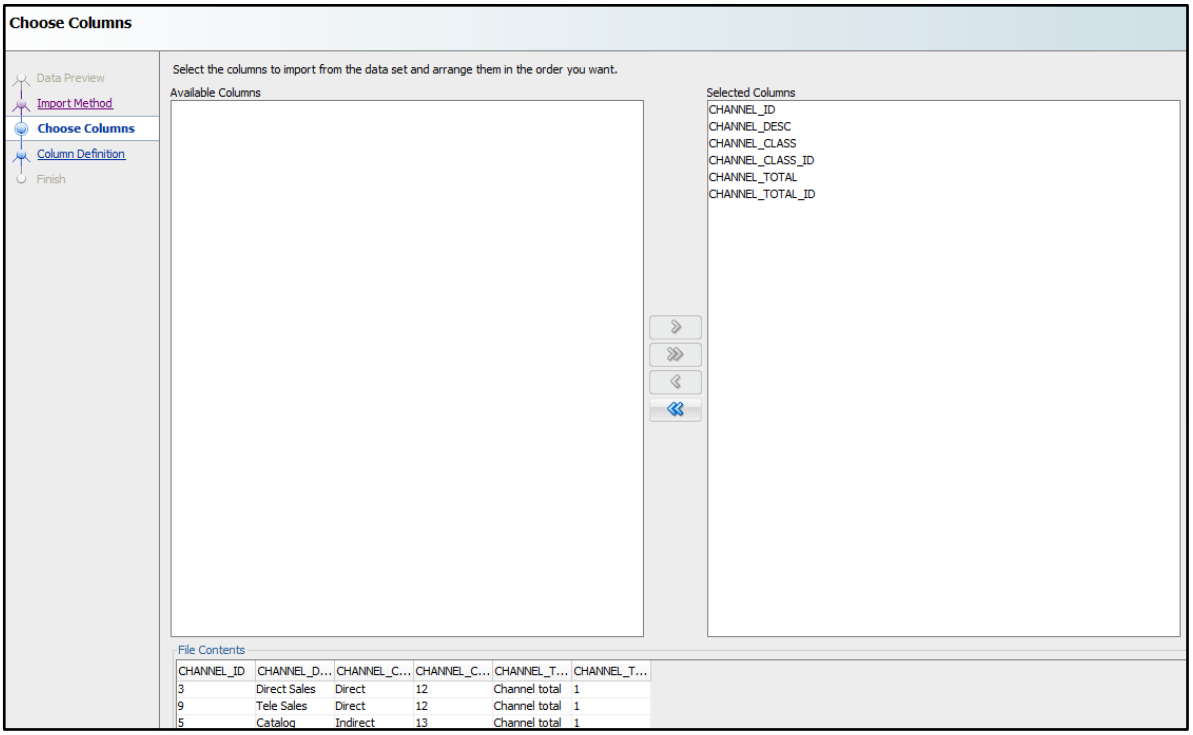
5. When you select the file, you will see the file contents in the import wizard.



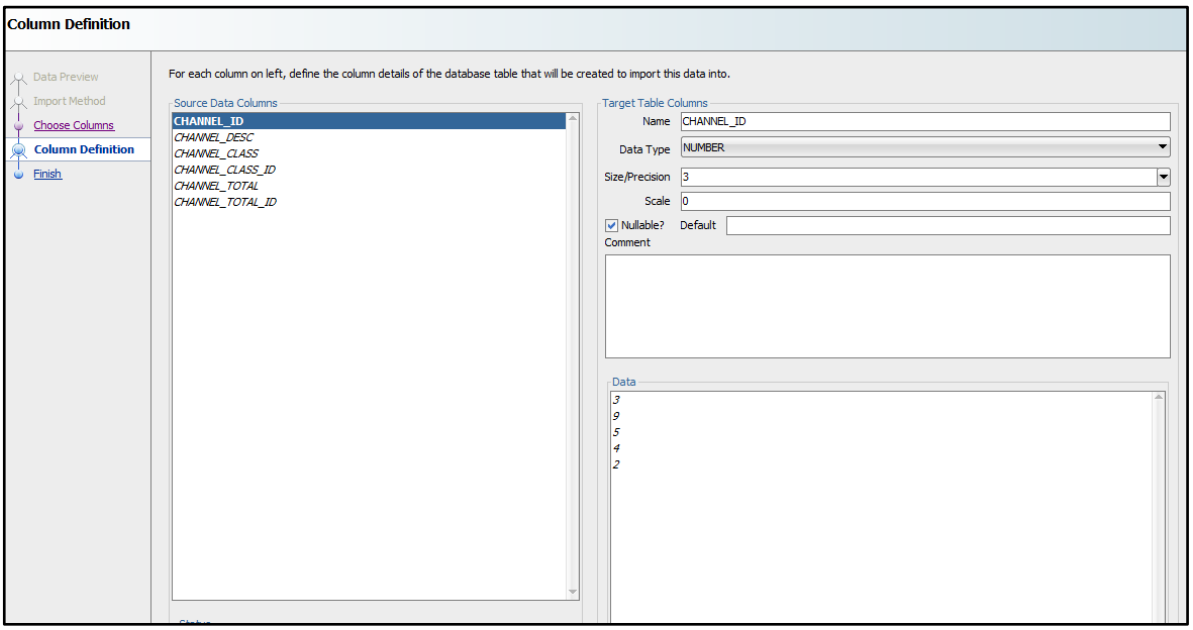
6. Click **Next.** In the next screen, enter CHANNELS as the table name you will create and load into.



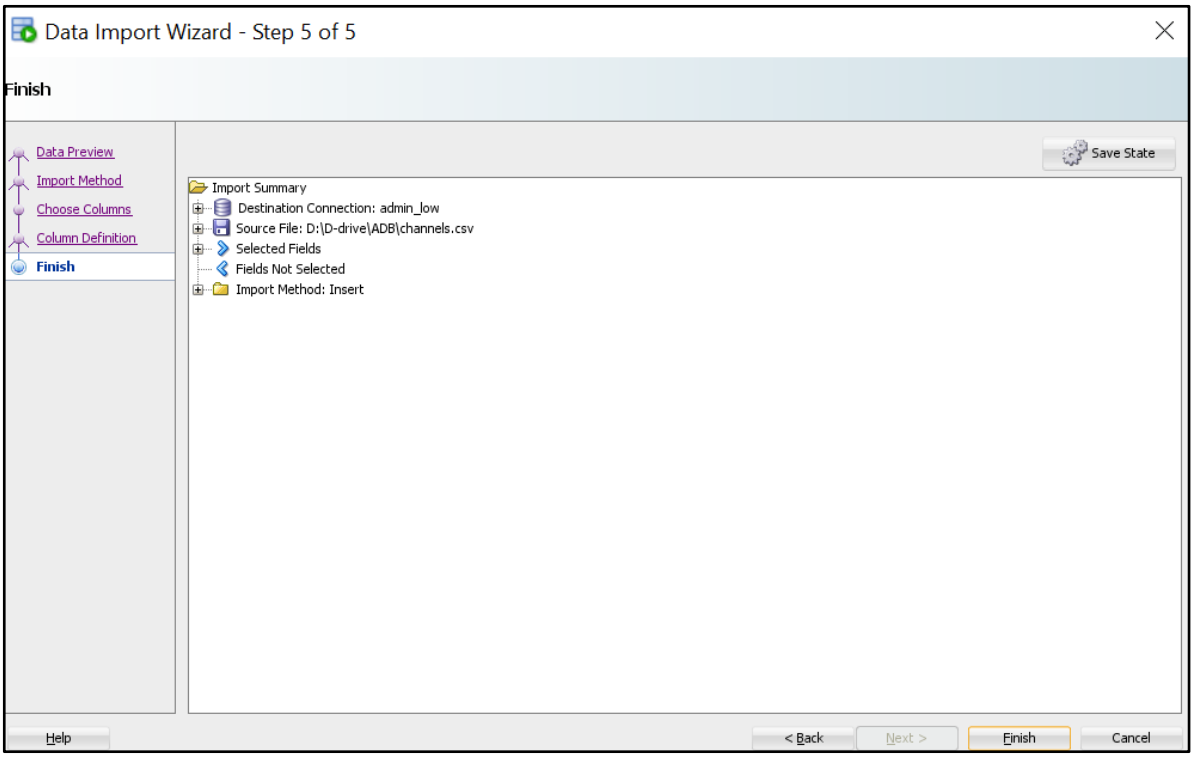
7. Click **Next.** The next screen allows you to select the columns you want for this table. For this practice, leave the columns as is, which means the table will have all columns available.



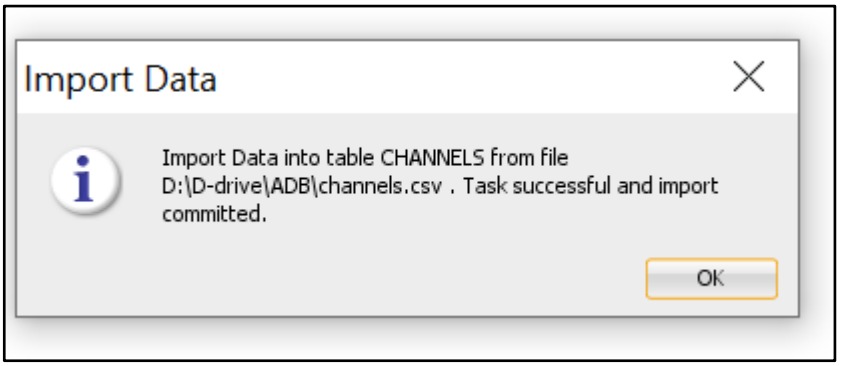
8. Click **Next**. The next screen allows you to look at the data types for each column. You can change the data types if you need to. For this practice, leave the data types as default.



9. Click **Next.** The next page will display a summary for the import operation.

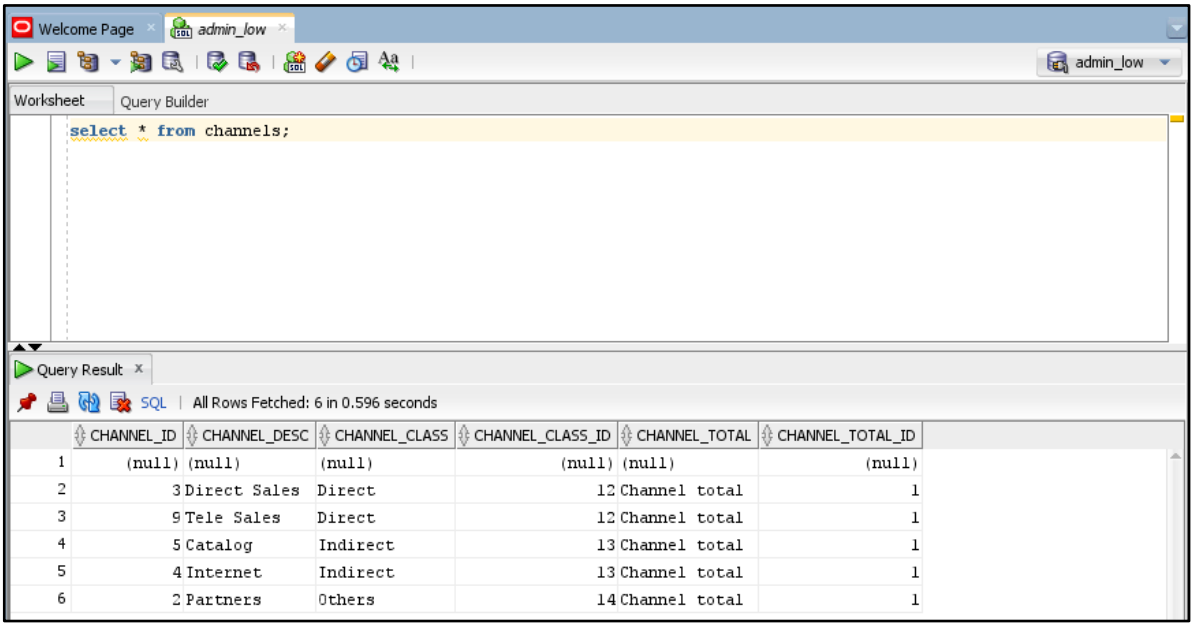


10. Click **Finish** to complete the import wizard and start the data load. When the data load finishes, you will see a message saying the import was completed.



11. Your source file is now loaded into ADB. You can run a query on the table to see your data.

```
select * from channels;
```



	CHANNEL_ID	CHANNEL_DESC	CHANNEL_CLASS	CHANNEL_CLASS_ID	CHANNEL_TOTAL	CHANNEL_TOTAL_ID
1	(null)	(null)	(null)	(null)	(null)	(null)
2	3	Direct Sales	Direct	12	Channel total	1
3	9	Tele Sales	Direct	12	Channel total	1
4	5	Catalog	Indirect	13	Channel total	1
5	4	Internet	Indirect	13	Channel total	1
6	2	Partners	Others	14	Channel total	1

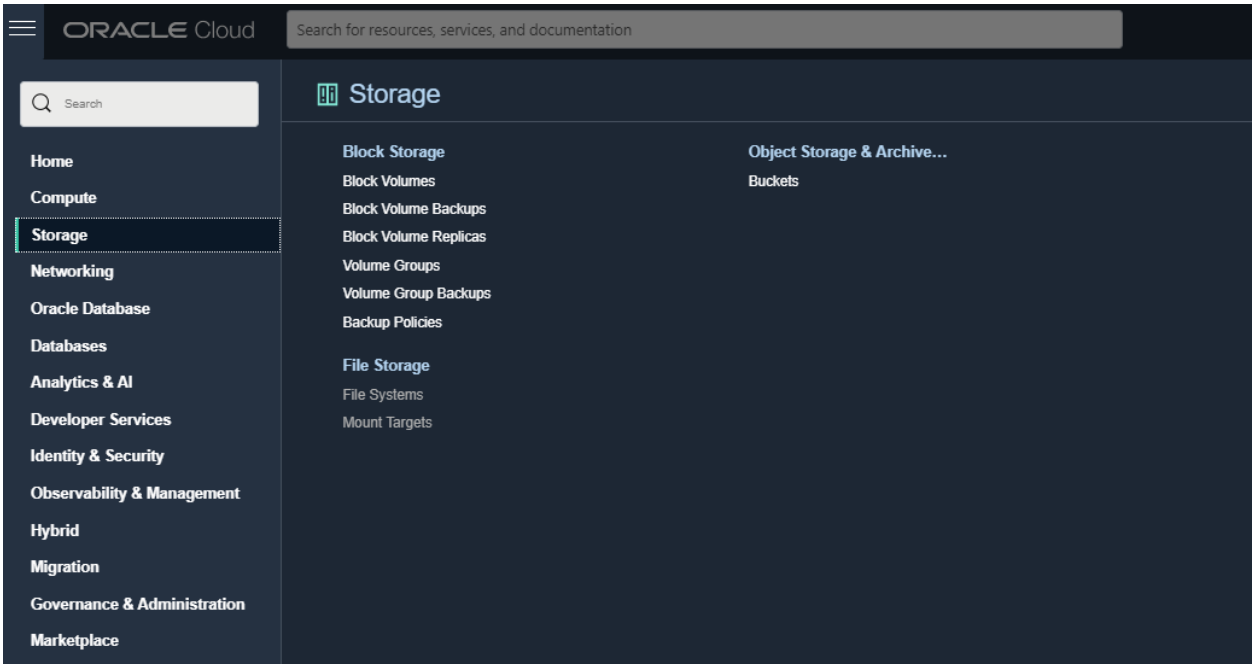
Practice: Upload Data Files to Your Object Store

Overview

In this practice, you connect to the ADB instance and upload various data sets into Object Store.

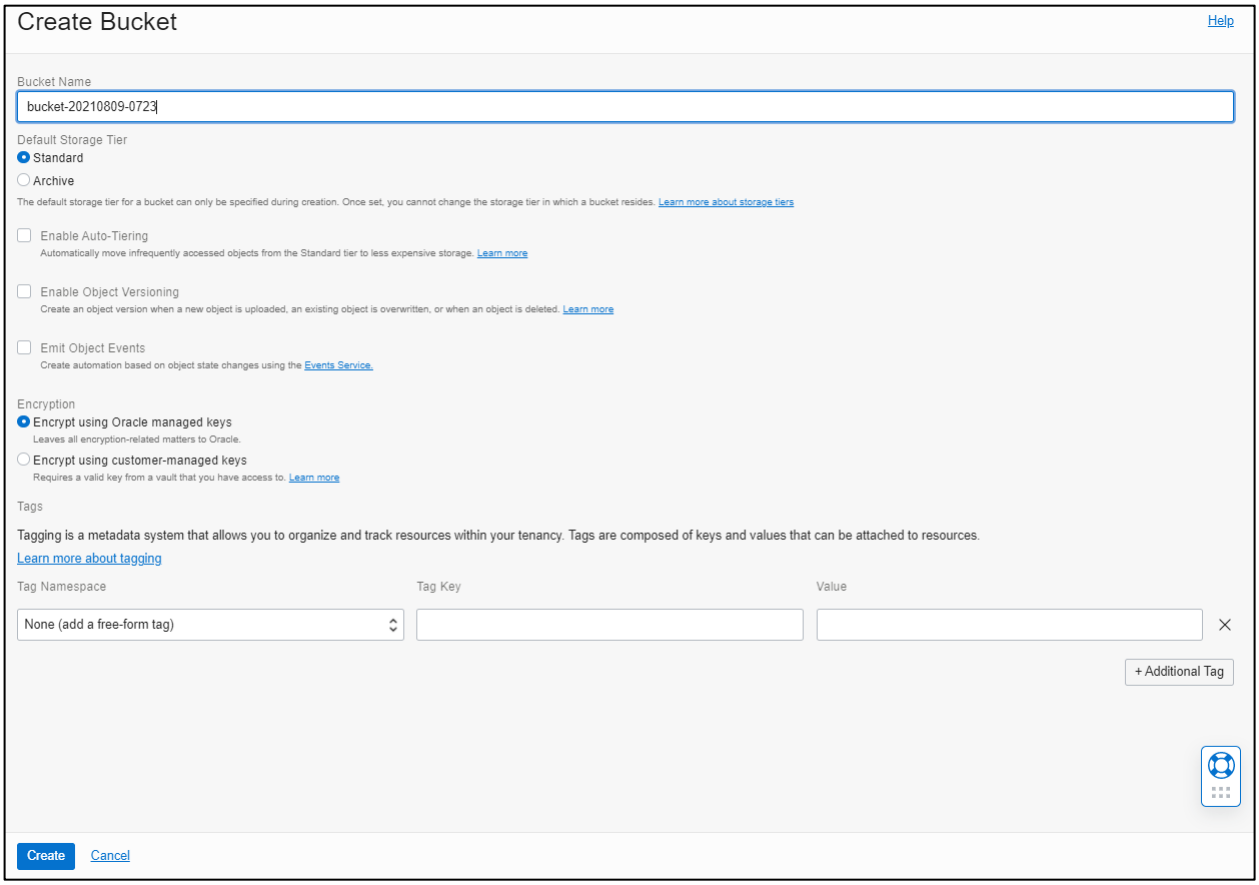
Tasks

1. Log in to your Oracle Cloud Free Tier Account.
2. Click the **Menu** option and select **Storage**.

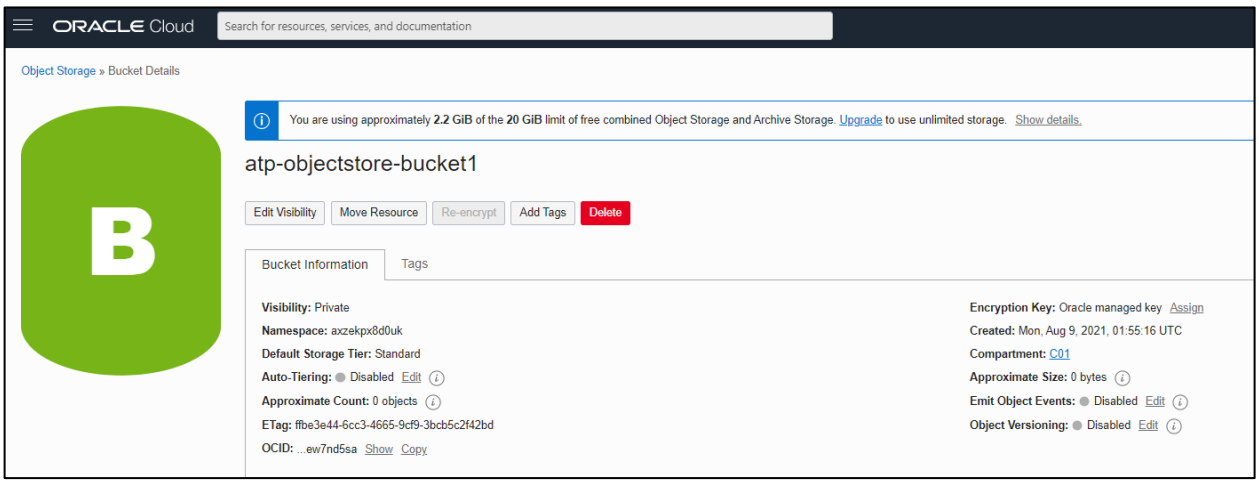


3. In the Object Storage menu, click **Buckets**.

4. Click **Create Bucket**, **provide a meaningful name to your bucket** (remember this name as you will load data into this), keep the storage type as **STANDARD,** and click **Create Bucket**.

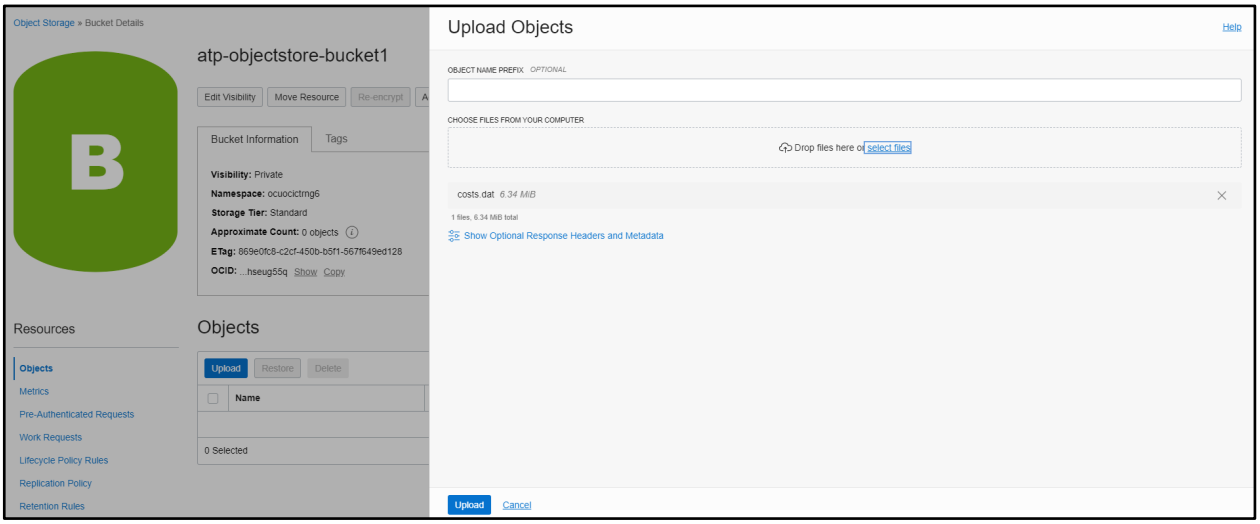


5. After the bucket is created, open the bucket by **clicking the bucket name** and start adding objects into the bucket. To do so, click **Upload**.



6. Click **Browse**, select a **.dat** file, and click to upload the file one after the other. Alternatively, you can also drag a **.dat** file one after the other from lab files location **\DAT_Files** on to the **Drop a file here** pane of the Upload Object window and upload them. You should upload all the 10 files.

   Important: You must ensure the files are uploaded one after the other. Do not try to drag and drop all the files at once.



7. This completes the task of uploading external data files on to Object Storage.

Practice: Create an Object Store Auth Token

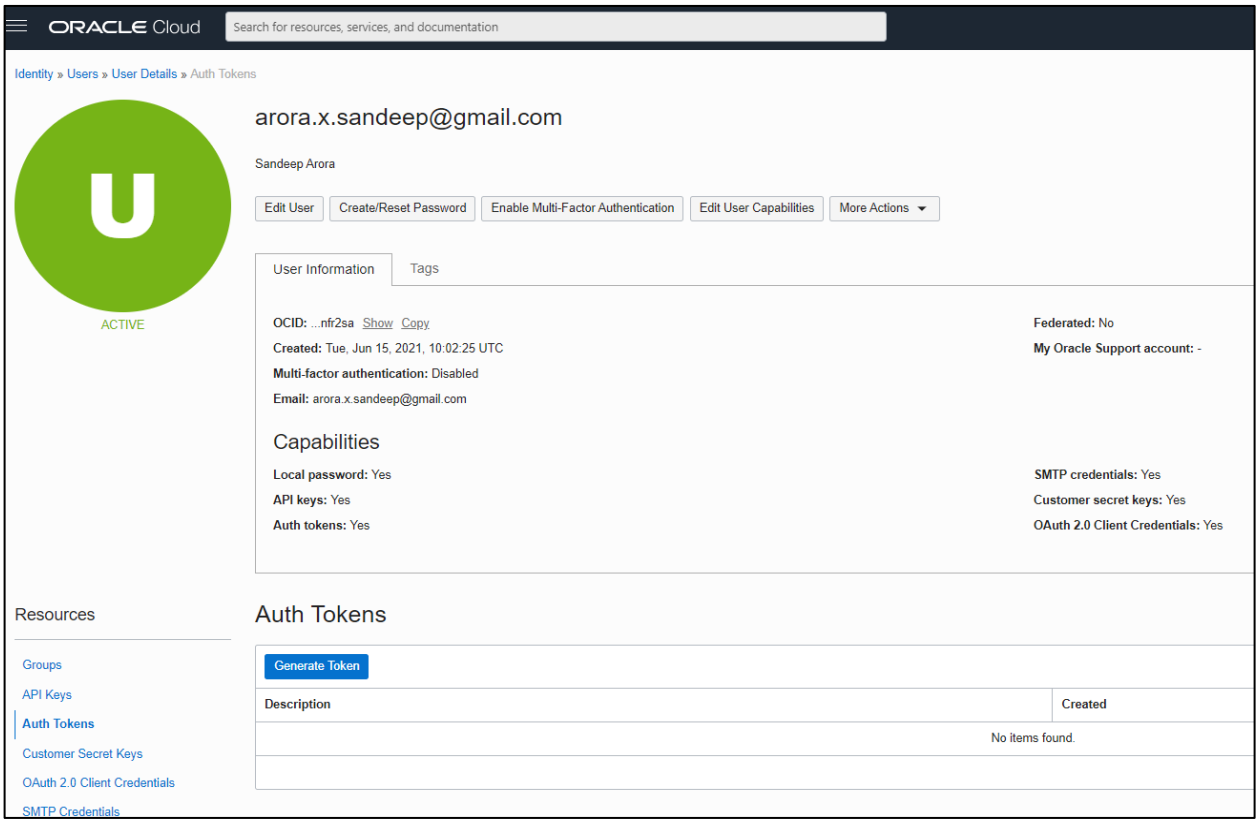
Overview

In this practice, you generate an Auth Token, and this token will be used while creating the credentials to access Object Storage objects.

Tasks

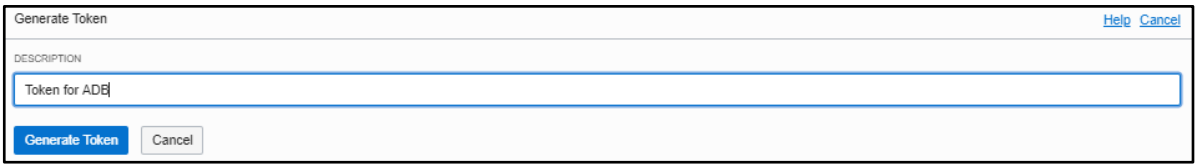
1. Click **User Settings** on the top menu where the username is being displayed.

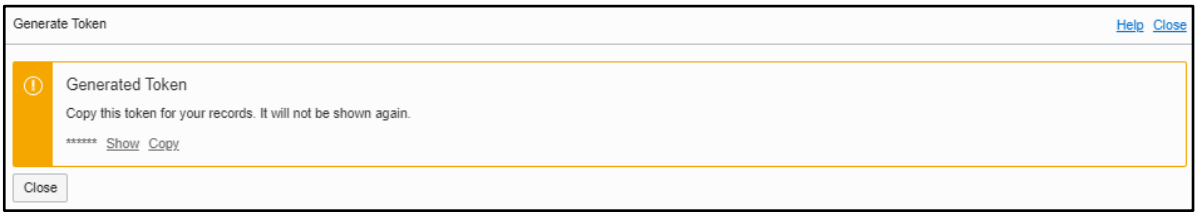
2. Click **Auth Tokens** under the **Resources** section on the left-hand side and then click **Generate Token**.



3. Provide your own meaningful description for the token and click **Generate Token** again. This will generate a token, which can be copied by clicking the **Copy** option.

   This completes the task of token generation. Make a note of the token as it will be used in the next practice to create the credentials.





Practice: Create Object Store Credentials

Overview

In this practice, you create the `atpc_user` user schema. You store the Auth Token created in the previous practice along with the username, as a credential under `atpc_user` schema. At the time of creating this credential, we will provide a credential name. This name is used to access the `.dat` files stored inside the object storage.

Tasks

1. Create the **`atpc_user`** user by logging in to SQL Developer as the **`ADMIN`** user. Grant the **`dwrole`** role to **`atpc_user`**. Granting this role is mandatory for this practice.
   a. Execute the following command to create the **`atpc_user`** user:

   ```
   create user atpc_user identified by "<password>";
   ```

   Note: The password should be at least 12 characters long, must contain one uppercase letter, and have at least one digit. If the password contains a special character, it needs to be enclosed in quotes.

   b. Execute the following commands to grant the **`dwrole`** role and allocate tablespace quota:

   ```
   grant dwrole to atpc_user;
   alter user atpc_user quota unlimited on DATA;
   ```

   You can enter both these commands in the SQL Developer worksheet and click the **Run Script** option to execute the commands.
2. Create a new SQL Developer connection for the new user **`atpc_user`** similar to the one created for the **admin** user. Name this connection **atpc_user**. You will use the connections created in this practice throughout the course practice sessions.
   a. Fill in the connection details as follows. Test and save the connection.
      - Connection Name: **atpc_user**
      - Username: **atpc_user**
      - Password: *Password for the above user*
      - Connection Type: **Cloud Wallet**
      - Configuration File: *Full path for the Client Credentials wallet file you downloaded earlier*
      - Service: **`myorcl_medium`** (Select the name corresponding to your instance)

3. In a SQL Developer worksheet, use the `create_credential` procedure of the `DBMS_CLOUD` package to store the object store credentials in your `atpc_user` schema.

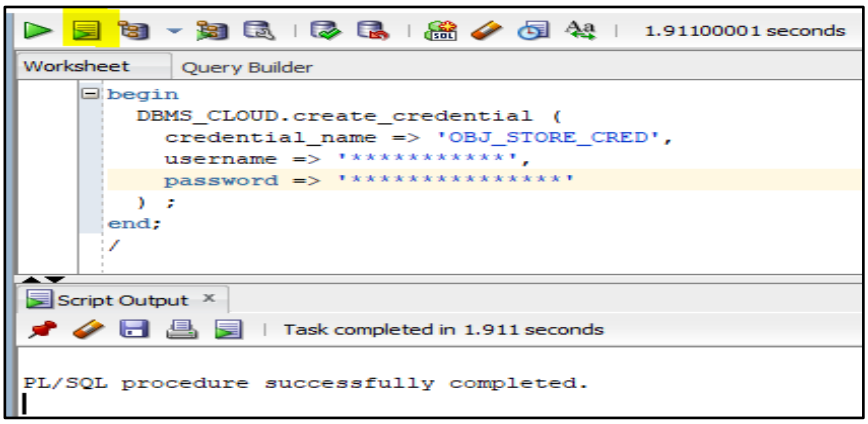
   Ensure you substitute your assigned Oracle Cloud account username and your own **Auth Token** for the password parameter value. This is the same Auth Token you created in the previous practice.

   a. After substituting your parameter values, execute the following command in SQL Developer by clicking the **Run Script** option:

```
begin
  DBMS_CLOUD.create_credential (
    credential_name => 'OBJ_STORE_CRED',
    username => '<your Oracle Cloud Account username>',
    password => '<your Auth Token>'
  ) ;
end;
/
```

   Ensure there is no space before or after the username and Auth Token values within the quotes.

   b. Ensure the command was executed successfully.



Practice: Copy Data from Object Store to Autonomous Transaction Processing Database Tables

Overview

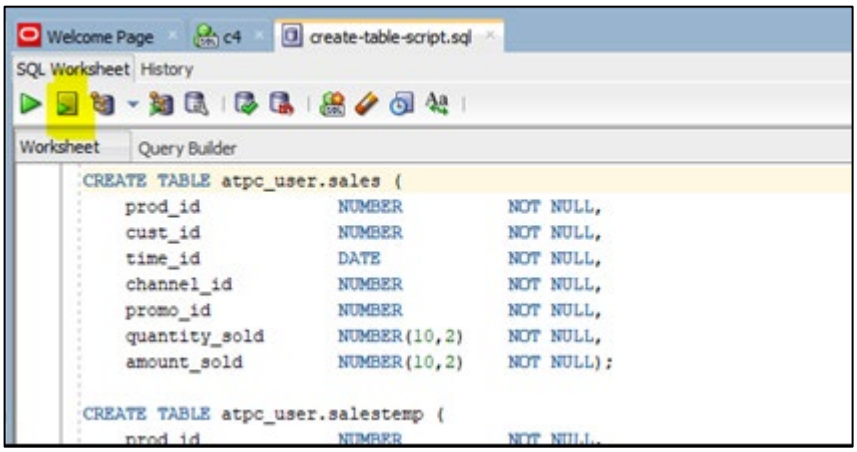
In this practice, you load data from external sources through Object Storage into the Autonomous Transaction Processing database.

Important: Ensure you edit the **`load-data-script.sql`** script to update your account details before executing it.

In this script, change the region name, tenant name, and bucket name with your respective names. We recommend keeping file names and table names untouched for this practice.

Tasks

1. Connect to the **`atpc_user`** user using SQL Developer connection.
2. Go to the **File** menu, select **Open,** and navigate to **`location`**, where the lab files were extracted.
3. Select the **`create-table-script.sql`** script and click **Open** in the file navigation window.
4. After the script is loaded into the SQL worksheet window in SQL Developer, select **`atpc_user`** from the connection drop-down list and click **Run Script**. The script execution should complete successfully before you proceed to the next step.



5. Open the **`load-data-script.sql`** script in the SQL Developer worksheet. Use the **find** and **replace** function to replace the following values with details corresponding to your assigned Oracle Cloud account.

   `<REGION NAME>`: *The region to which your cloud tenant belongs, you can find this as "Region Identifier" in manage regions page.*

   `<BUCKET NAME>`: *The name of the bucket you created in the earlier practice. In this example, we created the bucket "**`atp-objectstore-bucket1`**".*

   `<OBJECT STORAGE NAMESPACE>`: *The Object Storage namespace, you can find value for this in tenancy page.*

   `<TENANT NAME>`: *The tenant name assigned to you.*

   Sample of how the code looks **before** you replace your account details:

```
begin
 dbms_cloud.copy_data(
    table_name =>'CHANNELS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list
=>'https://swiftobjectstorage.<REGION_NAME>.oraclecloud.com/v1/<OBJECT_STORAGE_NAME
SPACE>/<BUCKET_NAME>/chan_v3.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'removequotes' value
'true')
 );
end;
/
```

   Sample of how the code looks **after** you replace your account details:

```
begin
 dbms_cloud.copy_data(
    table_name =>'COSTS',
    credential_name =>'OBJ_STORE_CRED',
    file_uri_list =>'https://swiftobjectstorage.ap-hyderabad-
1.oraclecloud.com/v1/axzldrkg4l2r/atp-objectstore-bucket1/costs.dat',
    format => json_object('ignoremissingcolumns' value 'true', 'dateformat' value
'YYYY-MM-DD', 'blankasnull' value 'true')
 );
end;
/
```

6. After the **`load-data-script.sql`** script is updated, select **atpc_user** from the connection drop-down list and click **Run Script**. The script execution should complete successfully before you proceed to the next step.

   Note: You might encounter these command errors:

   - **`ORA-27276`** error. Ensure you create the credential under the same user where you are executing the script. Check whether you have mentioned a different credential name.
   - **`ORA-20000`** error: You need to make sure that **`file_uri_list`** is properly mentioned with the region name, tenant name, and bucket name.

This completes the task of loading data into an Autonomous Database.

Practice: Clone Autonomous Database

Get your free cloud account: Click here

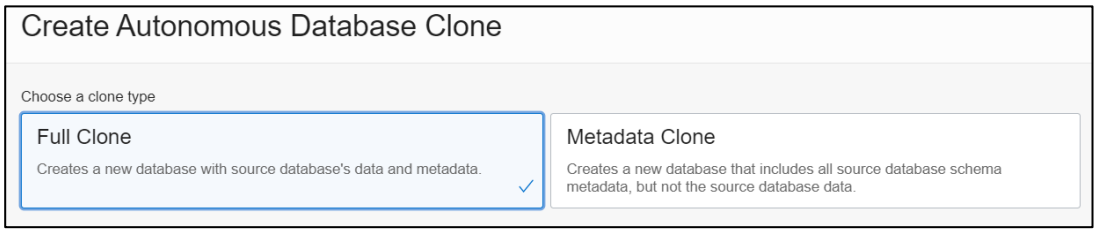
Overview

In this practice, you will clone Autonomous Database.

Tasks

1. Log in to your Oracle Cloud Free Tier Account.
2. Expand the options available under **Menu**.
3. Click **Autonomous Database.**
4. Select the compartment in which you have your ADB.
5. Select an Autonomous Database instance from the list in your compartment.
6. On the **Details** page, from the **More Actions** drop-down list, select **Create Clone**.
7. On the **Create Autonomous Database Clone** page, choose the clone type from the choices:
   - **Full Clone:** creates a new database with the source database's data and metadata.
   - **Metadata Clone:** creates a new database with the source database's metadata without the data

     Note: As part of this practice, choose the **Full Clone** option**.**



8. In the **Configure clone source** area, select the **Clone source** option:
   - **Clone from database instance:** This creates a clone from a running database.
   - **Clone from a backup:** This selection creates a database clone using a backup or lets you enter a point in time to create the clone. Select this option.

     Note: As part of this practice, choose the **Clone from database instance** option**.**

9. Select or fill in the following values in the Create Autonomous Database wizard:

   Important: Instance creation will fail if there is duplication in the name of the instance or the database name in a given tenancy. Hence, take extra care to ensure the names you enter are unique.

   a. Create in Compartment: **Ensure the right compartment is selected**

   b. Display Name: **MYCLONE**

   c. Database Name: **CLONEATP**

   d. Choose Database Version: **Select the latest version available**

   e. CPU Core Count: **1**

   f. Storage (TB): **20GB**

   g. Auto scaling: **Enabled** (Accept the default)

   h. Password: *Set the password for your Autonomous Database **ADMIN** user*

   i. The password must meet the strong password complexity criteria based on Oracle Cloud security standards. Example for an Admin password is "***Wwelcome#123***".

   j. Confirm password by reentering the same admin password.

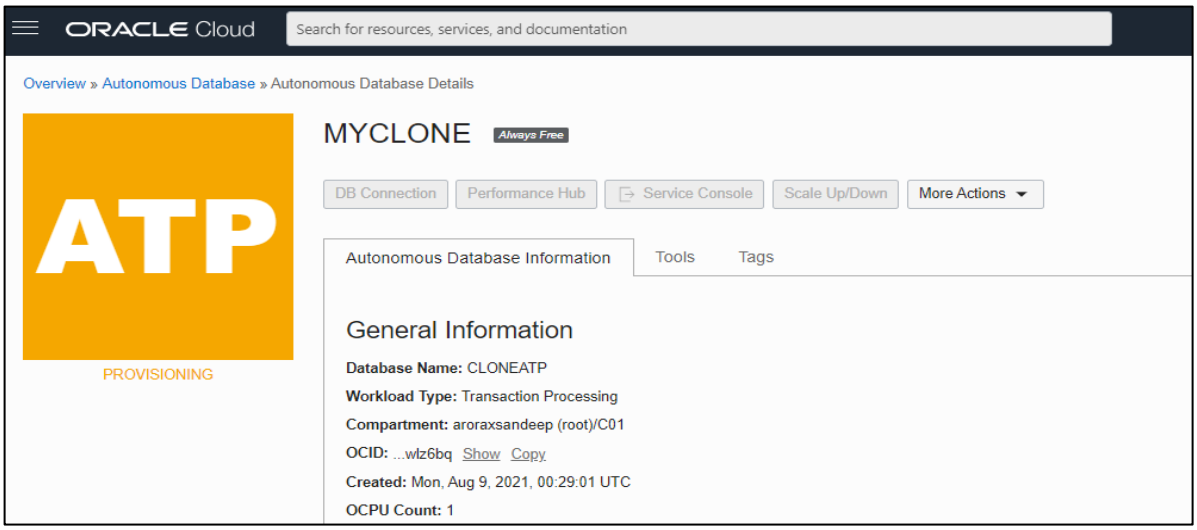
   k. Network Access: **Allow secure access from everywhere**

   l. License Type: **License Included**

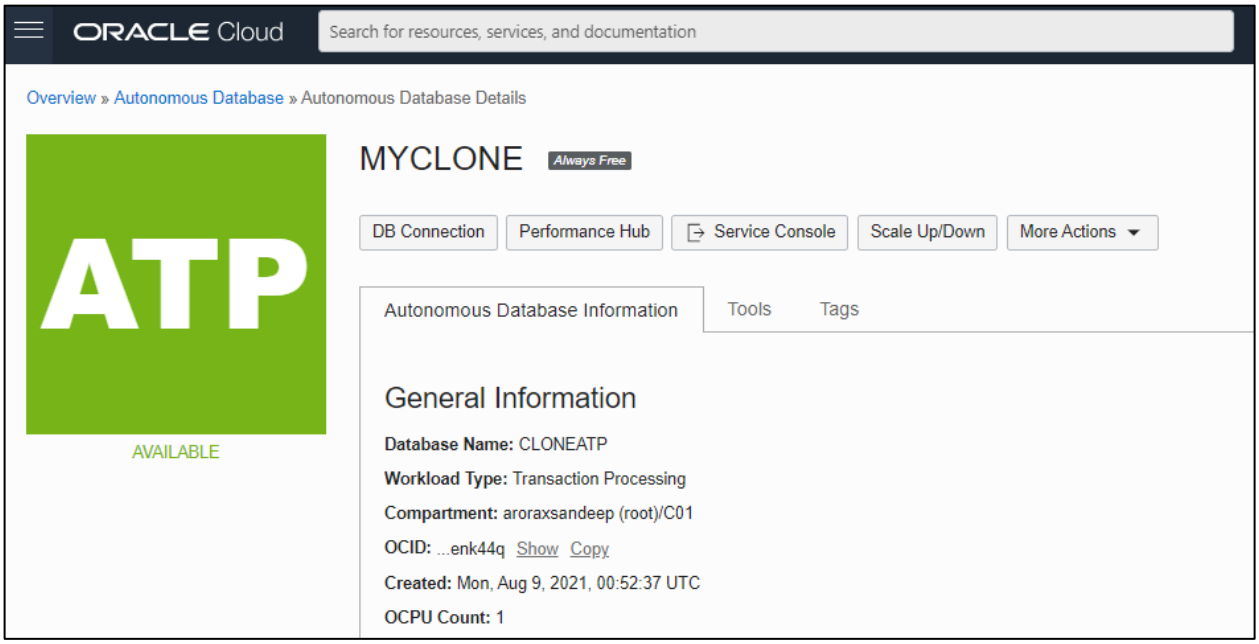
   m. Advance Options: Ignore this section (accept the default)

   n. Click **Create Autonomous Database Clone**.

   On the Oracle Cloud Infrastructure console, the **state** shows **provisioning** until the new Autonomous Database is available.

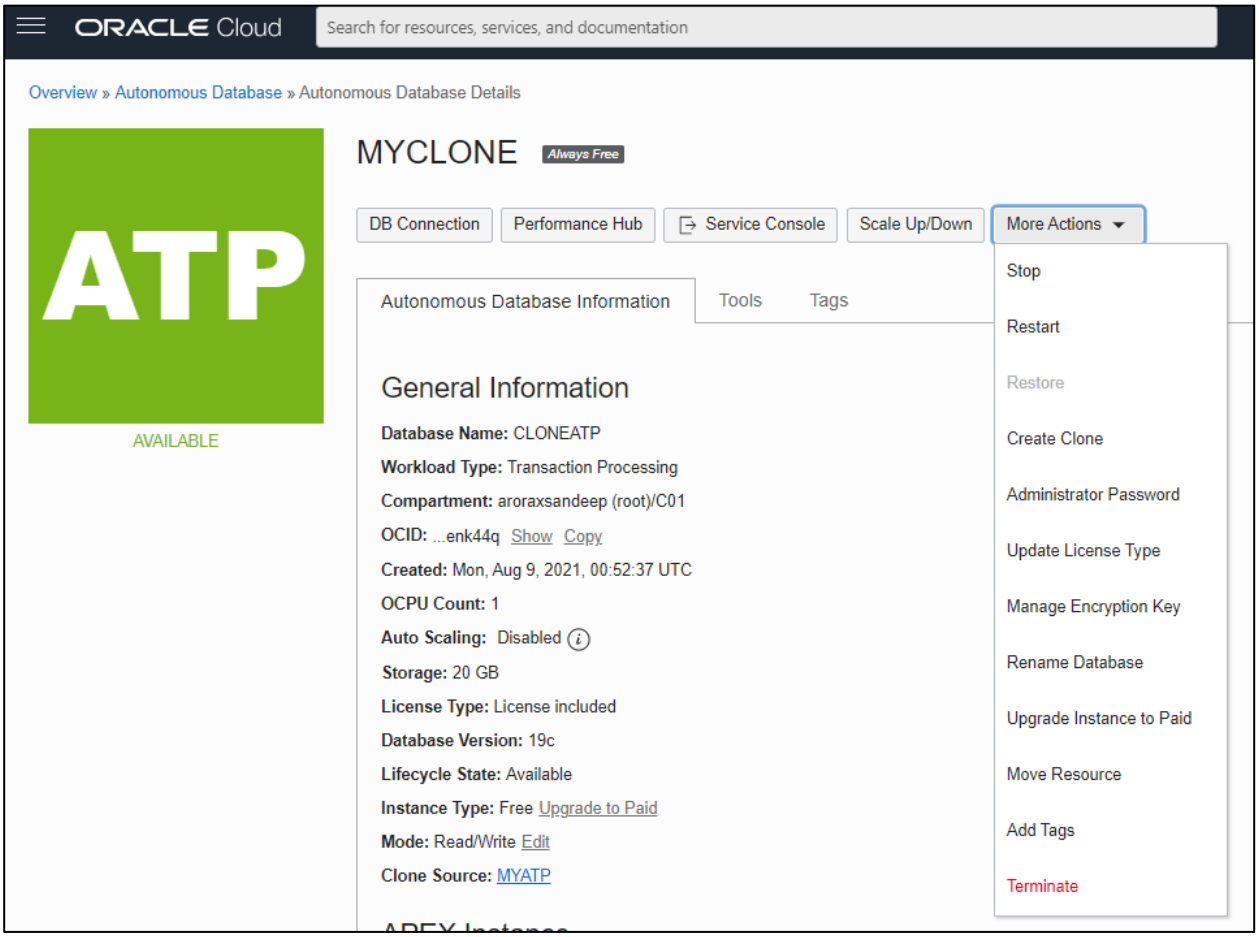


10. Ensure the cloning operation was executed successfully. The final **state** should show as **Available**.



11. Terminate the clone database.

    a. Click More Actions and click Terminate.



    b. Enter the display name “MYCLONE” and click Terminate Autonomous Database. It will take a few minutes to terminate.