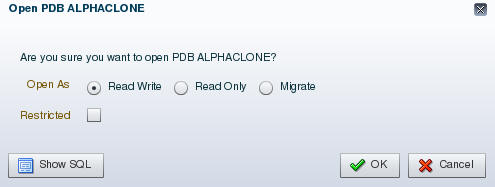


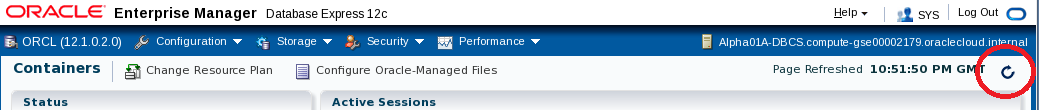
* Accept the defaults and Click **OK**.



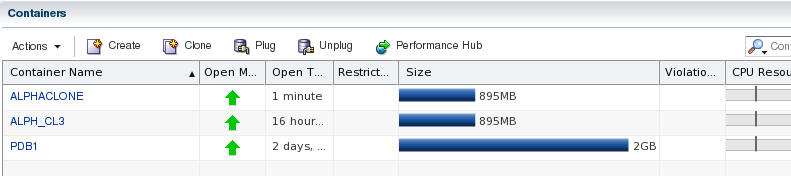
* Close the Confirmation pop up by clicking **OK**.
* Reopen the pluggable database via **Actions 🡪 Open**.
* Accept the default to open it Read / Write. Click **OK**.



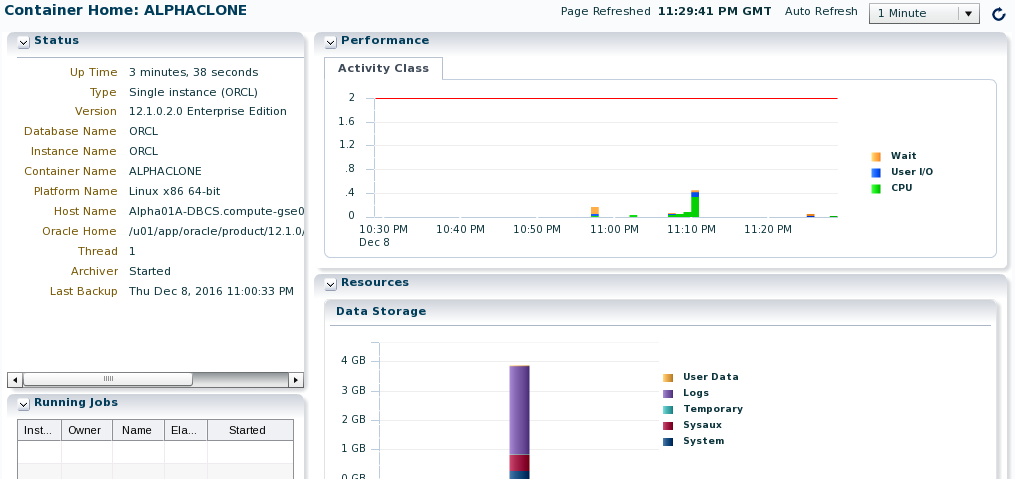
* Close the Confirmation pop up by clicking **OK**.
* Refresh the browser page using the refresh icon in the top right corner.



* Note that the violations are now gone.



* Click the ALPHACLONE container name link to review the database information.

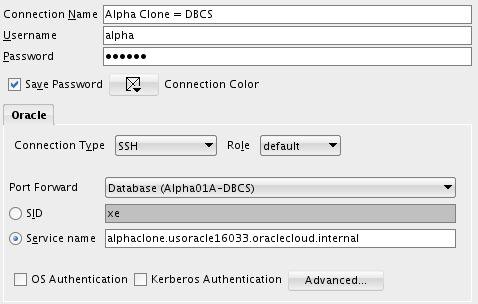


#### Create an SQL Developer connection to the Public Cloud database ALPHACLONE schema

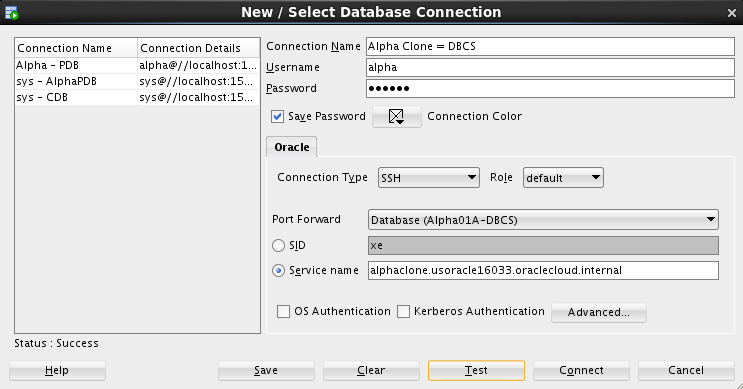
* In the SQL Developer application, click the green plus sign  in the Connections window to create a new connection; enter the following connection details:

|  |  |
| --- | --- |
| **Connection Name:** | Alpha Clone – DBCS |
| **Username:** | alpha |
| **Password:** | oracle |
| **Check:** | “Save Password” |
| **Connection Type:** | SSH |
| **Service Name:** | Alphaclone.<Your ID Domain>.oraclecloud.internal |

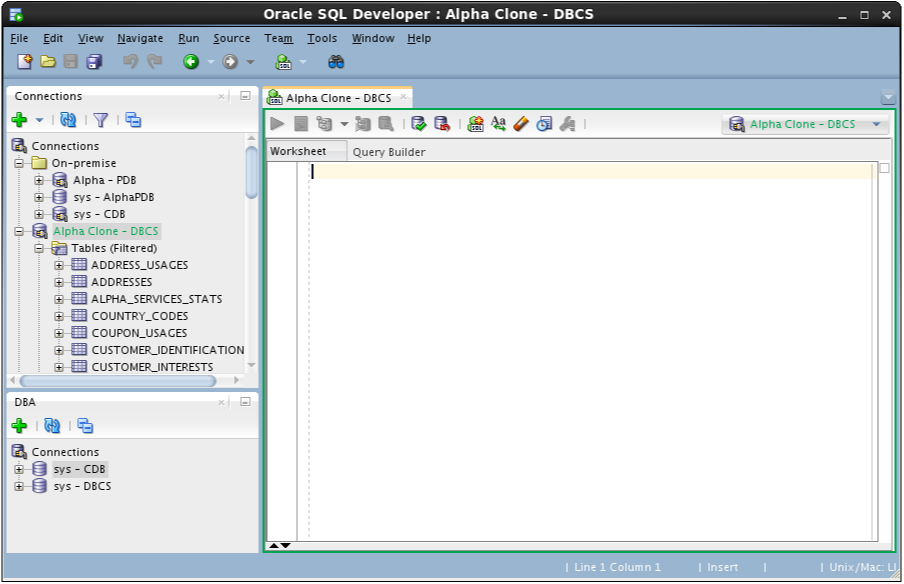
***Note:*** *You can optionally select a color for the connection to differentiate it from other connections.*



* Click Test to confirm the information was entered correctly.



* Click Connect to save the connection information and open a new SQL Worksheet.



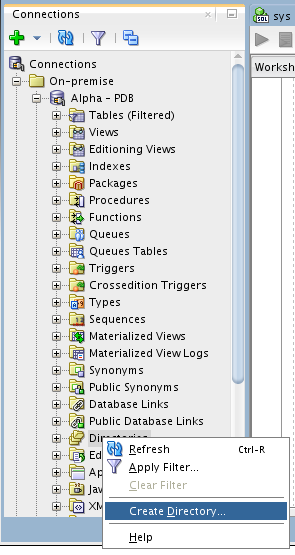
* You have successfully migrated a pluggable database from on premise to the cloud. In the next section we’ll migrate data using Data Pump.

### Cloud Migration Using Data Pump

#### Export the Alpha Schema

The first step will be to create a local Data Pump Directory.

* In the Connections Tab inside the "On-Premise" folder navigate to the Alpha - PDB 🡪 Directories item, right-mouse click and select Create Directory…

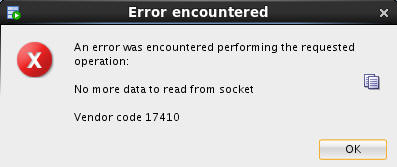


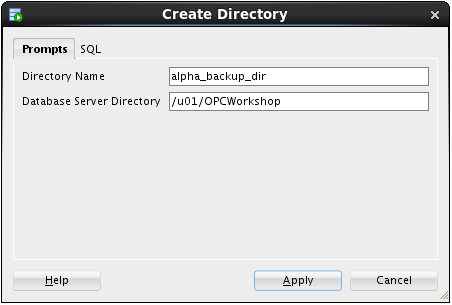
**Note**: The default Data Pump directory object, DATA\_PUMP\_DIR, does not work with PDBs. Data Pump requires an explicit directory object within the PDB for exporting or importing schemas or tables.

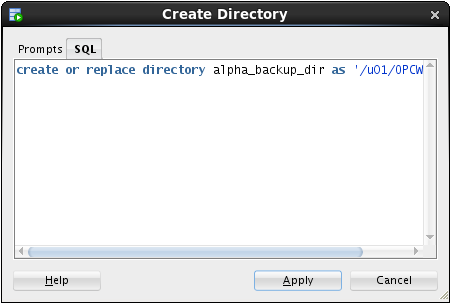
* Enter the following values and click Apply. Remember to use the SQL tab to review the actual DDL statement. Click OK to dismiss the confirmation.

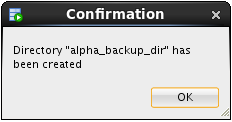
|  |  |
| --- | --- |
| **Directory Name:** | alpha\_backup\_dir (not case sensitive) |
| **Database Server Directory:** | /u01/OPCWorkshop |

**NOTE:** You may receive an error message stating that “An error was encountered performing the requested operation:” and that the directory cannot be created. To eliminate this error right-click on Alpha - PDB and choose Disconnect. Then Reconnect. The error occurs because you were connected earlier while performing the UNMOUNT / REMOUNT and during the previous “cloning” of the PDB container the connection information was lost. Reconnecting will normally solve this issue.

**

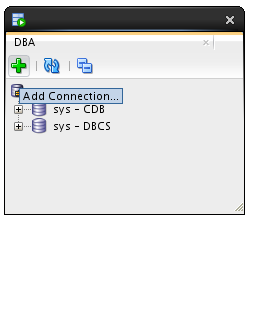
******

******

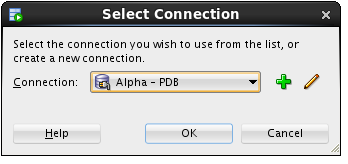
**

**Stop here-------**

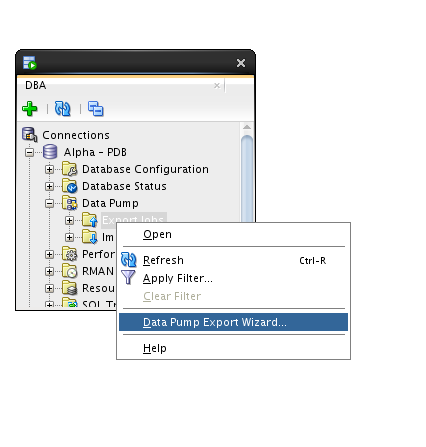
* Now that we’ve created the Data Pump export directory the next steps will outline how to create and run a Data Pump Export job using SQL Developer
* In the DBA Window, Add Connection by clicking on the Green Plus sign.



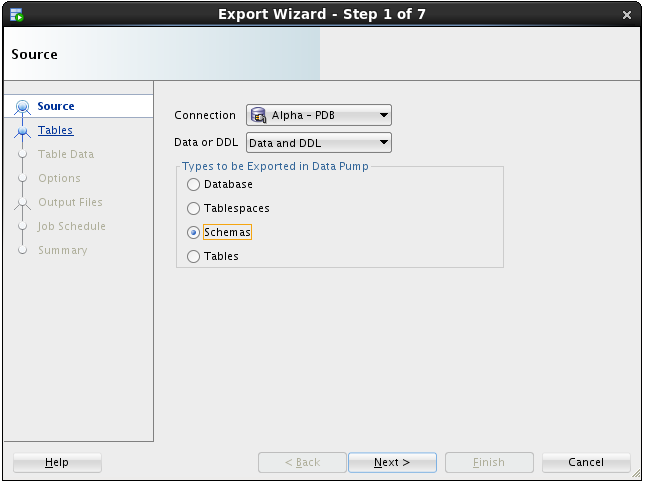
* Select the Alpha - PDB connection and click the OK button.



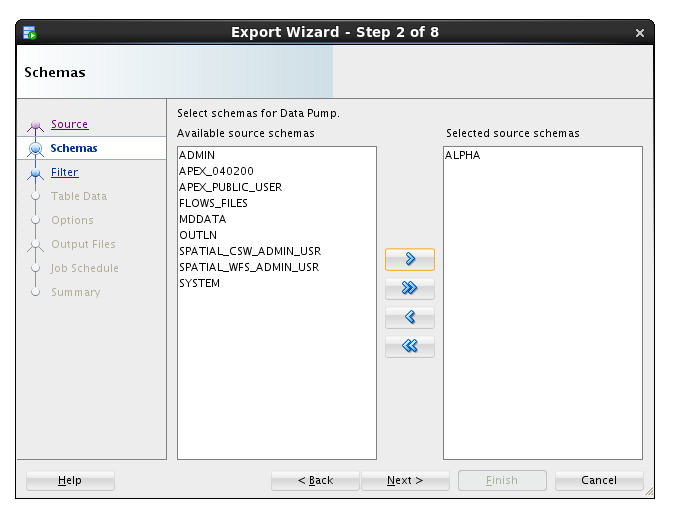
* Expand Alpha - PDB, expand Data Pump, then right-mouse-click on Export Jobs, and then select the Data Pump Export Wizard… menu item.



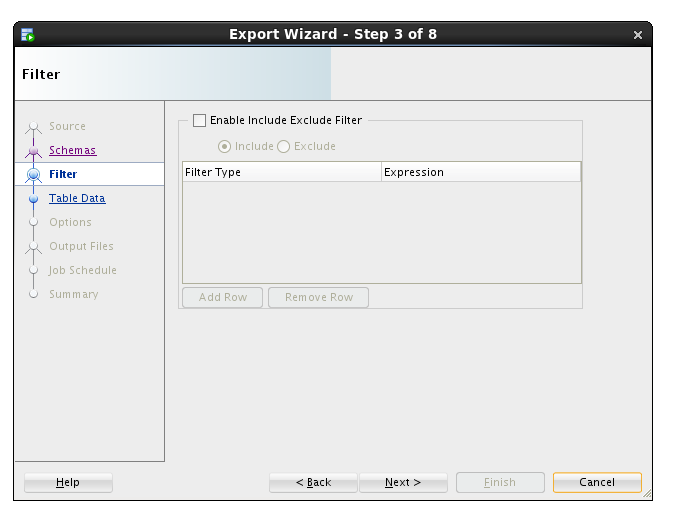
* Select the Schemas export type and click the Next button.



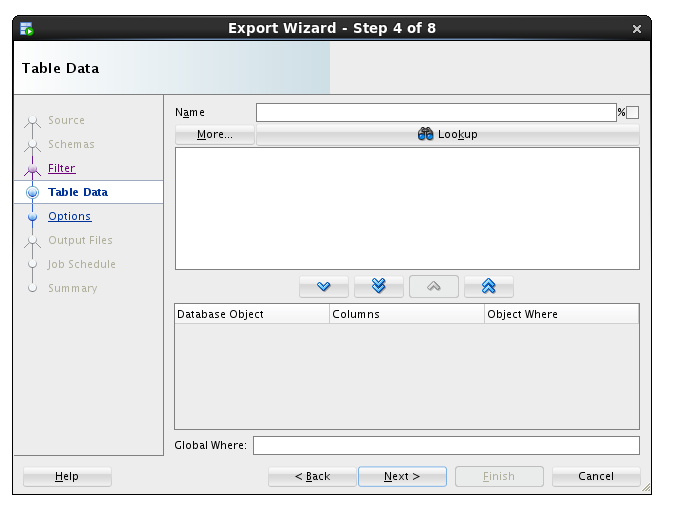
* Select the ALPHA schema and use the blue arrow to move it to the right-hand column. Click Next.



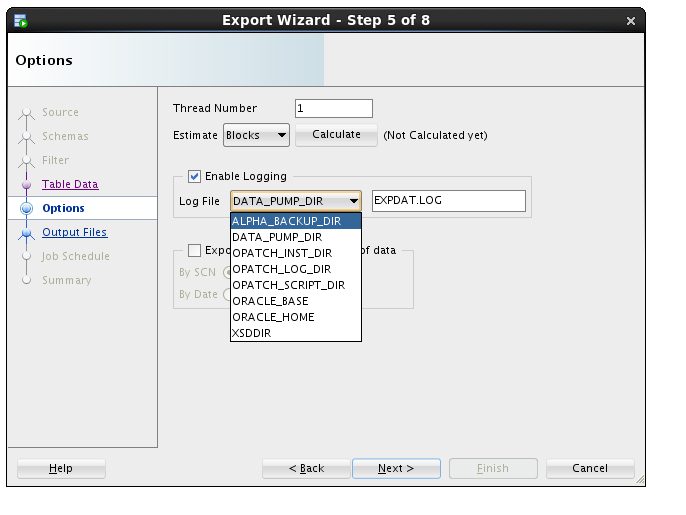
* We are not filtering out any objects, click the Next button.



* We are not applying where clauses to table data, click the Next button.

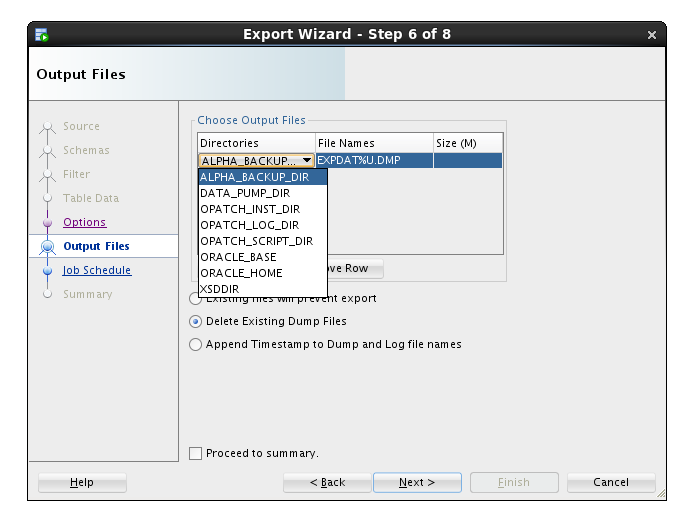


* We want a log for this export, and just like the actual export file, we must pick a directory from the list of directories in the database.
* Select ALPHA\_BACKUP\_DIR from the list and click the Next button.

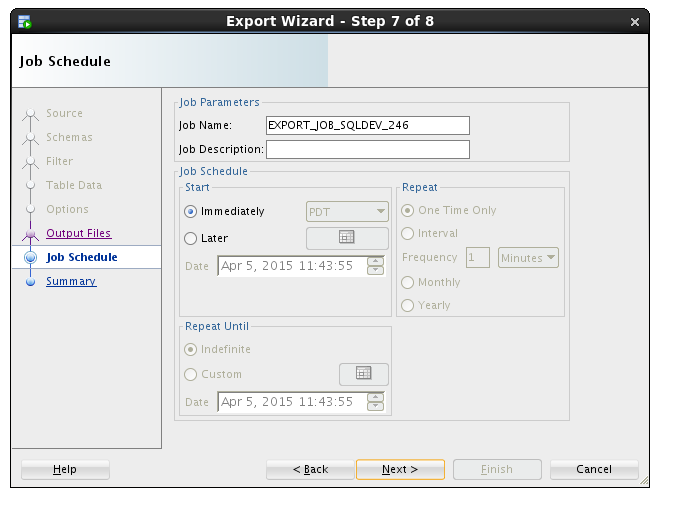


* The most important selection for any Data Pump operation is choosing the directory where the export file will be written.
* Select ALPHA\_BACKUP\_DIR from the Directories drop down list.
* Then, select the Delete Existing Dump Files radio button and click the Next button.

**Note**: Data Pump always uses a server side directory for all export or import operations.



* Data Pump jobs can be scheduled to run at any time and on any desired times of the day, week or year. We will run the job immediately - click the Next button.

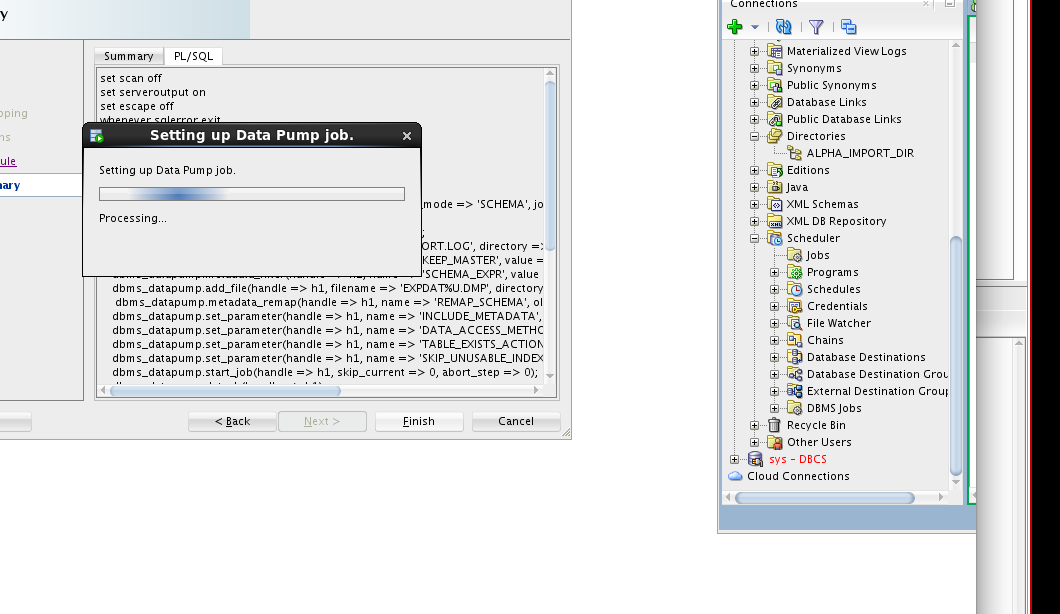


* On the Summary panel, click the PL/SQL tab to review the job definition. Review the PL/SQL use of Oracle Supplied PL/SQL subprograms for Data Pump. Click the Finish button to create the job.

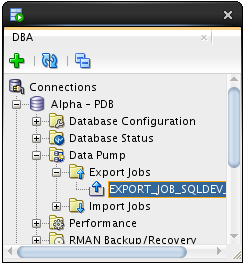
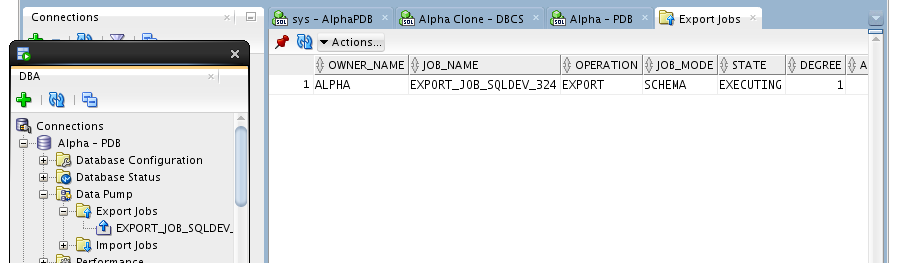
|  |  |
| --- | --- |
|  |  |

* For a brief time, SQL Developer shows a progress dialog while it creates the job in the database.

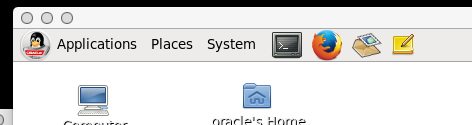
**Note:** the import actually runs as a job in the database so this message is only about creating and scheduling the export.



* While the job is running, you may view status information by clicking on the export job added to the DBA Navigator panel. It may take a couple of minutes so click the Refresh icon until the job is completed (NOT RUNNING).

* Now we’ll copy the export Data Pump file to the server
* Start a Terminal window using the top panel icon.

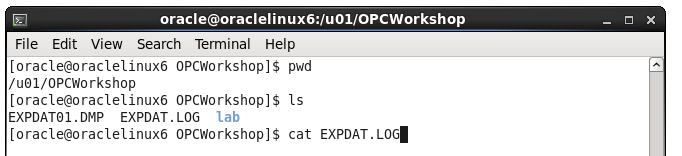


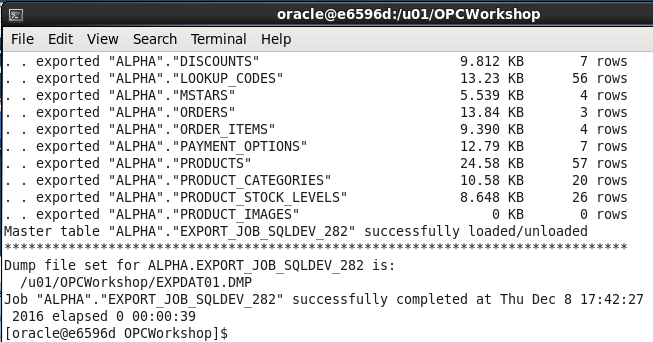
* Enter the following commands to print the working directory (pwd), list the directory (ls) contents and review the Data Pump log file.

$ pwd

$ ls

$ cat EXPDAT.LOG

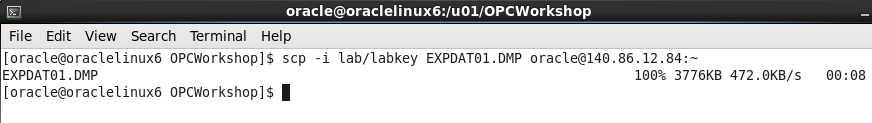




Use the following secure copy (scp) command to transfer the Data Pump export to the DBCS server. Use the Database Service Private IP address you identified in the first lab.

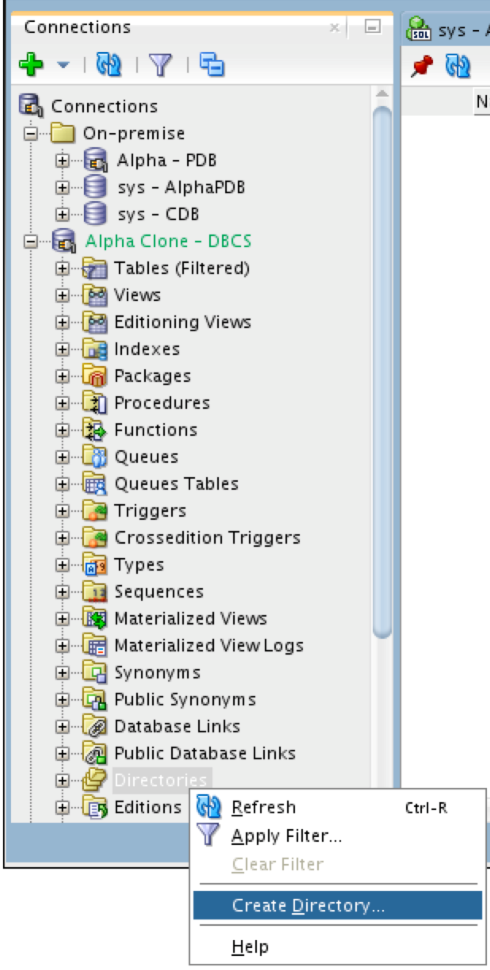
$ scp -i lab/labkey EXPDAT01.DMP oracle@<Alpha01A-DBCS public IP>:~

***Note:*** *the tilde (~) represents the oracle user's home directory.*



#### Import Alpha to a new Schema

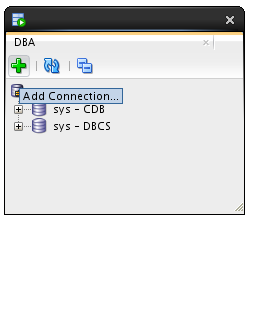
* As we begin the import phase of this example we’ll first create an import directory in the Alpha Clone PDB.
* Use SQL Developer and expand the Alpha Clone - DBCS connection.
* Right-mouse-click on the Directories tree item and select the Create  Directory… menu item.



* Enter the following values and click the Apply button.
* Click OK to dismiss the confirmation message. This lets the database access the same directory where the Data Pump export file was copied.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Directory Name:*** | alpha\_import\_dir | | |
| ***Database Server Directory:*** | /home/oracle | | |
|  | |  |  |

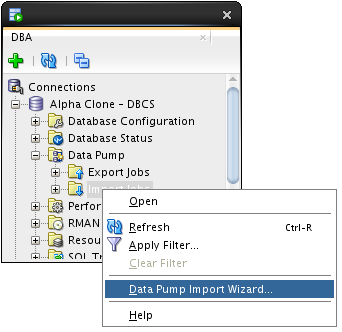
* The next few steps will outline creating the Data Pump Import job. To access the Data Pump features, we need to add the clone connection to the DBA Navigator.
* Click on the green plus sign, Add Connection icon on the DBA Navigator panel



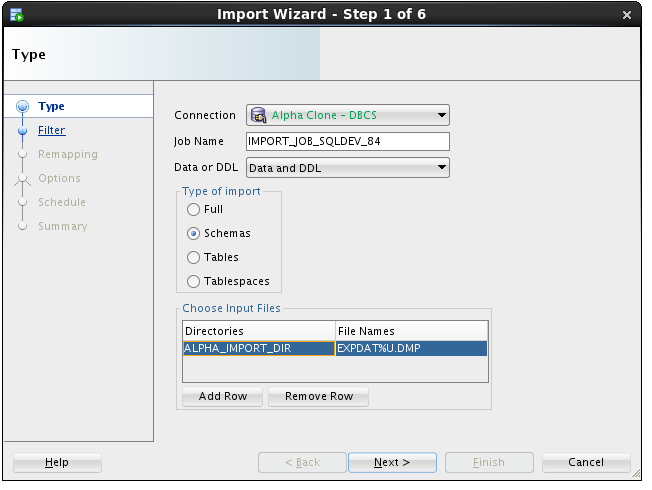
* Select Alpha Clone - DBCS connection and click OK.



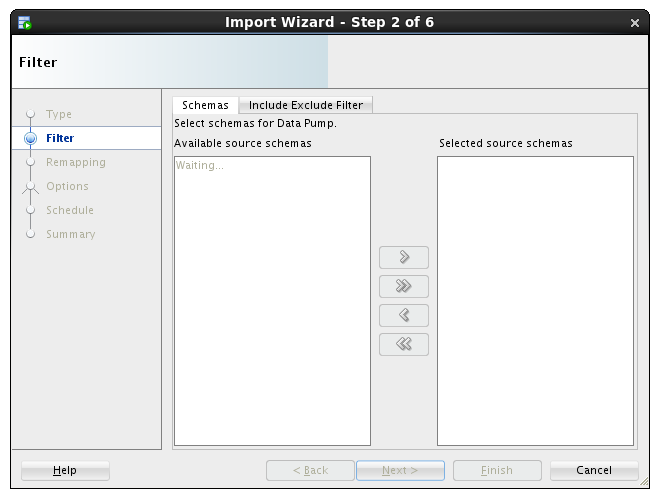
* Expand Alpha Clone - DBCS 🡪 Data Pump
* Right-mouse on the Import Jobs menu item, and select Data Pump Import Wizard… menu item.



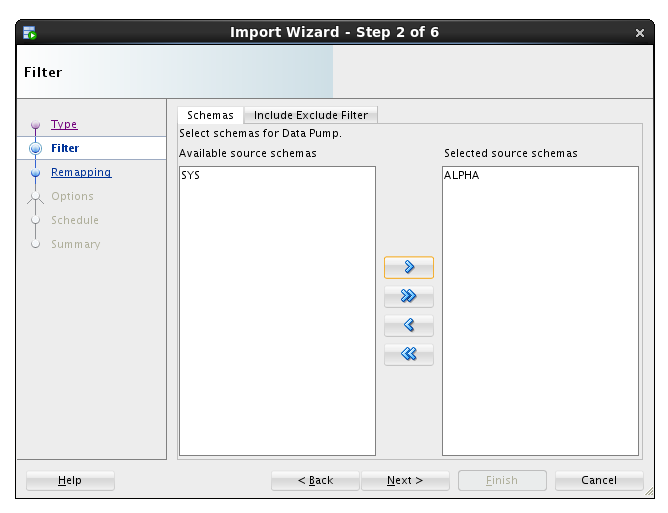
* Select Schemas from the ‘Type of Import box and Choose ALPHA\_IMPORT\_DIR from the ‘Choose Input Files’ drop down list, then click Next.



**Note:** This action might take a few minutes. There is some wait time while the database locates and scans the import file in the selected directory.

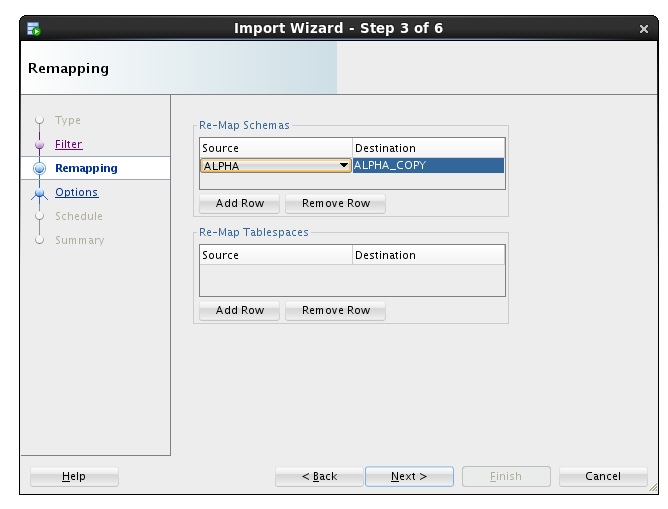


* Move the ALPHA schema from the left to the right column using the arrow button and click Next.

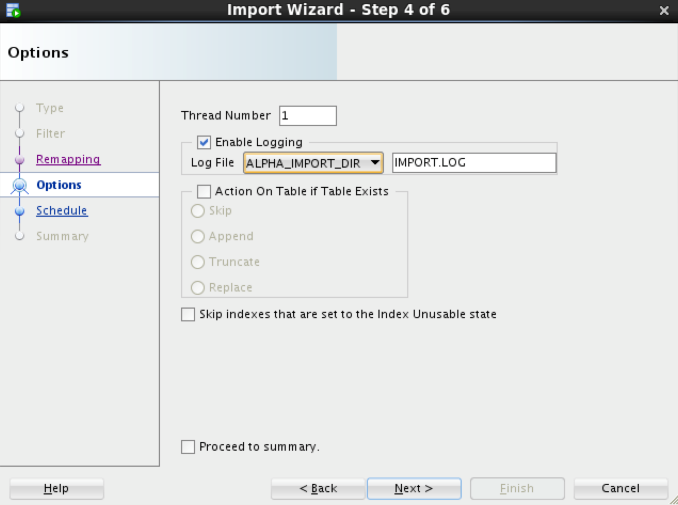


* For this lab, we are creating a new schema, so we will enter the new schema name as the destination.
* Under the Re-Map Schemas section click Add Row.
* Enter the following values and click the Next button.

|  |  |
| --- | --- |
| **Source:** | ALPHA (should be the default item) |
| **Destination:** | ALPHA\_COPY |

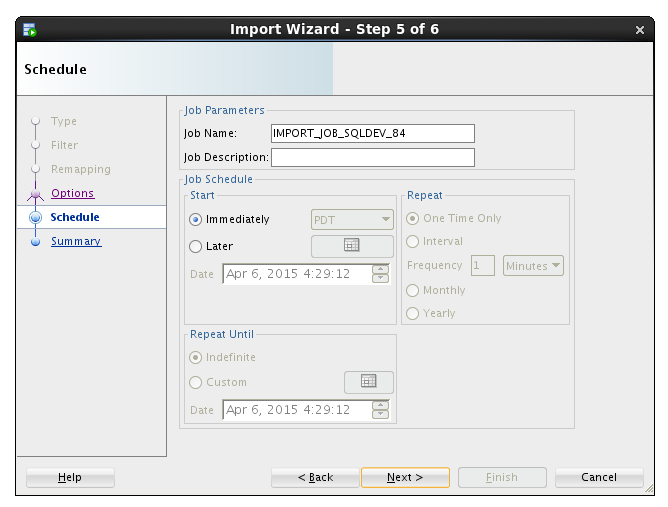


* We want to see the log output so we will select the same directory as the import file directory.
* Select ALPHA\_IMPORT\_DIR and click the Next button.

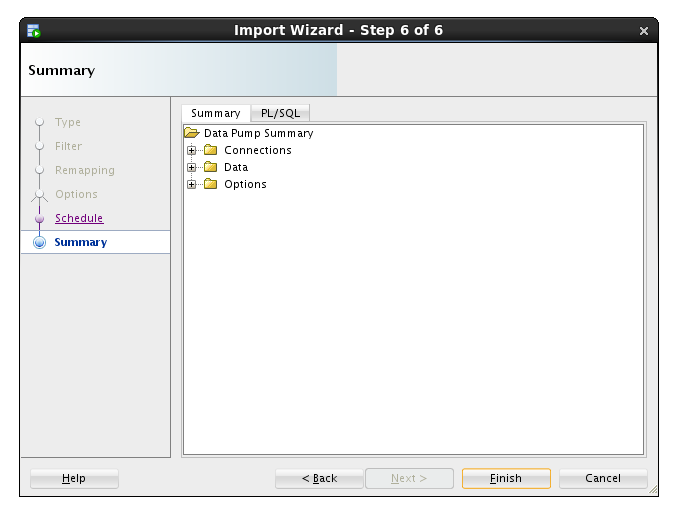
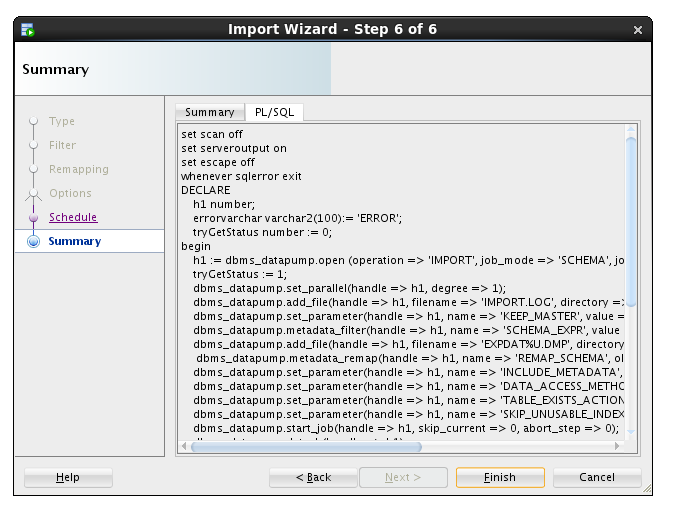


**Note:** For lab purposes we will execute the import immediately. In normal operations this job could be set up to refresh a development database on a daily basis.

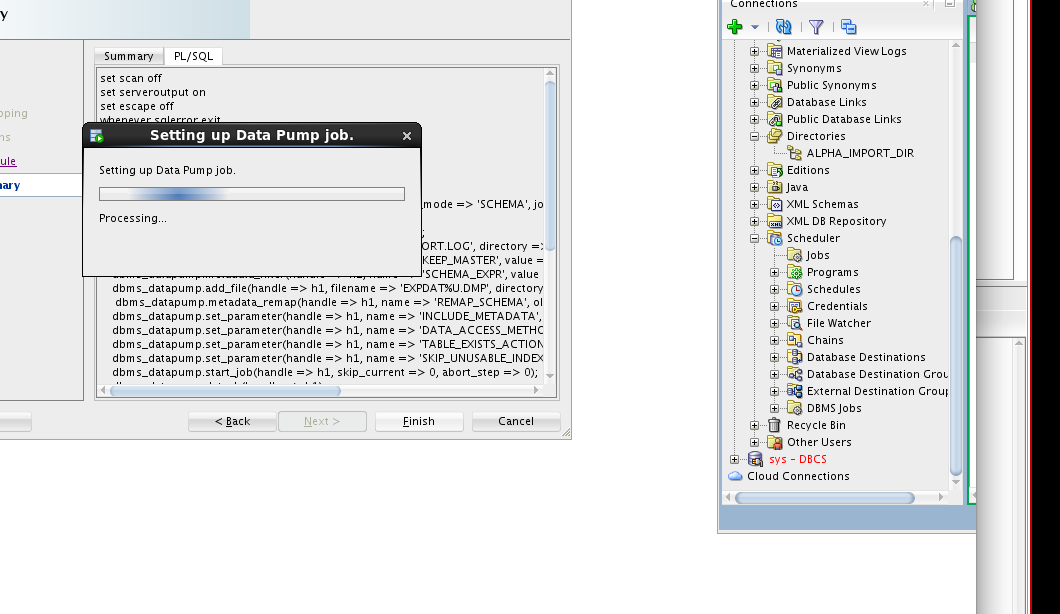
* Click the Next button.



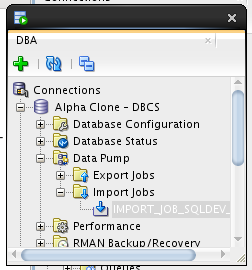
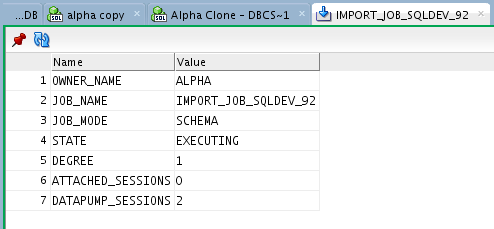
* Click the PL/SQL tab to review the small program that establishes the import job. Click the Finish button to create the job.

**Note:** For a period of time SQL Developer shows a progress dialog while the job is being created. The job does not run locally you’re seeing the progress of creating the job in the database.



* Locate and click on the job name to see the detailed status as the job runs. When the job completes, the database automatically removes the job. You will need to use the Refresh icon  to see when the job finishes.

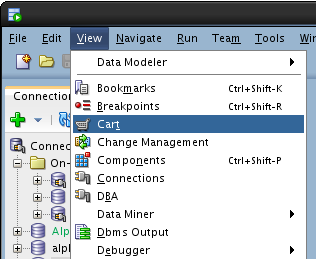
* If you are interested in verifying that the ALPHA\_COPY schema is the same as the ALPHA schema, feel free to create a connection and compare.

### Cloud Migration Using SQL Developer Carts

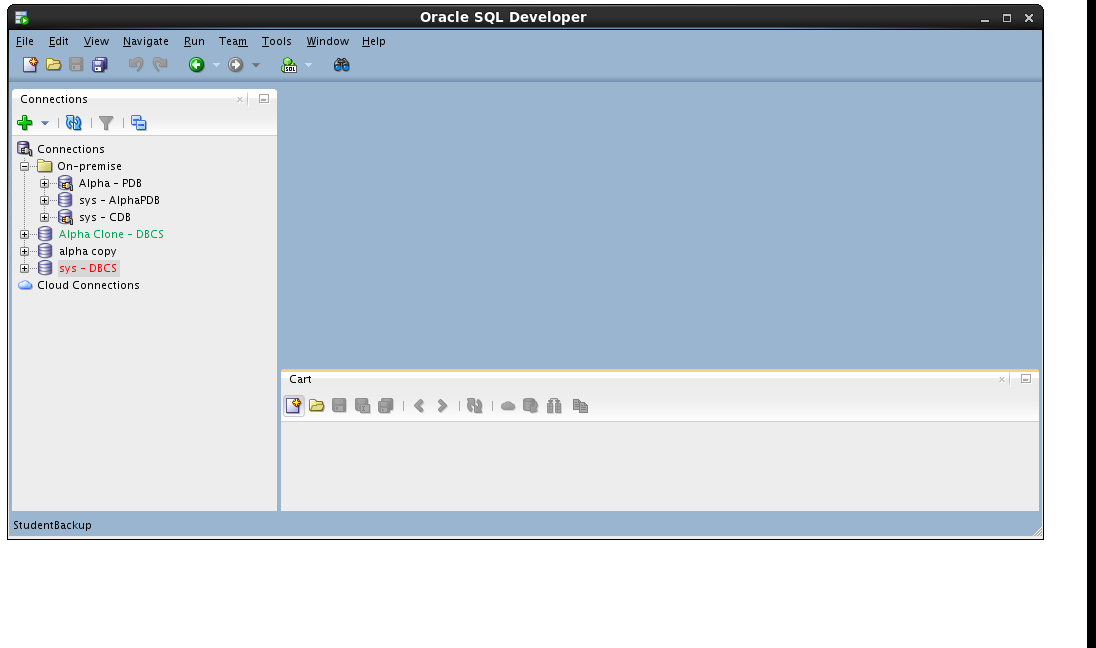
#### Creating an SQL Developer Cart

The SQL Developer Cart is a convenient method for organizing the deployment of database objects and data from one database to another. In this trivial example, we want to update the data of just the CUSTOMERS and PRODUCTS table in the development cloud database. More elaborate usages of the cart can help package entire application deployments, including pre and post processes from multiple data sources.

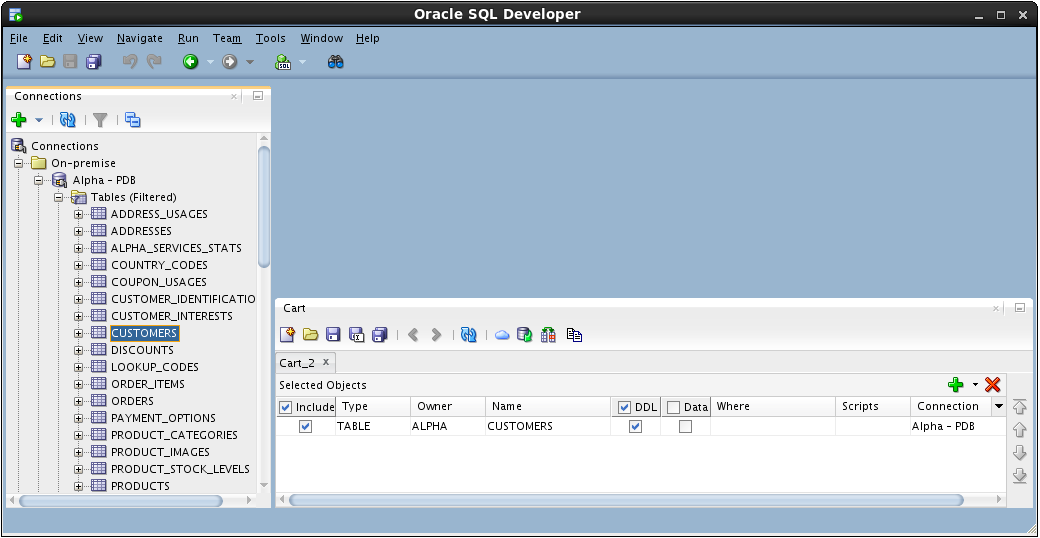
* Show the Cart using the View > Cart menu option.



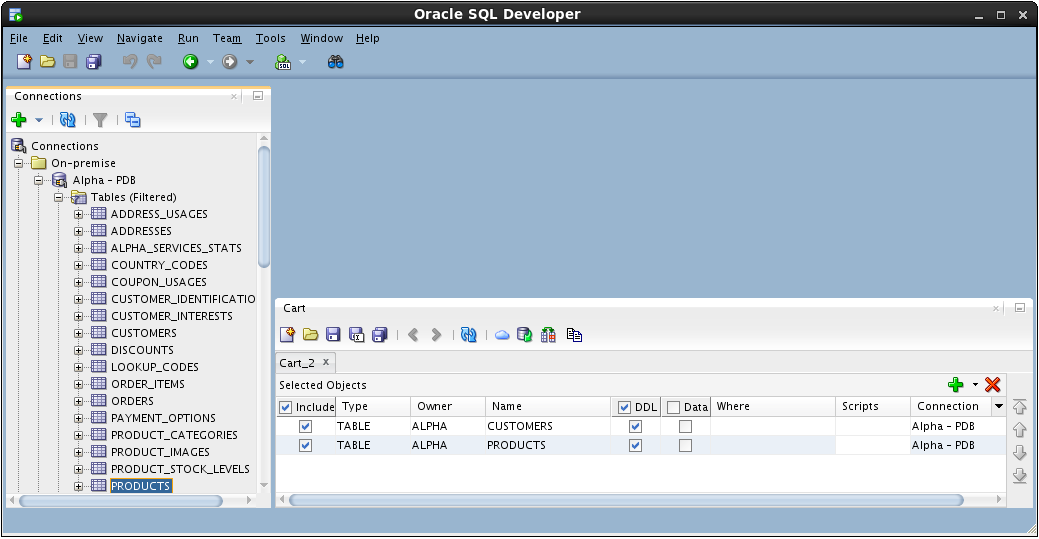
* If Cart\_1 is not already created (it should be), Click on the New Cart icon.



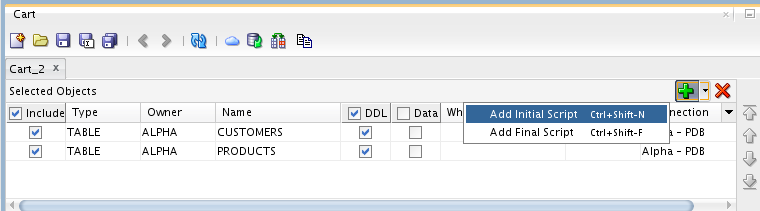
* Drag the CUSTOMERS table from the Alpha - PDB connection to the cart.

* Drag the PRODUCTS table to the cart.



* Include a script that runs before any other Cart activity. For this lab, we are disabling all the referential integrity constraints so we can delete and insert data without regard to foreign keys on our tables.
* In the Cart window click small down arrow next to the green plus “+” icon and select Add Initial Script

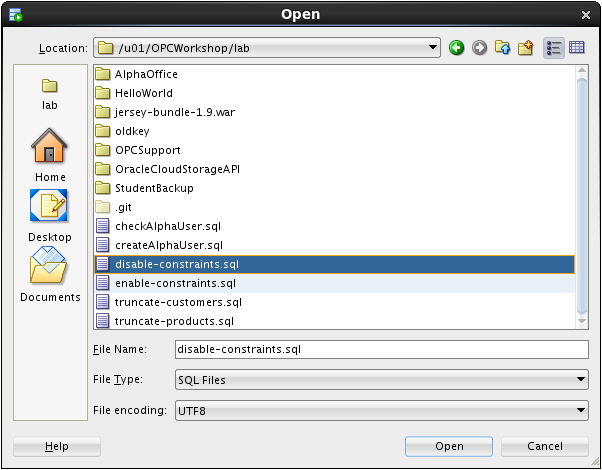


* Click the Browse… button.



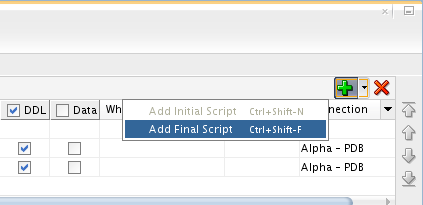
* Locate the following file and click Open:

/u01/OPCWorkshop/lab/disable-constraints.sql

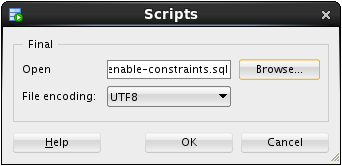


* Click OK.
* Click the down arrow next to the green plus “+” icon again and select Add Final Script which is included as the last operation performed during the cart operations.

**Note:** There can only be one Initial or Final script in a Cart.

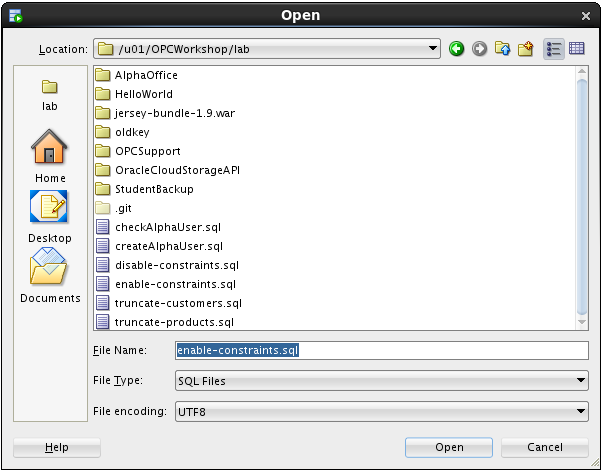


* Click the Browse… button

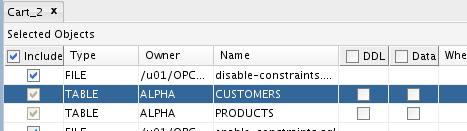


* Locate the following file and click Open:

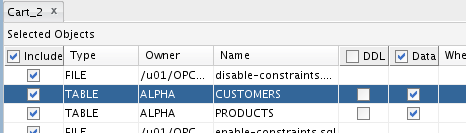
/u01/OPCWorkshop/lab/enable-constraints.sql



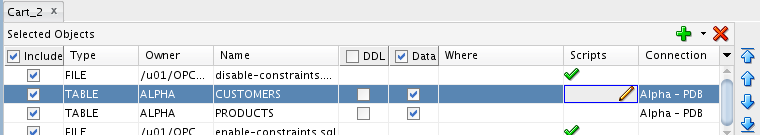
* Click OK.
* We are not creating any tables in this lab; uncheck the DDL column heading.



* We will move the data, include a check the Data column heading.



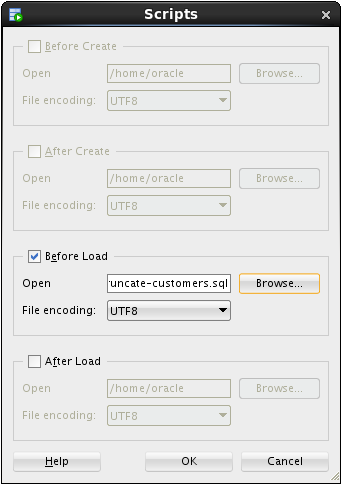
* Before we can overwrite the new rows in the CUSTOMERS table, we need to truncate the table.
* Click in the Scripts column cell for the CUSTOMERS table and then click the pencil icon.



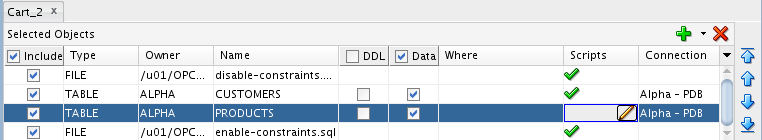
* Check the Before Load box, then click the Browse… button and select the following file:

/u01/OPCWorkshop/lab/truncate-customers.sql

* Click OK:



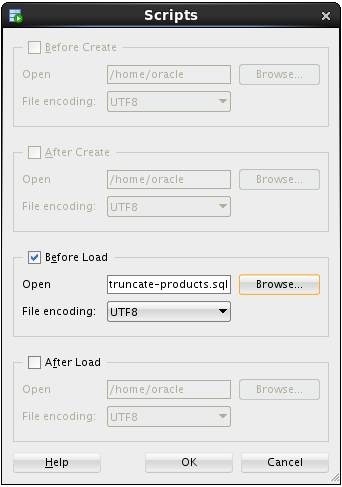
* Repeat the operation for the PRODUCTS table; click the pencil icon on the products row.



* Click the Before Load button, then click on the Browse… button and select the following file:

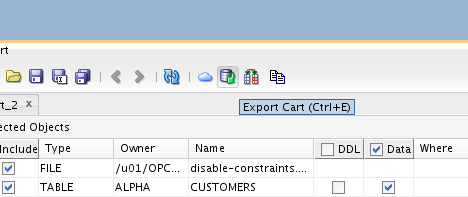
/u01/OPCWorkshop/lab/truncate-products.sql

* Click OK:



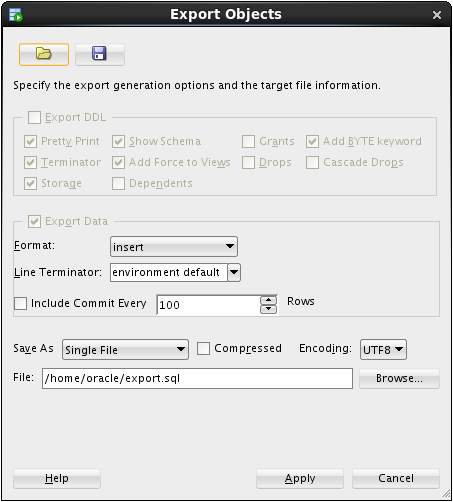
#### Export the SQL Developer Cart

* Now that the cart is complete, click the Export Cart toolbar icon to generate the script of all the elements we inserted in the cart.

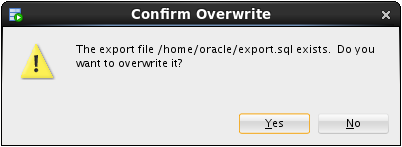


* Click the Apply button to generate script.

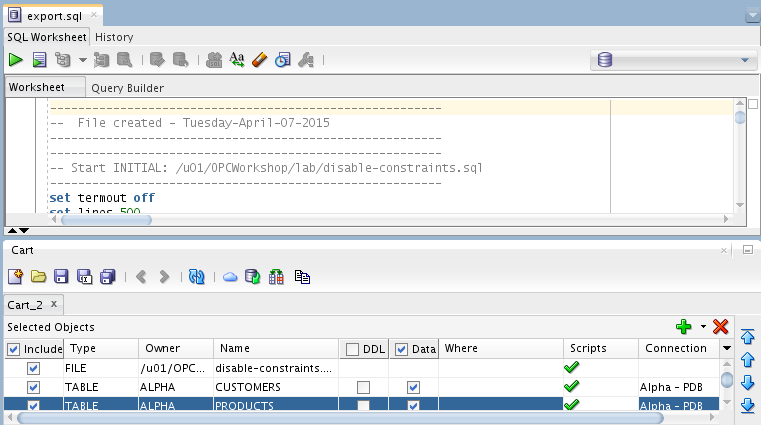
**Note:** The selections on this page may be saved and later reused if the cart is regularly used the same way.



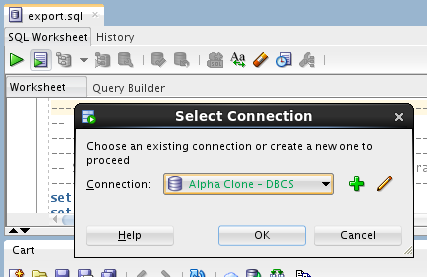
* If the file already exists, SQL Developer asks you to confirm overwriting it with new content. If you see this prompt, click the Yes button.



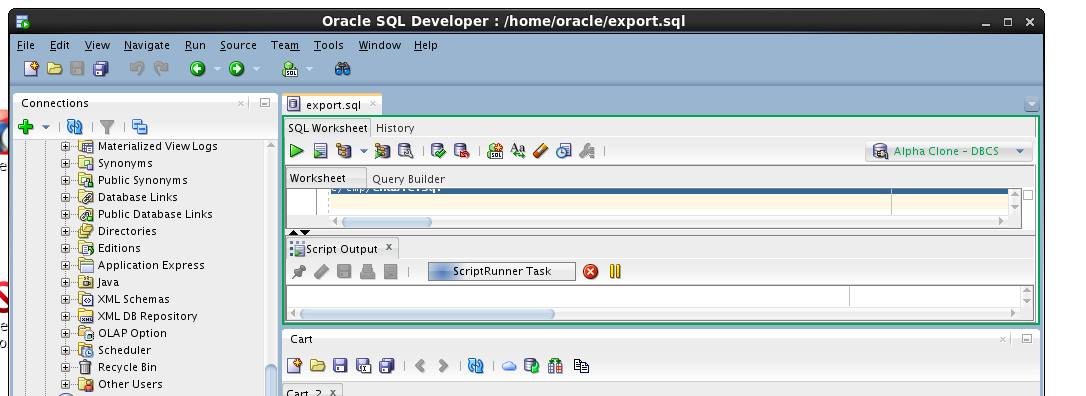
* Review the contents of the script with particular attention to the SQL statements that have been inserted based on the scripts we included.



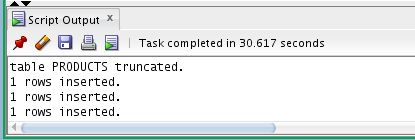
* Run the script by clicking the Run Script icon and selecting the Alpha Clone — DBCS connection.



* Click OK.
* SQL Developer shows a progress bar while the script runs. Depending on your window layout, you may see the command output scrolling by while the script runs.



* When the script is complete, review the script output looking for the execution of both the script elements and the DML statements.



* This concludes Lab 2 – Cloud Migration, proceed to the next lab when you’re ready.

# Backup and Recovery

### Introduction

Oracle Database Backup Service (ODBS) is a new backup-as-a-service offering that enables customers to store their backups securely in the Oracle cloud. ODBS provides a transparent, scalable, efficient, and elastic cloud storage platform for Oracle database backups. The Client side Oracle Database Cloud Backup Module which is used with Recovery Manager (RMAN) transparently handles the backup and restore operations.

Oracle Database Cloud Backup Module is the cloud backup module that is installed in the database server. During the install process, a platform specific backup module is downloaded and installed. The RMAN environment of the client database is configured to use the cloud backup module to perform backups to ODBS. Using familiar RMAN commands, backups and restores are transparently handled by the cloud backup module.

### Objectives

* Install the Oracle Database Cloud Backup Module onto the VM image provided in the workshop. The database provided in the VM represents the on premise database in a typical customer situation.
* Configure RMAN to support the Oracle Database Cloud Backup Module. Then, backup the database and take a restore point to be used for Point-In-Time-Recovery.
* Simulate a destructive database operation and then restore and recover to a specific Point-In-Time.

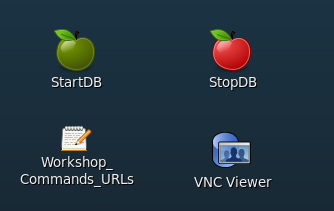
### Lab Requirements

* VNC Viewer to access the client system

### Oracle Public Cloud Backup Recovery

#### Start the On-Premise Oracle Database

* Access the Virtual Client image following the prior instructions regarding the VNC viewer.
* If your local database is not running for some reason (it should be at this point) locate and double-click the StartDB icon.



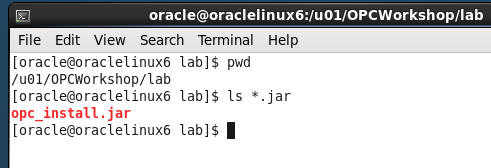
#### Install the Cloud Backup Module

* The .jar file (opc\_install.jar) used to install the Cloud Backup Module has already been placed into the /u01/OPCWorkshop/lab directory.
* Open a Terminal Window, cd into the lab directory and execute the following OS commands to verify that opc\_install.jar exists.

$ cd lab

$ pwd

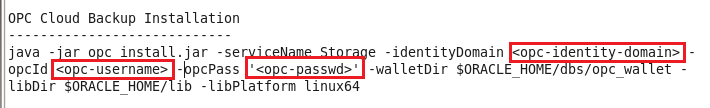
$ ls \*.jar



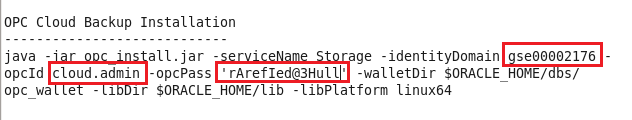
* The installation command with all of the options is rather lengthy. In order to make things easier for you and eliminate potential typos the installation command has been saved into a text file named Workshop\_Commands\_URLs.txt. The file is represented by an icon on the Client Image Desktop.
* Double click on the Workshop\_Commands\_URLs.txt icon to open up the file.



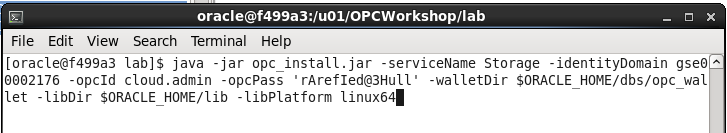
* Find the “**OPC Cloud Backup Installation**” section in the text file.
* Replace <opc-identity-domain> <opc-username> and <opc-passwd> (including replacing the <>) with the **Identity Domain, Username, and Password** student account information you were assigned. Also, be sure to put single quotes around your password to avoid any issues with special characters.
* Before:



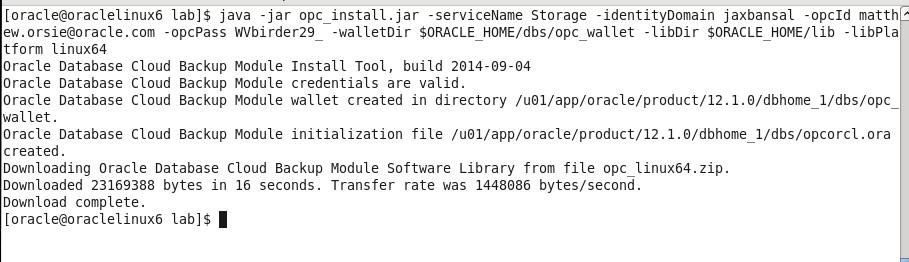
* After



* Copy and Paste the updated command from the text file into your terminal and hit Enter.



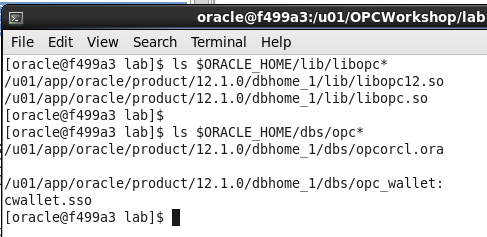
* The installation command creates a configuration file “**opcorcl.ora**” and wallet directory “**opc\_wallet**” and places these in $ORACLE\_HOME/dbs. It also downloads a library file “**opclib.so**” that RMAN uses to communicate with the Oracle Database Backup Service and places that in $ORACLE\_HOME/lib. You specified these locations in the syntax of the install command.



* Verify the required files have been created by entering the following commands

$ ls $ORACLE\_HOME/lib/libopc\*

$ ls $ORACLE\_HOME/dbs/opc\*



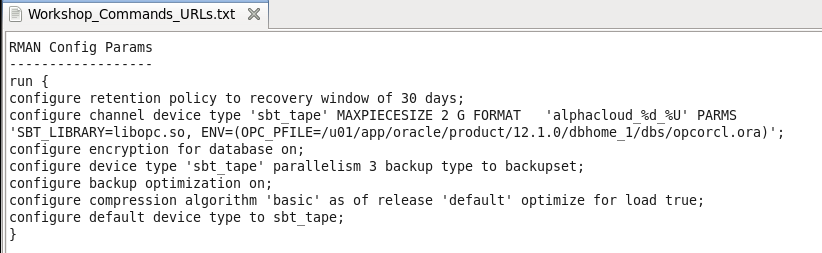
#### Configure RMAN to support Cloud Backups

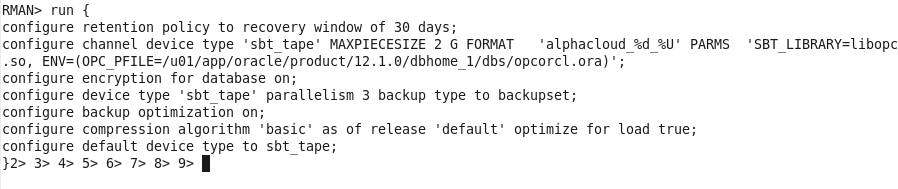
Before we can do backups to the Cloud storage location in your account, you need to configure a number of RMAN properties. These properties define:

* How long to retain the backups (30 days)
* Setting up a device type called “sbt\_tape” that uses the library and config files you just installed.
* Note that any defined storage chunks in the cloud will have names starting with “alphacloud\_”
* Turning on Encryption for data security. This is mandatory for an on premise to cloud backup scenario
* Set a degree of parallelism so that the backup/restore uses multiple threads. This is for performance.
* Setting backup optimization to ON so that RMAN will not unnecessarily transfer data to and from the cloud. (e.g. If a backup file is already present and has not had any changes before a “new” backup is performed, this file will not be dealt with, saving time).
* Setting a compression level for the files going to/from the cloud
* Configuring the sbt\_tape device as the default for all backups
* Connect RMAN to our local database using rman target /

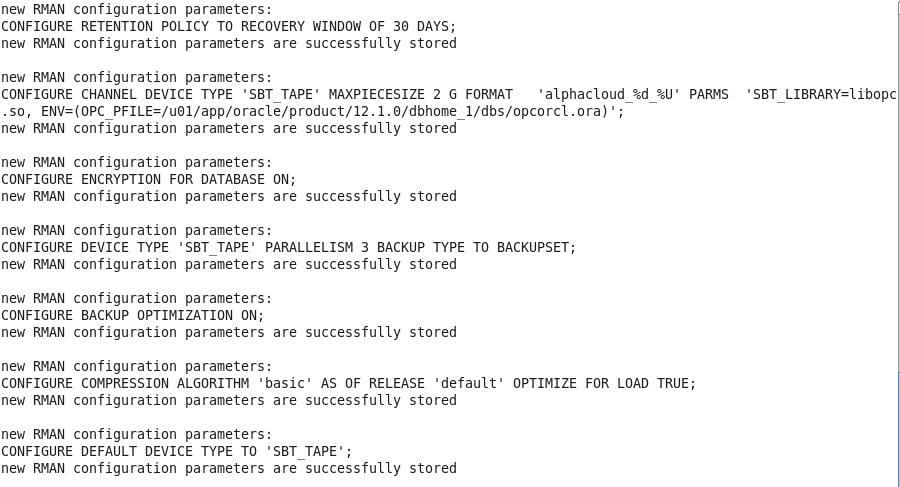


* Commands in RMAN can be run in blocks so you can do a sequence all at once.
* Copy and Paste from the entire run block in Workshop\_Commands\_URLS.txt under the RMAN Config Params section as shown below:

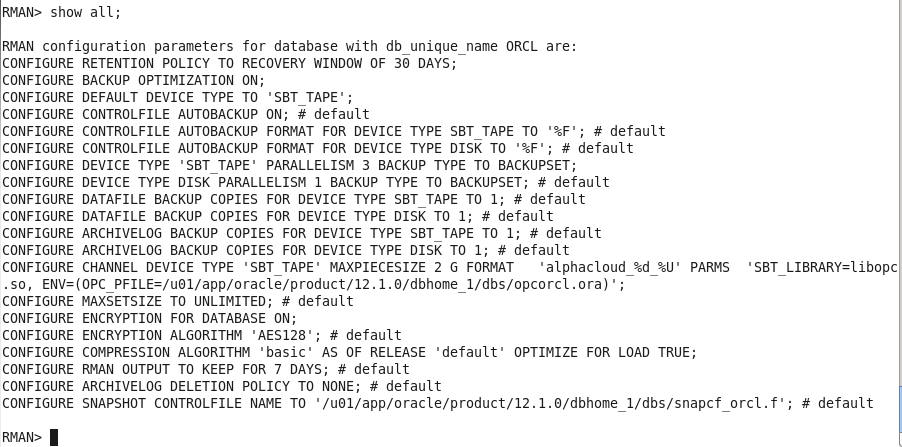




* Hit Enter and the parameters will be set to the following:



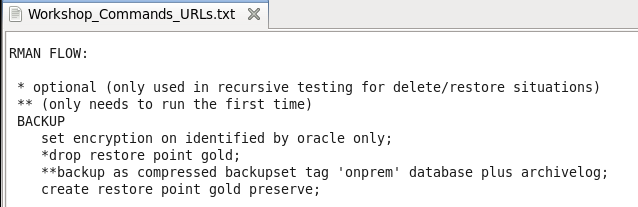
* Verify the changes in RMAN by typing show all;



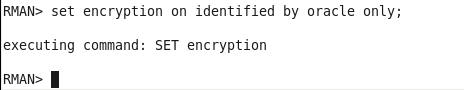
#### Backup the On Premise Database

For backup and recovery we would usually run the following sequence of commands from a shell script or an RMAN run block, but for lab illustration purposes we’ll copy and paste each individual command in sequence so you can get a better feel for what is going on.

* Under the RMAN FLOW 🡪 BACKUP section of the Workshop\_Commands\_URLS.txt file Copy the first line under the word BACKUP and paste it into your RMAN terminal session.

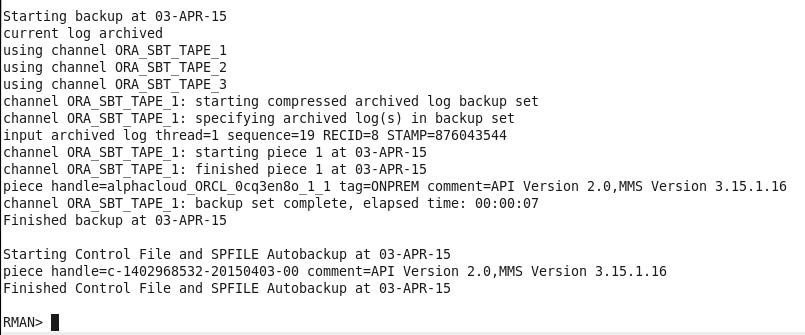


* RMAN> set encryption on identified by oracle only;

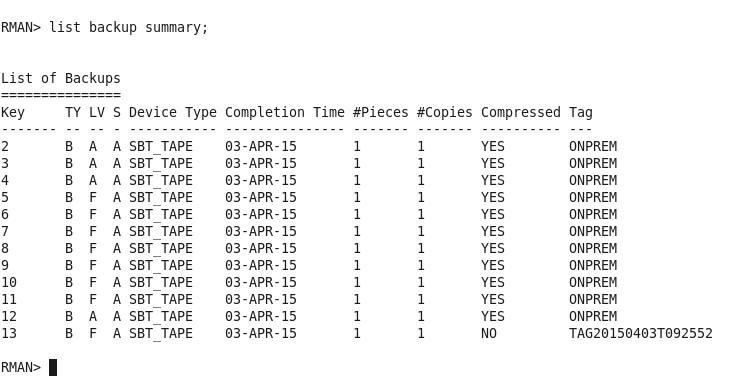


* For security reasons, backing up to the Oracle Public Cloud requires that encryption is used. The options are Transparent Data Encryption (TDE) and/or password encryption. We will be using password encryption in this lab.
* Copy and Paste the backup command (minus the asterisks)
* backup as compressed backupset tag 'onprem' database plus archivelog;
* The backup will commence. Depending on the speed of your network the backup job will take just a few minutes. The database control files and SPFILE are the last part to be backed up.

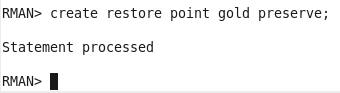
**NOTE:** If for some reason your backup does not finish properly because of network issues, there is a way to clean up the partial backup files and retry. This procedure is documented in the **Appendix** at the end of this lab.



* When creating a backup, the file chunks are placed in a user defined storage container in your account, OR they will be in a system generated container called oracle-data-storage-xxx. We can verify that the backup actually went to the cloud once the backup command has completed.
* You can use the RMAN list backup summary command to verify the backup files. Type the following command into the RMAN terminal session.
* list backup summary;



* Enter the following command at the RMAN prompt**:**
* create restore point gold preserve;



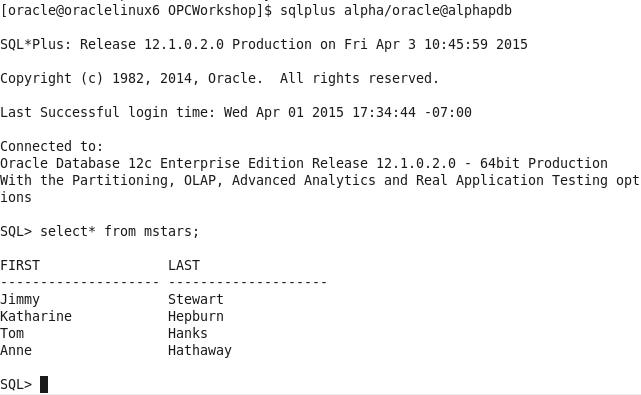
#### Oops…

* Now that we have a backup of our database we are going to “accidently” drop a table that will reappear once we perform the restore.
* Open up a new Terminal Window and use SQL\*Plus to connect to the **alpha schema** in the local AlphaPDB container database.

$ sqlplus alpha/oracle@alphapdb

* There is a table called **mstars** in the schema. Query the table to view the contents.

$ select \* from mstars;



* Drop the table with the drop table command

SQL> drop table mstars;

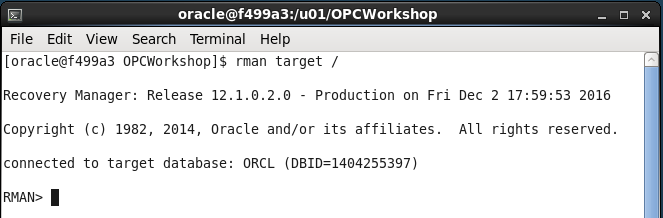


* Exit SQL\*Plus
* SQL> exit;

#### Restore and Recover the Database to a Point in Time

* In order to recover from the accidental table drop, we now need to restore the database to the point in time before the mstars table was accidentally deleted. We’ll use the cloud backup files to perform this restore.
* Open the RMAN session you used in the previous steps. If you’ve exited out of RMAN, it can be started again by entering the following within a terminal window:

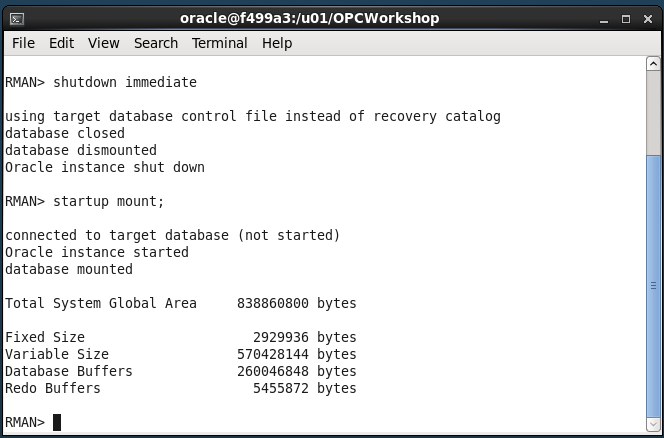
$ rman target /



Type the following commands to ready the database for a media recovery.

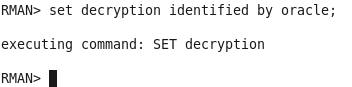
RMAN> shutdown immediate;

RMAN> startup mount;



* The next few steps will bring the entire database to a point where media recovery can occur. This takes the database offline. If you had multiple PDBs in the database and only needed to recover data in one PDB while leaving the others on-line, you could use the steps listed in the ALTERNATIVE RESTORE section in Workshop\_Commands\_URLS.txt under the RMAN FLOW heading. It takes a little longer using the ALTERNATIVE RESTORE method, so we’ll use the flow under the RESTORE section.
* Set the decryption user name by typing or copying the following command from within the RESTORE section of the Workshop\_Commands\_URLS.txt file.

RMAN> set decryption identified by oracle;



* Copy the run block and Paste it into the RMAN terminal session. The run command will perform the restore / recovery to our “gold” restore point. The final step opens the database and resets the logs since we’ve restored to a previous point in time.

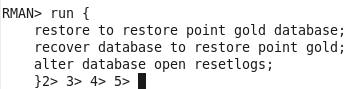
RMAN> run {

restore to restore point gold database;

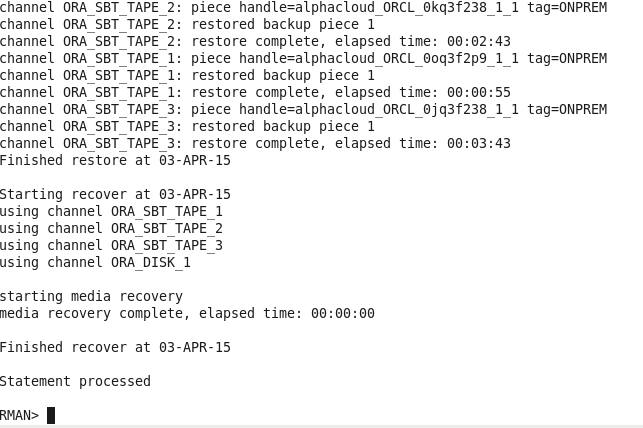
recover database to restore point gold;

alter database open resetlogs;

}



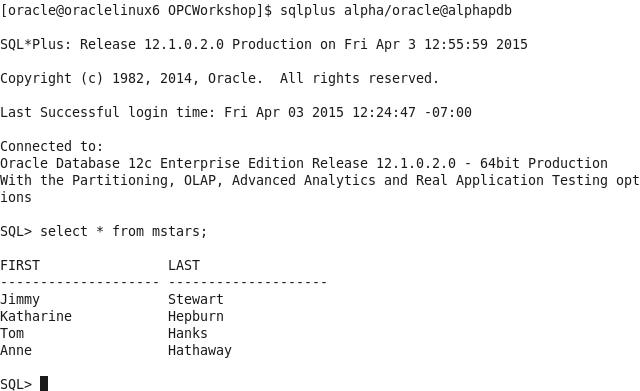
* Press Enter and the commands will be executed:



* Once the script completes, return to the Terminal Window that was used to connect with SQL\*Plus and connect back into the AlphaPDB container as alpha/oracle and run a query to see if the mstars table has been recovered.

$ sqlplus alpha/oracle@alphapdb

SQL> select \* from mstars;



This concludes lab 3 – Backup and Recovery. Proceed to the next lab when you’re ready.

**Appendix**

* In case your backup does not complete properly you can clean up the partial backupset and rerun the backup. You may have to wait a few minutes after the backup failure before the partial backup files can be deleted.
* Start up RMAN and type:
* RMAN> delete noprompt backupset tag ‘onprem’;
* Rerun the backup
* RMAN> backup as compressed backupset tag 'onprem' database plus archivelog;

# Database Development

### Introduction

In this lab you will deploy an APEX application to the Alpha Clone PDB and adjust the firewall rules to support access to the application from the Internet using a PC based browser or mobile device.

### Objectives

* Enable APEX in the Alpha Clone PDB.
* Create APEX REST services
* Deploy and access an Alpha Office APEX application.

### Lab Requirements

* The following lab assume that the steps outlined in lab guides 100 and 200 have been completed.
* VNC Viewer for access to the cloud client image
* The SSH tunnels must be active in a terminal window.

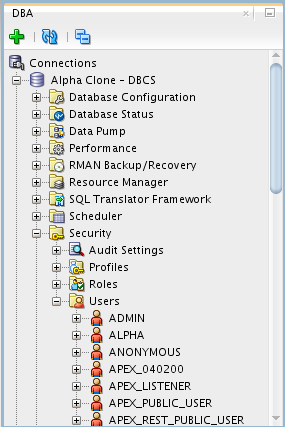
### Alpha Office and APEX

#### APEX Workspace Administration

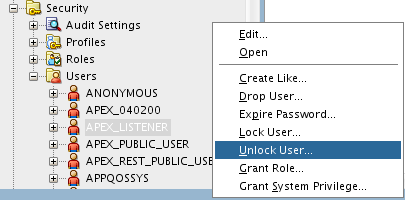
The Alpha Clone database contains an unused APEX configuration. As the first part of this lab we will complete the configuration of the cloned database APEX configuration.

**Note:** The standard install of APEX for a 12c database created many objects shared by both the container and pluggable database but user and password information is always local to the database we access. In other words, the APEX password we set in Lab 100 has not been in the cloned database.

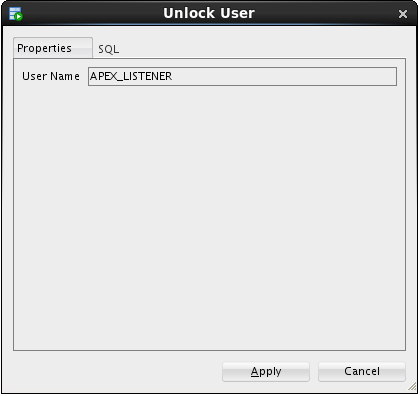
* Make sure the SSH tunnels you set up in lab 100 are still active in your terminal window, if not refer to lab 100 to set up the SSH tunnels.
* During the plug-in operation, many of the common objects in the pluggable database were evaluated by the database and some changes were made to the new database to work with its new container. One of these adjustments was locking the database accounts used to provide REST services. We will need to unlock the APEX\_LISTENER and APEX\_REST\_PUBLIC\_USER accounts.
* If it’s not already running, startup SQL Developer from the Cloud Client desktop on the VNC connection.
* Open the DBA Window and locate the Alpha Clone - DBCS item (created in Lab 200) in the **DBA Navigator**
* Expand it and click on the Security🡪Users item.



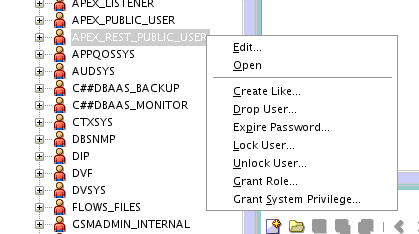
* Right-mouse on APEX\_LISTENER and select Unlock User…



* Click the **Apply** button to unlock APEX\_LISTENER. You may also use the SQL tab to review the unlock statement.

* Repeat the Unlock User… operation for the APEX\_REST\_PUBLIC\_USER.



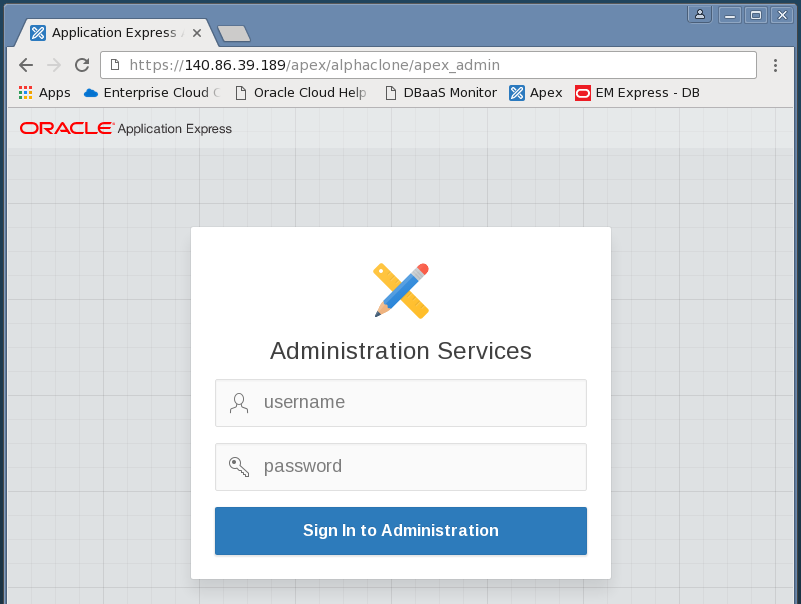


#### Create the Alpha Office workspace

* In the **Chrome** browser, open up a new tab and test the updated rule by accessing the APEX instance in the container database from the Internet. Use the Public IP address from the cloud instance we created in the first lab.

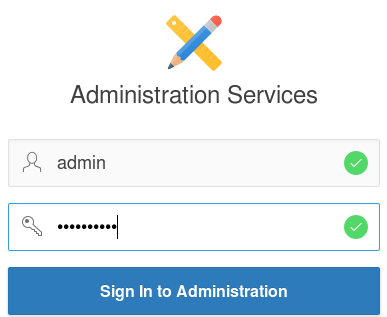
**Note:** Be sure to use the https protocol.

**https://<your-Public-IP>/apex/alphaclone/apex\_admin**



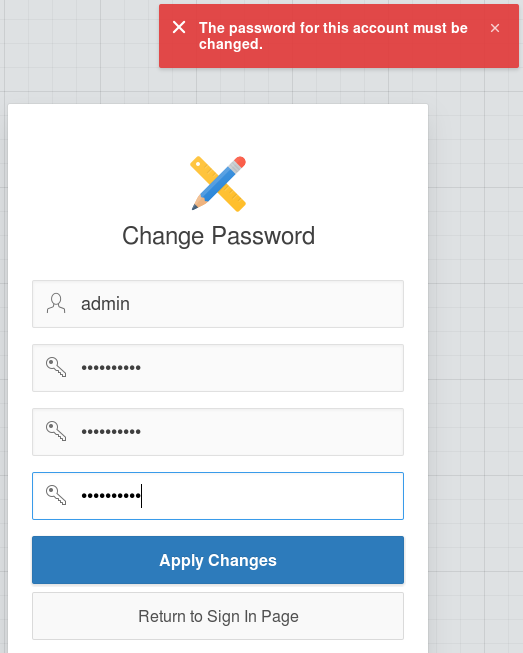
* After you’ve accepted the SSL certificate and see the APEX administration page, enter the following admin credentials and click the Login to Administration button:

|  |  |
| --- | --- |
| **Username:** | admin |
| **Password:** | Alpha2014\_ |

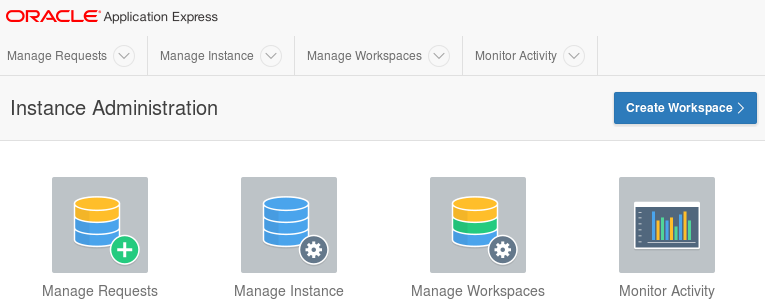


* You **may** be prompted to change the ADMIN user password, if not, skip to the next step. These credentials apply to the APEX objects local to the pluggable database. For convenience, we will enter the same password as the container database.
* Enter the following values and click the **Apply Changes** button.

|  |  |
| --- | --- |
| **Enter Current Password:** | Alpha2014\_ |
| **Enter New Password:** | Alpha2015! |
| **Confirm New Password:** | Alpha2015! |

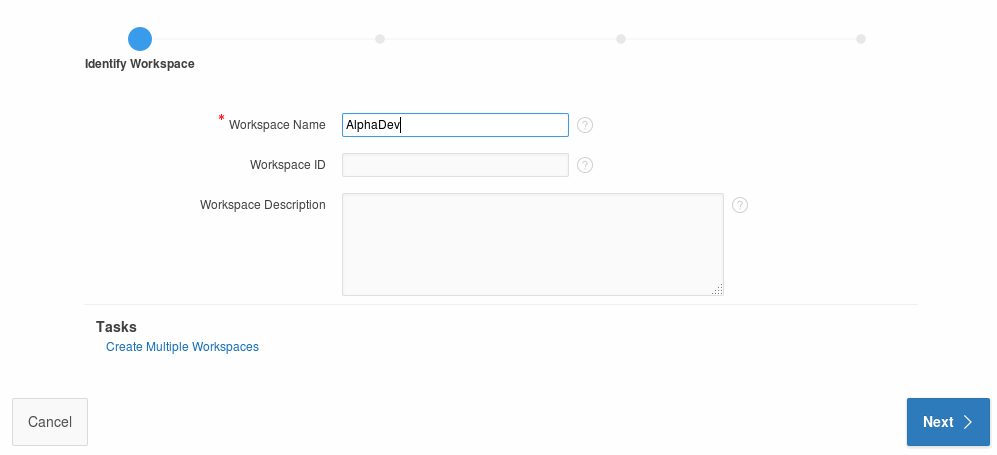


* After logging in successfully, feel free to click around in the APEX interface to get familiar with it.
* When you’re ready to begin, click the Create Workspace button



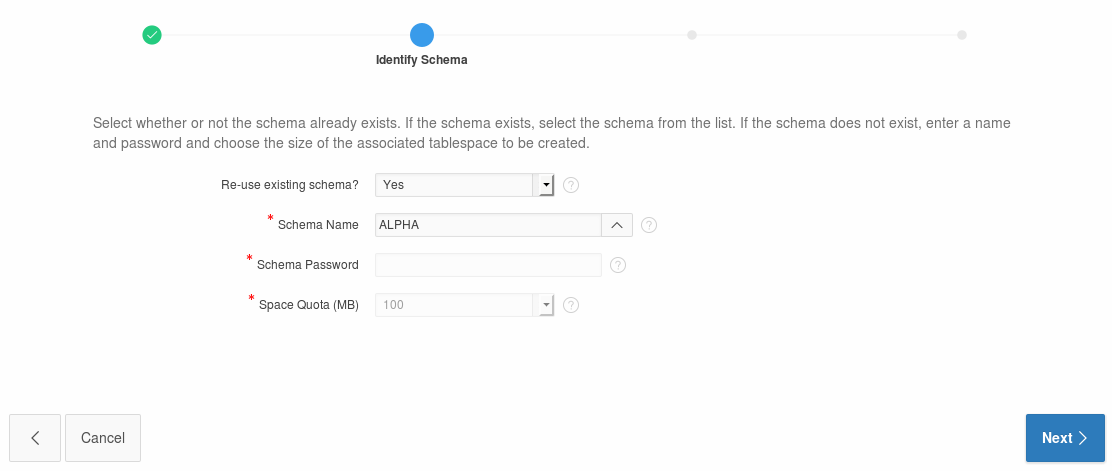
* At the Identify Workspace dialog, enter the following workspace name and click the Next button.

|  |  |
| --- | --- |
| **Workspace Name:** | AlphaDev |



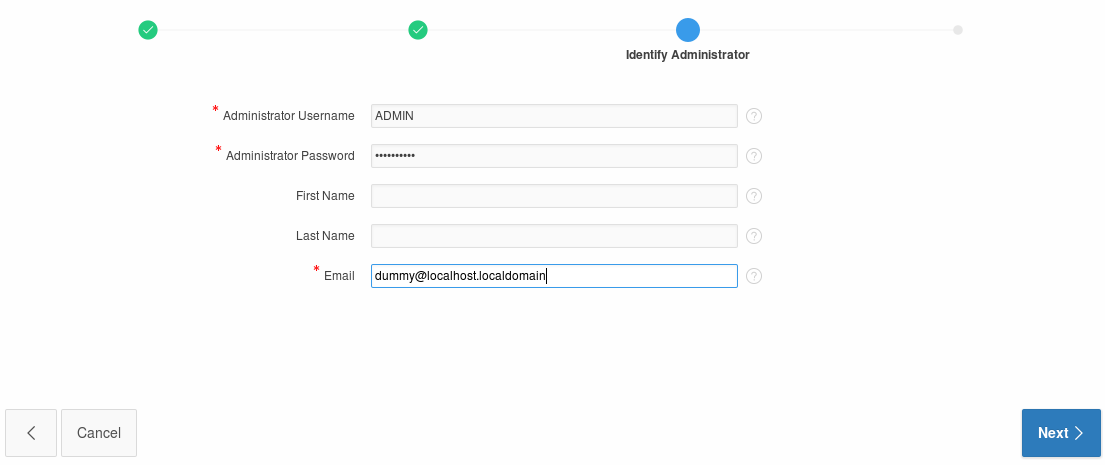
* At the Identify Schema dialog, select and enter the following values followed by the Next button.  
    
  **Note:** Use the search icon  to find the ALPHA schema.

|  |  |
| --- | --- |
| **Re-use existing schema?** | Yes |
| **Schema Name:** | ALPHA |

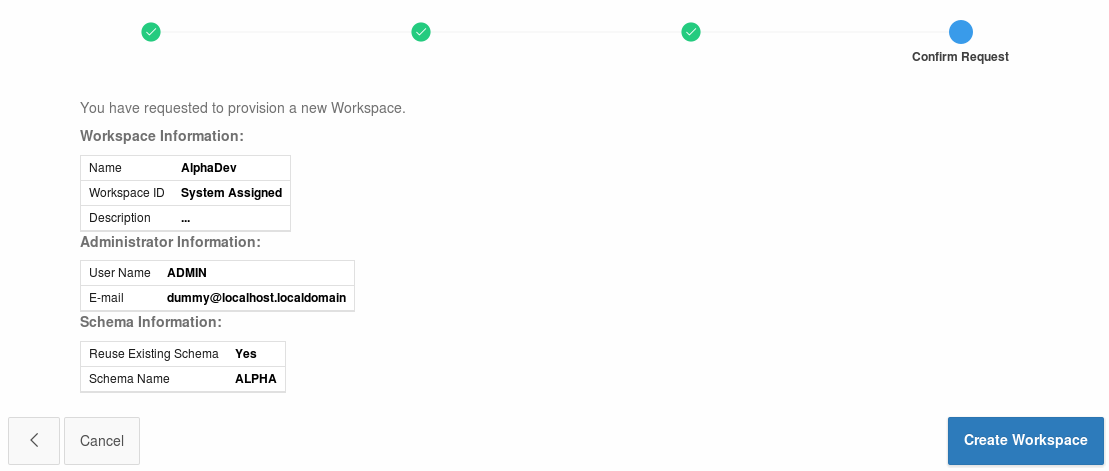
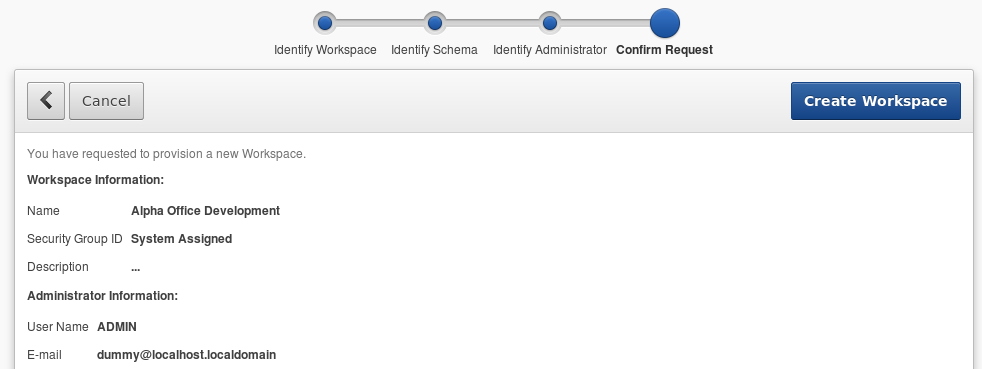


At the Identify Administrator dialog, enter the following values and click the Next button.

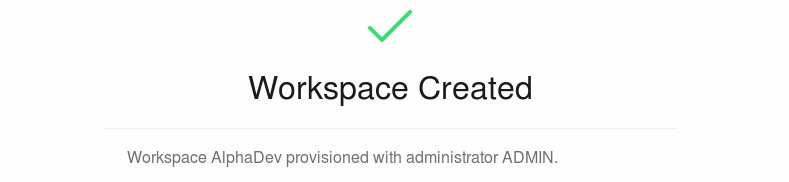
|  |  |
| --- | --- |
| **Administrator Username:** | ADMIN |
| **Administrator Password:** | Alpha2014\_ (May be prompted to change) |
| **Email:** | dummy@localhost.localdomain |



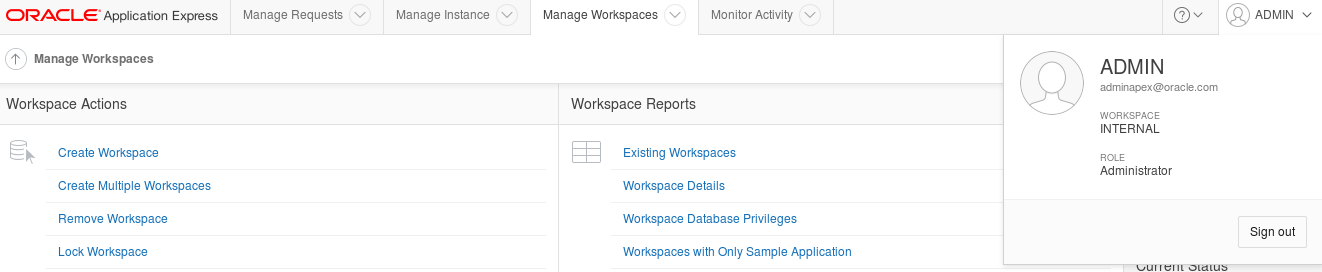
* Review the selections on the Confirm Request page and then click the Create Workspace button.

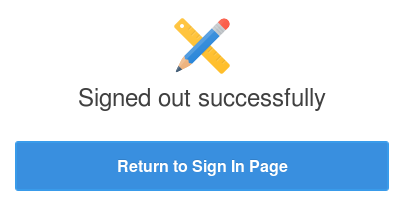
* APEX will display the ‘Workspace Created’ message
* Click Done



* Click the ADMIN dropdown in the upper right and select Signout



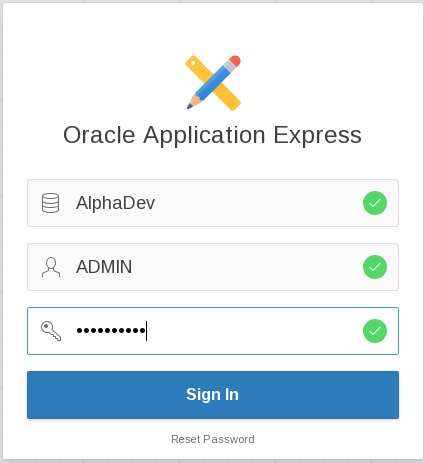
* Click the Return to **‘Sign In Page’** to continue



#### Build REST services

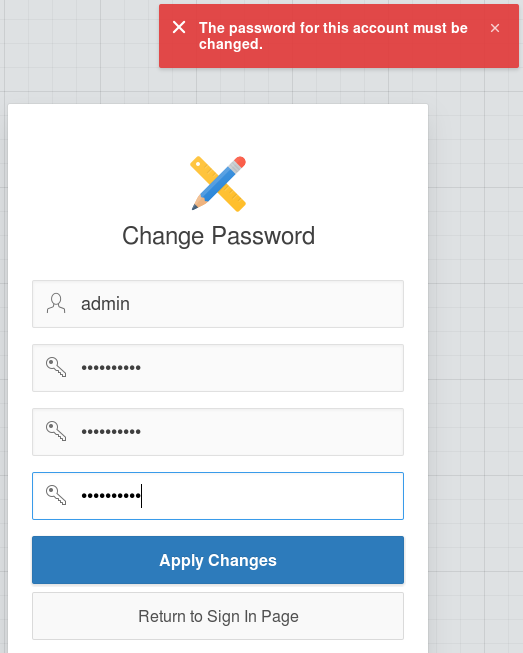
* Login to the Alpha Office APEX development workspace using the following credentials.

|  |  |
| --- | --- |
| **Workspace:** | AlphaDev |
| **Username:** | ADMIN |
| **Password:** | Alpha2014\_ |

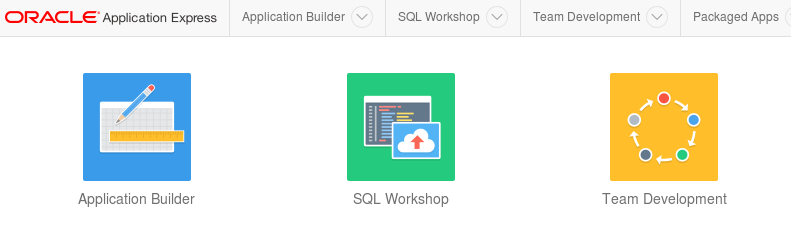


* You **may** be prompted to change your password. Enter the following values and click the Apply Changes button.

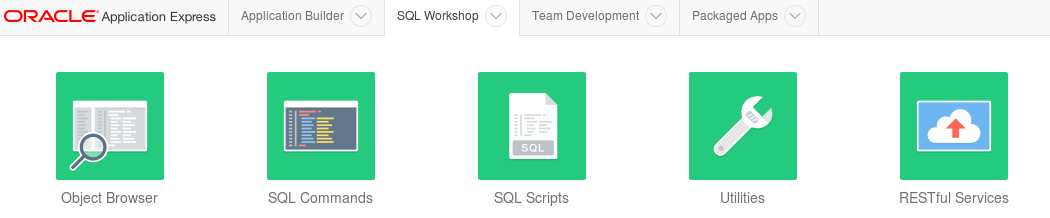
|  |  |
| --- | --- |
| **Enter Current Password** | Alpha2014\_ |
| **Enter New Password** | Alpah2015! |
| **Confirm New Password** | Alpha2015! |



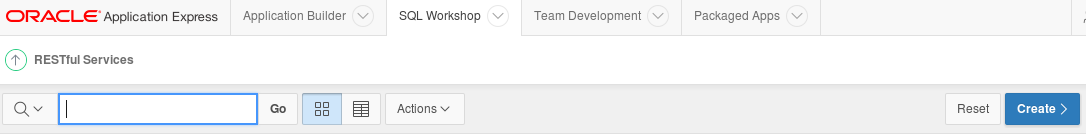
* Once you’ve logged in successfully, click the SQL Workshop button.



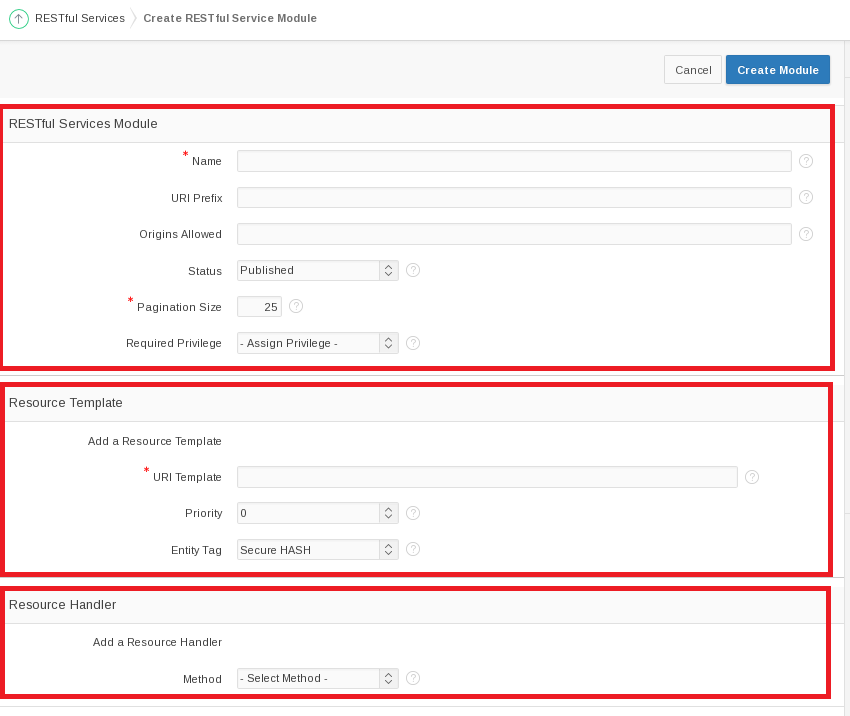
* Click the RESTful Services button.



* Click the Create > button

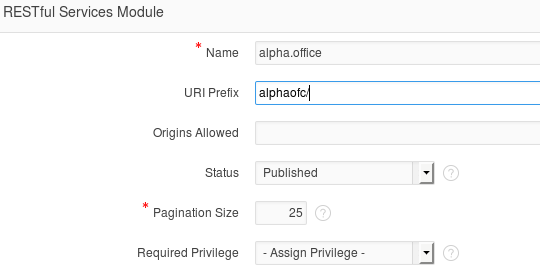


* There are three sections on the RESTful Services page:
* Restful Services Module
* Resource Template
* Resource Handler



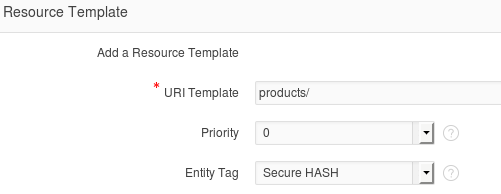
* Fill out the information for these sections using the information provided below.
* For the **RESTful Services Module** section, use the following values:

|  |  |
| --- | --- |
| **Name:** | alpha.office |
| **URI Prefix:** | alphaofc/ |



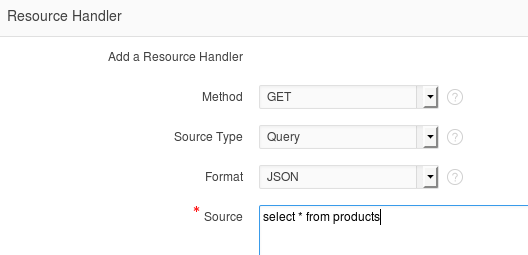
* In the **Resource Template** section enter the following value:

|  |  |
| --- | --- |
| **URI Template** | products/ |



* For the last section titled **Resource Handler** use the following values:

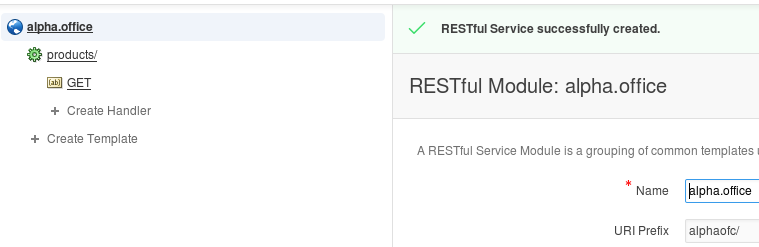
|  |  |
| --- | --- |
| **Method:** | GET |
| **Source Type:** | Query |
| **Format:** | JSON |
| **Source:** | select \* from products |



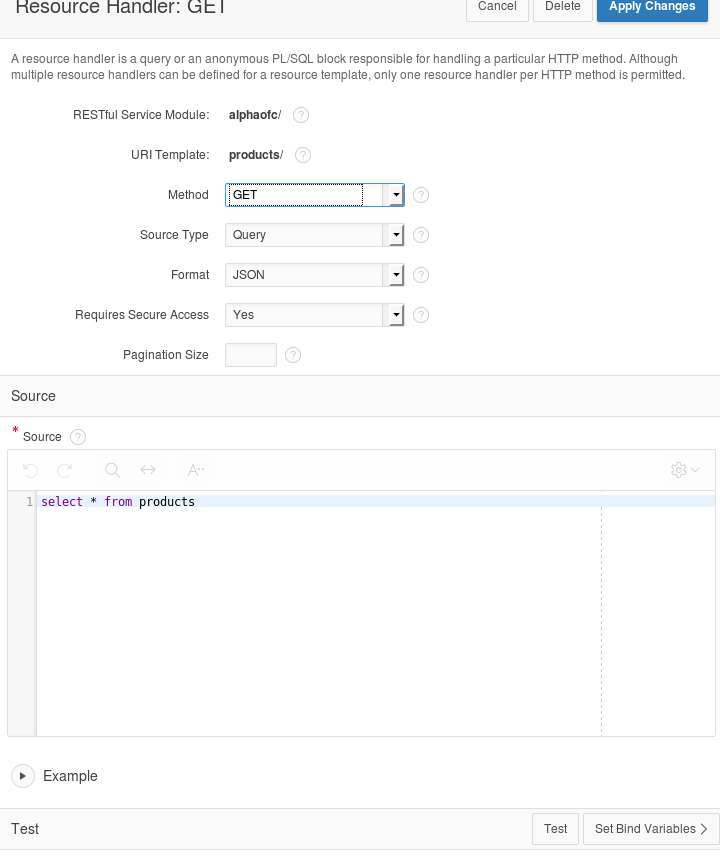
* Click Create Module to complete the REST service creation.



* APEX will show the new service module with a confirmation message.
* Click the GET handler for our template in the folder structure on the left of the screen.



* Review the definition.
* Since this operation has no parameters, we can easily test it by clicking the Test button.

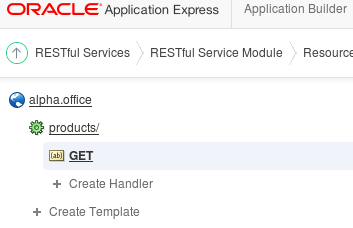


* Review the JSON produced by the service.
* Click the browser's back button to return to the APEX page.



#### Create a Parameterized REST Service

* In the next section we will create a REST service that takes a product number and returns only one database row as a JSON object.
* Click the Create Template link.

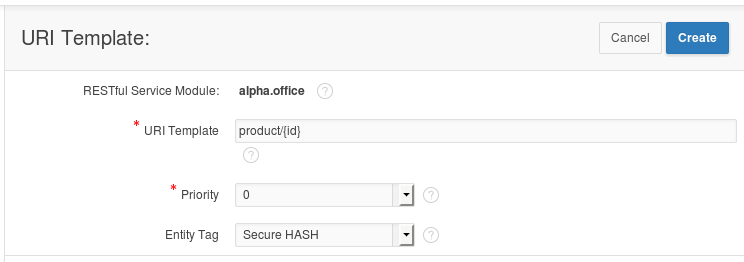


* Enter the following URI Template.

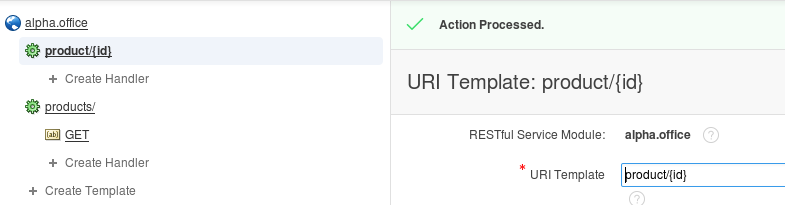
**Note:** The {id} syntax indicates the REST call accepts one parameter named "id" - this is automatically available in later for SQL queries.

* When the entry is complete, click the Create button.

|  |  |
| --- | --- |
| **URI Template:** | product/{id} |



* APEX displays a success message for the new template
* Click the Create Handler link under the product/{id} template on the left side of the screen.



* Enter the following SQL statement in the Source field of the Resource Handler page. Notice the use of the ":id" bind variable, this value comes from the URI template {id} provided when the service is invoked.

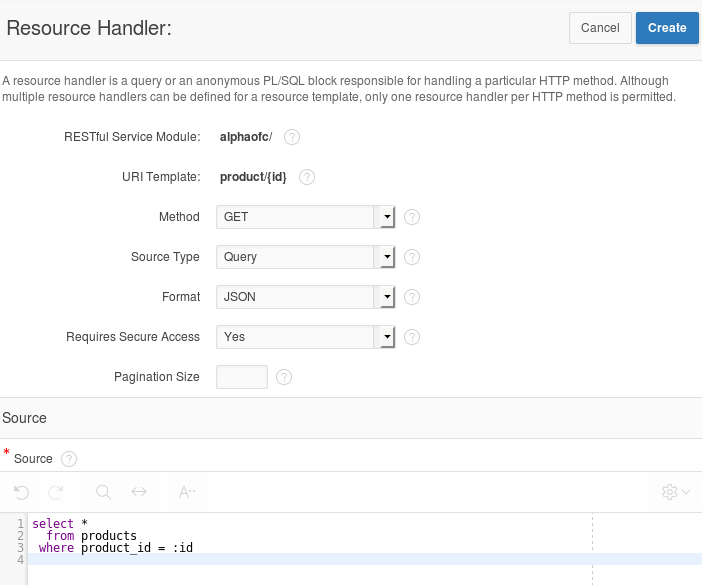
**Source:**

select \*

from products

where product\_id = :id

* Once you’ve finished entering the SQL statement, click the Create button.

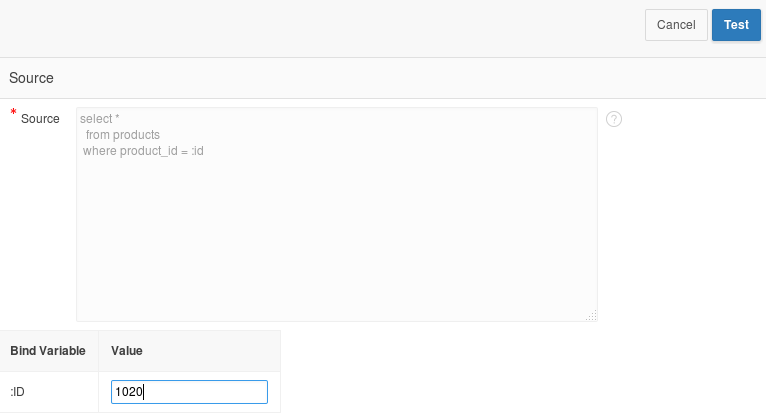


* Notice the ‘Action Processed’ at the top of your screen. We will test this service just like before, but we need to provide a product number to the call.
* Scroll to the bottom of the page and in the Test section, click the Set Bind Variables button.



* Enter the following product number and click the **Test** button.

|  |  |
| --- | --- |
| **:ID** | 1020 |

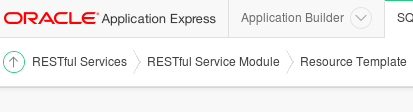


* In the new browser window, notice only the single product shows in the JSON object.
* Close this pop-up window.

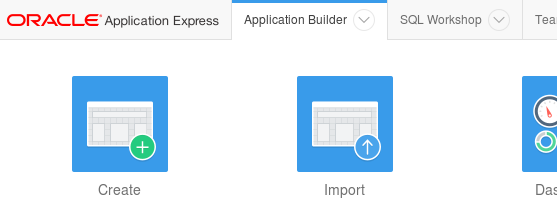


#### Install APEX Mobile Application

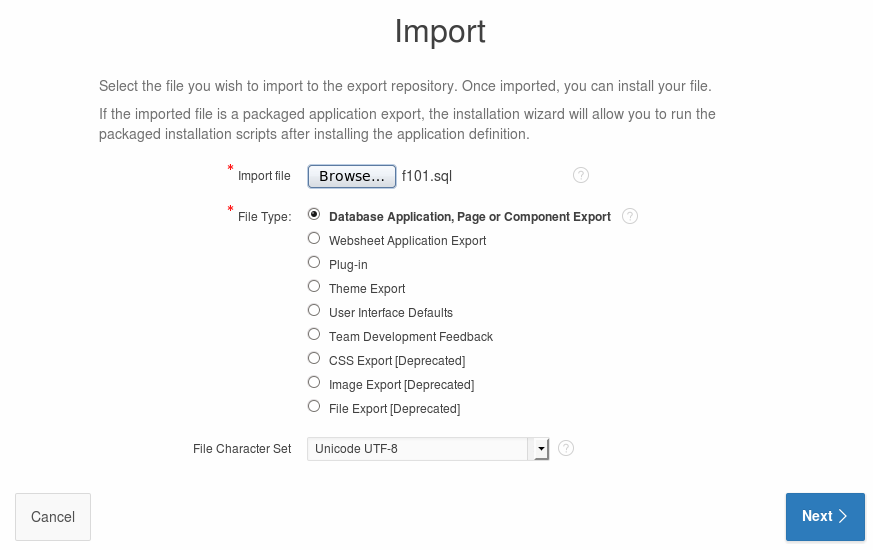
* Click the Application Builder menu item on the APEX page.



* Click the Import button on the Application Builder page.

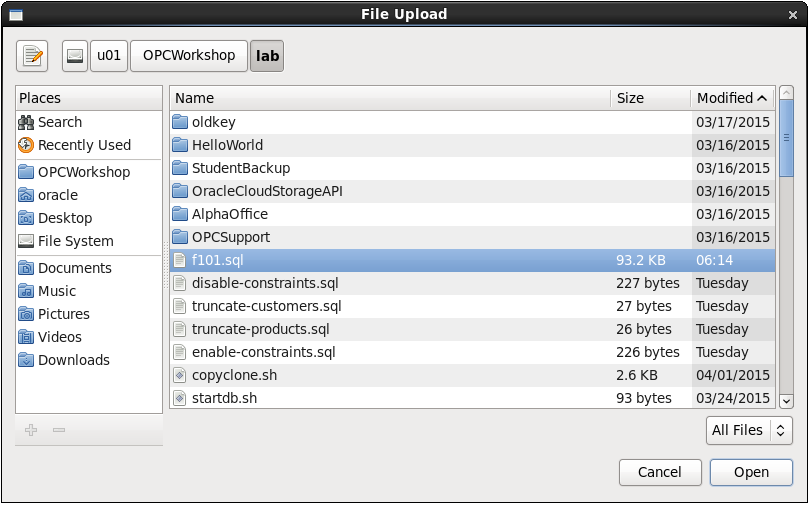


* Click the Browse button to locate the APEX application export file.



* Locate and open the following file and click the Open button:

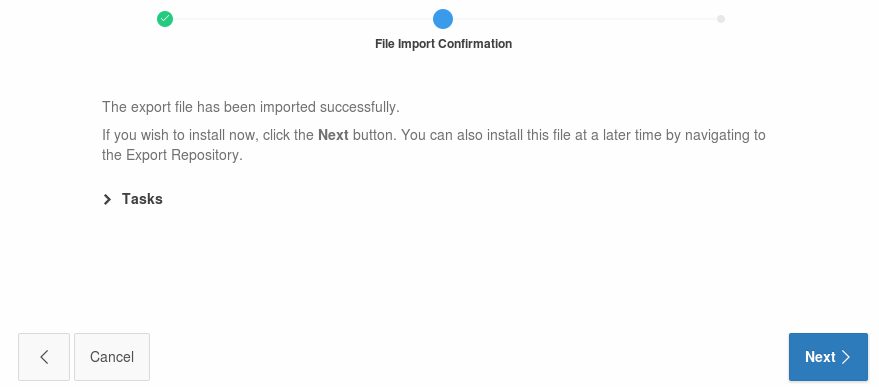
**/u01/OPCWorkshop/lab/f101.sql**

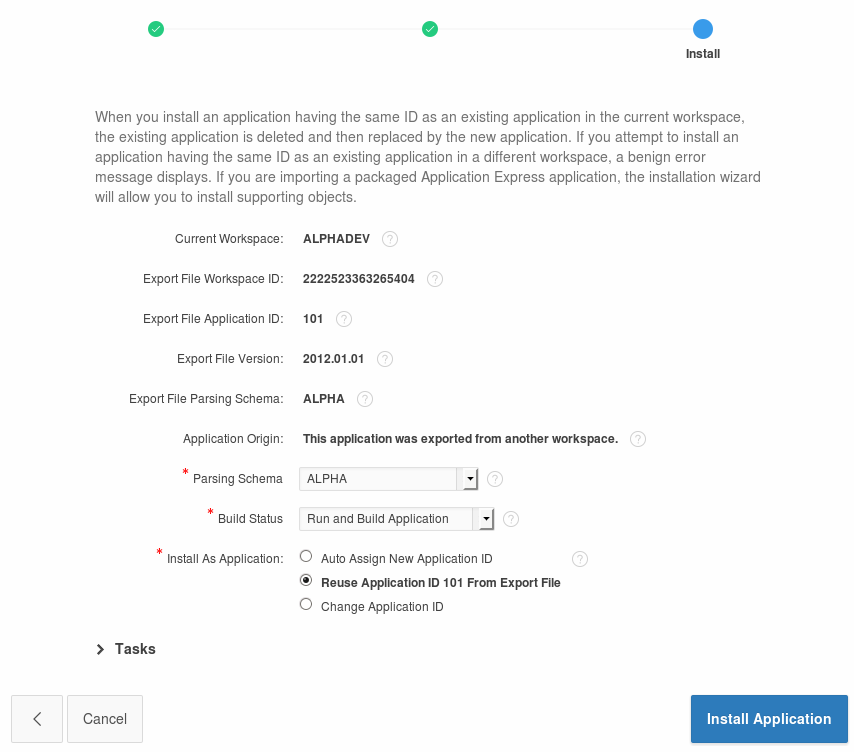


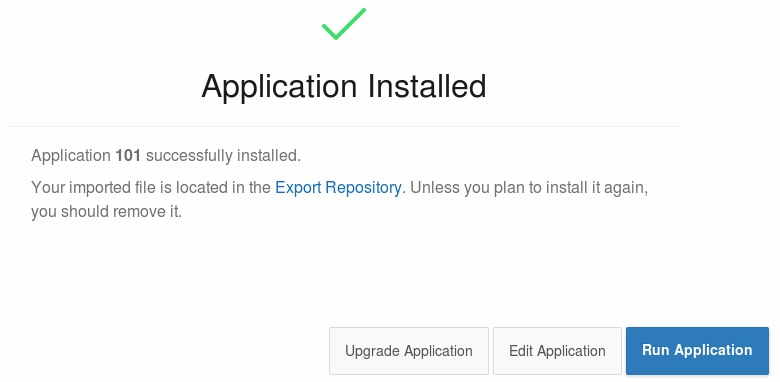
* Click the Next button to continue.



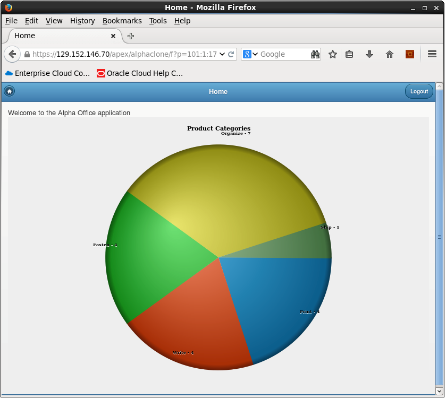
* After a brief pause while the application file is processed, click the Next button to continue.



* On the final page, select to Reuse Application ID 101 from Export File
* Click Install Application. 
* APEX displays a success message for the import
* Click **Run** Application.



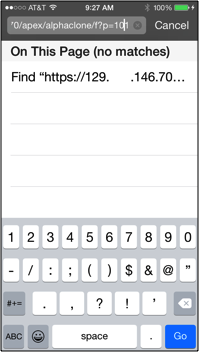
* APEX renders the first page of the mobile application in the browser – it might not look quite right since we are using a mobile template.



#### Access the Alpha Office Mobile Application on your Smart Device

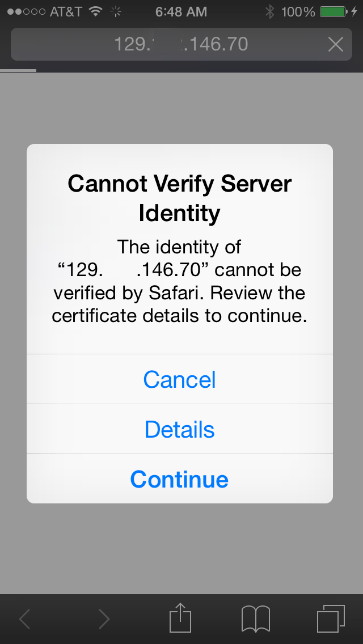
* Using any Internet connected smart phone or tablet we will access the mobile application using the port we opened earlier in the lab. This example is using an Apple iPhone 5s.
* Use your device's browser and navigate to the following URL:

**https://< Public IP Address of Alpha01A-DBCS>/apex/alphaclone/f?p=101**

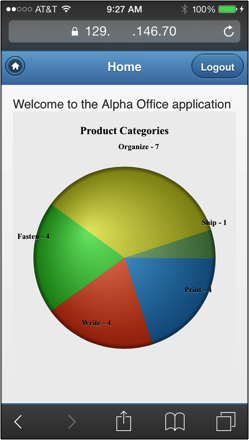


***Note:*** *It may be more convenient to email the link to your device.*

* The browser should prompt you to accept the unknown certificate.
* Click of touch Continue.



* Touch the screen to explore the application. On the device, touching one of the pie slices highlights the slice; a second tap drills into that slice.



* Congratulations, you’ve created an application on the Oracle Database Cloud. This is the final lab for the DBCS Workshop.