

Accelerate Innovation
Oracle Cloud Platform as a Service

Overview Data App Dev Integration Management Content Analytics Security Dev Portal What's New Customers

Oracle Data Management Cloud Workshop

Database Cloud Service

Updated: July 12, 2017



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Oracle Training Materials – Usage Agreement

Use of this Site ("Site") or Materials constitutes agreement with the following terms and conditions:

1. Oracle Corporation ("Oracle") is pleased to allow its business partner ("Partner") to download and copy the information, documents, and the online training courses (collectively, "Materials") found on this Site. The use of the Materials is restricted to the non-commercial, internal training of the Partner's employees only. The Materials may not be used for training, promotion, or sales to customers or other partners or third parties.
2. All the Materials are trademarks of Oracle and are proprietary information of Oracle. Partner or other third party at no time has any right to resell, redistribute or create derivative works from the Materials.
3. Oracle disclaims any warranties or representations as to the accuracy or completeness of any Materials. Materials are provided "as is" without warranty of any kind, either express or implied, including without limitation warranties of merchantability, fitness for a particular purpose, and non-infringement.
4. Under no circumstances shall Oracle or the Oracle Authorized Delivery Partner be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this Site of Materials. As a condition of use of the Materials, Partner agrees to indemnify Oracle from and against any and all actions, claims, losses, damages, liabilities and expenses (including reasonable attorneys' fees) arising out of Partner's use of the Materials.
5. Reference materials including but not limited to those identified in the Boot Camp manifest cannot be redistributed in any format without Oracle written consent.

Contents

Section 1: Database Cloud Service Overview	4
1.1: Introduction.....	4
1.2: Objectives	4
1.3: Lab Requirements	4
1.4: Retrieve Public IP for Client Image.....	5
1.5: Create Database Cloud Service Instance	13
1.6: Configuration and Image Exploration	25
Section 2: Cloud Migration	41
2.1: Introduction.....	41
2.2: Objectives	41
2.3: System Requirements	41
2.4: Cloud Migration Using Pluggable Databases	41
2.5: Cloud Migration Using Data Pump	65
2.6: Cloud Migration Using SQL Developer Carts	79
Section 3: Backup and Recovery	88
3.1: Introduction.....	88
3.2: Objectives	88
3.3: Lab Requirements	88
3.4: Oracle Public Cloud Backup Recovery	88
Section 4: Database Development.....	101
4.1: Introduction.....	101
4.2: Objectives	101
4.3: Lab Requirements	101
4.4: Alpha Office and APEX.....	101

Section 1: Database Cloud Service Overview

1.1: Introduction

This lab will help give you a basic understanding of the Oracle Database Cloud Service and its capabilities around administration and database development.

We will walk through creating a new Database Cloud Service instance. After the database has been created, you will connect into the Database image using an SSH private key and familiarize yourself with the image layout. Next you will learn about SSH tunneling using an SSH configuration file. This file will be used to tunnel multiple ports to a remote OPC instance. Using the tunnels, you will learn how to access various Database consoles.

1.2: Objectives

- ✓ Create Database Cloud Service
- ✓ Configure security with SSH
- ✓ Explore VM and cloud consoles

1.3: Lab Requirements

- ✓ VNC Viewer to connect to an Image running on Oracle's IaaS Compute Service.
- ✓ Laptop capable of connecting to the internet and running VNC Viewer
- ✓ Cloud environment access details provided by instructor in advance of the class

Note: Use the table below and fill in the blanks as you go through the labs. Create a text file or note on your VNC desktop to keep track of important information you'll need throughout the lab exercises so that you can easily copy and paste the information.

Cloud Data Center:	
Identity Domain:	
Login Username (s):	
Login Password:	
Alpha01A-DBCS Cloud Public IP Address:	
Alpha01A-DBCS Cloud Private IP Address:	
Site Location ID:	
Alpha01B-DBCS Cloud Public IP Address:	
Alpha01B-DBCS Cloud Private IP Address:	
Client Public IP Address:	

Client Private IP Address:	
VNC Viewer Port #:	
Client Image VNC Viewer Password:	

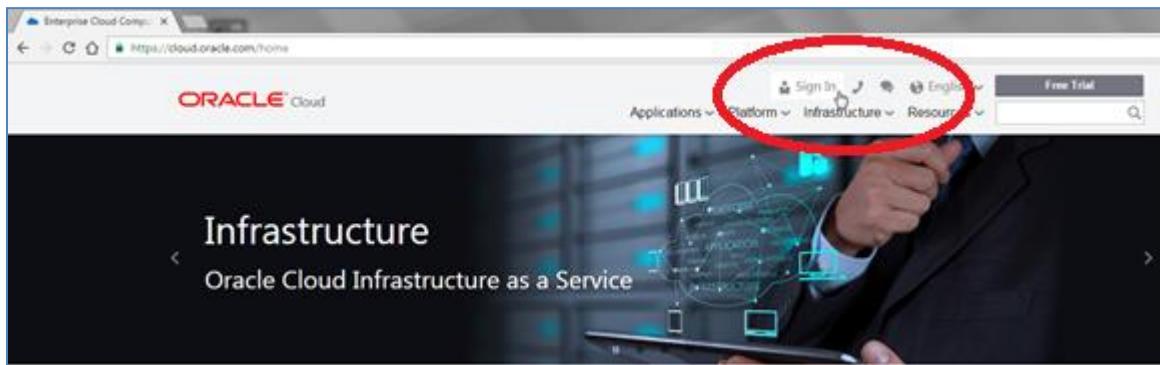
1.4: Retrieve Public IP for Client Image

For the Database Cloud Service Workshop we will be using a Client Image running in the Oracle Compute cloud to simulate the on premise environment. This client image is running Linux and contains a preinstalled Oracle 12.1.0.2 database with a pluggable database that we will migrate to the Oracle Public Cloud Database instance. The client image contains SQL Developer 4.1, SQL*Plus and other utilities that will be used to connect and manipulate both the local and cloud database instances.

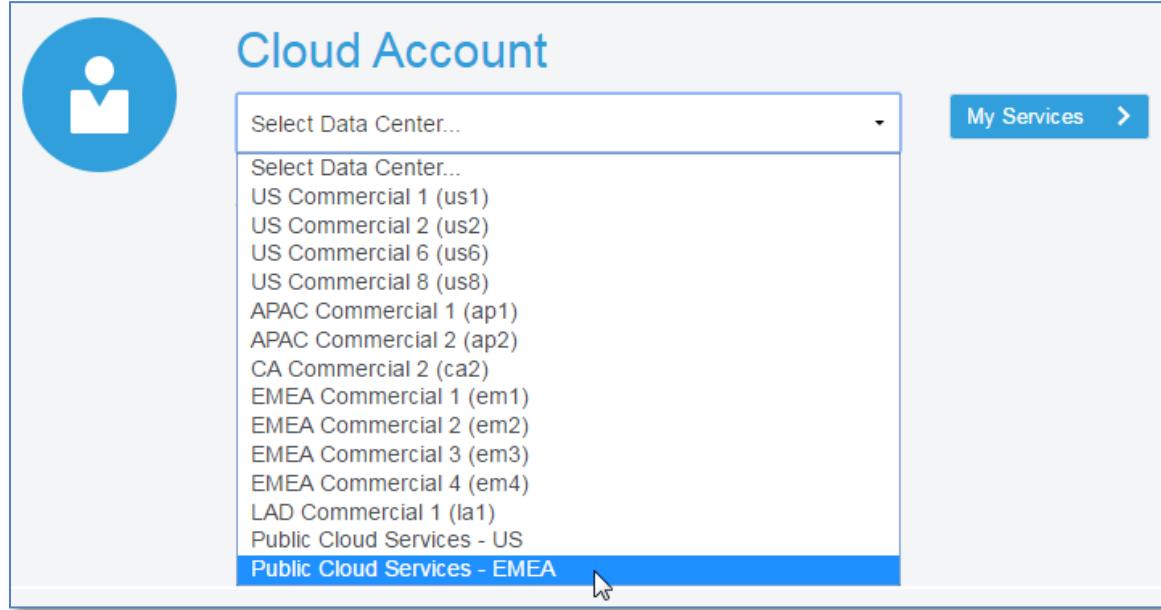
The Client Image is a VM that is running on Oracle's IaaS Compute service.

1.4.1: RETRIEVE PUBLIC IP FOR CLIENT IMAGE

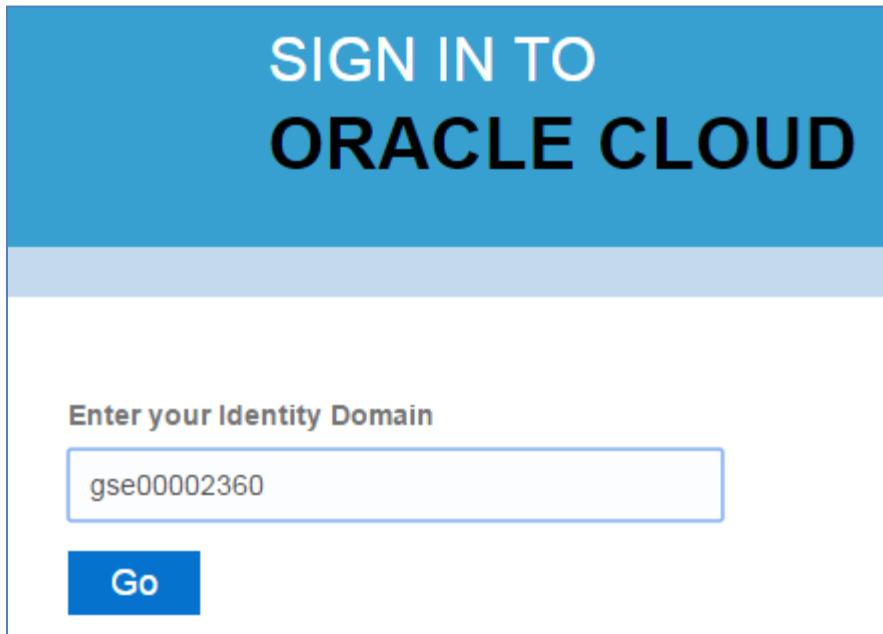
- Login to your Oracle Cloud account
- Open a browser and go to the following URL: <https://cloud.oracle.com>
- Click **Sign In** in the upper right hand corner of the browser



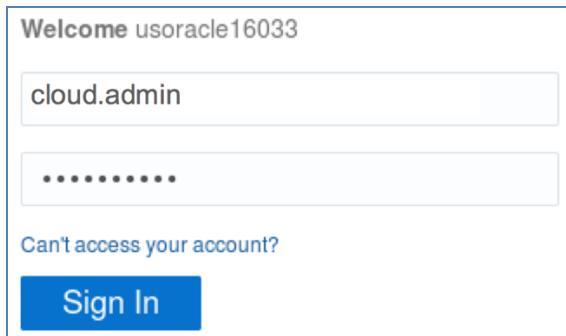
- Under My Services > Select Data Center ... select the region from the drop down list. Your instructor will provide this information to you prior the course.



- Enter the **identity domain** and click **Go**

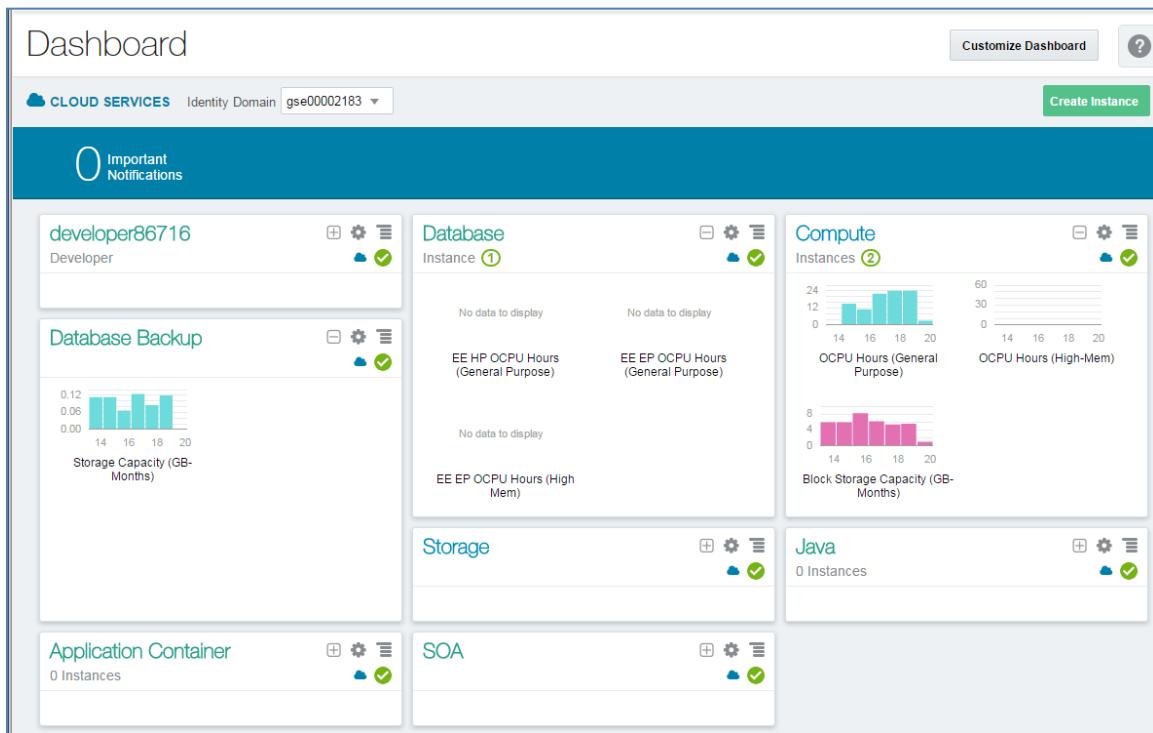


- After entering the Identity Domain you will enter the User Name and Password
- Click **Sign In**

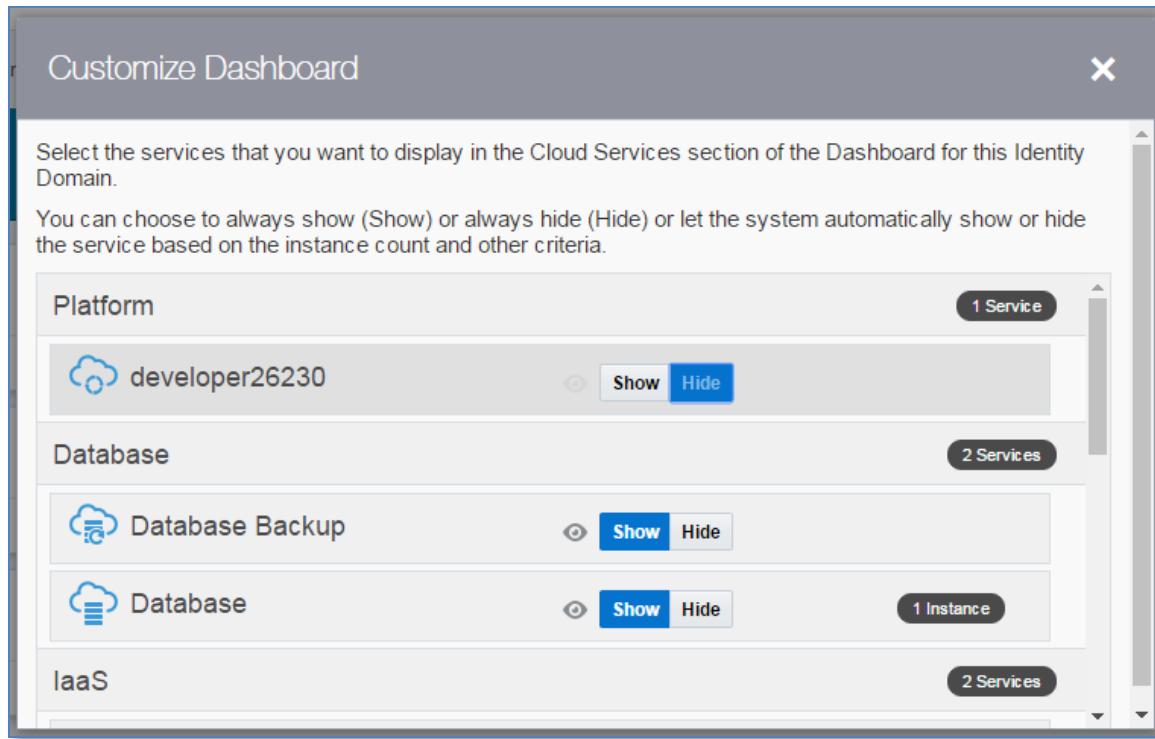


NOTE: The **Identity Domain**, **User Name** and **Password** values were provided to you by your instructor prior to starting this lab.

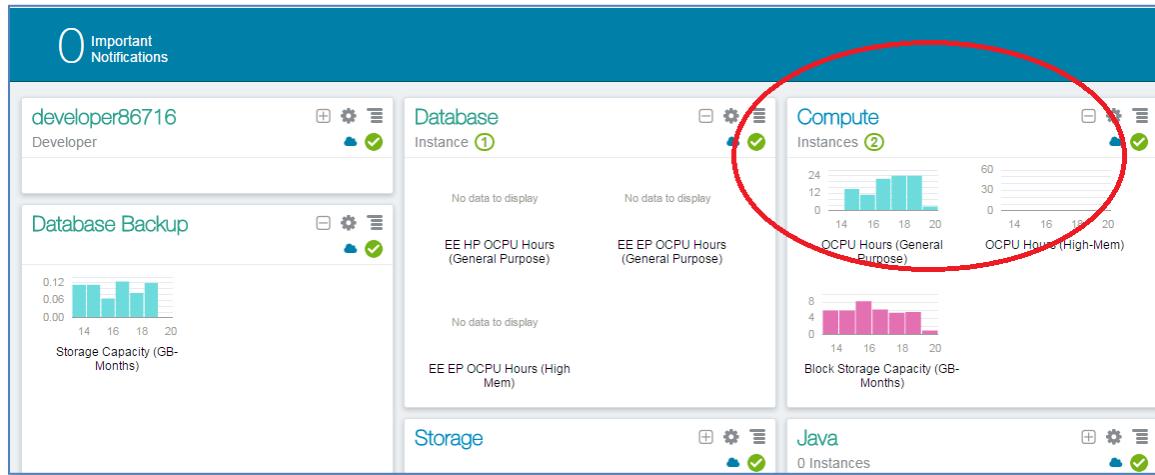
- ✓ You should see the Dashboard summarizing all of your available services. Compute, backup, and storage are all related to the Database Cloud Service.



- If there's a service that's not visible, **click** on the **Customize Dashboard** dropdown and add each service as a **favorite** by clicking on the **Show** button next to the service.



- From the main dashboard, click on the **Compute** service link to access the Compute Service Console. This is where we will get the IP address of our database service.



- From this page you can view general information about this Compute Cloud Service. Click on the **Open Service Console** button.

Service Details: Oracle Compute Cloud Service

Overview

Additional Information

Plan:	Oracle Compute Cloud Service	CSI Number:	Not available
Service Start Date:	Jul 27, 16	Data Center:	EMEA Commercial 2 - Amsterdam
Service End Date:	Jul 27, 18	Status:	Active
Subscription ID:	536845894	Domain SFTP Host & Port:	sftp2.em2.cloud.oracle.com:22
Service Instance ID:	536847002	Domain SFTP User Name:	em237317 ?
Customer Account:	gse00002176 (US)	REST Endpoint:	https://api-z17.compute.em2

Billing Metrics

Resource Quotas

Monitoring Metrics

Documents

- The Compute Service Console will give you a summary of the resources your cloud service is using and a list all running VM's.

ORACLE® CLOUD My Services

Compute

Summary

2 instances	2 OCPUs	15 GB memory	185 GB volume size in use
-------------	---------	--------------	---------------------------

Instances

An Oracle Compute Cloud Service instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. [Learn more](#).

Name	Status	OCPUs	Memory	Volumes	Public IP	Private IP
Alpha01A-DBCS/db_...	Running	1	7.5 GB	185 GB	140.86.32.50	10.196.172.106
ClientImage01	Running	1	7.5 GB	None	140.86.35.187	10.196.198.242

- Identity Domains have multiple sites. If you don't see your VM images in the Compute Console, you may be in the wrong site.

The screenshot shows the Oracle Compute Cloud Service interface. At the top, there's a navigation bar with tabs: Compute (selected), Instances, Network, and Storage. Below the navigation bar is a summary section with four metrics: instances (0), OCPUs (0), memory (0), and volume size in use (0). To the right of the summary are three icons: a question mark, a speech bubble, and a phone. The main content area is titled 'Instances' and contains a sub-instruction: 'An Oracle Compute Cloud Service instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. Learn more.' Below this are search and filter controls (Category: All, Show: All) and a message: 'No instances available matching the filters.' There's also a small placeholder icon for instances.

- Please ask your instructor which site the Client Image is running on. If necessary, click the **Site** dropdown at the top of the page to access the **Site Selector** and choose the correct site.

The screenshot shows the Oracle Cloud My Services interface. At the top, there's a header with the Oracle logo and the text 'CLOUD My Services'. To the right of the header are several buttons: 'Dashboard', 'Users', 'Notifications', and 'Monitoring'. Above these buttons is a dropdown menu labeled 'Site: EM002_Z17'. This dropdown is highlighted with a red circle.

The screenshot shows the 'Site Selector' dialog box. At the top, it displays the current site selection: 'Site: EM002_Z17'. Below this is a 'Select a Site' dropdown menu with three options: 'EM002_Z17' (selected), 'EM002_Z16', and 'EM002_Z17'. Underneath the dropdown are two other dropdown menus: 'Data Center:' (set to 'EM002_Z16') and 'REST Endpoint:' (set to 'http://10.0.0.1:8080'). Below these dropdowns is a section titled 'Site Usage' containing three progress bars: 'OCPU' (59.8%), 'Memory' (36.1%), and 'IP Reservations' (26.9%). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- Once the correct site has been selected, locate the instance named **ClientImage01** and copy the **Public IP**.
- **Note:** Keep this IP Address somewhere accessible, like a notepad or text document. We will use this IP with VNC to connect to the cloud client desktop.

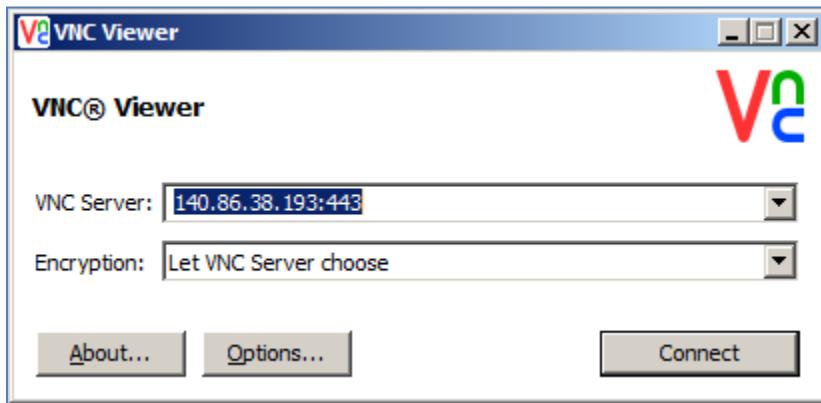
Instances						
An Oracle Compute Cloud Service instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. Learn more .						
Name	Status	OCPUs	Memory	Volumes	Public IP	Private IP
Alpha01A-DBCS/db_...	Running	1	7.5 GB	185 GB	140.86.32.50	10.196.172.106
ClientImage01	Running	1	7.5 GB	None	140.86.35.187	10.196.198.242

1.4.2: CONNECT TO CLIENT IMAGE USING VNC VIEWER

- From your desktop run the VNC Viewer application and enter the Public IP address you just obtained, with a display port separated from the IP address by a colon <:443> and click **Connect**

Note 1: If connecting inside an Oracle office through the ‘clear-guest’ network, and the VNC session won’t connect or times out, try port :10.

Note 2: If you do not already have a VNC Viewer installed on your computer you will need to download it. Or ask the instructor for the Real VNC Viewer executable.



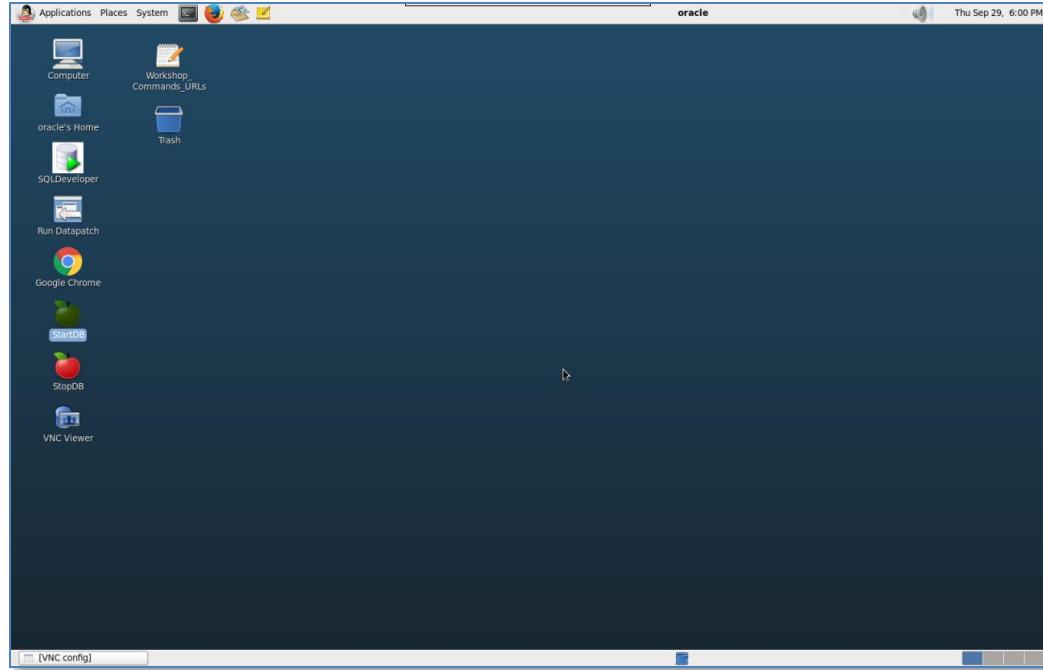
- Click Continue on the encryption message.



- Enter the password supplied by your instructor and click **OK**



- Verify that you can see and interact with the Linux desktop. You are now connected to the Client Image that will be used for all labs.

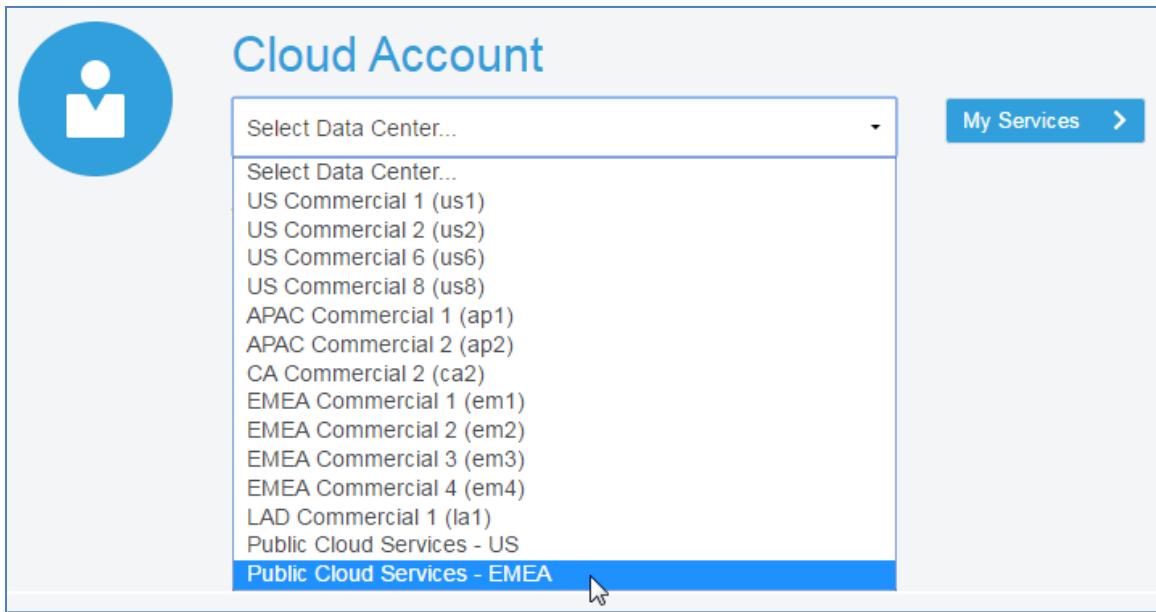


1.5: Create Database Cloud Service Instance

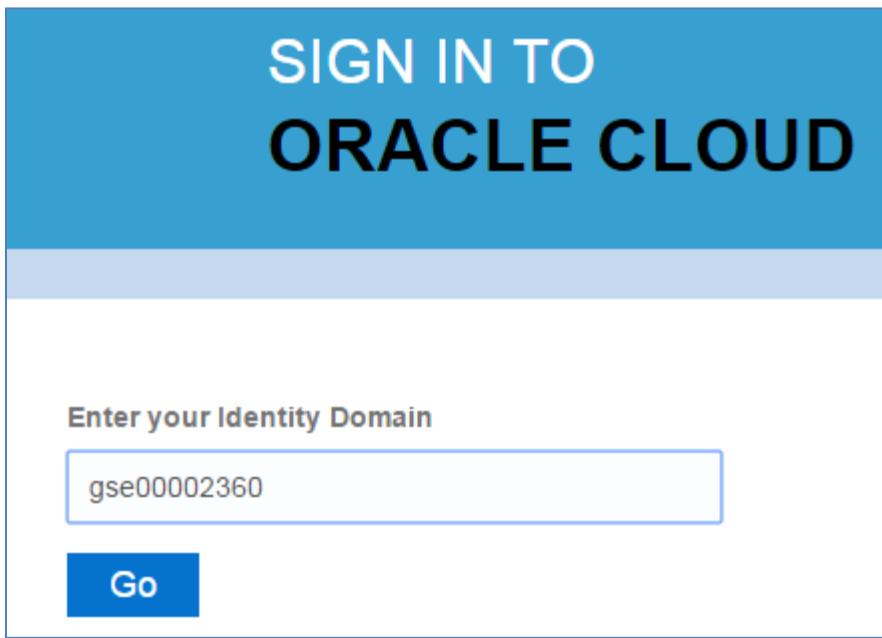
1.5.1: LOGIN TO YOUR ORACLE CLOUD ACCOUNT

- From within the VNC Session open the **Google Chrome browser**  and go to the following URL: <https://cloud.oracle.com>
- Click **Sign In** in the upper right hand corner of the browser
- Under the Cloud Account field click on the Select Data Center drop down, choose the data center location (for demos and trials, it's usually EMEA) then click on the My Services > button.

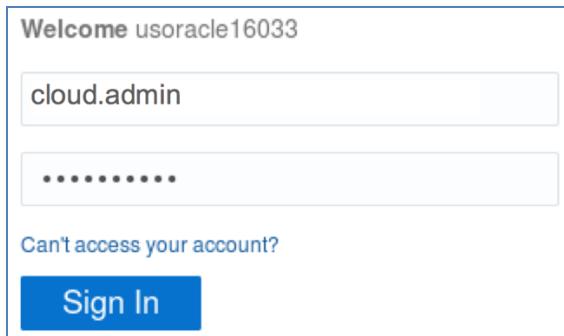
IMPORTANT - Under My Services, the **Data Center location is provided to you in the lab connection instructions.**



- In the next screen enter the **identity domain** and click **GO**



- Enter the cloud User Name and Password and click **Sign In**



NOTE: The **Identity Domain**, **User Name** and **Password** values are provided in the [lab connection instructions](#).

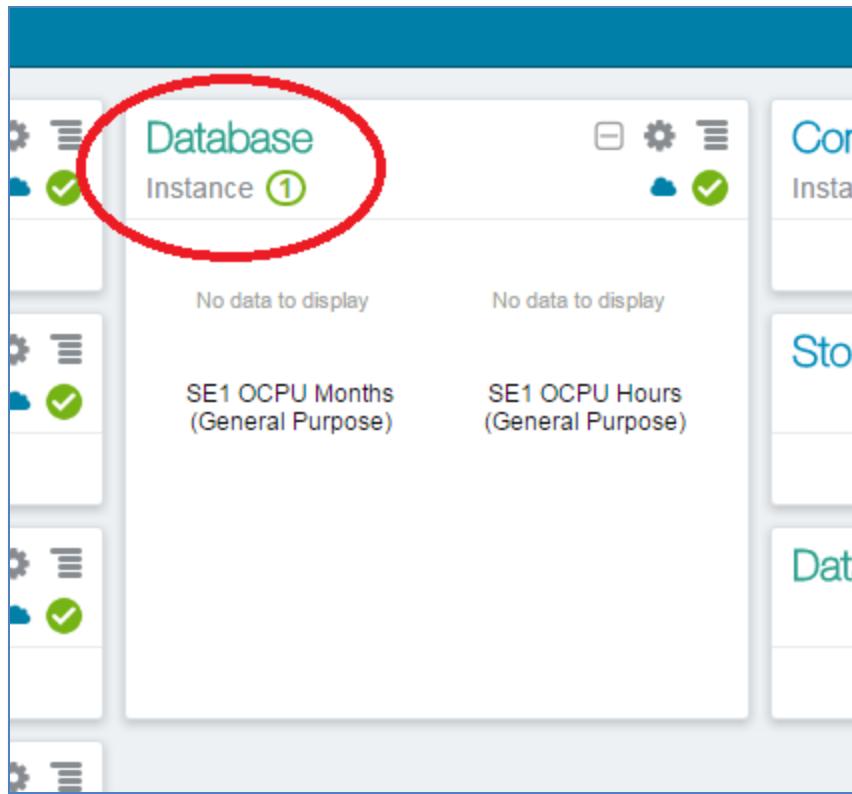
- You should see the Dashboard summarizing all of the available services.

The dashboard displays various service metrics and links:

- Identity Domain: gse00002183
- Dashboard, Users, Notifications buttons
- Customize Dashboard, Create Instance buttons
- Important Notifications: 0
- Developer: developer86716 (Developer)
- Database: Instance ① (EE HP OCPU Hours (General Purpose), EE EP OCPU Hours (General Purpose), No data to display, No data to display)
- Compute: Instances ② (OCPU Hours (General Purpose), OCPU Hours (High-Mem), Block Storage Capacity (GB-Months))
- Storage
- Java: 0 Instances
- Application Container: 0 Instances
- SOA

1.5.2: CREATE DATABASE CLOUD SERVICE

- From the main dashboard, click on the [Database](#) service link



- From this page you can view general information about this Database Cloud Service. Click on the **Open Service Console** button.

This screenshot shows the "Service Details: Oracle Database Cloud Service" page. At the top right, there's a button labeled "Open Service Console" with a red oval around it. The page has tabs for "Overview", "Billing Metrics", and "Monitoring Metrics". Under "Additional Information", there are several details:

- Plan: Oracle Database Cloud Service
- Service Start Date: 26-Jul-2016
- Service End Date: 26-Jul-2018
- Subscription ID: 536845804
- Service Instance ID: 536846905
- Customer Account: gse00002183 (US)
- CSI Number: Not available
- Data Center: EMEA Commercial 2 - Amsterdam
- Status: Active
- Domain SFTP Host & Port: sftp2.em2.cloud.oracle.com:22
- Domain SFTP User Name: em223482
- REST Endpoint: <https://dbcs.emea.oraclecl>

- From the console, click the **Create Service** button
- **Note:** There should already be a service provisioned (Alpha01A-DBCS). It was created earlier in order to save lab time. This service will be utilized later in the lab, don't interact with it at this point.

The screenshot shows the Oracle Database Cloud Service dashboard. At the top, there's a navigation bar with a cloud icon, the service name, and links for Services, Activity, SSH Access, Welcome!, and REST APIs. Below the header is a summary section with metrics: 1 Service, 1 OCPUs, 7.5 GB Memory, 185 GB Storage, and 1 Public IP. A search bar for services is present, along with a timestamp (As of Nov 2, 2016 10:48:45 PM UTC) and a red-outlined 'Create Service' button. Below the summary is a table listing a single service: Alpha01A-DBCS, Version 12.1.0.2, Edition Enterprise Edition, created on Nov 2, 2016 at 6:10:21 AM UTC, with 1 OCPUs, 7.5 GB Memory, and 185 GB Storage.

 **Important Step!** In the next steps, you will provide identity service configuration details and select the compute shape for your new Database instance.

Note: Service Name must be unique. When providing a name, please note you may have another service instance already created in your account.

- Use the information from the following table for the Service Configuration details:

Basic Service Information	
Service Name	Alpha01B-DBCS
Description	Alpha Office Database Cloud Service
Service Level	Oracle Database Cloud Service
Metering Frequency	Hourly
Software Release	Oracle Database 12c Release 1
Software Edition	Enterprise Edition
Database Type	Single Instance

Below are examples of the alternate selections for each

* Service Level	Oracle Database Cloud Service Oracle Database Cloud Service Oracle Database Cloud Service - Virtual Image	Software Edition	Enterprise Edition Standard Edition Enterprise Edition Enterprise Edition - High Performance Enterprise Edition - Extreme Performance
* Metering Frequency	Hourly Hourly Monthly	* Database Type	Single Instance Single Instance Database Clustering with RAC Single Instance with Data Guard Standby Database Clustering with RAC and Data Guard Standby
* Software Release	Oracle Database 12c Release 1 Oracle Database 11g Release 2 Oracle Database 12c Release 1 Oracle Database 12c Release 2		

- Your screen should look like this ...

Oracle Database Cloud Service
Create Service

Cancel Service Details Confirmation Next >

Service
Provide basic service instance information.

* Service Name	Alpha01B-DBCS	* Service Level	Oracle Database Cloud Service
Description	Alpha Office Database Cloud Service	* Metering Frequency	Hourly
		* Software Release	Oracle Database 12c Release 1
		* Software Edition	Enterprise Edition
		* Database Type	Single Instance

- Click **Next** to continue
- In the next screen we will fill out the **Service Details** for our Database Cloud instance. The screen will be broken down into sections to make the information easier to understand.

- ✓ Here is an example of the completed Service Details Screen. Continue to the next step for details on how to fill this form in correctly.

Oracle Database Cloud Service
Create Service

Service Details

Provide details for this Oracle Database Cloud Service instance.

Database Configuration

- * DB Name (SID) ORCL
- * PDB Name PDB1
- * Administration Password
- * Confirm Password
- * Usable Database Storage (GB) 25
- Total Data File Storage (GB) 88.5
- * Compute Shape OC3 - 1 OCPU, 7.5 GB RAM
- * SSH Public Key labkey.pub

Backup and Recovery Configuration

- * Backup Destination Both Cloud Storage and Local
- * Cloud Storage Container Storage-gse00002177/Alpha
- * Cloud Storage Username cloud.admin
- * Cloud Storage Password
- Create Cloud Storage Container
- Total Estimated Monthly Storage (GB) 140

Initialize Data From Backup

- * Create Instance from Existing Backup No

Advanced Settings

- * Listener Port 1521
- * Timezone (UTC) Coordinated Universal
- * Character Set AL32UTF8 - Unicode Universe
- * National Character Set AL16UTF16 - Unicode UTF-16
- Enable Oracle GoldenGate
- Include "Demos" PDB

Section 1: Database Configuration:

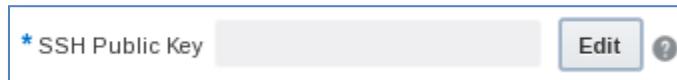
Database Configuration

- * DB Name (SID) ORCL
- * PDB Name PDB1
- * Administration Password
- * Confirm Password
- * Usable Database Storage (GB) 25
- Total Data File Storage (GB) 88.5
- * Compute Shape OC3 - 1 OCPU, 7.5 GB RAM
- * SSH Public Key labkey.pub

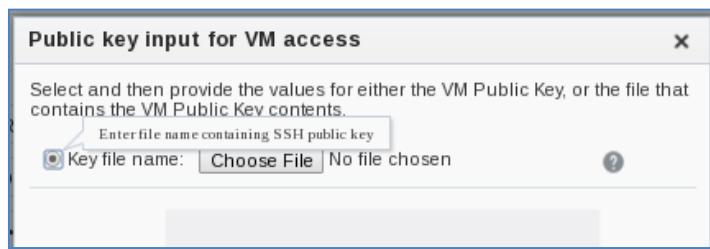
Section 1: Database Configuration	
DB Name (SID)	ORCL
PDB Name	PDB1
Administration Password	Alpha2014_
Usable Database Storage (GB)	25
Total Data File Storage (GB)	88.5
Compute Shape	OC3 – 1 OCPU, 7.5 GB RAM
SSH Public Key	labkey.pub

Note: The SSH Key has already been created for you. It's named labkey.pub and can be found on the client image under /u01/OPCWorkshop/lab/

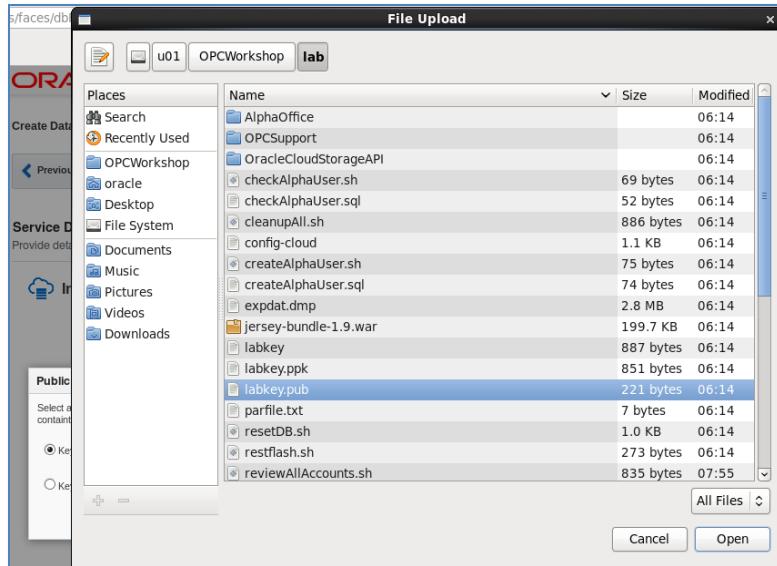
- On the SSH Public Key section, click Edit



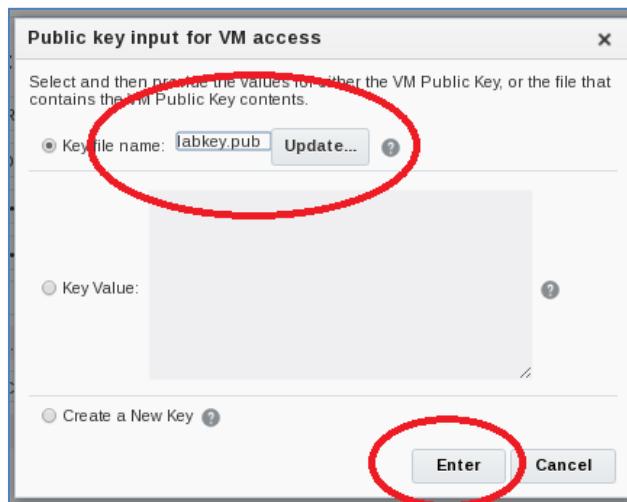
- Click on the **Edit** button to browse for the labkey.pub



- Make sure the Key File name: radio button is checked and click **Choose File**
- Navigate to the following subdirectory on your system: /u01/OPCWorkshop/lab/
- Choose the file named labkey.pub



- Click Open and confirm that **labkey.pub** is shown on the screen



- Click **Enter** and continue filling out the Details form. Continue on to the **Backup and Recovery** and **Initialize Data From Backup Configuration** Sections

Backup and Recovery Configuration

* Backup Destination: Both Cloud Storage and Local

* Cloud Storage Container: Storage-gse00002177/Alpha

* Cloud Storage Username: cloud.admin

* Cloud Storage Password: [REDACTED]

Create Cloud Storage Container: ?

Total Estimated Monthly Storage (GB): 140

Initialize Data From Backup

* Create Instance from Existing Backup: No

- Use the information from the following table to help fill out the **Backup and Recovery Configuration** and **Initialize Data From Backup** sections.

Backup and Recovery Configuration Information	
Backup Destination	Both Cloud and Local Storage
Cloud Storage Container	Storage-<your domain name>/Alpha01B_DBCS_SC
Cloud Storage Username	cloud.admin
Cloud Storage Password	Use <cloud.admin> assigned password
Create Cloud Storage Container	Check box – Yes
Total Estimated Monthly Storage (GB)	Default 140
Create Instance from Existing Backup	No

Note: By checking the box labeled - **Create Cloud Storage Container**, if the container does not exist it will be created automatically. This will be the location for all backups

- Confirm the following information from the **Advanced Settings** section. This information is default and doesn't need to be altered.

Advanced Settings

* Listener Port: 1521
 * Timezone: (UTC) Coordinated Universal
 * Character Set: AL32UTF8 - Unicode Universa
 * National Character Set: AL16UTF16 - Unicode UTF-16
 Enable Oracle GoldenGate:
 Include "Demos" PDB:

- After all fields have been filled out , click on **Next**

Oracle Database Cloud Service
Create Service

Previous Cancel Service Details Confirmation **Next >** (circled in red)

Service Details
 Provide details for this Oracle Database Cloud Service instance.

Database Configuration

* DB Name (SID): ORCL
 * PDB Name: PDB1
 * Administration Password:
 * Confirm Password:
 * Usable Database Storage (GB): 25
 Total Data File Storage (GB): 88.5
 * Compute Shape: OC3 - 1 OCPU, 7.5 GB RAM
 * SSH Public Key: labkey.pub

Backup and Recovery Configuration

* Backup Destination: Both Cloud Storage and Local
 * Cloud Storage Container: Storage-gse00002177/Alpha
 * Cloud Storage Username: cloud.admin
 * Cloud Storage Password:
 Create Cloud Storage Container:

Initialize Data From Backup

* Create Instance from Existing Backup: No

Advanced Settings

* Listener Port: 1521
 * Timezone: (UTC) Coordinated Universal
 * Character Set: AL32UTF8 - Unicode Universa
 * National Character Set: AL16UTF16 - Unicode UTF-16
 Enable Oracle GoldenGate:
 Include "Demos" PDB:

- Review the Confirmation screen and confirm that your entries are correct. When you're satisfied that everything is in order click **Create**.

Oracle Database Cloud Service
Create Service

[Previous](#) [Cancel](#)

Service Details Confirmation

Confirmation
Confirm your responses and create this Oracle Database Cloud Service instance.

Subscription Details	Database Configuration Details
Service Level: Oracle Database Cloud Service	Usable Database Storage: 25
Metering Frequency: Hourly	Total Data File Storage: 88.5
Software Release: Oracle Database 12c Release 1	DB Name (SID): ORCL
Software Edition: Enterprise Edition	PDB Name: PDB1
Service Details	Listener Port: 1521
Service Name: Alpha01B-DBCS	Character Set: AL32UTF8 - Unicode Universal character set UTF-8 form 32-bit
Description: Alpha Office Database Cloud Service	National Character Set: AL16UTF16 - Unicode UTF-16 Universal character set
Compute Shape: OC3 - 1 OCPU, 7.5 GB RAM	Include "Demos" PDB: No
Timezone: (UTC) Coordinated Universal Time(UTC)	Include GoldenGate: No
Key: labkey.pub	
Backup and Recovery Details	Standby Database Configuration Details
Backup Destination: Both Cloud Storage and Local Storage	Standby Database with Data Guard: No
Username: cloud.admin	
Cloud Storage Container:	Storage- gse00002177/Alpha01B_DBCS_SC

- You should be returned to the dashboard and your new Database Cloud Service instance **Alpha01B-DBCS** should be visible and labeled as 'In Progress'

Services

Enter a full or partial service name

[Create Service](#)

Alpha01B-DBCS
Status: In Progress
Version: 12.1.0.2
Edition: Enterprise Edition
Submitted On: Jan 6, 2016 2:01:26 AM UTC
OCpus: 1
Memory: 7.5 GB
Storage:

Alpha01A-DBCS
Version: 12.1.0.2
Edition: Enterprise Edition
Created On: Jan 5, 2016 3:30:08 PM UTC
OCpus: 1
Memory: 7.5 GB
Storage: 101 GB

- The creation of the DBCS instance will take approximately 20 minutes. While your DBCS instance is being created, you can view the current status clicking on the **In Progress** link.

NOTE: To save time we will continue with the lab by utilizing **Alpha01A-DBCS** DBCS instance already created. **Alpha01A-DBCS** was created following the exact steps you just performed.

The screenshot shows a 'Services' section in the Oracle Database Cloud Service console. A search bar at the top has 'Enter a full or partial service name'. Below it, a service named 'Alpha' is listed. The service icon is a blue cloud with a database icon. The status is 'In Progress' with a progress bar. The service details are: Version: 12.1.0.2, Edition: Enterprise Edition. A message 'Starting Compute resources...' is displayed above the service entry. To the right, the text 'Submitted On: Jan 6, 2016 2:01:26 AM UTC' is shown.

1.6: Configuration and Image Exploration

1.6.1: RECORD THE IP ADDRESS OF THE DATABASE CLOUD SERVICE

In the following steps you will record the IP addresses of the Virtual Machine on which the cloud service runs and configure SSH connectivity to **Alpha01A-DBCS**. Then you will access the different monitoring, configuration, and development consoles available on Oracle Database Cloud Service.

- Click on **Alpha01A-DBCS** link or cloud icon from the list of Database Instances to get further details on Alpha01A-DBCS

The screenshot shows the details of the 'Alpha01A-DBCS' instance. It includes a cloud icon with a red arrow pointing to it, the instance name 'Alpha01A-DBCS', its version '12.1.0.2', and its edition 'Enterprise Edition'. To the right, creation details are shown: 'Created On: Feb 23, 2017 12:53:38 AM UTC', and resource specifications: 'OCPUs: 1', 'Memory: 7.5 GB', and 'Storage: 185 GB'. A three-dot menu icon is also present.

- Note the Public IP address of **Alpha01A-DBCS**. In the below example the address is 140.86.12.71, yours will be different.

The screenshot shows the 'Resources' section of the Oracle Database Cloud Service console. An instance named 'Alpha01A-DBCS' is highlighted with a red oval. Its public IP address is listed as 'Public IP: 140.86.12.71'. To the right, detailed resource information is provided: SQL *Net Port: 1521, SID: ORCL, PDB Name: PDB1, OCPUs: 1, Memory: 7.5 GB, and Storage: 185 GB. A three-dot menu icon is also present.

1.6.2: CREATE THE SSH CONFIGURATION FILE AND START TUNNELS

For obvious reasons the default Oracle Public Cloud network configuration is very secure. Customers can open individual ports to the various servers in the cloud environment through the Compute Cloud Service Console or they can create SSH tunnels to the specific server/port combinations as needed.

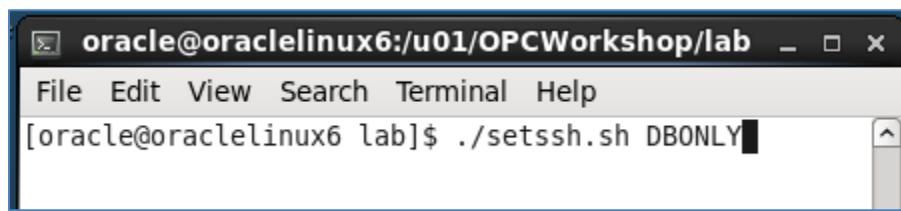
In this section we will use **SSH Tunnels** to communicate securely between the client and the Cloud Database instance. We have built a *script that automatically creates these tunnels for you*. You will review the script later in the exercise so you can learn how to create them on your own.

We will show you how to use secure tunnels as well as how to open ports for the following exercises.

In step 1.6.2 you will run the script that creates and configures an SSH file that will be used to connect to your various servers. Once the SSH file is created, it is used as a script to create the SSH tunnels in the background with connections to selected ports used in this and other labs.



- Open a terminal Window using the Utilities menu or by clicking the terminal icon.
- Change into the lab directory `cd lab`
- Run the SSH script to open the ports. Type `./setssh.sh DBONLY` in the terminal window

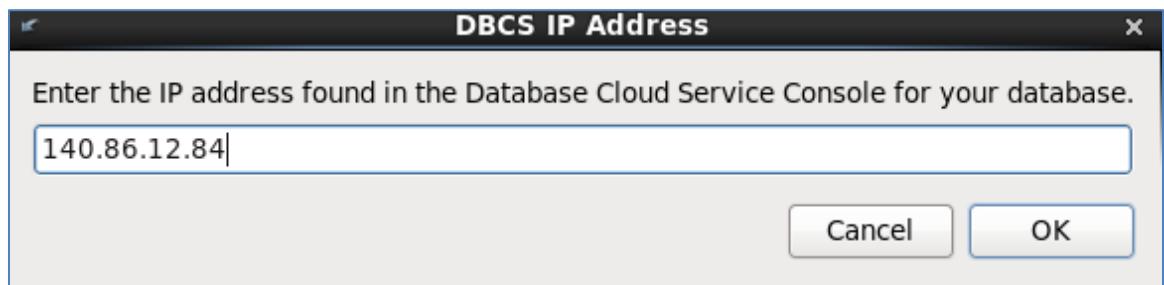


A screenshot of a terminal window titled "oracle@oracelinux6:u01/OPCWorkshop/lab". The window shows a menu bar with File, Edit, View, Search, Terminal, Help. The command line shows the user's prompt "[oracle@oracelinux6 lab]" followed by the command `./setssh.sh DBONLY`.

- If you make an error you'll see:



- Enter the Cloud Database IP address obtained in Step 1.6.1 and Click **OK**.



The script will try a test connection to your server. If successful, an SSH configuration file is created and an SSH session to the DB server will be started in the background.

If the script encounters any problems, or you entered the wrong address, you will be re-prompted to enter the IP address.

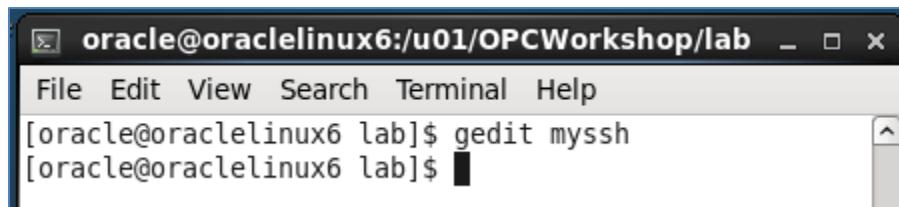
If the tunnels are created successfully you will see a Success Information Dialog.



- Click OK on the dialog box to clear it.

Note: Do not close the terminal window that you used to run the SSH Configuration script. Closing the terminal window will shut down the background SSH tunnel processes. If you accidentally shut the terminal window, you can start over again at step 1.6.2 and run the script again. You might want to minimize this terminal window until you need to stop the tunnels.

- Examine the script to view and understand the commands used to start the tunnels.
- Type `gedit myssh` in the terminal window



Notice under the **Host AlphaDBCS** section of the configuration file that the DBCS IP address has been inserted in 5 areas defining LocalForward ports (1526, 443, 4848, and 5500). Which correspond to host direct access, MySQL, APEX, DB Monitor, Glassfish, and Enterprise Manager. These ports can now be accessed locally using "localhost" in connection information or URLs.

Tunneling Script Example:

```
Host AlphaDBCS
hostname 140.86.12.84

user oracle
IdentityFile /u01/OPCWorkshop/lab/labkey

# Database Access
LocalForward 1526 140.86.12.84:1521

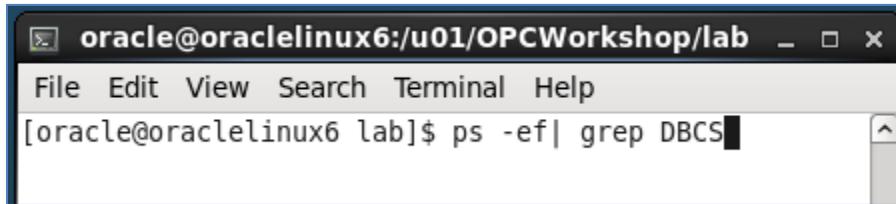
# Apex and DB Monitor
# https://localhost:443/apex/apex/pdb1/ .... workspace=internal, username=admin
# https://localhost:443/dbaas_monitor ..... username=dbaas_monitor
LocalForward 443 140.86.12.84:443

# Glass Fish
# https://localhost:4848 .... username=admin
LocalForward 4848 140.86.12.84:4848

# Enterprise Manager
# https://localhost:5500/em ..... username=sys
LocalForward 5500 140.86.12.84:5500

ServerAliveInterval 60
StrictHostKeyChecking no
```

- Close the edit session.
- Type `ps -ef | grep DBCS` to see the background session created.

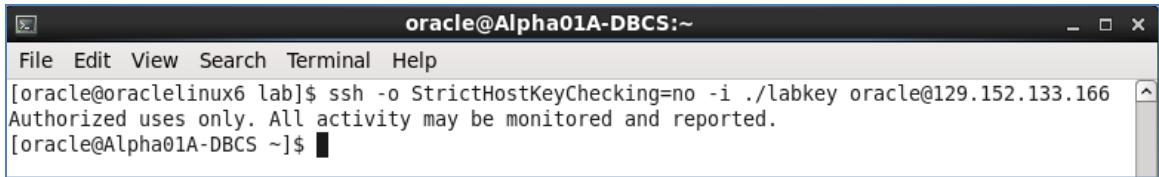


```
[oracle@oraclelinux6 lab]$ ps -ef|grep DBCS
root      3715     1  0 17:44 pts/0    00:00:00 sudo ssh -t -t -F myssh AlphaDBCS
root      3717  3715  0 17:44 pts/0    00:00:00 ssh -t -t -F myssh AlphaDBCS
oracle    3741  3055  0 17:53 pts/0    00:00:00 grep DBCS
[oracle@oraclelinux6 lab]$
```

1.6.3: EXPLORE DB IMAGE VIA SSH

In the previous step, you started SSH tunnels in the background for access to development tools and the browser. In this step we will demonstrate how to do an explicit SSH connection to your DB cloud server.

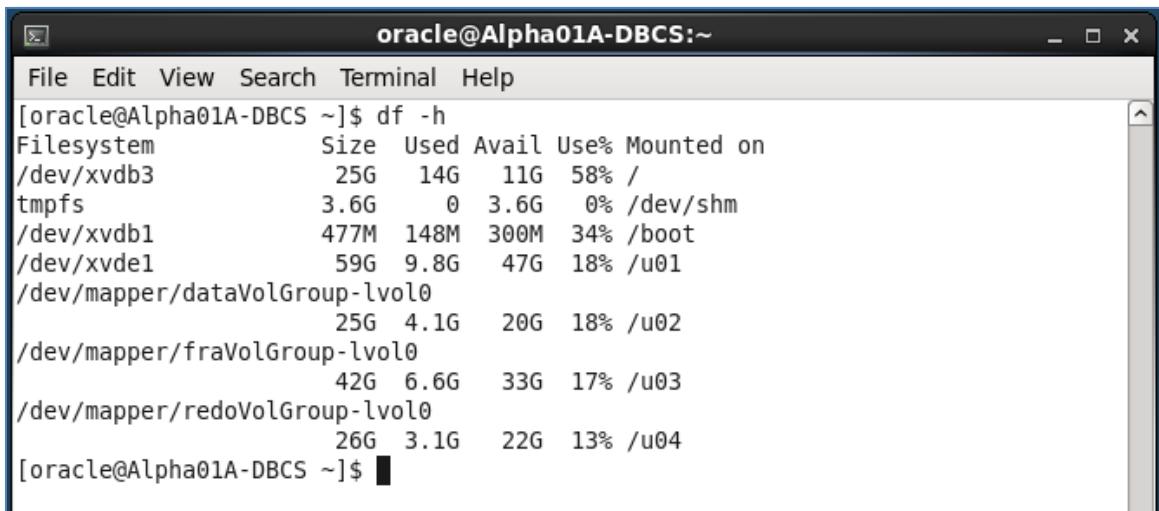
- Enter the below command in the same terminal window you ran the `setssh.sh` script, using the Public IP address of your DBCS cloud instance.
- `$ ssh -o StrictHostKeyChecking=no -i ./labkey oracle@<your-DB-Public-IP>`



```
oracle@Alpha01A-DBCS:~  
File Edit View Search Terminal Help  
[oracle@oraclelinux6 lab]$ ssh -o StrictHostKeyChecking=no -i ./labkey oracle@129.152.133.166  
Authorized uses only. All activity may be monitored and reported.  
[oracle@Alpha01A-DBCS ~]$
```

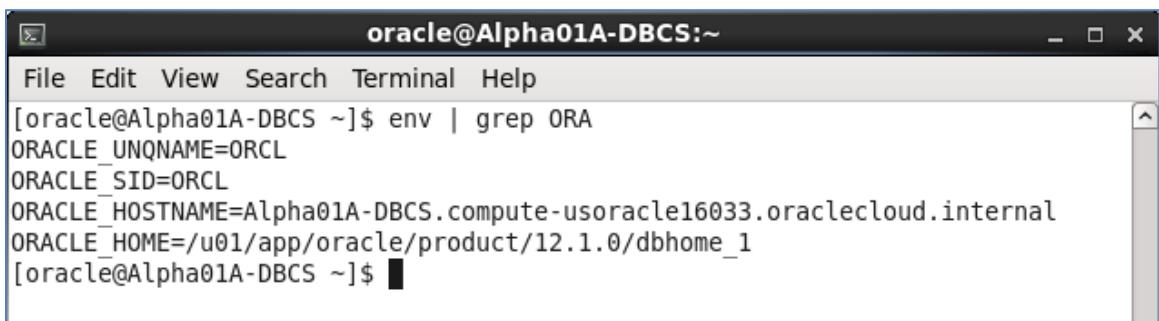
After successfully connecting to Alpha01A-DBCS you can browse the image.

- Issue a **df** command to see mounted disks and existing space.
- \$ df -h**



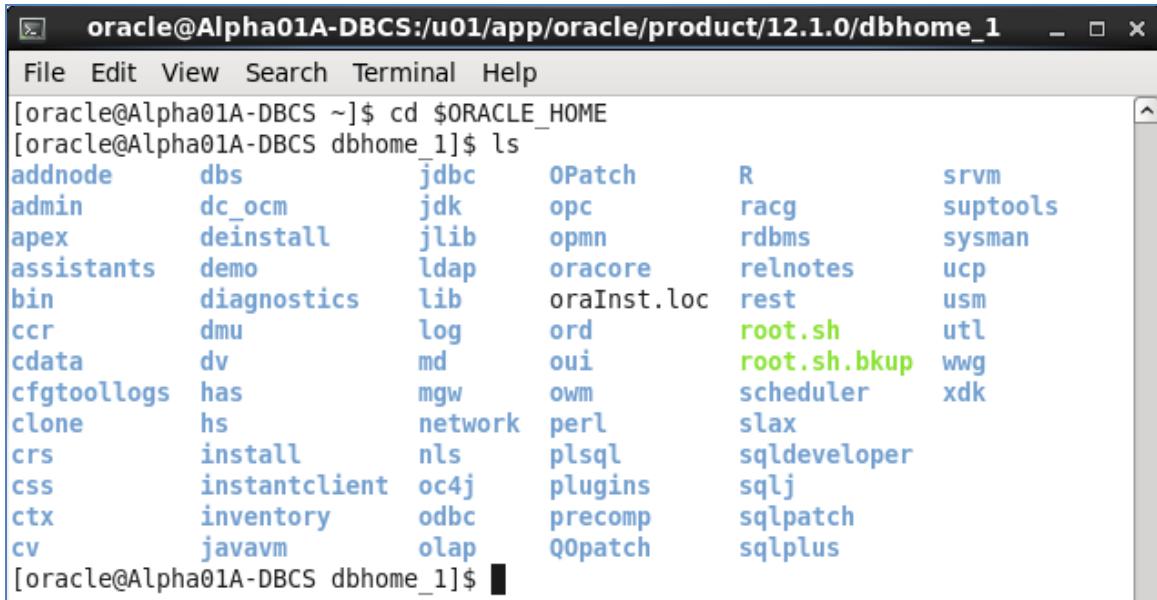
```
oracle@Alpha01A-DBCS:~  
File Edit View Search Terminal Help  
[oracle@Alpha01A-DBCS ~]$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/xvdb3       25G   14G   11G  58% /  
tmpfs          3.6G    0  3.6G  0% /dev/shm  
/dev/xvdb1      477M  148M  300M 34% /boot  
/dev/xvde1       59G   9.8G   47G  18% /u01  
/dev/mapper/dataVolGroup-lvol0  
                  25G  4.1G   20G  18% /u02  
/dev/mapper/fraVolGroup-lvol0  
                  42G  6.6G   33G  17% /u03  
/dev/mapper/redoVolGroup-lvol0  
                  26G  3.1G   22G  13% /u04  
[oracle@Alpha01A-DBCS ~]$
```

- View the Database environment variables.
- \$ env | grep ORA**



```
oracle@Alpha01A-DBCS:~  
File Edit View Search Terminal Help  
[oracle@Alpha01A-DBCS ~]$ env | grep ORA  
ORACLE_UNQNAME=ORCL  
ORACLE_SID=ORCL  
ORACLE_HOSTNAME=Alpha01A-DBCS.compute-usoracle16033.oraclecloud.internal  
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1  
[oracle@Alpha01A-DBCS ~]$
```

- Change directories to the Oracle Home and list the directories.
- \$ cd \$ORACLE_HOME**
- \$ ls**



The screenshot shows a terminal window titled "oracle@Alpha01A-DBCS:/u01/app/oracle/product/12.1.0/dbhome_1". The window displays a file listing in the "/u01/app/oracle/product/12.1.0/dbhome_1" directory. The files listed include addnode, admin, apex, assistants, bin, ccr, cdata, cfgtoollogs, clone, crs, css, ctx, cv, dbs, dc_ocm, deinstall, demo, diagnostics, dm, dv, has, hs, install, instantclient, inventory, javavm, jdbc, jdk, jlib, ldap, lib, log, md, mgw, network, nls, oc4j, odbc, olap, opc, opmn, oraInst.loc, oui, owm, perl, plsql, plugins, precomp, QOPatch, racg, rdbms, relnotes, rest, root.sh, root.sh.bkup, scheduler, slax, sqldeveloper, sqlj, sqlpatch, sqlplus, srvm, suptools, sysman, ucp, usm, utl, wwg, and xdk. The "root.sh" file is highlighted in green.

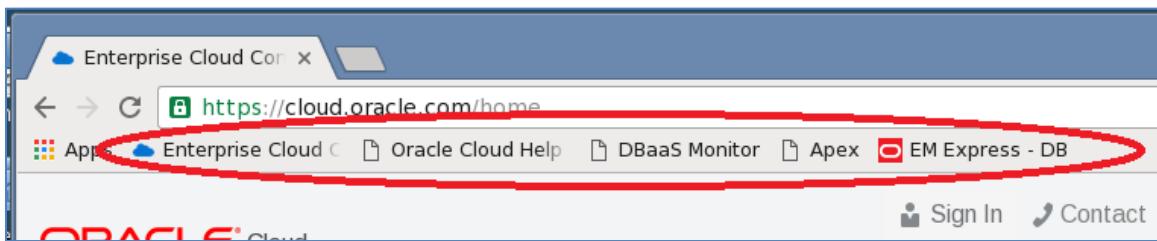
Type **exit** once to close the session.

```
$ exit
```

NOTE: By typing *exit* once, you are exiting the Cloud DBCS Service SSH session. If you are running commands in the same window you started the SSH tunnel script in and you type *exit* twice you will exit the terminal window and you will terminate the SSH tunnels. Do not exit the Terminal window, these SSH Tunnels will be used during the remainder of the lab. Feel free to minimize the SSH Tunnel terminal and start a new one to avoid accidentally closing the tunnels.

1.6.4: ACCESS THE COMPUTE CLOUD SERVICE AND OPEN SSH PORTS FOR ACCESS.

NOTE: For any of the URLs mentioned in this section you can also use the predefined browser links. If blocked ports on the network cause an issue loading any of these consoles, the predefined browser links will help by routing the traffic through the SSH tunnel that should already be open:



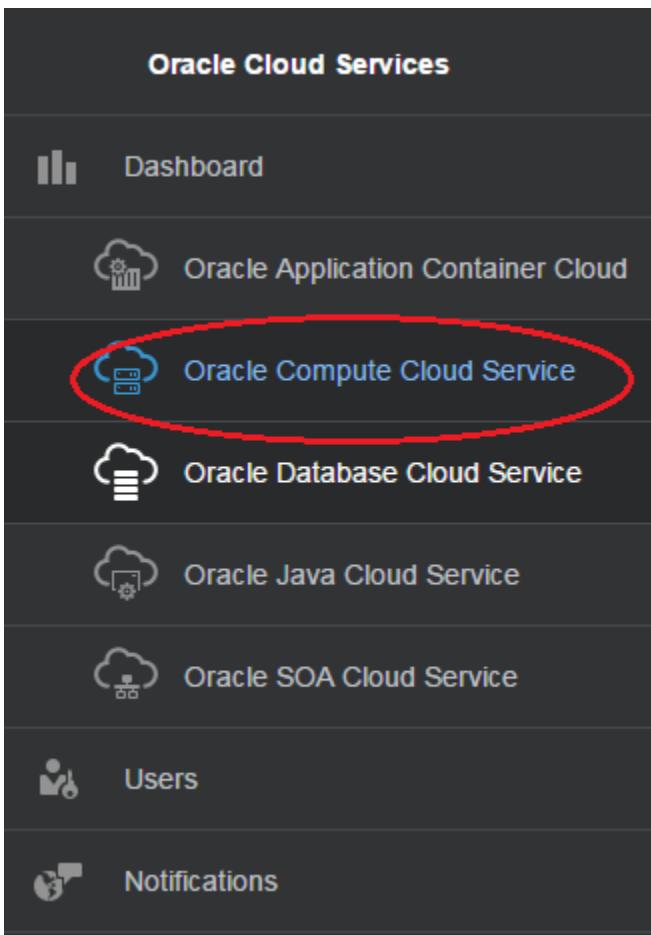
To gain access to the various consoles used by the Database Cloud Service you have two options. You can open up the port on which the monitor is listening, or you can create an SSH tunnel to the specific server/port combinations as needed. We have already created the SSH tunnels and those will be used to access the consoles we need for this lab.

In this step you will open the port 443 on the VM using a pre-created access rule and protocol definition.

- Return to your Cloud.oracle.com session on the Google browser. If you've logged out for any reason, log back into the cloud account using the details at the start of the lab.
- Click the header menu in the upper left next to Oracle Cloud My Services and choose the Compute Cloud console



- Click on **Compute Cloud Service Console** menu item



- Verify that you're on the correct site and that the Alpha01A-DBCS and Alpha01B-DBCS instance are visible.

ORACLE CLOUD My Services

Compute Instances Network Storage Orchestrations Images

Instances

Summary

3 instances 3 OCPUs 22.5GB memory 370GB volume size in use

Instances
An Oracle Compute Cloud Service instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. [Learn more.](#)

Name	Status	OCPUs	Memory	Volumes	Public IP	Private IP
Alpha01A-DBCS/db...	Running	1	7.5 GB	185 GB	140.86.32.50	10.196.172.106
Alpha01B-DBCS/db...	Running	1	7.5 GB	185 GB	140.86.32.207	10.196.199.138
ClientImage01	Running	1	7.5 GB	None	140.86.35.187	10.196.198.242

- From the top menu bar click on **Network** tab.

Compute Instances Network Storage Orchestrations Images

Security

Security Rules

Summary

21 security rules 10 enabled

Security Rules
You can use security rules to control network access between your instances and the Internet. On this page, you can create, view, update, and delete security rules. [Learn more.](#)

Name	Status	Security Application	Source	Destination
Alpha01A-DBCS/d...	Enabled	Alpha01A-DBCS/db_1/or...	public-internet	Alpha01A-DBCS/d...
Alpha01B-DBCS/d...	Disabled	Alpha01B-DBCS/db_1/or...	public-internet	Alpha01B-DBCS/d...

The Network section will allow you to manage security access, IP networking, and SSH information. There can be many entries so we'll filter out what we're looking for.

- With the Security > Security Rules section selected, type **https://** into the search box and hit enter or click the magnifying glass icon.

Compute Instances Network Storage Orchestrations Images

Security

Security Rules

Summary

21 security rules 10 enabled

Security Rules
You can use security rules to control network access between your instances and the Internet. On this page, you can create, view, update, and delete security rules. [Learn more.](#)

Name	Status	Security Application	Source	Destination
Alpha01A-DBCS/d...	Enabled	Alpha01A-DBCS/db_1/or...	public-internet	Alpha01A-DBCS/d...
Alpha01B-DBCS/d...	Disabled	Alpha01B-DBCS/db_1/or...	public-internet	Alpha01B-DBCS/d...

Note: If you can't see the particular rule clearly, widen your browser screen or click on the table icon to change the view.

Security Rules
You can use security rules to control network access between your instances and the Internet. On this page, you can create, view, update, and delete security rules. [Learn more](#).

https:// Category: All Show: All

- For each instance there is an https:// rule called '[ora_p2_https://](#)'.
- To change the rule for the **Alpha01A-DBCS** instance click on the **hamburger menu**  or on the colored arrows and select **Update**

Name	Description	Status	Action
Alpha01A-DBCS/db_1/ora_p2_https://	Security Application: Alpha01A-DBCS/db_1/ora_https:// Source: public-internet Destination: Alpha01A-DBCS/db_1/	Disabled	<input type="button" value="Update"/> <input type="button" value="Delete"/>
Alpha01B-DBCS/db_1/ora_p2_https://	Security Application: Alpha01B-DBCS/db_1/ora_https:// Source: public-internet Destination: Alpha01B-DBCS/db_1/	Disabled	<input type="button" value="Update"/> <input type="button" value="Delete"/>

- Change Status to **Enabled** and click **Update**

Update Security Rule

Update your security rule as required. You can enable or disable this rule or modify the description. [Learn More](#)

* Name: Alpha01A-DBCS/db_1/ora_p2_https://

Status:

Security Application: Alpha01A-DBCS/db_1/ora_https://

* Source: Security Lists Security IP Lists

Security IP Lists
public-internet

Destination: Alpha01A-DBCS/db_1/ora_db

Description:

- Notice the status change for the [ora_p2_https://](#) Access rule to **Enabled**. The port is now open and can be accessed without the tunnels if necessary.

The screenshot shows a table of security rules. The columns are: Name, Status, Security Application, Source, and Destination. There are two entries:

Name	Status	Security Application	Source	Destination
Alpha01A-DBCS/...	Enabled	Alpha01A-DBCS/db_1...	public-internet	Alpha01A-DBCS/...
Alpha01B-DBCS/...	Disabled	Alpha01B-DBCS/db_1...	public-internet	Alpha01B-DBCS/...

- Navigate back to the **Database Cloud Service Console** by clicking the header icon and selecting **Database Cloud Service**.

The screenshot shows the Oracle Cloud My Services header. It features the Oracle logo and the text "CLOUD My Services". A red circle highlights the hamburger menu icon (three horizontal lines) located to the left of the logo.

The screenshot shows the Oracle Cloud Services dashboard. The main title is "Cloud Services". Below it is a list of services, each with an icon and a name. The "Oracle Database Cloud Service" option is highlighted with a large red oval.

Icon	Service Name
Dashboard	Oracle Database Cloud Service
Cloud	Oracle Compute Cloud Service
Cloud	Oracle Java Cloud Service
Cloud	Oracle Application Container Cloud
Cloud	Oracle SOA Cloud Service
User	Users
Notification	Notifications
Monitoring	Monitoring

- For the **Alpha01A-DBCS** instance click the **hamburger menu**  and select **Open DBaaS Monitor Console**.

Services

Enter a full or partial service name

Create Service

Service Name	Created On	OCpus	Memory	Storage
Alpha01B-DBCS	Jan 6, 2016 2:01:26 AM UTC	1	7.5 GB	128 GB
Alpha01A-DBCS	Jan 5, 2016 3:30:08 PM UTC	1	7.5 GB	128 GB

Instance create and delete history

- You will likely get a security warning, click **ADVANCED** followed by **Proceed to <your IP Address> (unsafe)**

Your connection is not private

Attackers might be trying to steal your information from **140.86.13.52** (for example, passwords, messages, or credit cards).

NET::ERR_CERT_AUTHORITY_INVALID

Automatically report details of possible security incidents to Google. [Privacy policy](#)

[HIDE ADVANCED](#)

[Back to safety](#)

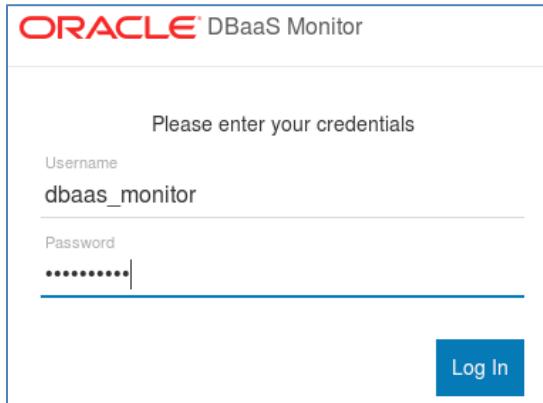
This server could not prove that it is **140.86.13.52**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

[Proceed to 140.86.13.52 \(unsafe\)](#)

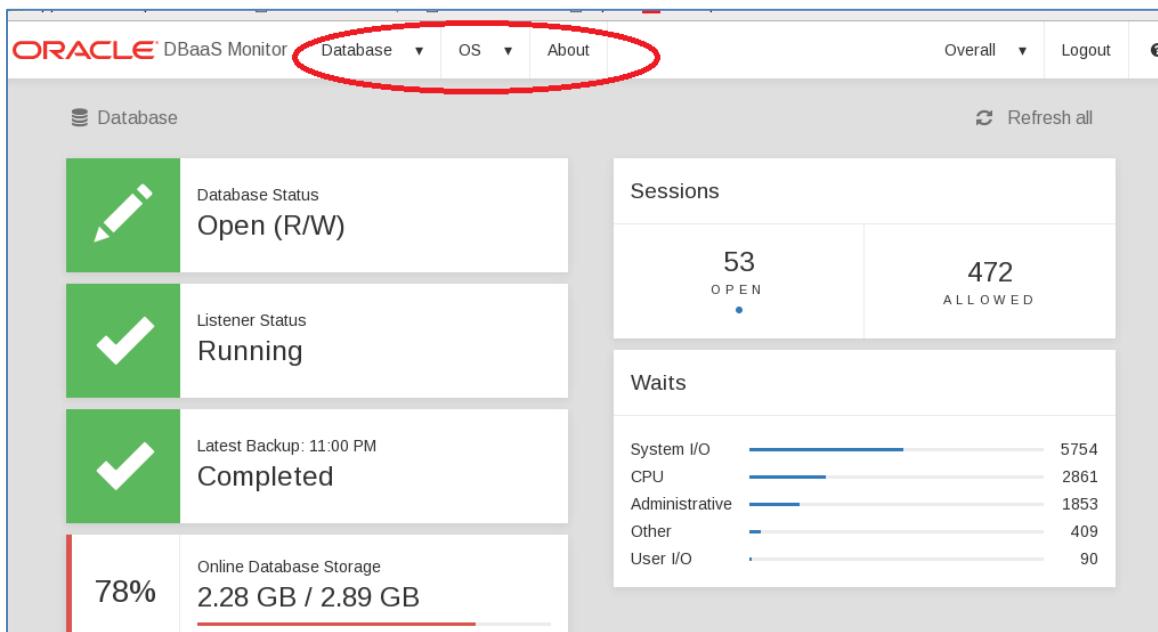
- Login to the DBaaS monitor

User Name:	dbaas_monitor
------------	---------------

Password:	Alpha2014_
-----------	------------



- Once connected to the Database Monitor Console, feel free to explore the various screens. Click on the header items to familiarize yourself with DBaaS monitor.
- Logout when finished

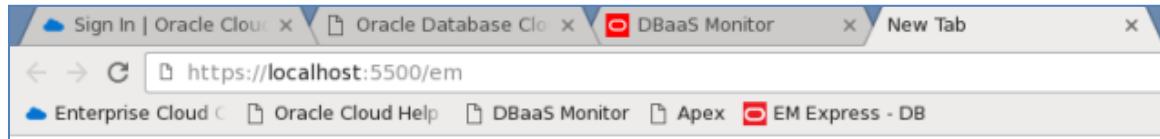


1.6.5: ACCESS ENTERPRISE MANAGER DB EXPRESS

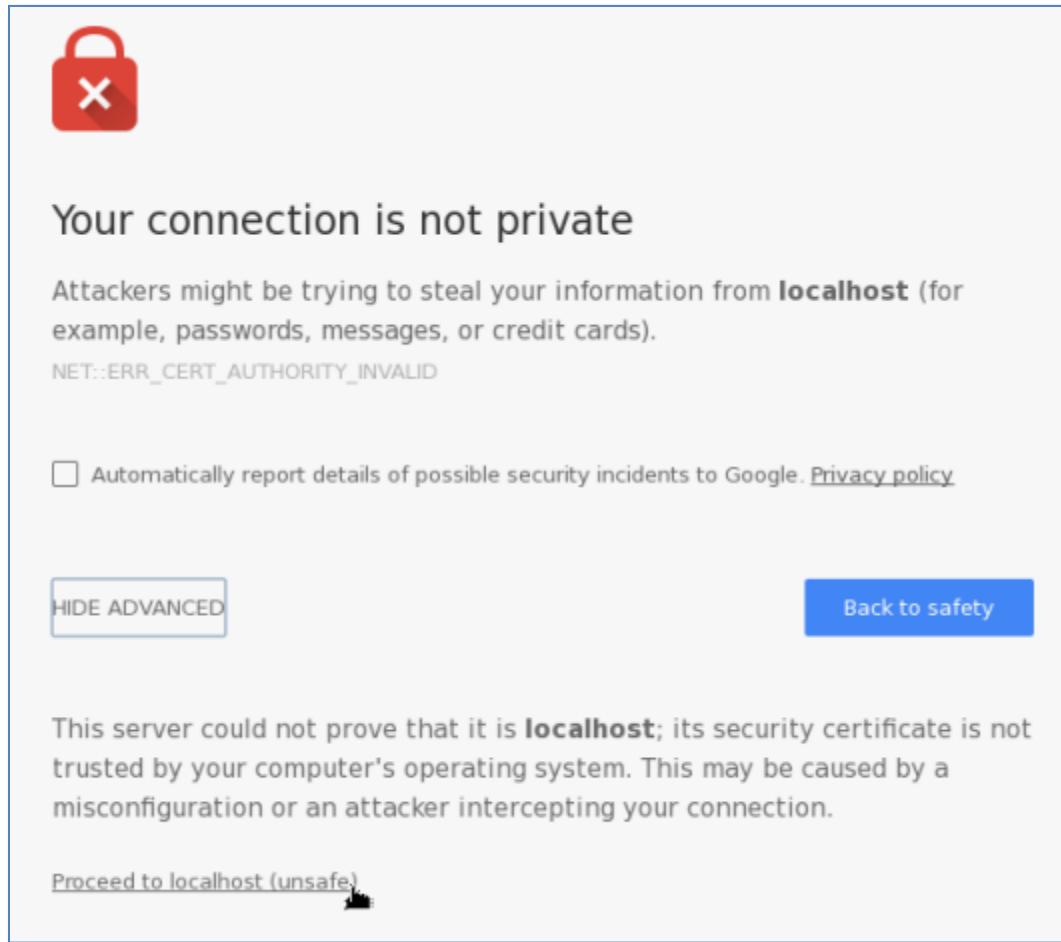
- On the browser, open a new tab and **enter** the following **URL** to access the **EM Express** page. The first time the URL is used, it can take a minute for the console to load.

Note: The `setssh.sh` script that you ran earlier in this lab created the ssh tunnel which routes all traffic over `localhost:5500` to the database cloud service.

- <https://localhost:5500/em>

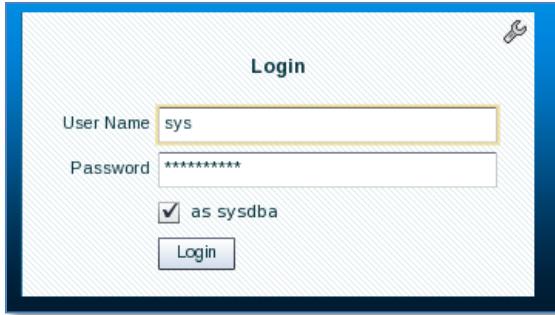


- If you get a security warning, click **ADVANCED** followed by **Proceed to localhost (unsafe)**

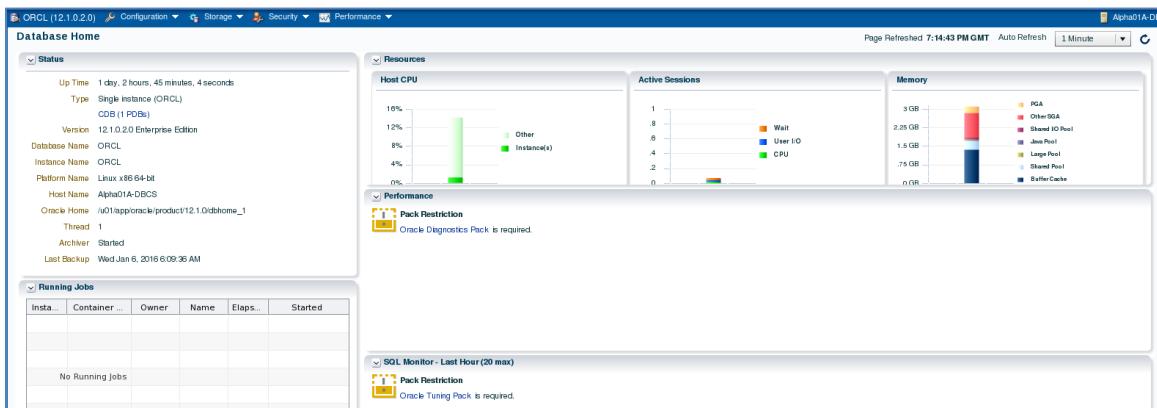


- On the EM login page enter the following and click on **Login**:

User Name:	sys
Password:	Alpha2014_
Checkbox:	"as sysdba"



- Click on different items on the Enterprise Manager console to get familiar with its capabilities.
- Logout when finished.

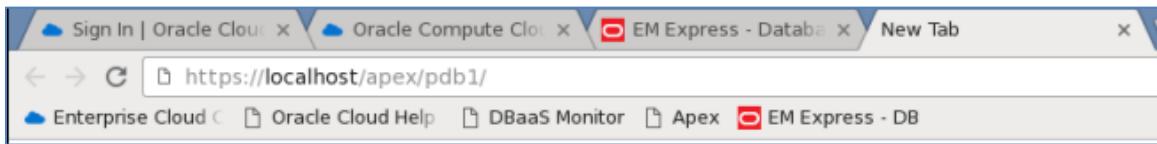


1.6.6: ACCESS APEX MONITOR

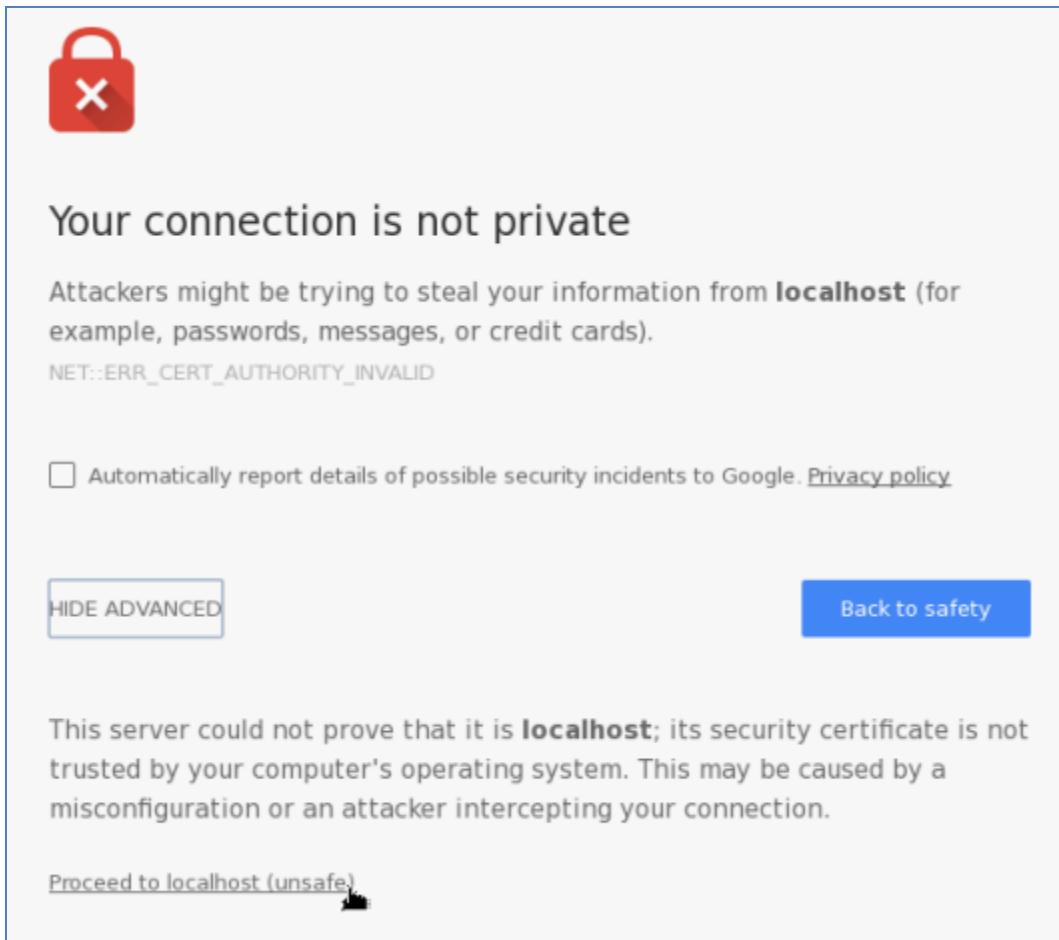
- Enter the following URL into a browser window to access the **Apex console** (click to add a security exception if necessary).

Note: All traffic to the default https port (443) on local host is also routed through the SSH tunnel to the database cloud service.

- <https://localhost/apex/pdb1/>

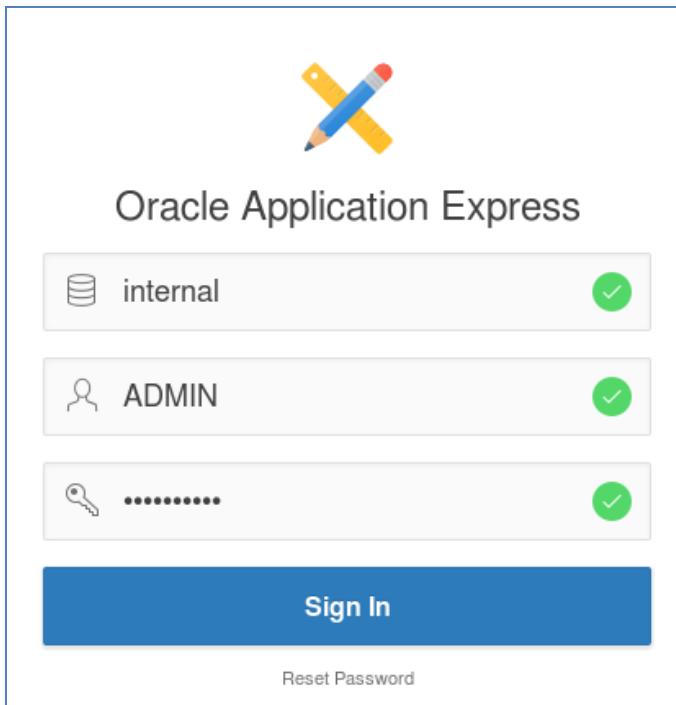


- If you get a security warning, click **ADVANCED** followed by **Proceed to localhost (unsafe)**

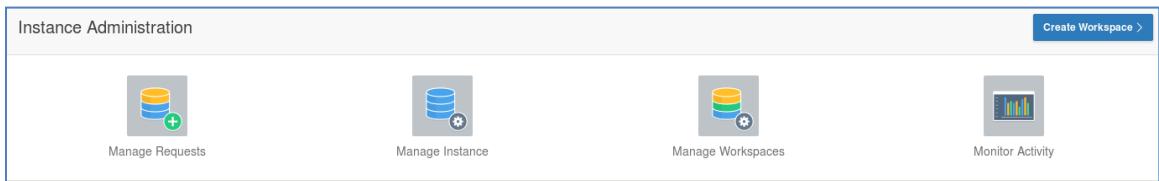


- Once the Apex login window is displayed, **enter the following** and click on **Sign In**:

Workspace:	internal
Username:	ADMIN
Password:	Alpha2014_



- You are now connected to Apex. Feel free to explore the various menu options. The last lab in this workshop will walk you through the creation of a simple application.
- Log out of APEX when you are finished.



- This concludes the first lab, Overview of DBCS. Proceed to the next lab when ready.

Section 2: Cloud Migration

2.1: Introduction

In this lab you will explore some common methods for moving data from on premise to the cloud. There are multiple options for solving this data movement challenge. In this lab we will use SQL*Developer and command line tools to clone and move a pluggable database from an on premise database (in the provided Virtual Machine) to a cloud database. We will also use standard Oracle Data Pump tools to export a schema from the on premise database, and then import that data to a cloud database in a new schema. The final exercise will use the SQL Developer cart feature to quickly move data from the local database to the cloud using only the privileges of a normal schema owner.

2.2: Objectives

- ✓ Clone, unplug, transfer, and plug the AlphaPDB pluggable database using SQL Developer.
- ✓ Export and import a schema using SQL Developer to execute Oracle Data Pump jobs.
- ✓ Export and import a small collection of tables using SQL Developer.

2.3: System Requirements

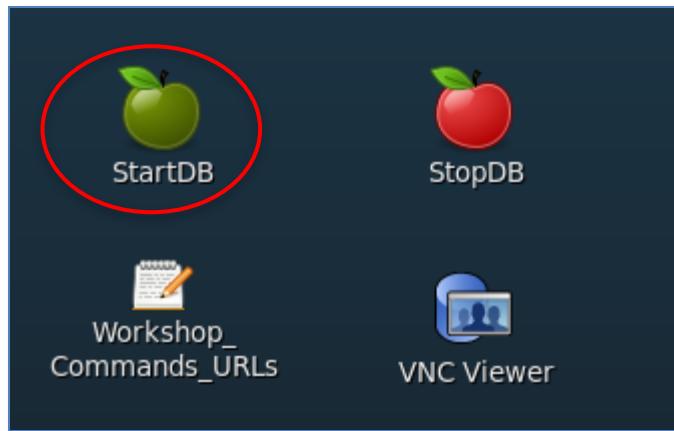
- ✓ VNC Viewer for connection to client system
- ✓ Successful completion of Section 1: Database Cloud Service Overview lab.
- ✓ The SSH tunnels must be active in a terminal window in the client system

2.4: Cloud Migration Using Pluggable Databases

In this section of the lab we will prepare the on premise environment and create connections. Then we'll clone the database. We'll create SSH connections to the cloud instance and finally we'll copy the clone to the public cloud using Oracle SQL Developer. Once the database has been cloned we will patch the instance to bring it up to date with the Cloud Container database. We will verify successful patching and login in to the newly migrated Database instance.

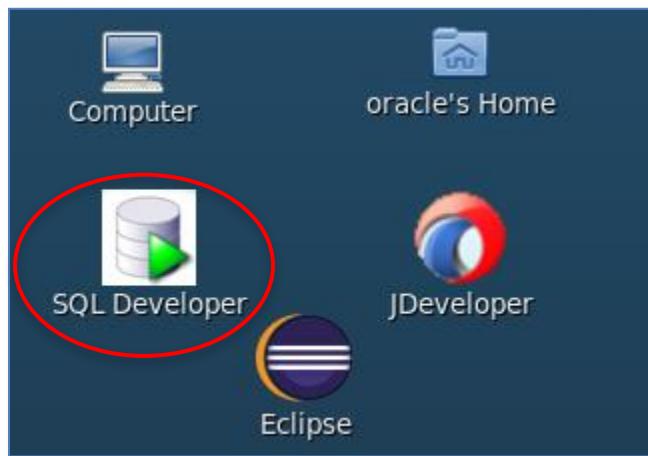
2.4.1: CONFIGURE THE ENVIRONMENT

- From the VNC Session desktop, locate and double-click on the **StartDB** icon. It will take a minute for the Database to fully start. Once started, the Terminal Log Window will automatically close.



- From the VNC Session desktop, locate and double-click on the **SQL Developer** icon. We will use SQL Developer to connect to the on premise database.

NOTE: The first time SQL Developer is brought up, it may a few minutes to start up.



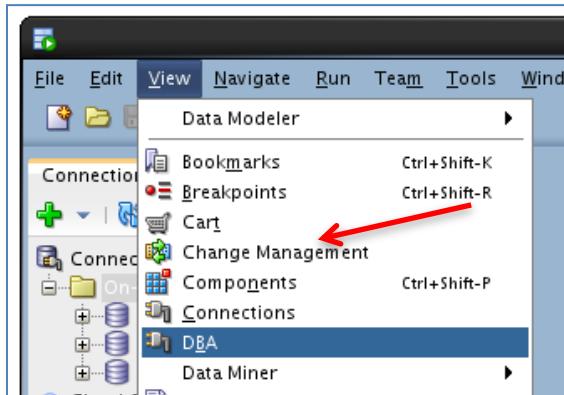
- Double-click the **On-premise** folder or click on the plus sign next the folder to expand the list of database connections. Please note the pre-configured connections to the on-premise database, Alpha – PDB, sys- AlphaPDB and sys-CDB.

Alpha - PDB
This is a normal database connection to the *alpha* schema inside the Alpha pluggable database.

sys - AlphaPDB
Privileged sys account in the Alpha pluggable database.

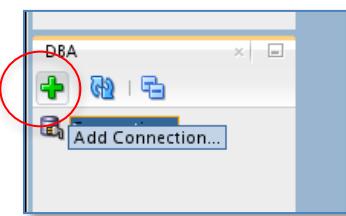
sys - CDB
Privileged sys account for the on-premise container database.

- Select the **View -> DBA** menu option from the top dropdown menu to open up the DBA navigator panel.



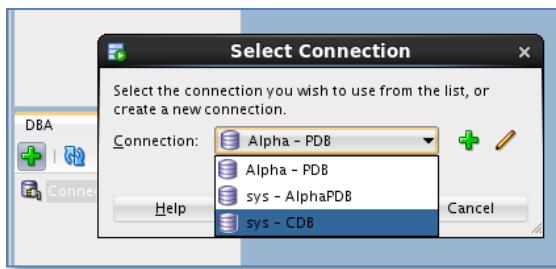
- On the DBA panel, click the green plus icon to create a new connection.

Note: you can also right-click on the green plus sign under Connections and select Add Connection.

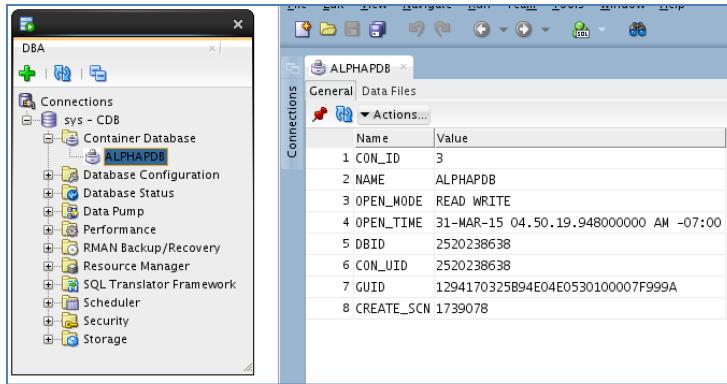


- Select the **sys - CDB** connection and click **OK**.

Note: “sys – CDB” is the on premise database located on the virtual client Image.

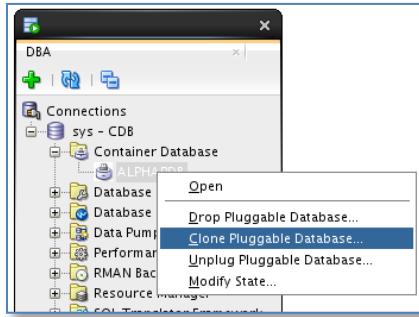


- Expand the **sys - CDB** connection by double clicking or by clicking on the plus sign, then expand the **Container Database** tree item. Click on the **ALPHAPDB** pluggable database item and view the details for the pluggable database.

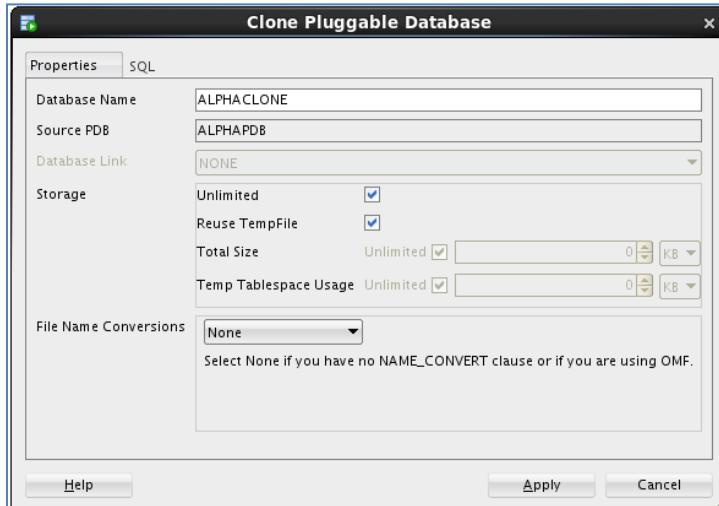


2.4.2: CLONE THE ALPHAPDB

- In the DBA Navigator panel, right click on the ALPHAPDB pluggable database and select the **Clone Pluggable Database...** menu option.

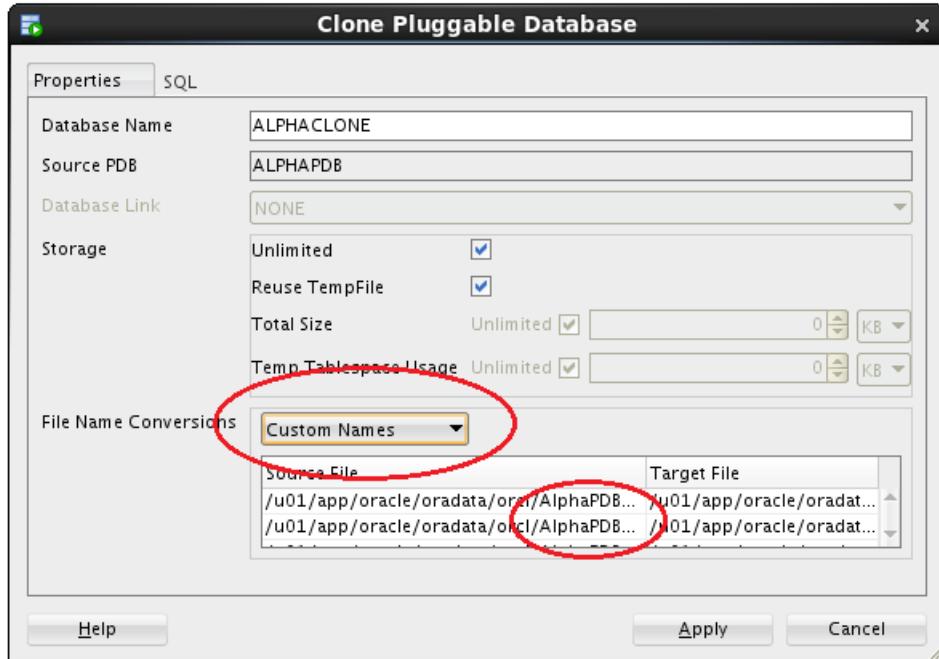


- Enter the following: Database Name: **ALPHACLONE**



- Select **Custom Names** from the File Name Conversions section and review the Source File names. Note the name "/AlphaPDB/" for the directory of the source file names. For

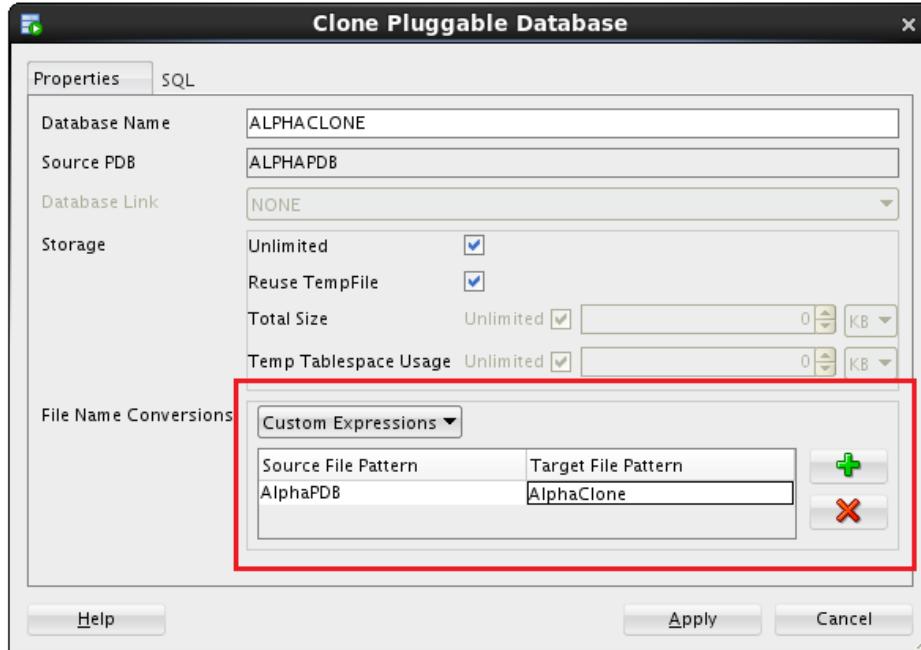
a successful cloning operation, we must create new target files that are different than the source files.



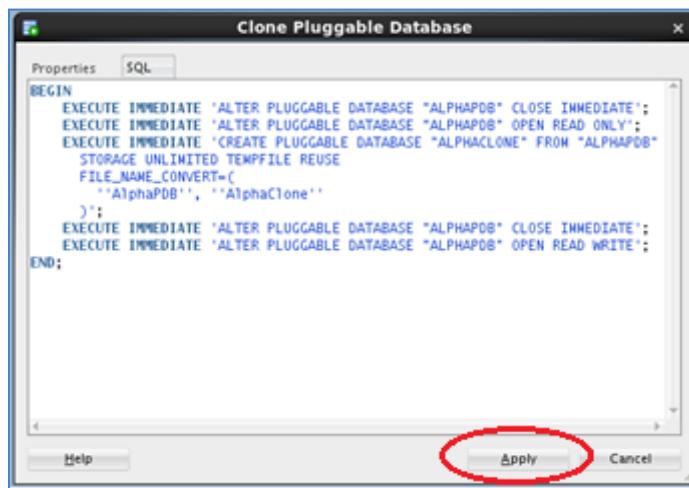
Note: At this point we could manually change all the target files to point to a different directory, but following are directions for an easier way to do this.

- Select **Custom Expressions** from the File Name Conversions drop down.
- Click the **green plus icon**
- Enter the source directory portion we want to change, followed by the target for the new files. Use the information from the table below being sure to use the indicated upper and lower case letters:

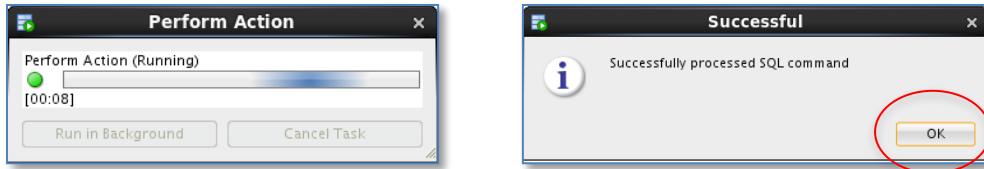
Source File Pattern	AlphaPDB
Target File Pattern	AlphaClone



- Review the SQL statement by clicking on the SQL tab - note the **FILE_NAME_CONVERT** clause mapping the existing files to new files in a new directory.
- Click the **Apply** button.

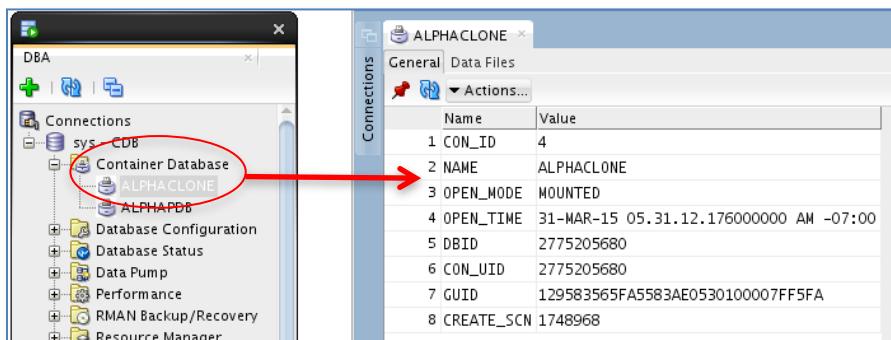


- SQL Developer shows the action in-progress message for roughly 1-2 minutes followed by a success message.
- Click **OK** on the success message.

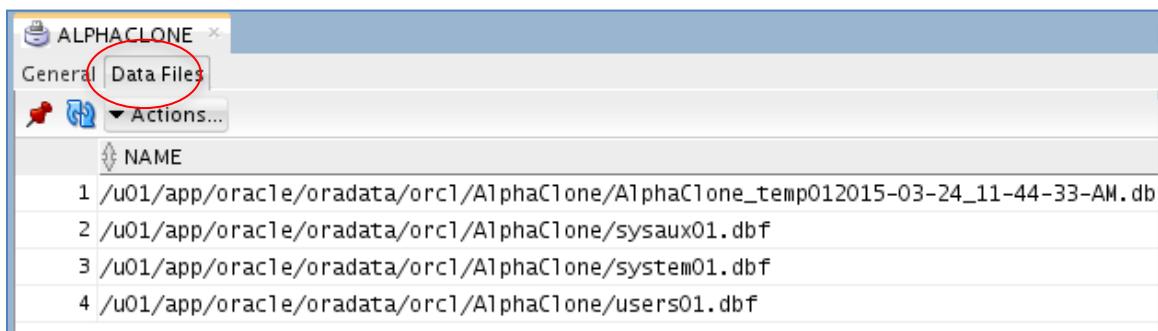


- Click on the **ALPHACLONE** database in the DBA navigator to see the status of the database.

Note: the cloned database shows an **OPEN_MODE** of **MOUNTED** indicating the database is plugged-in but is not open for access.



- Click on the **Data Files** tab for the ALPHACLONE to review the data files created during the cloning operation.



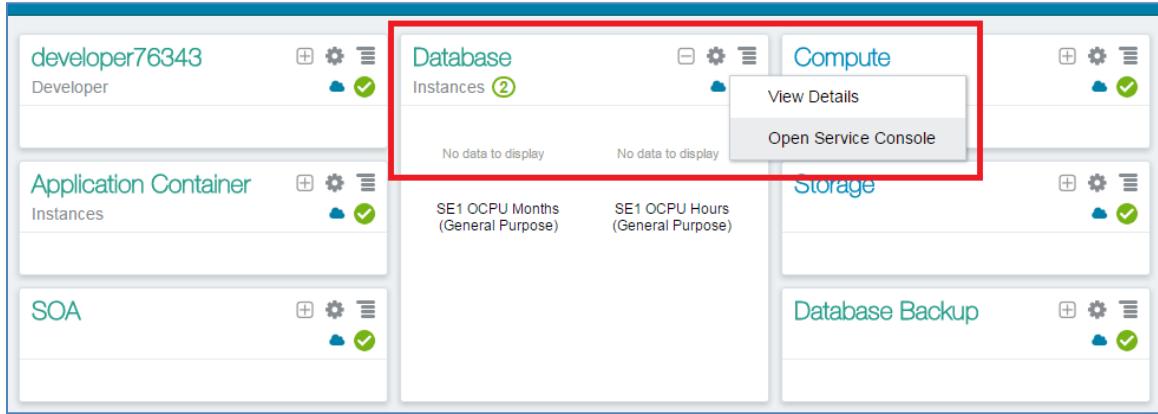
2.4.3: CREATE SSH HOST

In this section you will create SSH and SYS database cloud connections.

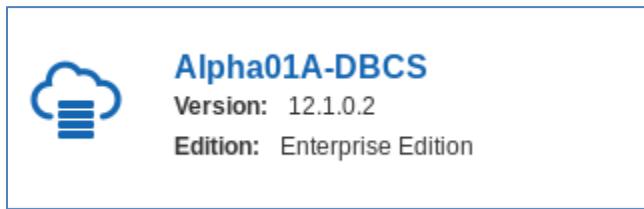
In the following steps you will record the IP addresses of the Virtual Machine on which the cloud service runs and configure SSH connectivity to **Alpha01A-DBCS**. Then you will access the different monitoring, configuration, and development consoles available on Oracle Database Cloud Service.

- ✓ If you haven't already noted down from the last lab, determine the Public IP address of Alpha01A-DBCS as we'll be using that IP to set up our SSH connection for the data migration exercises.

- Open a browser and login to cloud.oracle.com using the credentials provided by your instructor.
- Click on the hamburger menu at the top right of the Database service section in the Dashboard and choose “**Open Service Console**”



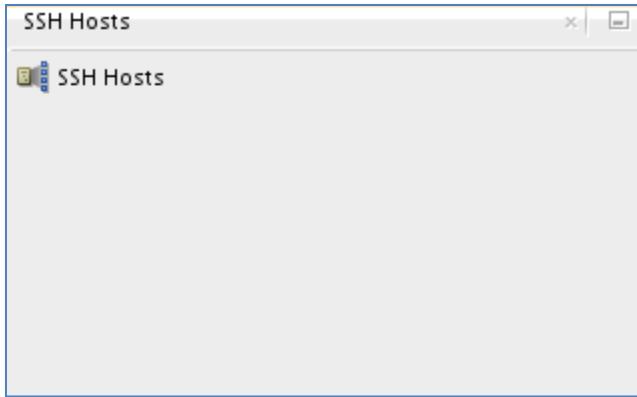
- Click on **Alpha01A-DBCS** from the list of Database Instances



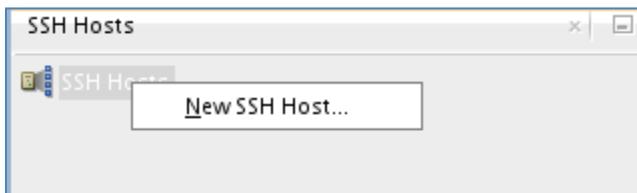
- Note the Public IP address of **Alpha01A-DBCS**. The IP address for your lab will be different than the one in the below screenshot.

Nodes		As of Dec 2, 2016 6:1		
 Alpha01A-DBCS	Public IP: 140.86.39.189	SQL *Net Port: 1521	OCPUs: 1	Memory: 7.5 GB

- ✓ Now we can setup an SSH host connection to the Database Cloud Service instance.
 - Return to SQL Developer. From the top menu select **View -> SSH** to display SSH hosts panel on the left.

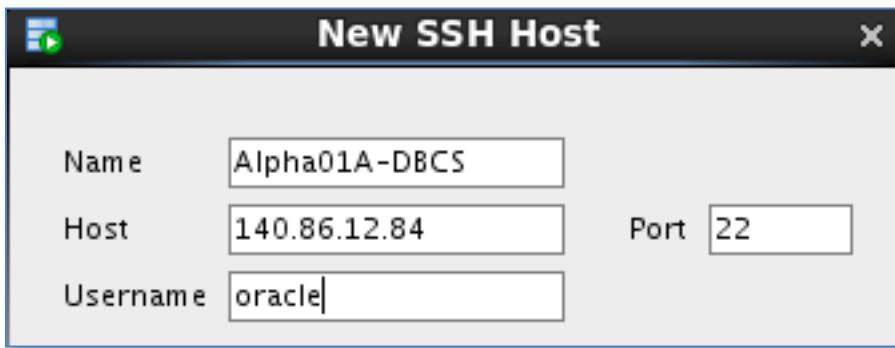


- Right click on **SSH Hosts** and select **New SSH Host**.

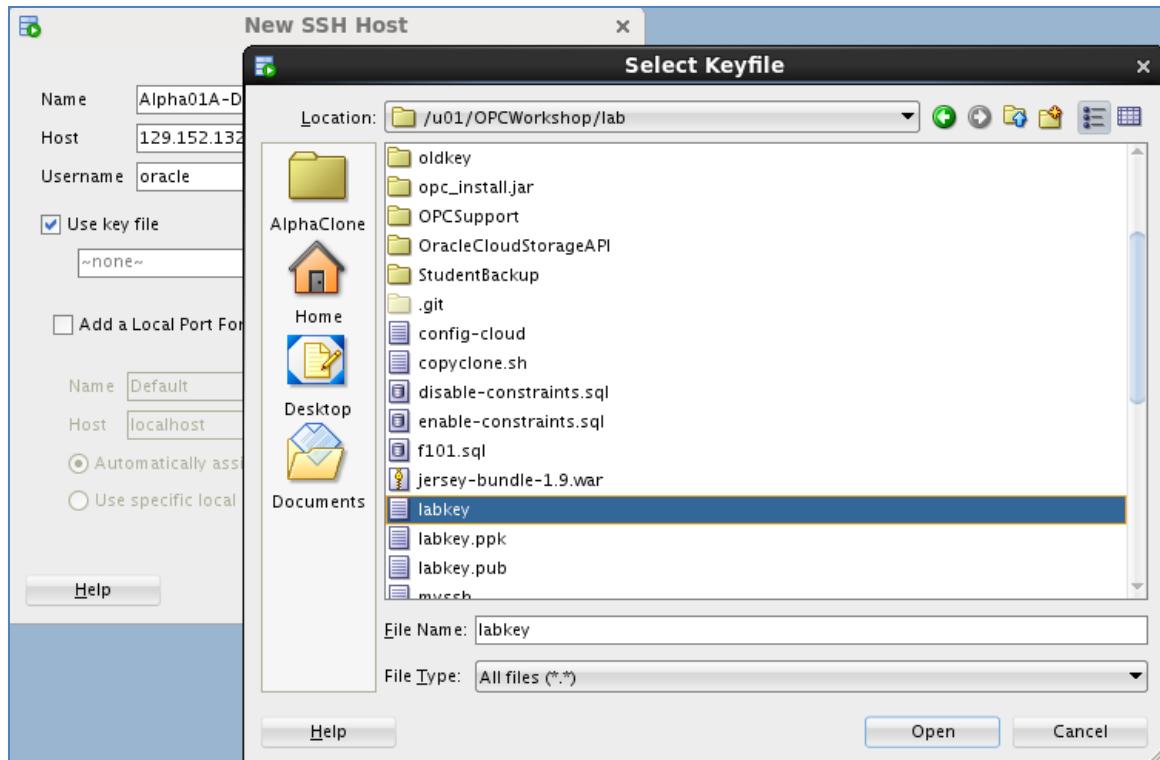


Enter the following information to configure the SSH connection to the DBCS instance

Name	Alpha01A-DBCS
Host	Public IP of Alpha01A-DBCS
Username	oracle



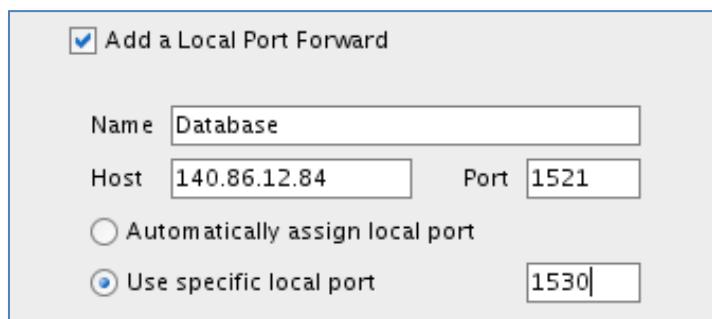
- Select **Use key file** and click **Browse...** Select file **/u01/OPCWorkshop/lab/labkey** and click **Open**.



- Click **Add a Local Port Forward** and enter the following values:

Name	Database
Host	Public IP of Alpha01A-DBCS
Use specific local port	1530

NOTE: We are using port 1530 since 1521 is already in use for our local database.



- Verify the configuration and click **OK**
- Click the green plus sign  in the Connections window to create a new SQL Developer connection to the Public Cloud Database SYS schema

- Enter the following connection details:

Connection Name: sys - OPCDBCS

Username: sys

Password: *****

Save Password Connection Color

Oracle

Connection Type: SSH Role: SYSDBA

Port Forward: Database (Alpha01A-DBCS)

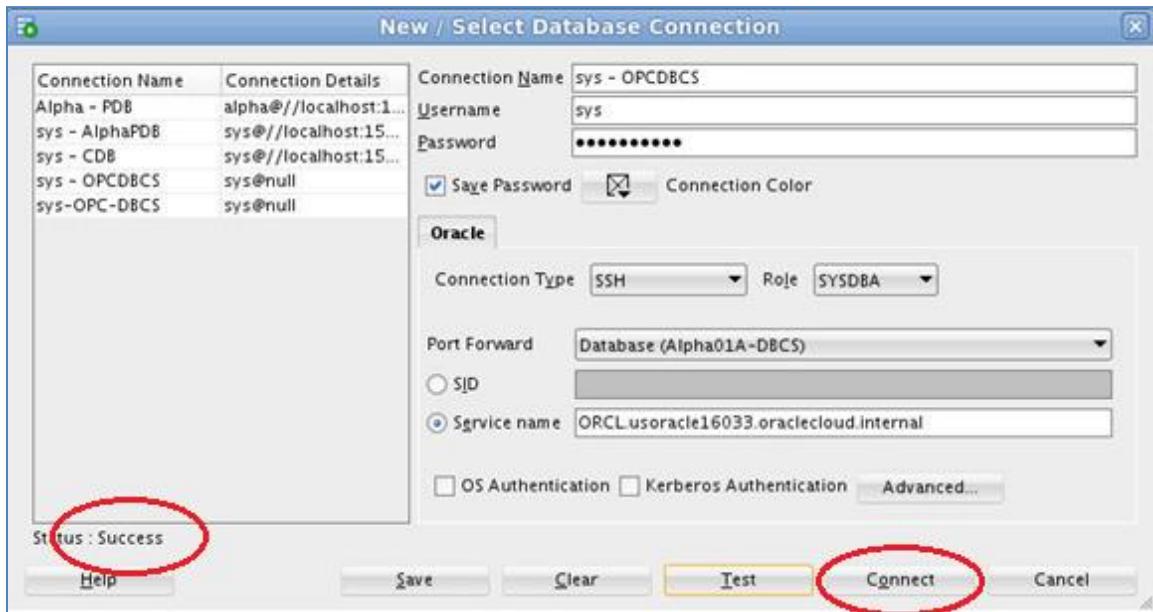
SID Service name: ORCL.usoracle16033.oraclecloud.internal

OS Authentication Kerberos Authentication

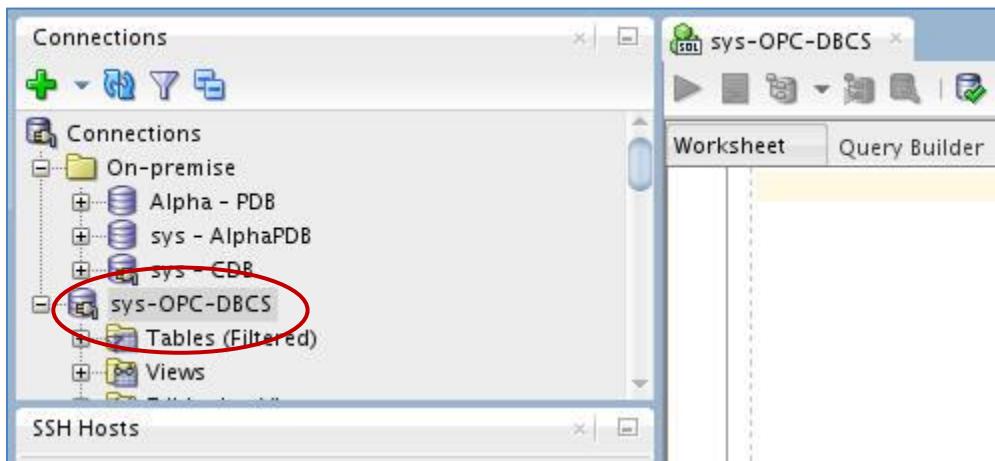
Connection Name:	sys - OPCDBCS
Username:	sys
Password:	Alpha2014_
Check:	“Save Password”
Connection Type:	SSH
Role:	SYSDBA
Service Name:	ORCL.<Your Domain ID>.oraclecloud.internal

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. Verify that you receive a ‘Success’ status.



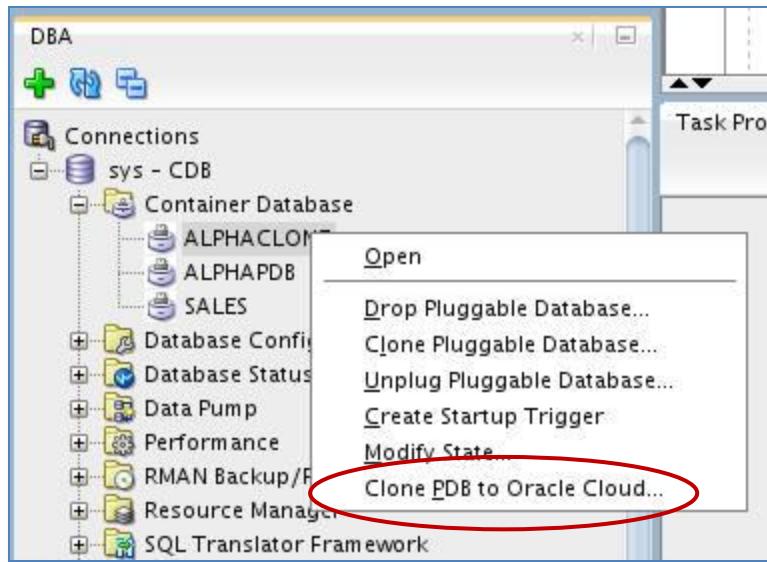
- Click **Connect** to save the connection information which opens a new SQL Worksheet.



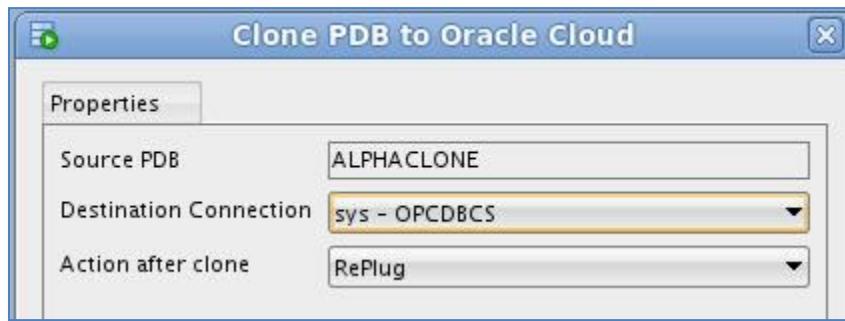
2.4.4: COPY THE CLONE PLUGGABLE DATABASE TO THE CLOUD

In this step we will copy the cloned pluggable database to the cloud using SQL Developer.

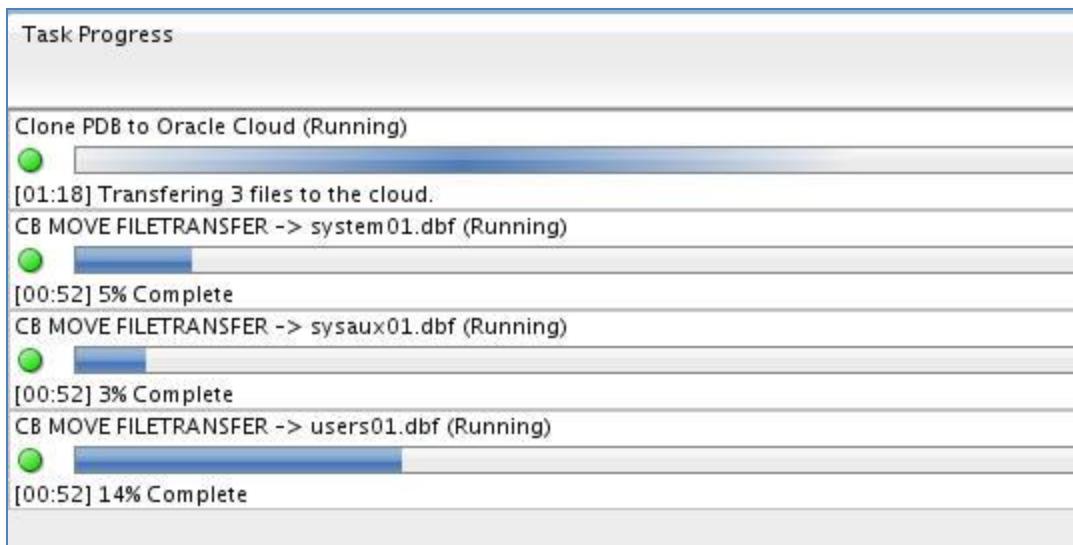
- Click on **View→Task Progress** to open up the Task Progress window.
- In the DBA window expand ‘sys – CDB’ and expand ‘Container Database’, then right-click on **ALPHACLONE** and select “**Clone PDB to Oracle Cloud**”



- Nothing needs to be changed in this window, verify that the default properties include your Public Cloud Connection. Click **Apply**.
- ✓ Source PDB: **ALPHACLONE**
- ✓ Destination Connection: **sys - OPCDBCS**
- ✓ Action after clone: **RePlug**

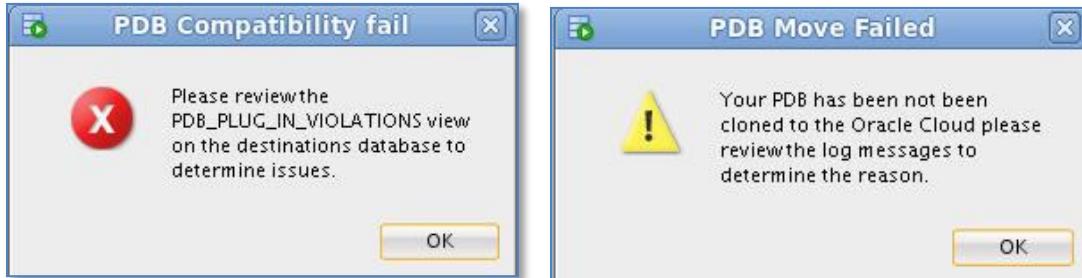


- You will note in the Task Progress window the progress of moving the datafiles over to the cloud database. This task will take about a minute to complete.



- Upon completion of the transfer you will be alerted to at least two Plugin Violations. This is because the patch level of the local ALPHACLONE pluggable database is different than the Container database in the cloud. We will remedy this in the next few steps.
- Click **OK** for **each** popup.

Note: The datafiles will be transferred despite what the pop up implies.



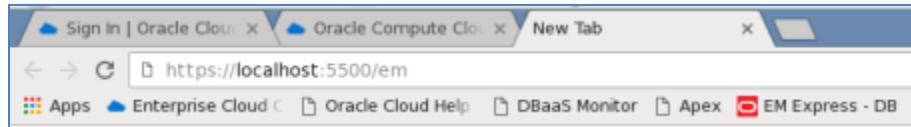
2.4.5: USE EM EXPRESS TO PLUG THE TRANSFERRED DATABASE

- Open Chrome by clicking the icon on the menu bar or the Desktop.



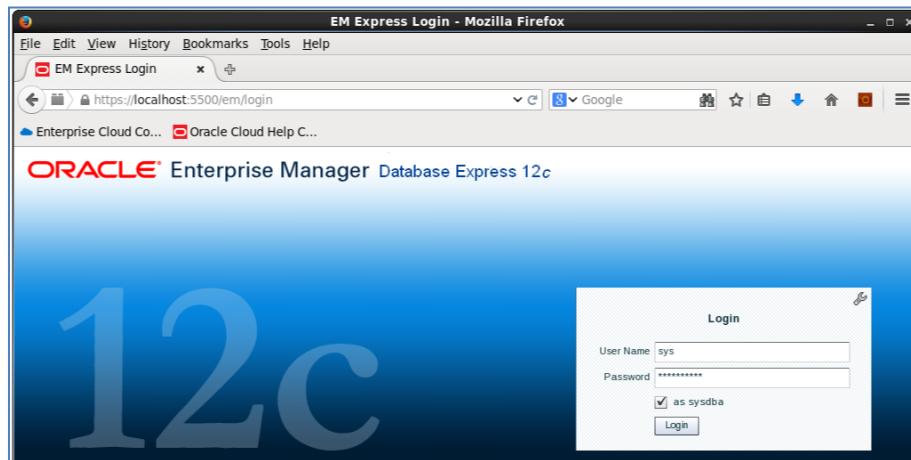
- Enter the following URL into the Address bar or click the "**EM Express - DB**" link in the header bar – <https://localhost:5500/em>

Note: When using localhost:5500 in the URL below, your browser request is routed through the SSH proxy that we loaded in a terminal window in the first lab. If for some reason that window was closed, or is not working, you should refer to the first lab in step 1.6.2

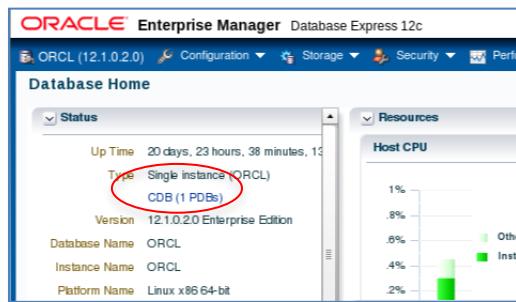


- Enter the following login credentials, check the "as sysdba" box and click the Login button:

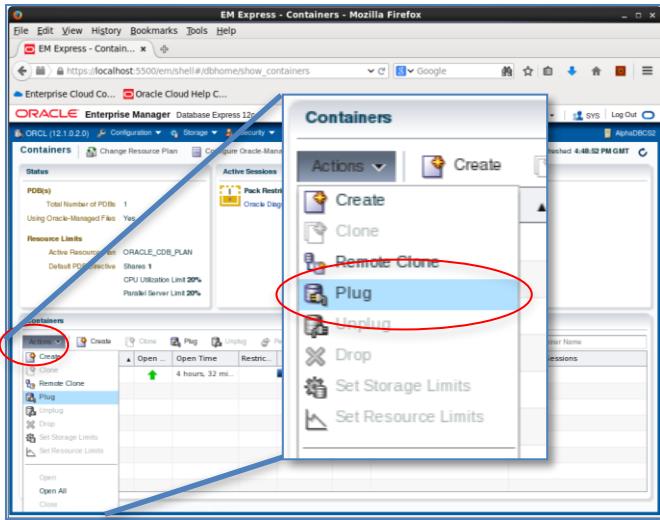
User Name:	sys
Password:	Alpha2014_
Check:	as sysdba



- We will now plug the Alpha Clone database into the Cloud database. From the Database Home page, click the **CDB(1 PDBs)** link.



- Open the **Actions** list and select the **Plug** command



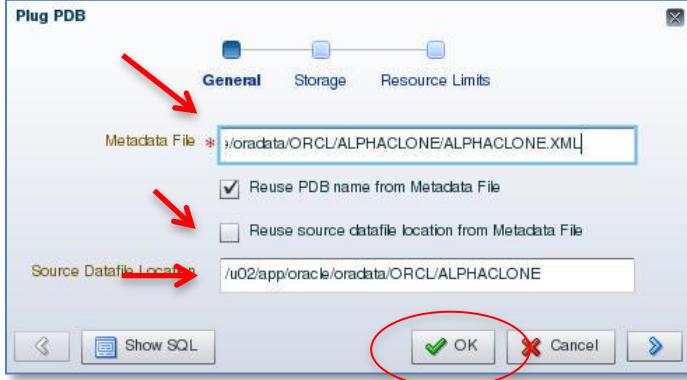
- Enter the following filename and directory location in the Metadata File field in the Plug PDB dialog box.

Metadata File: /u02/app/oracle/oradata/ORCL/ALPHACLONE/ALPHACLONE.XML

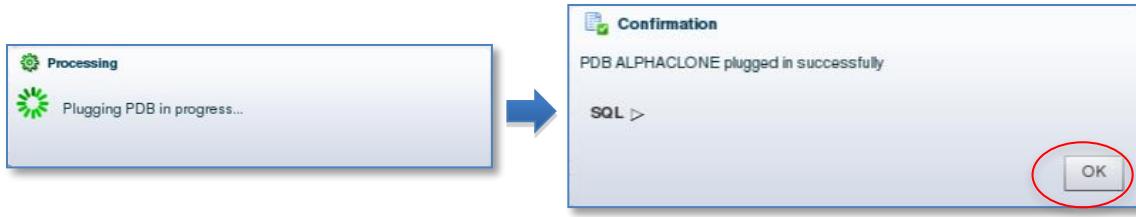
- Uncheck** the Reuse source datafile location from Metadata File check box
- Enter the Source Datafile Location:

Source Datafile Location: /u02/app/oracle/oradata/ORCL/ALPHACLONE

- Click OK



- The Processing message displays for the 2 minutes (approximately) required to plug the database into the container. Click the **OK** button when the Confirmation message displays.



- Notice the database is now in the list of Containers.

Note: There will be Violations because of the patch level mismatch between the original source Pluggable database and the Cloud Container database.

Containers									
Container Name	Open...	Open Ti...	Restr...	Size	Viola...	CPU Resource Limits	Runn...	Container Name	
ALPHAclone	▲	6 seconds	●	895MB	✖	100%	100%		
ALPH_CL3	▲	15 hours,...	●	895MB		100%	100%		
PDB1	▲	2 days, 2...	●	2GB		100%	100%		

- You now need to SSH into the Cloud database server in order to patch the database. Example is shown below. Substitute your Cloud database server IP address (Alpha01A-DBCS)
- Open a Terminal and type the following SSH command to connect to the cloud database server.
- `ssh -o ServerAliveInterval=60 -i /u01/OPCWorkshop/lab/labkey oracle@<Alpha01A-DBCS-IP-address>`

```
oracle@Alpha01A-DBCS:~$ ssh -o ServerAliveInterval=60 -i /u01/OPCWorkshop/lab/labkey oracle@140.86.39.189
[oracle@f499a3 OPCWorkshop]$
```

- We'll need to run the **datapatch** script to apply any missing patches. It should run with no errors.

Note: If it complains the first time about not being able to determine the current opatch status then wait a minute until it's had time to pick up the newly cloned pluggable database and retry.

- `$ORACLE_HOME/OPatch/datapatch -verbose`

```
[oracle@Alpha01A-DBCS ~]$ $ORACLE_HOME/OPatch/datapatch -verbose
SQL Patching tool version 12.1.0.2.0 on Thu Aug 27 18:52:50 2015
Copyright (c) 2015, Oracle. All rights reserved.

Log file for this invocation: /u01/app/oracle/cfgtoollogs/sqlpatch/sqlpatch_1652
3_2015_08_27_18_52_50/sqlpatch_invocation.log

Connecting to database...OK
Note: Datapatch will only apply or rollback SQL fixes for PDBs
      that are in an open state, no patches will be applied to closed PDBs.
      Please refer to Note: Datapatch: Database 12c Post Patch SQL Automation
      (Doc ID 1585822.1)
Bootstrapping registry and package to current versions...■
```

```
ID 3 in the binary registry and ID 3 in PDB CDB$ROOT, ID 3 in PDB PDB$SEED, ID
3 in PDB PDB1, ID 3 in PDB ALPH_CL3, ID 3 in PDB ALPHAclone

Adding patches to installation queue and performing prreq checks...
Installation queue:
For the following PDBs: CDB$ROOT PDB$SEED PDB1 ALPH_CL3
  Nothing to roll back
  Nothing to apply
For the following PDBs: ALPHAclone
  Nothing to roll back
  The following patches will be applied:
    19665921 ()

Installing patches...
Patch installation complete. Total patches installed: 1

Validating logfiles...
Patch 19665921 apply (pdb ALPHAclone): SUCCESS
  logfile: /u01/app/oracle/cfgtoollogs/sqlpatch/19665921/18730325/19665921_apply
  ORCL_ALPHAclone_2015Aug27_18_53_49.log (no errors)
```

In the next few steps we'll upgrade the PDB.

Note: If you receive an error message like, “**The pluggable databases that need to be patched must be in upgrade mode**” complete the following upgrade PDB Step. If not, proceed directly to the next step (Close and Reopen ALPHAclone PDB).

```
Adding patches to installation queue and performing prereq checks...
Installation queue:
  For the following PDBs: CDB$ROOT PDB$SEED PDB1
    Nothing to roll back
    Nothing to apply
  For the following PDBs: ALPHACLONE
    The following patches will be rolled back:
      21555660 (Database PSU 12.1.0.2.5, Oracle JavaVM Component (Oct2015))
    The following patches will be applied:
      22139226 (Database PSU 12.1.0.2.160119, Oracle JavaVM Component (Jan2016))
      21948354 (Database Patch Set Update : 12.1.0.2.160119 (21948354))
      22543975 ()

Error: prereq checks failed!
patch 21555660: The pluggable databases that need to be patched must be in upgrade mode
patch 22139226: The pluggable databases that need to be patched must be in upgrade mode
Prereq check failed, exiting without installing any patches.

Please refer to MOS Note 1609718.1 and/or the invocation log
/u01/app/oracle/cfgtoollogs/sqlpatch/sqlpatch_21674_2016_03_31_20_08_16/sqlpatch_invocation.log
for information on how to resolve the above errors.

SQL Patching tool complete on Thu Mar 31 20:08:59 2016
[oracle@Alpha01A-DBCS ~]$
```

- ✓ Put the database in upgrade mode to correct the patch errors.
- Connect to container database using SQL Plus and place the database in **upgrade mode**. Once completed run **datapatch** again and you should have no errors. Run the following commands to complete this step.
 - sqlplus / as sysdba
 - alter pluggable database ALPHACLONE close;
 - alter pluggable database ALPHACLONE open upgrade;
 - exit
 - \$ORACLE_HOME/OPatch/datapatch -verbose

```
[oracle@Alpha01A-DBCS ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Thu Mar 31 21:48:39 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Oracle Label Security and Real Application Testing options

SQL> alter pluggable database ALPHAACLONE close;
Pluggable database altered.

SQL> alter pluggable database ALPHAACLONE open upgrade;
Pluggable database altered.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Oracle Label Security and Real Application Testing options
[oracle@Alpha01A-DBCS ~]$ ■

For the following PDBs: ALPHAACLONE
The following patches will be rolled back:
  21555660 (Database PSU 12.1.0.2.5, Oracle JavaVM Component (Oct2015))
The following patches will be applied:
  22139226 (Database PSU 12.1.0.2.160119, Oracle JavaVM Component (Jan2016))
  21948354 (Database Patch Set Update : 12.1.0.2.160119 (21948354))
  22543975 ()

Installing patches...
Patch installation complete. Total patches installed: 4

Validating logfiles...
Patch 21555660 rollback (pdb ALPHAACLONE): SUCCESS
  logfile: /u01/app/oracle/cfgtoollogs/sqlpatch/21555660/19361790/21555660_rollback_ORCL_ALPH
ACLONE_2016Mar31_21_50_57.log (no errors)
Patch 22139226 apply (pdb ALPHAACLONE): SUCCESS
  logfile: /u01/app/oracle/cfgtoollogs/sqlpatch/22139226/19729684/22139226_apply_ORCL_ALPHACL
ONE_2016Mar31_21_51_35.log (no errors)
Patch 21948354 apply (pdb ALPHAACLONE): SUCCESS
  logfile: /u01/app/oracle/cfgtoollogs/sqlpatch/21948354/19553095/21948354_apply_ORCL_ALPHACL
ONE_2016Mar31_21_51_35.log (no errors)
Patch 22543975 apply (pdb ALPHAACLONE): SUCCESS
  logfile: /u01/app/oracle/cfgtoollogs/sqlpatch/22543975/19772638/22543975_apply_ORCL_ALPHACL
ONE_2016Mar31_21_51_36.log (no errors)
SQL Patching tool complete on Thu Mar 31 21:51:40 2016
[oracle@Alpha01A-DBCS ~]$ ■
```

✓ **Close and Reopen ALPHAACLONE PDB**

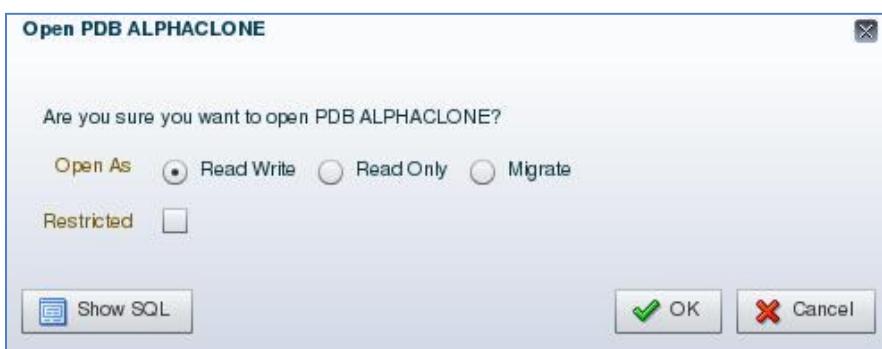
- The final step is to close and reopen the ALPHAACLONE pluggable database. Go back to EM Express, with the ALPHAACLONE row highlighted. (DO NOT CLICK THE ALPHAACLONE LINK).
- Select **Actions → Close**.



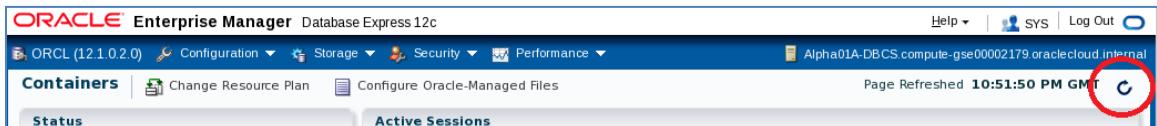
- Accept the defaults and Click **OK**.



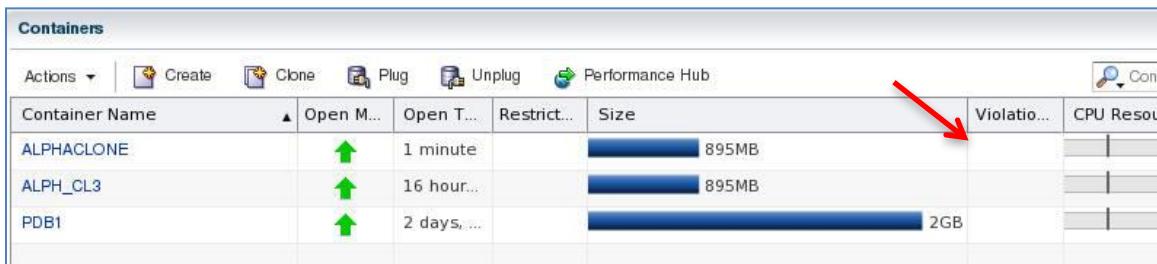
- Close the Confirmation pop up by clicking **OK**.
- Reopen the pluggable database via **Actions → Open**.
- Accept the default to open it Read / Write. Click **OK**.



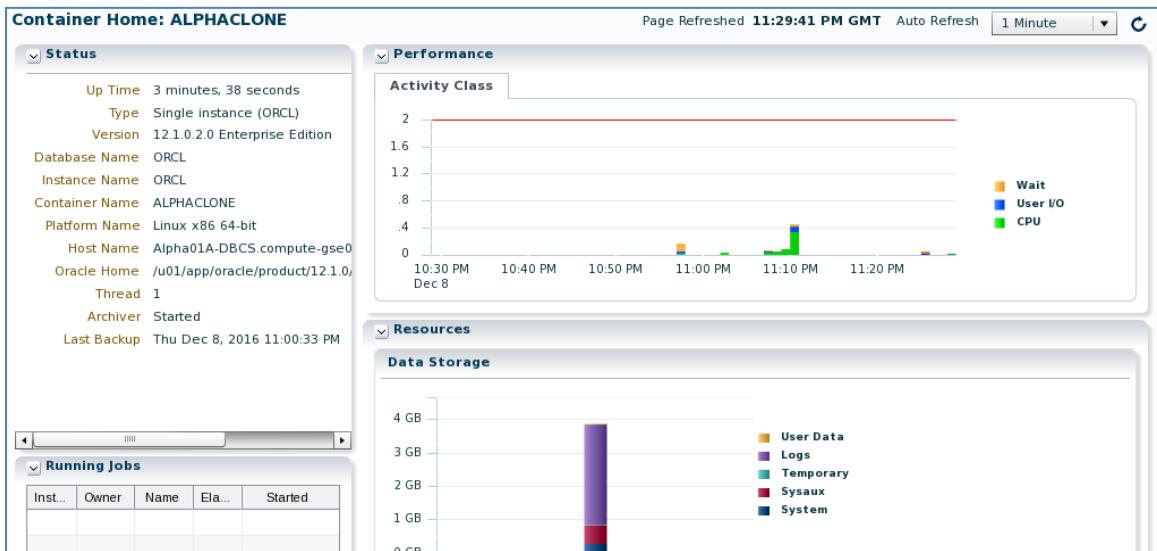
- Close the Confirmation pop up by clicking **OK**.
- Refresh the browser page using the refresh icon in the top right corner.



- Note that the violations are now gone.



- Click the **ALPHAclone** container name link to review the database information.



2.4.6: 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:

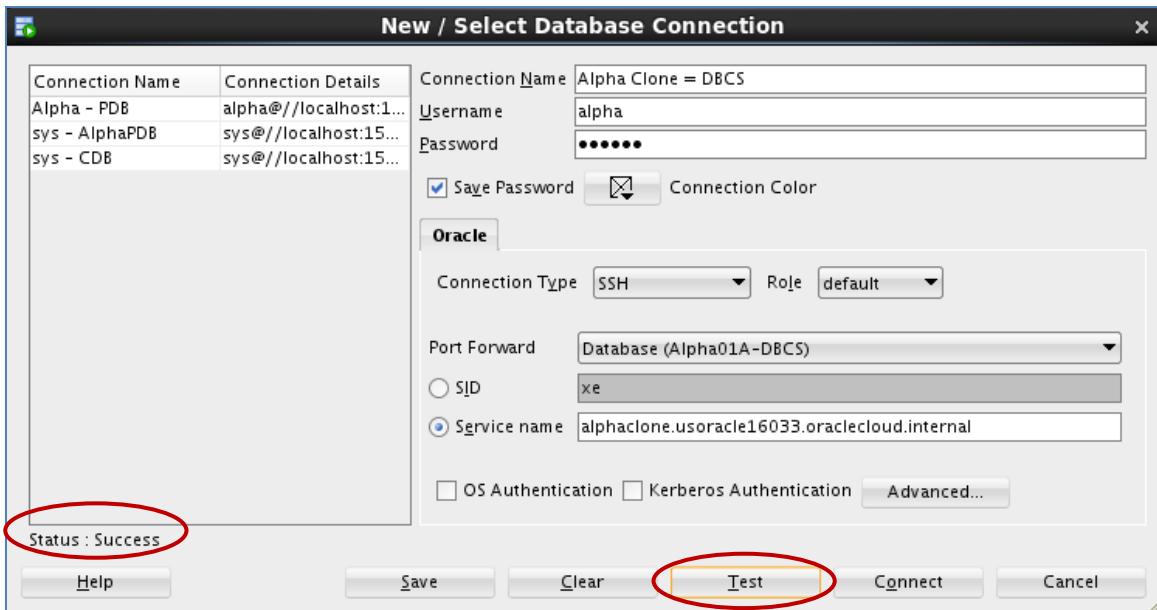
Connection Name:	Alpha Clone – DBCS
-------------------------	--------------------

Username:	alpha
Password:	oracle
Check:	“Save Password”
Connection Type:	SSH
Service Name:	Alphaclone.<Your ID Domain>.oraclecloud.internal

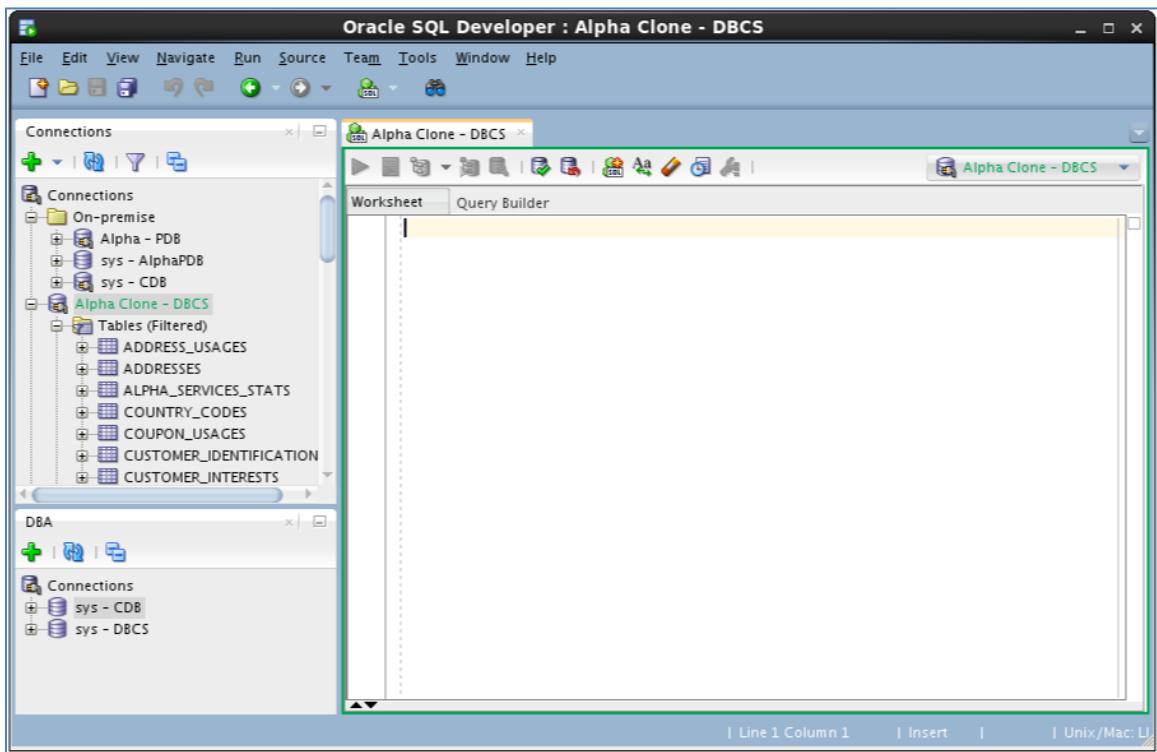
Note: You can optionally select a color for the connection to differentiate it from other connections.

The screenshot shows the Oracle Connection dialog box. At the top, the Connection Name is set to "Alpha Clone = DBCS". Below that, the Username is "alpha" and the Password is masked as "*****". There is a checked checkbox for "Save Password" and a dropdown for "Connection Color". A tab labeled "Oracle" is selected, showing the Connection Type as "SSH" and the Role as "default". Under "Port Forward", the dropdown is set to "Database (Alpha01A-DBCS)". Below that, there are two radio buttons: "SID" (selected) with value "xe" and "Service name" with value "alphaclone.usoracle16033.oraclecloud.internal". At the bottom, there are checkboxes for "OS Authentication" and "Kerberos Authentication", both of which are unchecked. A "Advanced..." button is also present.

- Click **Test** to confirm the information was entered correctly.



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



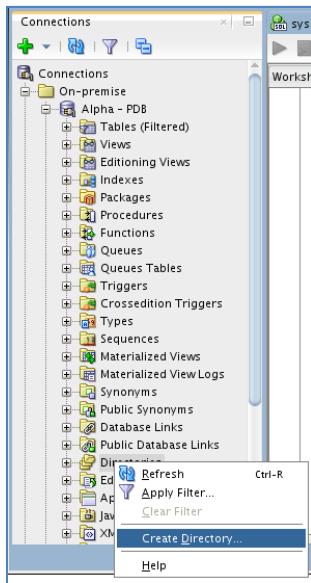
- ✓ You have successfully migrated a pluggable database from on-premise to the cloud. In the next section we'll migrate data using Data Pump.

2.5: Cloud Migration Using Data Pump

2.5.1: 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...**

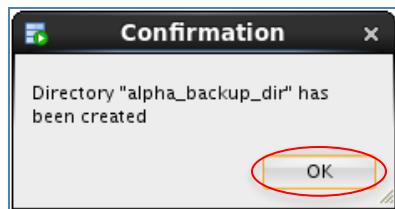
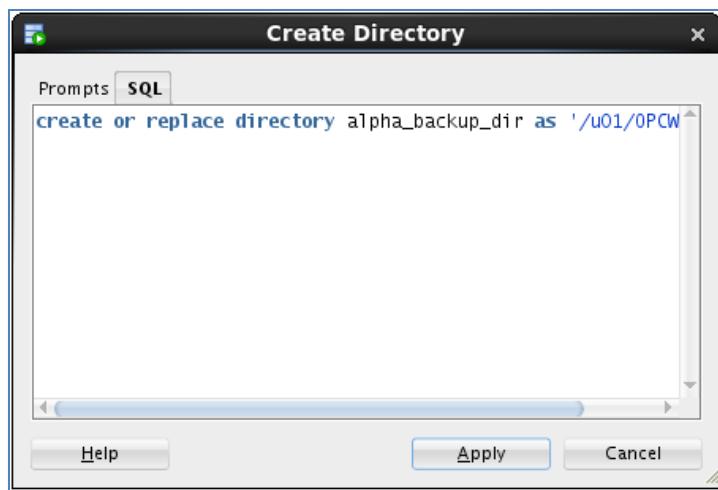
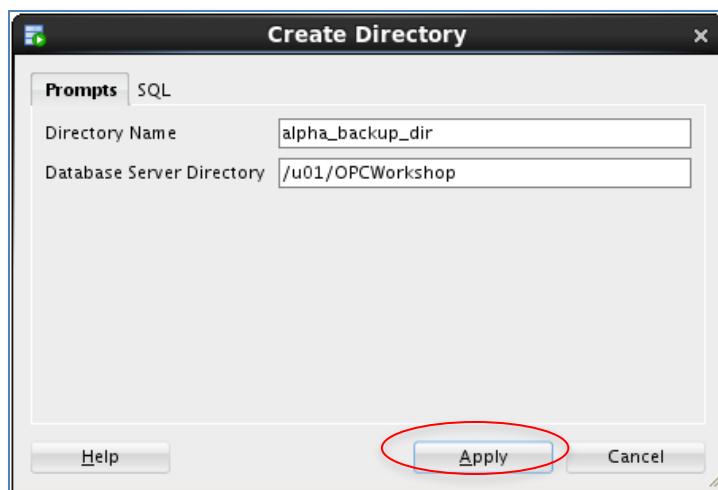


Note: The default Data Pump directory object, `DATA_PUMP_DIR`, does not work with PDBs. Data Pump requires an explicit directory object within the PDB for exporting or importing schemas or tables.

- 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

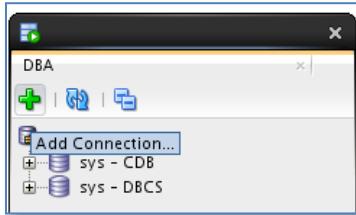
NOTE: You may receive an error message stating that “An error was encountered performing the requested operation.” and that the directory cannot be created. To eliminate this error right-click on Alpha - PDB and choose Disconnect. Then Reconnect. The error occurs because you were connected earlier while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



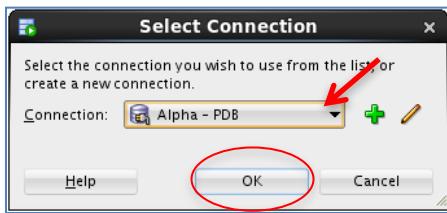
Stop here-----

- ✓ 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

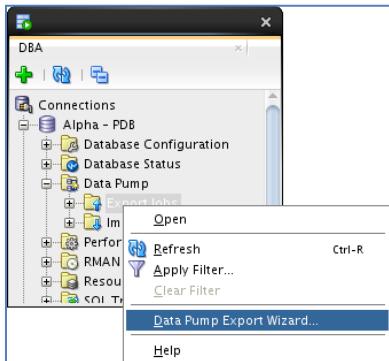
- In the DBA Window, **Add Connection** by clicking on the Green Plus sign.



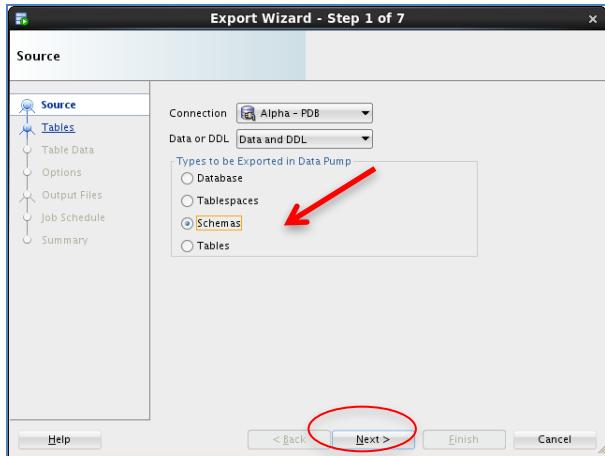
- Select the **Alpha - PDB** connection and click the **OK** button.



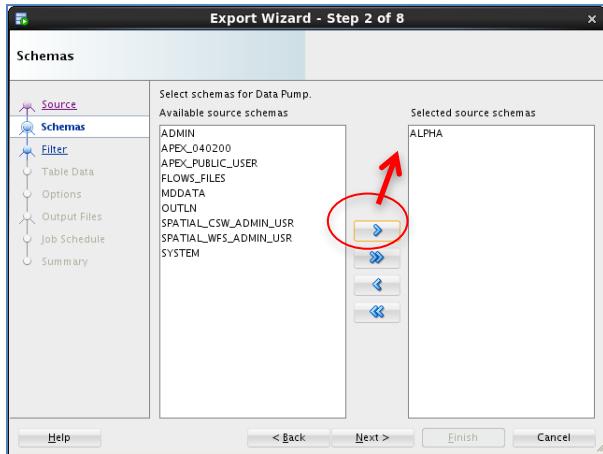
- Expand **Alpha - PDB**, expand **Data Pump**, then right-mouse-click on **Export Jobs**, and then select the **Data Pump Export Wizard...** menu item.



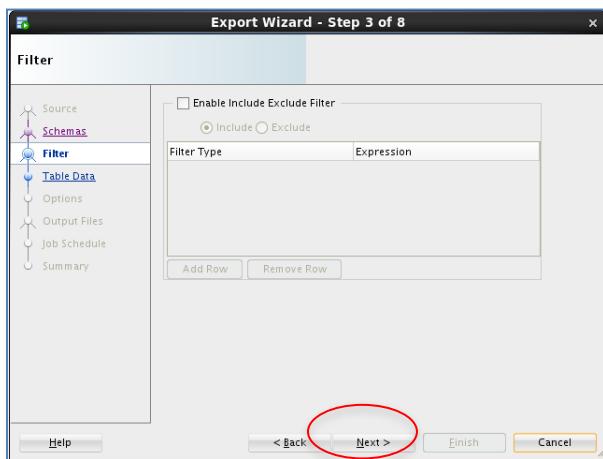
- Select the **Schemas** export type and click the **Next** button.



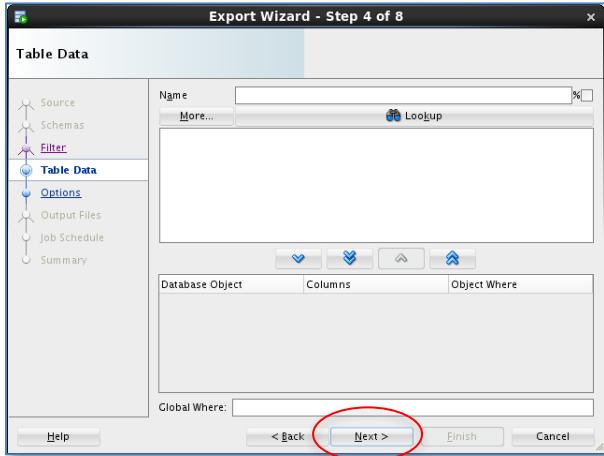
- Select the **ALPHA** schema and use the blue arrow to move it to the right-hand column. Click **Next**.



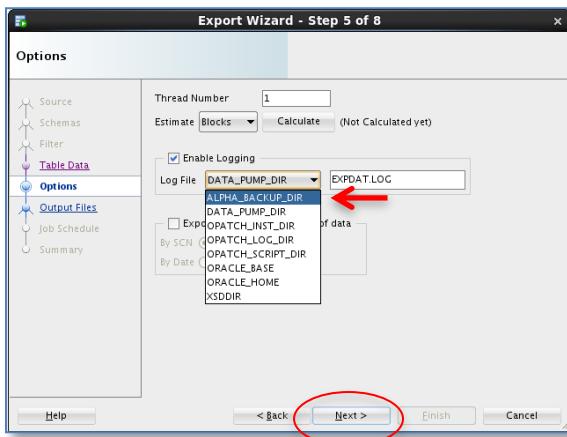
- We are not filtering out any objects, click the **Next** button.



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

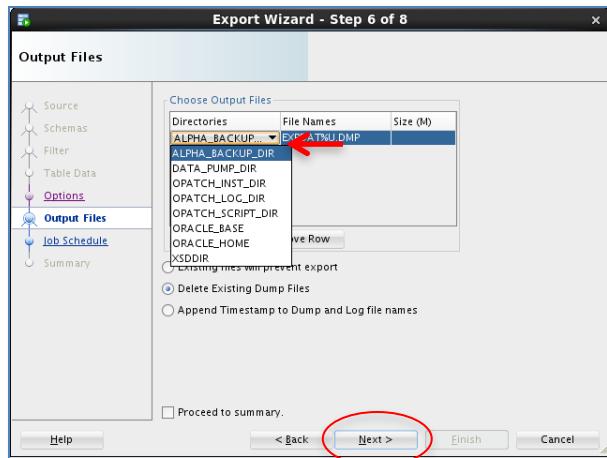


- ✓ We want a log for this export, and just like the actual export file, we must pick a directory from the list of directories in the database.
 - Select **ALPHA_BACKUP_DIR** from the list and click the **Next** button.

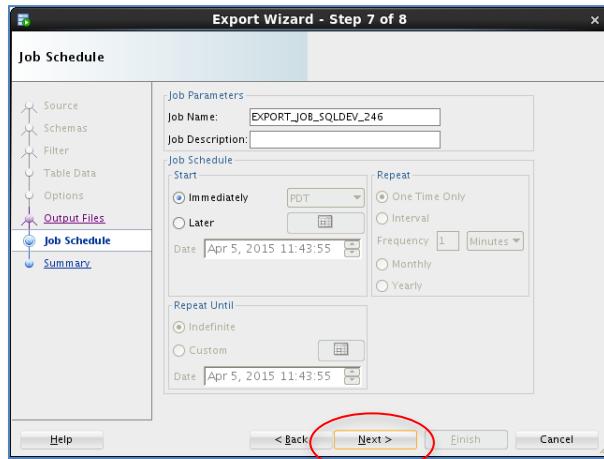


- The most important selection for any Data Pump operation is choosing the directory where the export file will be written.
 - Select **ALPHA_BACKUP_DIR** from the Directories drop down list.
 - Then, select the **Delete Existing Dump Files** radio button and click the **Next** button.

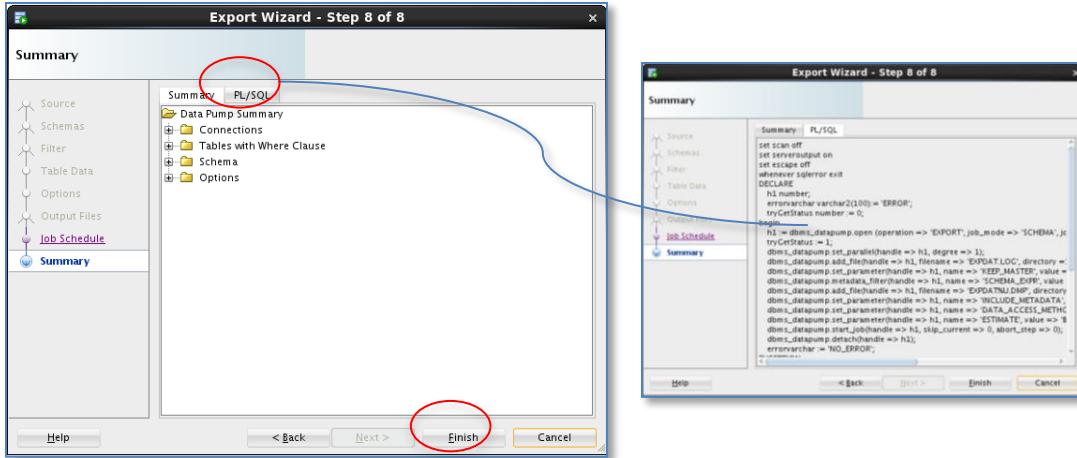
Note: *Data Pump always uses a server side directory for all export or import operations.*



- 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.

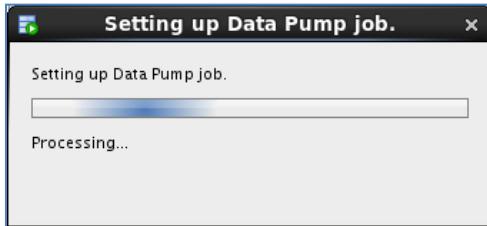


- On the Summary panel, click the **PL/SQL** tab to review the job definition. Review the PL/SQL use of Oracle Supplied PL/SQL subprograms for Data Pump. Click the **Finish** button to create the job.

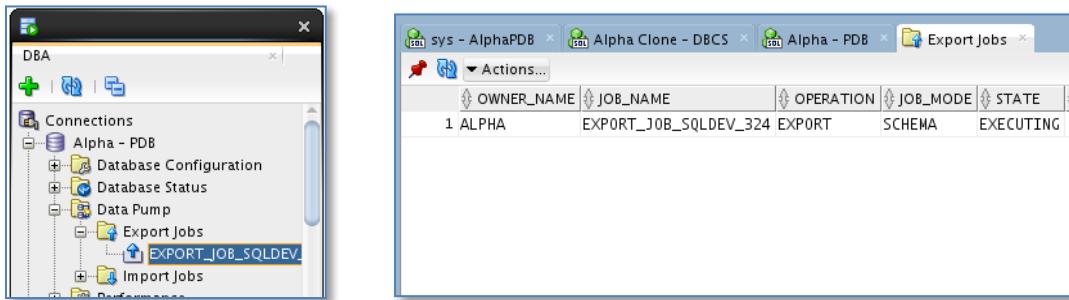


- ✓ For a brief time, SQL Developer shows a progress dialog while it creates the job in the database.

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

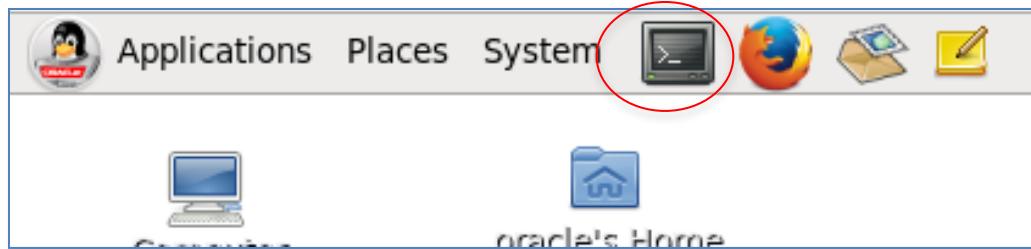


- While the job is running, you may view status information by clicking on the export job added to the DBA Navigator panel. It may take a couple of minutes so click the Refresh icon until the job is completed (NOT RUNNING).



- ✓ 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
```

A screenshot of a terminal window titled "oracle@oraclelinux6:u01/OPCWorkshop". The window shows the following command history:

```
File Edit View Search Terminal Help  
[oracle@oraclelinux6 OPCWorkshop]$ pwd  
/u01/OPCWorkshop  
[oracle@oraclelinux6 OPCWorkshop]$ ls  
EXPDAT01.DMP EXPDAT.LOG lab  
[oracle@oraclelinux6 OPCWorkshop]$ cat EXPDAT.LOG
```

The command "cat EXPDAT.LOG" is highlighted with a red box.A screenshot of a terminal window titled "oracle@e6596d:u01/OPCWorkshop". The window shows the output of a Data Pump export job:

```
File Edit View Search Terminal Help  
. . exported "ALPHA"."DISCOUNTS" 9.812 KB 7 rows  
. . exported "ALPHA"."LOOKUP_CODES" 13.23 KB 56 rows  
. . exported "ALPHA"."MSTARS" 5.539 KB 4 rows  
. . exported "ALPHA"."ORDERS" 13.84 KB 3 rows  
. . exported "ALPHA"."ORDER_ITEMS" 9.390 KB 4 rows  
. . exported "ALPHA"."PAYMENT_OPTIONS" 12.79 KB 7 rows  
. . exported "ALPHA"."PRODUCTS" 24.58 KB 57 rows  
. . exported "ALPHA"."PRODUCT_CATEGORIES" 10.58 KB 20 rows  
. . exported "ALPHA"."PRODUCT_STOCK_LEVELS" 8.648 KB 26 rows  
. . exported "ALPHA"."PRODUCT_IMAGES" 0 KB 0 rows  
Master table "ALPHA"."EXPORT_JOB_SQLDEV_282" successfully loaded/unloaded  
*****  
Dump file set for ALPHA.EXPORT_JOB_SQLDEV_282 is:  
/u01/OPCWorkshop/EXPDAT01.DMP  
Job "ALPHA"."EXPORT_JOB_SQLDEV_282" successfully completed at Thu Dec 8 17:42:27  
2016 elapsed 0 00:00:39  
[oracle@e6596d OPCWorkshop]$
```

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>:~
```

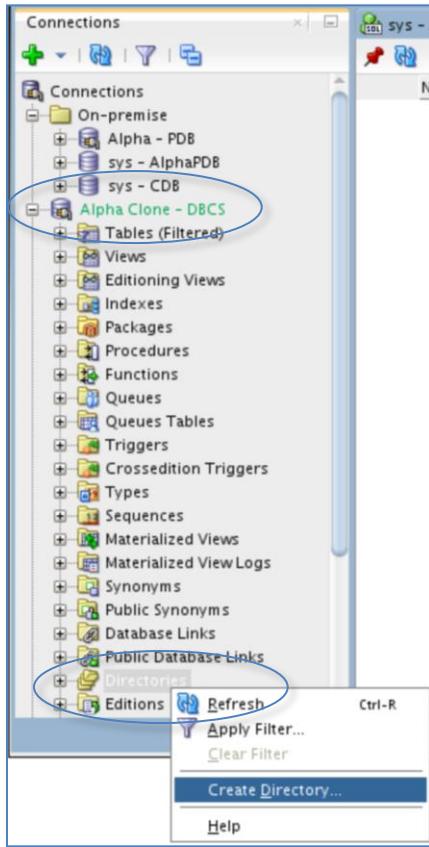
Note: the tilde (~) represents the oracle user's home directory.



```
oracle@oraclelinux6:u01/OPCWorkshop
File Edit View Search Terminal Help
[oracle@oraclelinux6 OPCWorkshop]$ scp -i lab/labkey EXPDAT01.DMP oracle@140.86.12.84:~
EXPDAT01.DMP                                         100% 3776KB 472.0KB/s   00:08
[oracle@oraclelinux6 OPCWorkshop]$
```

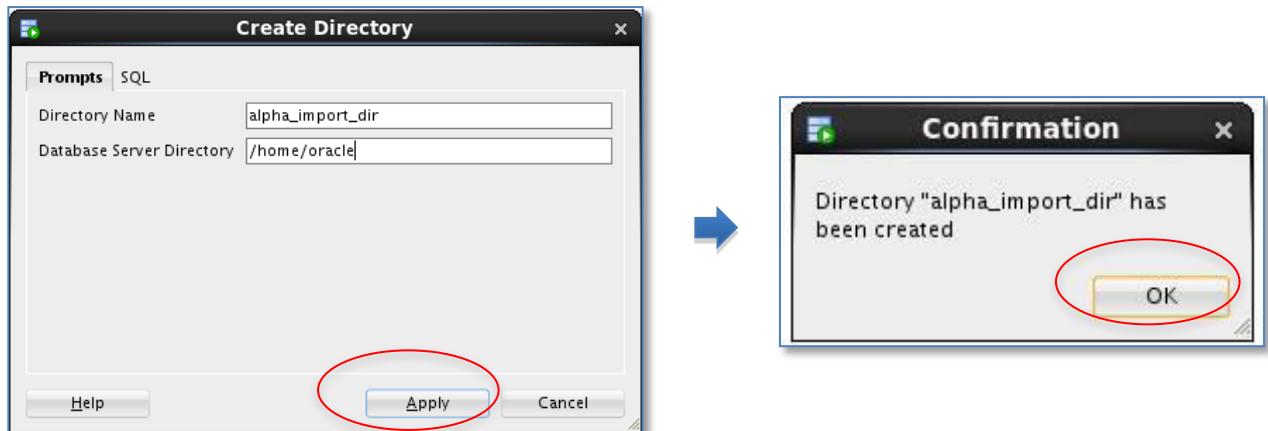
2.5.2: IMPORT ALPHA TO A NEW SCHEMA

- ✓ 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.

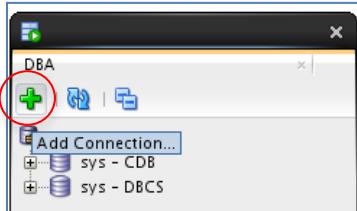


- Enter the following values and click the **Apply** button.
- Click **OK** to dismiss the confirmation message. This lets the database access the same directory where the Data Pump export file was copied.

Directory Name:	alpha_import_dir
Database Server Directory:	/home/oracle



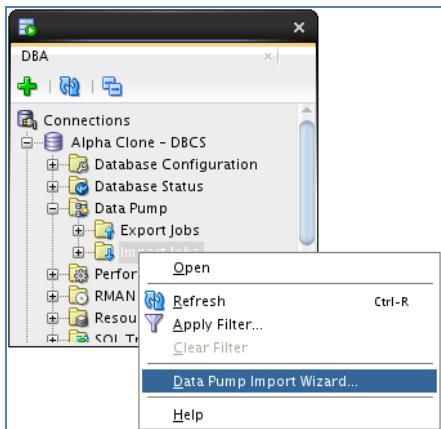
- ✓ The next few steps will outline creating the Data Pump Import job. To access the Data Pump features, we need to add the clone connection to the DBA Navigator.
- Click on the green plus sign, Add Connection icon on the DBA Navigator panel



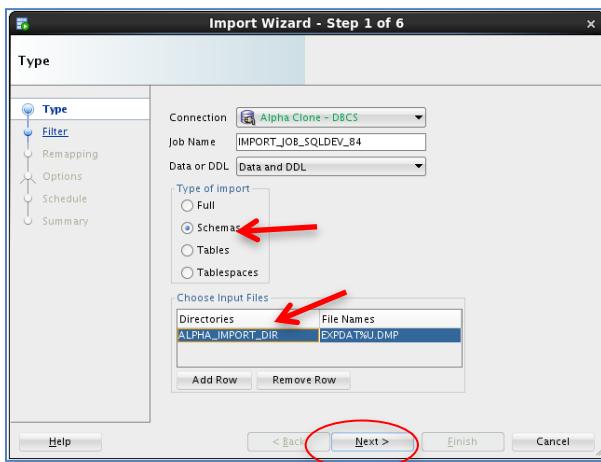
- Select **Alpha Clone - DBCS** connection and click **OK**.



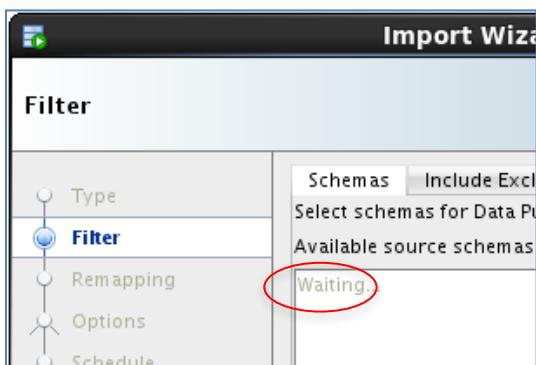
- Expand **Alpha Clone - DBCS → Data Pump**
- Right-mouse on the **Import Jobs** menu item, and select **Data Pump Import Wizard...** menu item.



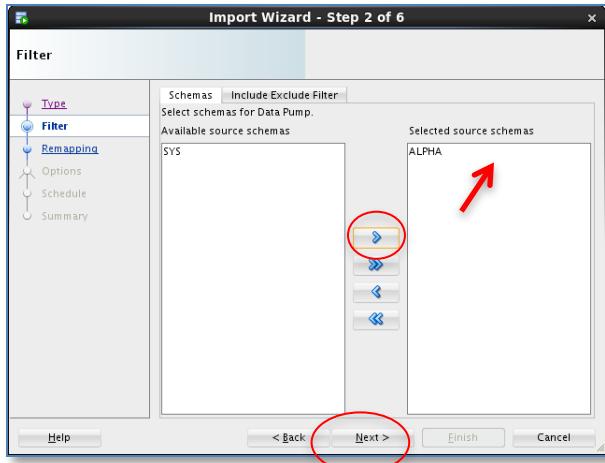
- Select **Schemas** from the 'Type of Import' box and Choose **ALPHA_IMPORT_DIR** from the 'Choose Input Files' drop down list, then click **Next**.



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.

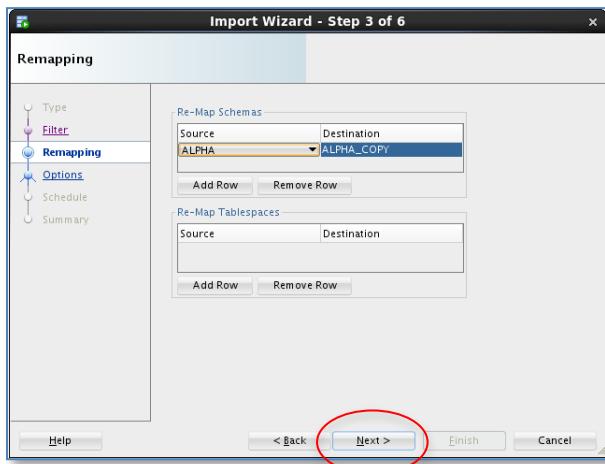


- Move the **ALPHA** schema from the left to the right column using the arrow button and click **Next**.

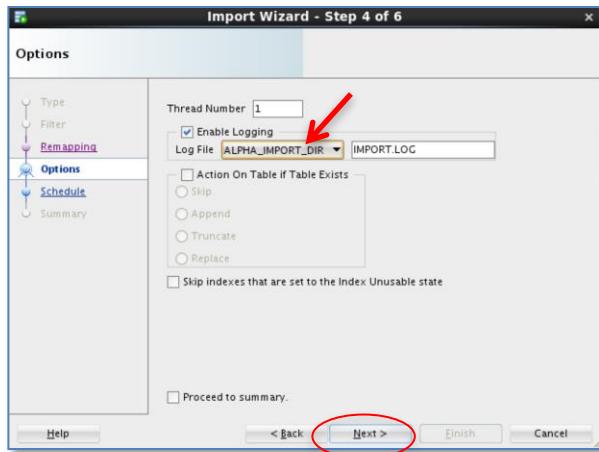


- For this lab, we are creating a new schema, so we will enter the new schema name as the destination.
- Under the Re-Map Schemas section click **Add Row**.
- Enter the following values and click the **Next** button.

Source:	ALPHA (should be the default item)
Destination:	ALPHA_COPY

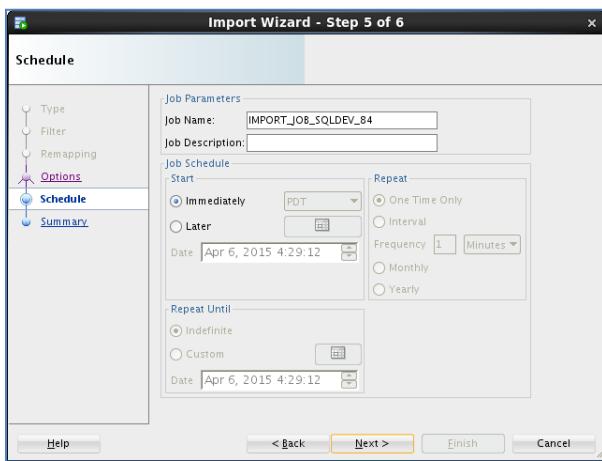


- ✓ We want to see the log output so we will select the same directory as the import file directory.
- Select **ALPHA_IMPORT_DIR** and click the **Next** button.

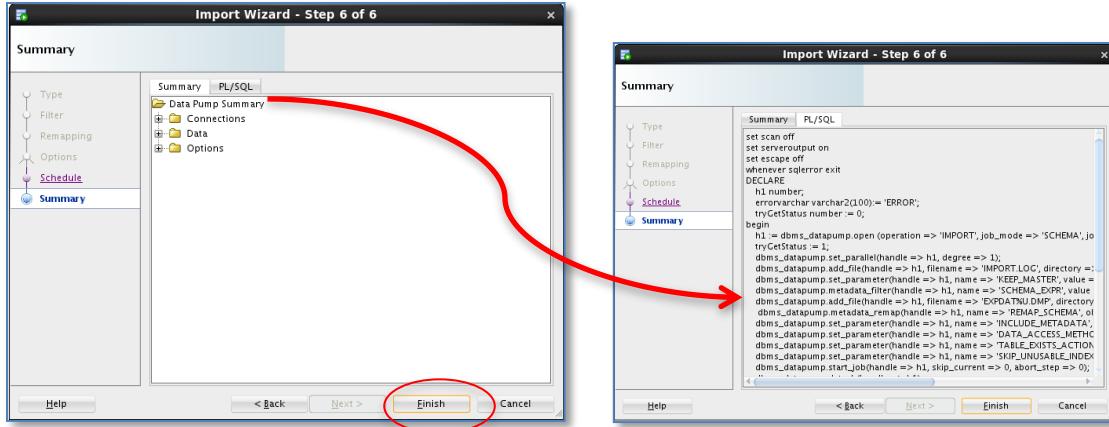


Note: For lab purposes we will execute the import immediately. In normal operations this job could be set up to refresh a development database on a daily basis.

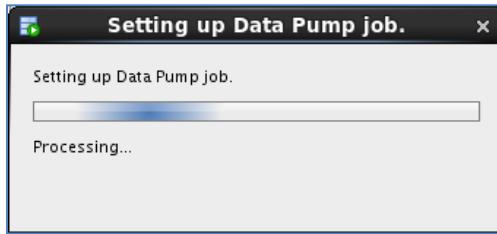
- Click the **Next** button.



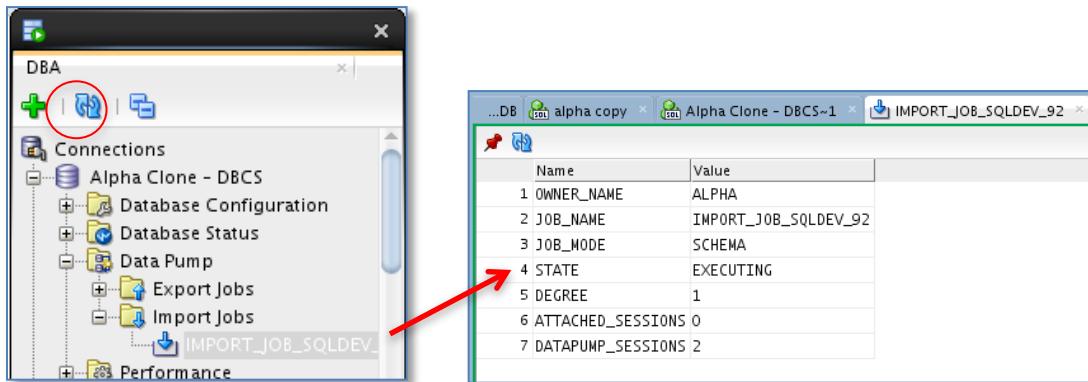
- Click the PL/SQL tab to review the small program that establishes the import job. Click the **Finish** button to create the job.



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.



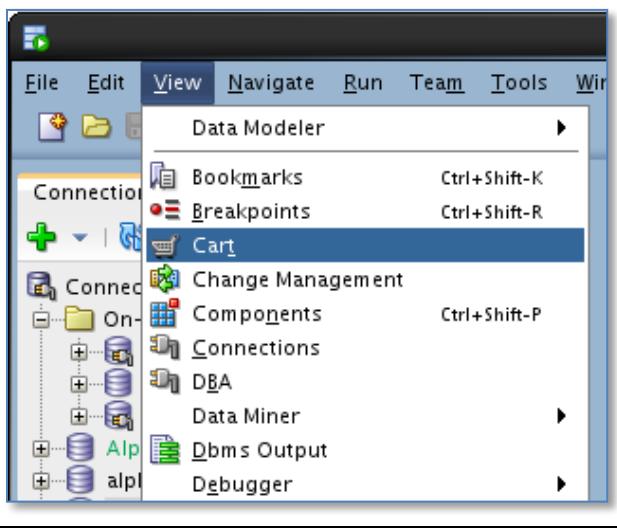
- If you are interested in verifying that the ALPHA_COPY schema is the same as the ALPHA schema, feel free to create a connection and compare.

2.6: Cloud Migration Using SQL Developer Carts

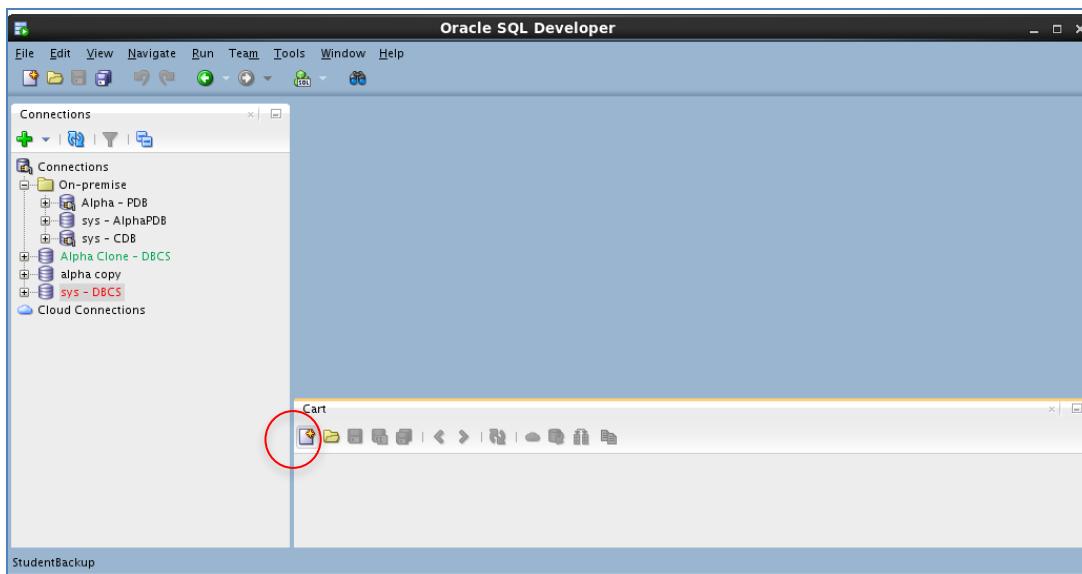
2.6.1: CREATING AN SQL DEVELOPER CART

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.

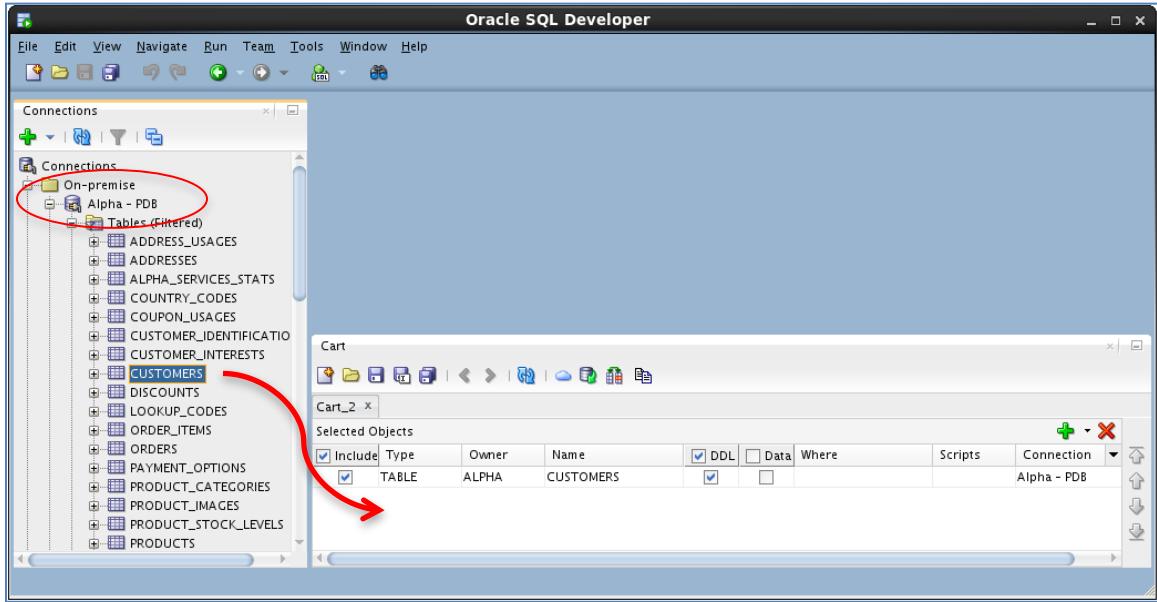
- Show the Cart using the **View > Cart** menu option.



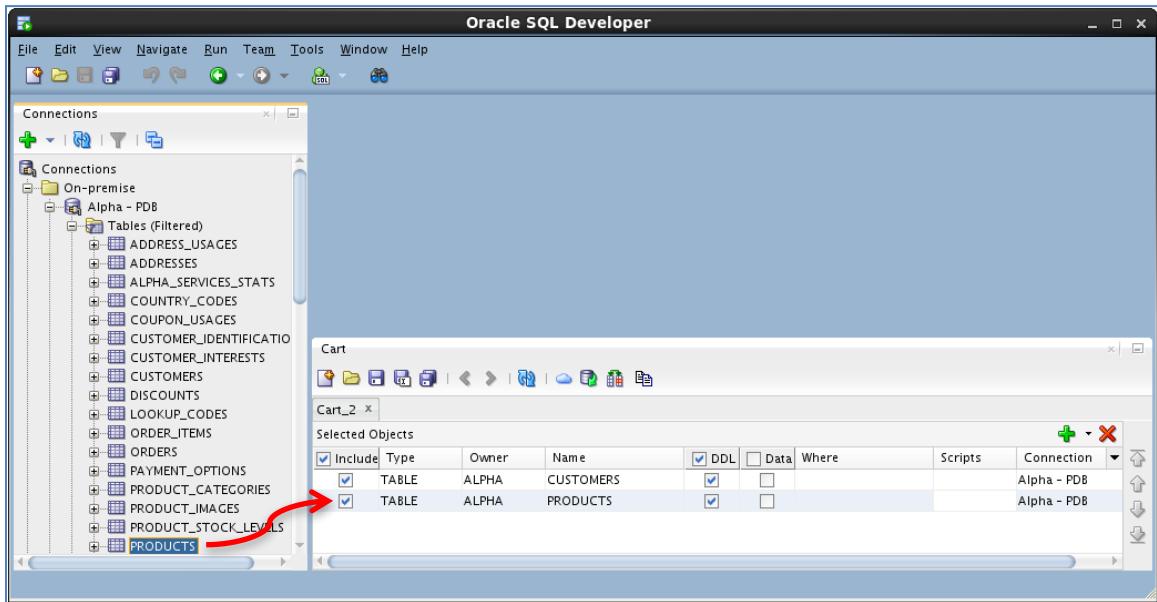
- If Cart_1 is not already created (it should be), Click on the **New Cart** icon.



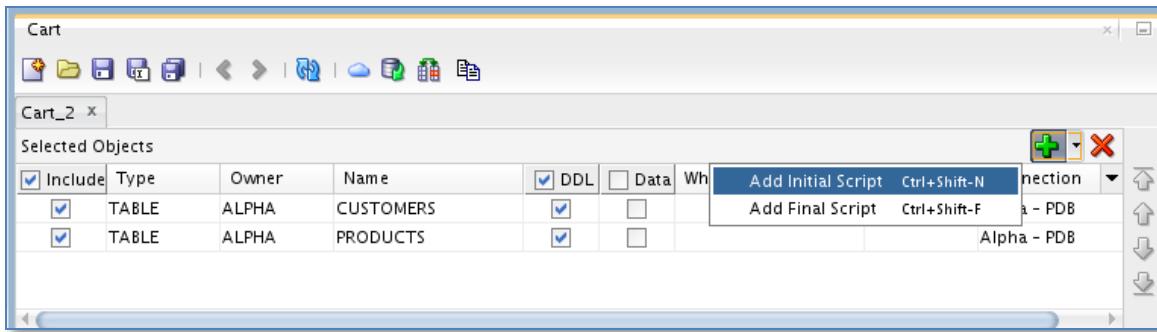
- Drag the **CUSTOMERS** table from the **Alpha - PDB** connection to the cart.



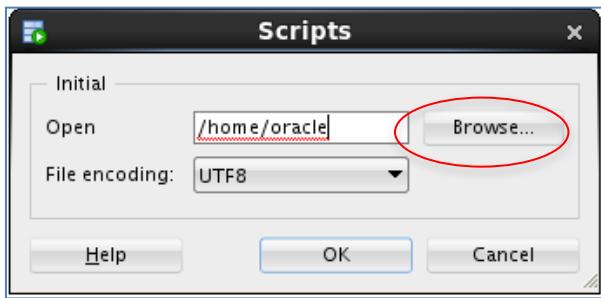
- Drag the **PRODUCTS** table to the cart.



- 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**

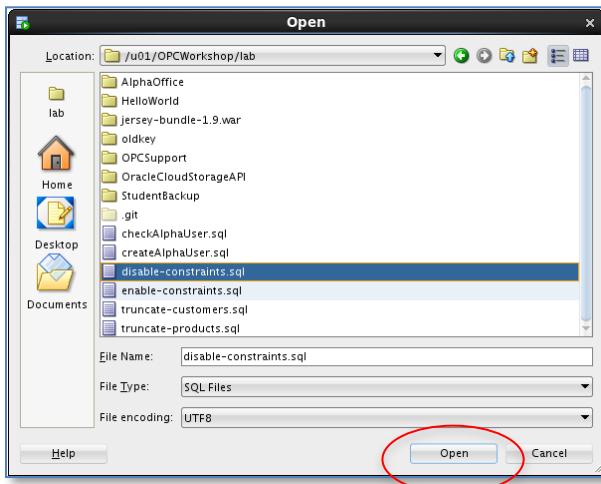


- ✓ Click the **Browse...** button.



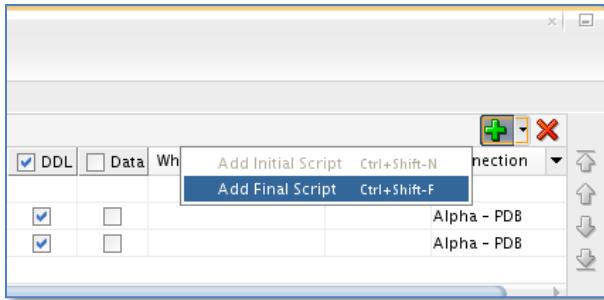
- Locate the following file and click **Open**:

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

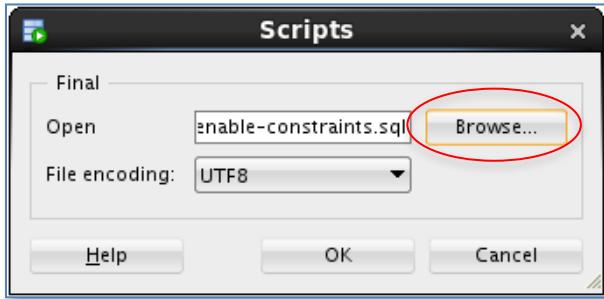


- Click **OK**.
- Click the down arrow next to the **green plus “+” icon** again and select **Add Final Script** which is included as the last operation performed during the cart operations.

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

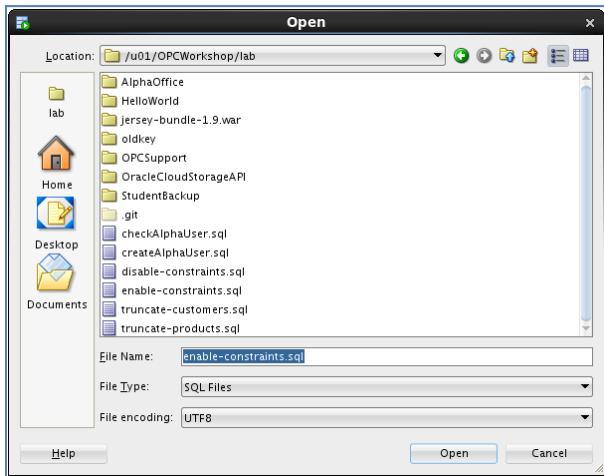


- Click the **Browse...** button



- 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.

Include	Type	Owner	Name	DDL	Data	Where
<input checked="" type="checkbox"/>	FILE	/u01/OPC...	disable-constraints....	<input type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	TABLE	ALPHA	CUSTOMERS	<input type="checkbox"/>	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	TABLE	ALPHA	PRODUCTS	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	FILE	/u01/OPC...	enable-constraints.sql	<input type="checkbox"/>	<input type="checkbox"/>	

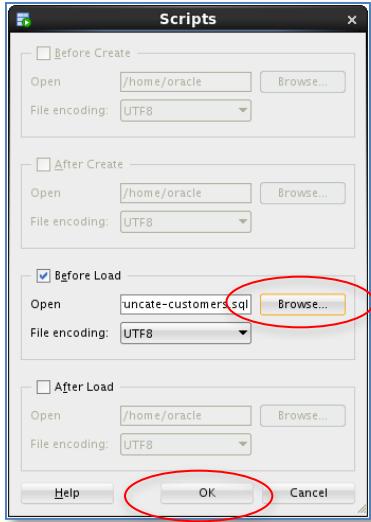
- We will move the data, include a check the **Data** column heading.

Include	Type	Owner	Name	DDL	Data	Where
<input checked="" type="checkbox"/>	FILE	/u01/OPC...	disable-constraints....	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	TABLE	ALPHA	CUSTOMERS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	TABLE	ALPHA	PRODUCTS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	FILE	/u01/OPC...	enable-constraints.sql	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

- ✓ 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**.

Include	Type	Owner	Name	DDL	Data	Where	Scripts	Connection
<input checked="" type="checkbox"/>	FILE	/u01/OPC...	disable-constraints....	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	TABLE	ALPHA	CUSTOMERS	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	Alpha - PDB
<input checked="" type="checkbox"/>	TABLE	ALPHA	PRODUCTS	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	Alpha - PDB
<input type="checkbox"/>	FILE	/u01/OPC...	enable-constraints.sql	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	

- Check the **Before Load** box, then click the **Browse...** button and select the following file:
/u01/OPCWorkshop/lab/truncate-customers.sql
- Click **OK**:



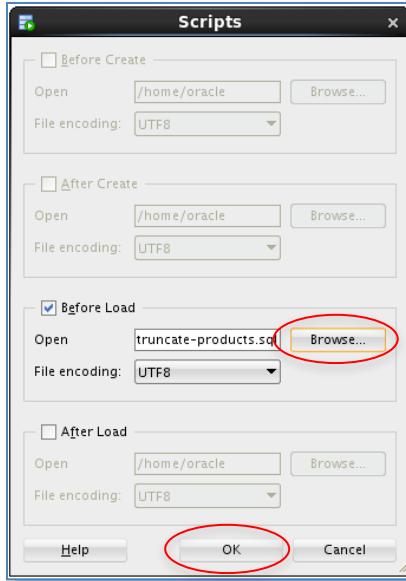
- Repeat the operation for the PRODUCTS table; click the **pencil icon** on the products row.

Cart_2									
Selected Objects									
Include	Type	Owner	Name	DDL	Data	Where	Scripts	Connection	
<input checked="" type="checkbox"/>	FILE	/u01/OPC...	disable-constraints....	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	TABLE	ALPHA	CUSTOMERS	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	Alpha - PDB	
<input checked="" type="checkbox"/>	TABLE	ALPHA	PRODUCTS	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	Alpha - PDB	
<input checked="" type="checkbox"/>	FILE	/u01/OPC...	enable-constraints.sql				<input checked="" type="checkbox"/>		

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

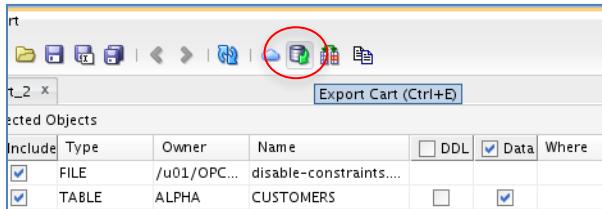
/u01/OPCWorkshop/lab/truncate-products.sql

- Click **OK**:



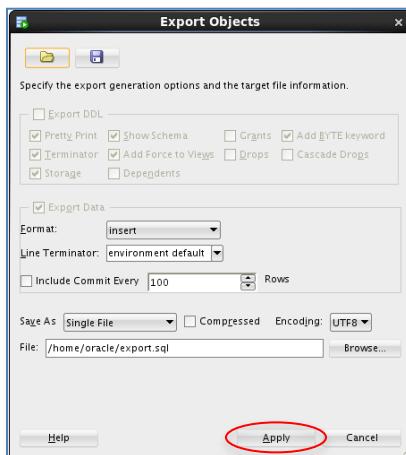
2.6.2: EXPORT THE SQL DEVELOPER CART

- 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.

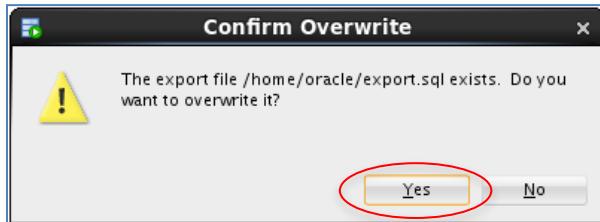


- Click the **Apply** button to generate script.

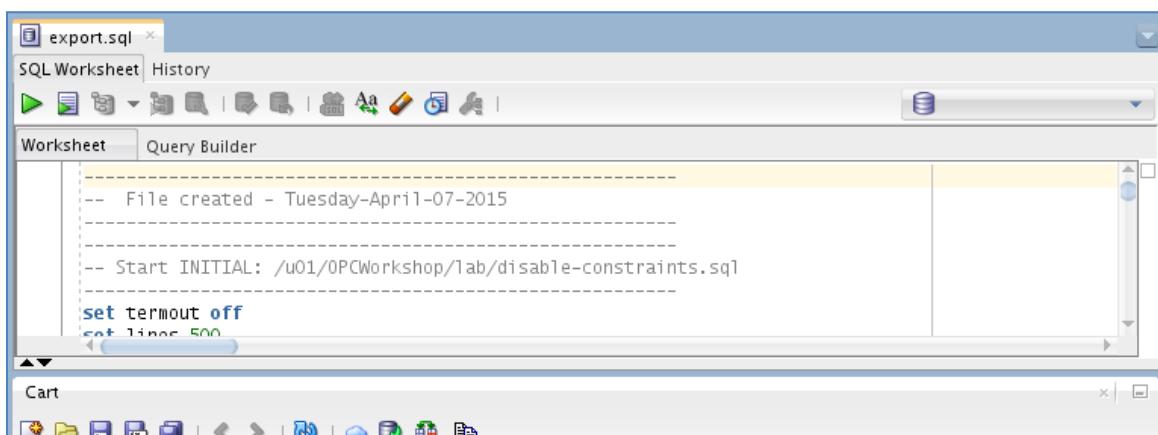
Note: The selections on this page may be saved and later reused if the cart is regularly used the same way.



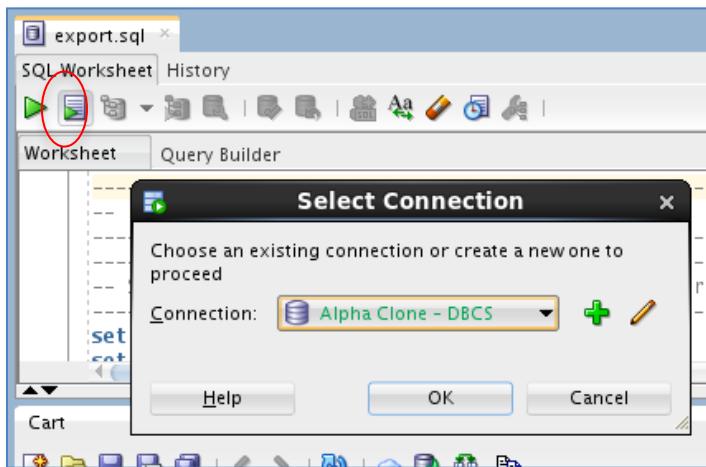
- If the file already exists, SQL Developer asks you to confirm overwriting it with new content. If you see this prompt, click the **Yes** button.



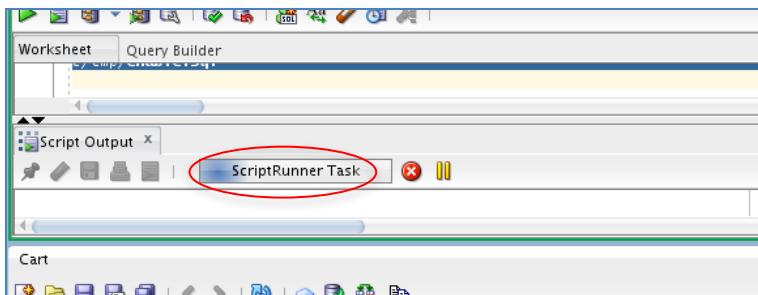
- Review the contents of the script with particular attention to the SQL statements that have been inserted based on the scripts we included.



- 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.

```
table PRODUCTS truncated.  
1 rows inserted.  
1 rows inserted.  
1 rows inserted.
```

A screenshot of the 'Script Output' window from Oracle SQL Developer. The window title is 'Script Output'. Inside, there is a message 'Task completed in 30.617 seconds'. Below that, the output of a task is shown: 'table PRODUCTS truncated.' followed by three lines of '1 rows inserted.'. The window has a standard Windows-style border with a title bar and a scroll bar on the right side.

- ✓ This concludes Lab 2 – Cloud Migration, proceed to the next lab when you're ready.

Section 3: Backup and Recovery

3.1: 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.

3.2: 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.

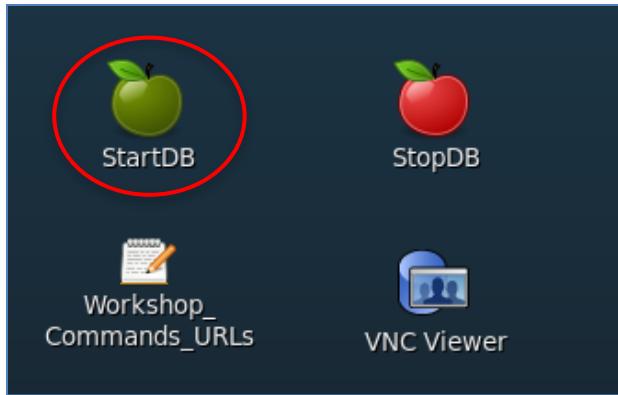
3.3: Lab Requirements

- ✓ VNC Viewer to access the client system

3.4: Oracle Public Cloud Backup Recovery

3.4.1: 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.



3.4.2: 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
```

```
oracle@oracelinux6:/u01/OPCWorkshop/lab
File Edit View Search Terminal Help
[oracle@oracelinux6 lab]$ pwd
/u01/OPCWorkshop/lab
[oracle@oracelinux6 lab]$ ls *.jar
opc_install.jar
[oracle@oracelinux6 lab]$
```

A terminal window titled "oracle@oracelinux6:/u01/OPCWorkshop/lab". The window shows the user's command history. The user typed "pwd" to show the current working directory, then "ls *.jar" to list files ending in ".jar". The file "opc_install.jar" is listed in red, indicating it is the file of interest.

- ✓ 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.



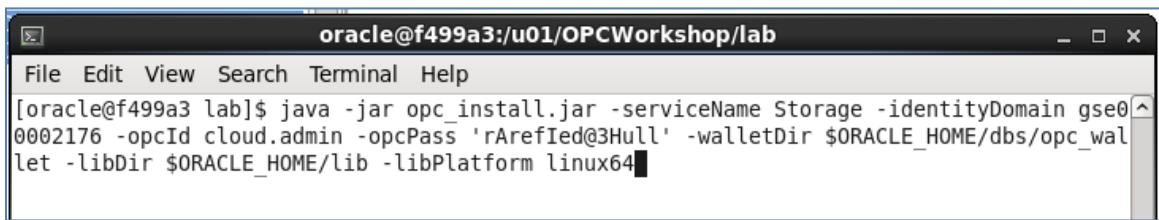
- Find the “**OPC Cloud Backup Installation**” section in the text file.
- Replace **<opc-identity-domain>** **<opc-username>** and **<opc-passwd>** (including replacing the <>) with the **Identity Domain, Username, and Password** student account information you were assigned. Also, be sure to put single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

```
OPC Cloud Backup Installation
-----
java -jar opc_install.jar -serviceName Storage -identityDomain <opc-identity-domain> -
opcId <opc-username> -opcPass '<opc-passwd>' -walletDir $ORACLE_HOME/dbs/opc_wallet -
libDir $ORACLE_HOME/lib -libPlatform linux64
```

- ✓ After

```
OPC Cloud Backup Installation
-----
java -jar opc_install.jar -serviceName Storage -identityDomain gse00002176 -
opcId cloud.admin -opcPass 'rArefied@3Hull' -walletDir $ORACLE_HOME/dbs/
opc_wallet -libDir $ORACLE_HOME/lib -libPlatform linux64
```

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



A screenshot of a terminal window titled "oracle@f499a3:/u01/OPCWorkshop/lab". The window shows the command "java -jar opc_install.jar -serviceName Storage -identityDomain gse00002176 -opcId cloud.admin -opcPass 'rArefied@3Hull' -walletDir \$ORACLE_HOME/dbs/opc_wallet -libDir \$ORACLE_HOME/lib -libPlatform linux64" entered at the prompt.

- ✓ The installation command creates a configuration file “**opcocl.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.

```
[oracle@oraclelinux6 lab]$ java -jar opc_install.jar -serviceName Storage -identityDomain jaxbansal -opcId mattew.orsie@oracle.com -opcPass WVbirder29_ -walletDir $ORACLE_HOME/dbs/opc_wallet -libDir $ORACLE_HOME/lib -libPlatform linux64
Oracle Database Cloud Backup Module Install Tool, build 2014-09-04
Oracle Database Cloud Backup Module credentials are valid.
Oracle Database Cloud Backup Module wallet created in directory /u01/app/oracle/product/12.1.0/dbhome_1/dbs/opc_wallet.
Oracle Database Cloud Backup Module initialization file /u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcorcl.ora created.
Downloading Oracle Database Cloud Backup Module Software Library from file opc_linux64.zip.
Downloaded 23169388 bytes in 16 seconds. Transfer rate was 1448086 bytes/second.
Download complete.
[oracle@oraclelinux6 lab]$
```

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

```
$ ls $ORACLE_HOME/lib/libopc*
$ ls $ORACLE_HOME/dbs/opc*
```

```
oracle@f499a3:/u01/OPCWorkshop/lab
File Edit View Search Terminal Help
[oracle@f499a3 lab]$ ls $ORACLE_HOME/lib/libopc*
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libopc12.so
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libopc.so
[oracle@f499a3 lab]$
[oracle@f499a3 lab]$ ls $ORACLE_HOME/dbs/opc*
/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcorcl.ora

/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opc_wallet:
cwallet.sso
[oracle@f499a3 lab]$
```

3.4.3: 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 /**

```
[oracle@oraclelinux6 lab]$ rman target /  
Recovery Manager: Release 12.1.0.2.0 - Production on Fri Apr 3 08:28:28 2015  
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.  
connected to target database: ORCL (DBID=1402968532)  
  
RMAN> █
```

- ✓ 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:

```
Workshop_Commands_URLs.txt █  
RMAN Config Params  
-----  
run {  
configure retention policy to recovery window of 30 days;  
configure channel device type 'sbt_tape' MAXPIECESIZE 2 G FORMAT 'alphacloud %d %U' PARMS 'SBT_LIBRARY=libopc.so, ENV=(OPC_PFILE=/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcorcl.ora)';  
configure encryption for database on;  
configure device type 'sbt_tape' parallelism 3 backup type to backupset;  
configure backup optimization on;  
configure compression algorithm 'basic' as of release 'default' optimize for load true;  
configure default device type to sbt_tape;  
}
```

```
RMAN> run {  
configure retention policy to recovery window of 30 days;  
configure channel device type 'sbt_tape' MAXPIECESIZE 2 G FORMAT 'alphacloud_%d_%U' PARMS 'SBT_LIBRARY=libopc.so, ENV=(OPC_PFILE=/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcorcl.ora)';  
configure encryption for database on;  
configure device type 'sbt_tape' parallelism 3 backup type to backupset;  
configure backup optimization on;  
configure compression algorithm 'basic' as of release 'default' optimize for load true;  
configure default device type to sbt_tape;  
}2> 3> 4> 5> 6> 7> 8> 9> █
```

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

```
new RMAN configuration parameters:  
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 30 DAYS;  
new RMAN configuration parameters are successfully stored  
  
new RMAN configuration parameters:  
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' MAXPIECESIZE 2 G FORMAT  'alphacloud_%d_%U' PARMs  'SBT_LIBRARY=libopc.  
.so, ENV=(OPC_PFILE=/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcorcl.ora)';  
new RMAN configuration parameters are successfully stored  
  
new RMAN configuration parameters:  
CONFIGURE ENCRYPTION FOR DATABASE ON;  
new RMAN configuration parameters are successfully stored  
  
new RMAN configuration parameters:  
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 3 BACKUP TYPE TO BACKUPSET;  
new RMAN configuration parameters are successfully stored  
  
new RMAN configuration parameters:  
CONFIGURE BACKUP OPTIMIZATION ON;  
new RMAN configuration parameters are successfully stored  
  
new RMAN configuration parameters:  
CONFIGURE COMPRESSION ALGORITHM 'basic' AS OF RELEASE 'default' OPTIMIZE FOR LOAD TRUE;  
new RMAN configuration parameters are successfully stored  
  
new RMAN configuration parameters:  
CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';  
new RMAN configuration parameters are successfully stored
```

- Verify the changes in RMAN by typing **show all**;

```
RMAN> show all;  
  
RMAN configuration parameters for database with db_unique_name ORCL are:  
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 30 DAYS;  
CONFIGURE BACKUP OPTIMIZATION ON;  
CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';  
CONFIGURE CONTROLFILE AUTOBACKUP ON; # default  
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F'; # default  
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default  
CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 3 BACKUP TYPE TO BACKUPSET;  
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default  
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT TAPE TO 1; # default  
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default  
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT TAPE TO 1; # default  
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default  
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' MAXPIECESIZE 2 G FORMAT  'alphacloud_%d_%U' PARMs  'SBT_LIBRARY=libopc.  
.so, ENV=(OPC_PFILE=/u01/app/oracle/product/12.1.0/dbhome_1/dbs/opcorcl.ora)';  
CONFIGURE MAXSETSIZE TO UNLIMITED; # default  
CONFIGURE ENCRYPTION FOR DATABASE ON;  
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default  
CONFIGURE COMPRESSION ALGORITHM 'basic' AS OF RELEASE 'default' OPTIMIZE FOR LOAD TRUE;  
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default  
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default  
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/snapcf_orcl.f'; # default  
  
RMAN> ■
```

3.4.4: 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.

```
Workshop_Commands_URLs.txt X

RMAN FLOW:

* optional (only used in recursive testing for delete/restore situations)
** (only needs to run the first time)
BACKUP
set encryption on identified by oracle only;
*drop restore point gold;
**backup as compressed backupset tag 'onprem' database plus archivelog;
create restore point gold preserve;
```

- RMAN> set encryption on identified by oracle only;

```
RMAN> set encryption on identified by oracle only;
```

```
executing command: SET encryption
```

```
RMAN> █
```

- ✓ 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.

```
Starting backup at 03-APR-15
current log archived
using channel ORA_SBT_TAPE_1
using channel ORA_SBT_TAPE_2
using channel ORA_SBT_TAPE_3
channel ORA_SBT_TAPE_1: starting compressed archived log backup set
channel ORA_SBT_TAPE_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=19 RECID=8 STAMP=876043544
channel ORA_SBT_TAPE_1: starting piece 1 at 03-APR-15
channel ORA_SBT_TAPE_1: finished piece 1 at 03-APR-15
piece handle=alphacloud_ORCL_0cq3en8o_1_1 tag=ONPREM comment=API Version 2.0,MMS Version 3.15.1.16
channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:07
Finished backup at 03-APR-15

Starting Control File and SPFILE Autobackup at 03-APR-15
piece handle=c-1402968532-20150403-00 comment=API Version 2.0,MMS Version 3.15.1.16
Finished Control File and SPFILE Autobackup at 03-APR-15

RMAN> ■
```

- ✓ 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;`

```
RMAN> list backup summary;

List of Backups
=====
Key    TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
-----+
2      B  A  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
3      B  A  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
4      B  A  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
5      B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
6      B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
7      B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
8      B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
9      B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
10     B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
11     B  F  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
12     B  A  A SBT_TAPE  03-APR-15      1      1     YES  ONPREM
13     B  F  A SBT_TAPE  03-APR-15      1      1     NO   TAG20150403T092552

RMAN> ■
```

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

```
RMAN> create restore point gold preserve;
```

```
Statement processed
```

```
RMAN> █
```

3.4.5: 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;
```

```
[oracle@oraclelinux6 OPCWorkshop]$ sqlplus alpha/oracle@alphapdb
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Apr 3 10:45:59 2015
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
Last Successful login time: Wed Apr 01 2015 17:34:44 -07:00
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

```
SQL> select* from mstars;
```

FIRST	LAST
Jimmy	Stewart
Katharine	Hepburn
Tom	Hanks
Anne	Hathaway

```
SQL> █
```

- Drop the table with the drop table command

```
SQL> drop table mstars;
```

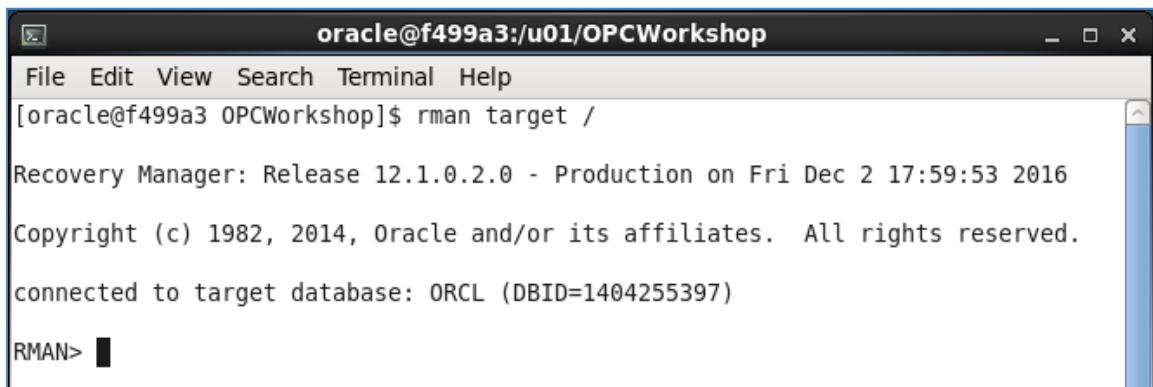
```
SQL> drop table mstars;  
Table dropped.  
SQL> ■
```

- Exit SQL*Plus
- SQL> exit;

3.4.6: 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:

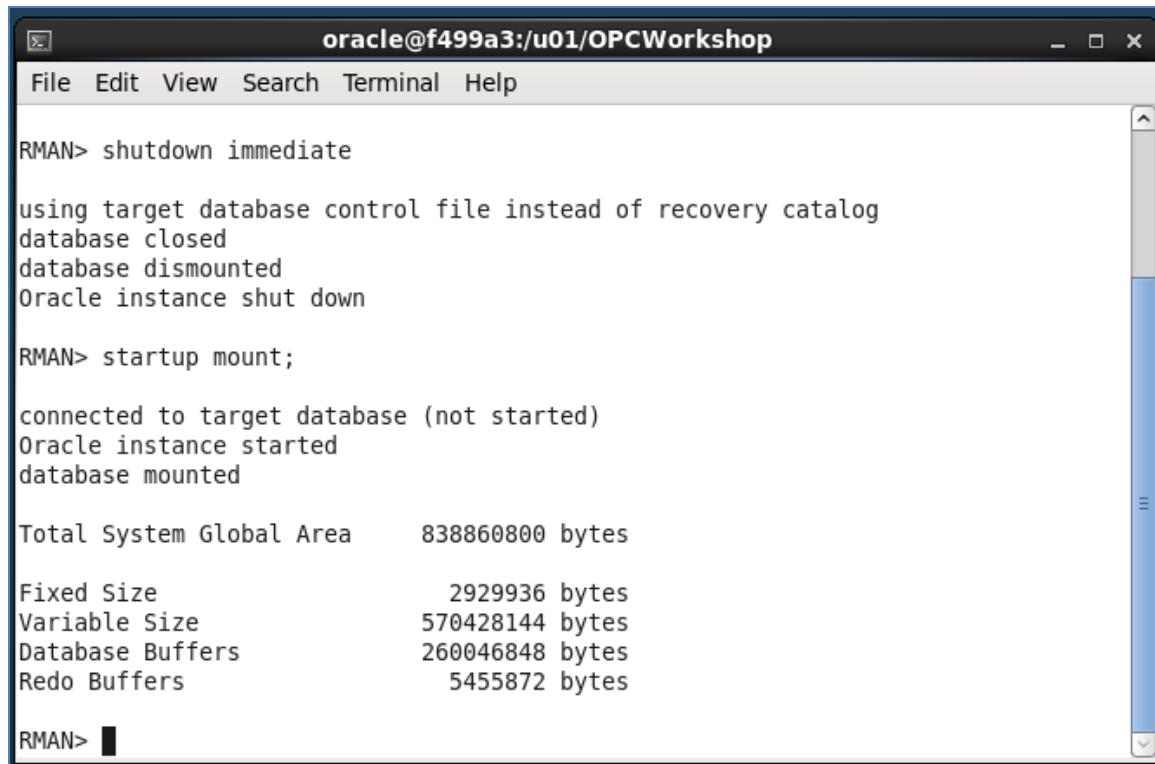
```
$ rman target /
```



```
oracle@f499a3:/u01/OPCWorkshop  
File Edit View Search Terminal Help  
[oracle@f499a3 OPCWorkshop]$ rman target /  
Recovery Manager: Release 12.1.0.2.0 - Production on Fri Dec 2 17:59:53 2016  
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.  
connected to target database: ORCL (DBID=1404255397)  
RMAN> ■
```

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

```
RMAN> shutdown immediate;  
RMAN> startup mount;
```



```
oracle@f499a3:/u01/OPCWorkshop
File Edit View Search Terminal Help

RMAN> shutdown immediate
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down

RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted

Total System Global Area     838860800 bytes
Fixed Size                  2929936 bytes
Variable Size                570428144 bytes
Database Buffers             260046848 bytes
Redo Buffers                 5455872 bytes

RMAN>
```

- ✓ 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;
```

```
RMAN> set decryption identified by oracle;
executing command: SET decryption
RMAN>
```

- 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;  
}
```

```
RMAN> run {  
    restore to restore point gold database;  
    recover database to restore point gold;  
    alter database open resetlogs;  
}2> 3> 4> 5> █
```

- Press Enter and the commands will be executed:

```
channel ORA_SBT_TAPE_2: piece handle=alphacloud_ORCL_0kq3f238_1_1 tag=ONPREM  
channel ORA_SBT_TAPE_2: restored backup piece 1  
channel ORA_SBT_TAPE_2: restore complete, elapsed time: 00:02:43  
channel ORA_SBT_TAPE_1: piece handle=alphacloud_ORCL_0oq3f2p9_1_1 tag=ONPREM  
channel ORA_SBT_TAPE_1: restored backup piece 1  
channel ORA_SBT_TAPE_1: restore complete, elapsed time: 00:00:55  
channel ORA_SBT_TAPE_3: piece handle=alphacloud_ORCL_0jq3f238_1_1 tag=ONPREM  
channel ORA_SBT_TAPE_3: restored backup piece 1  
channel ORA_SBT_TAPE_3: restore complete, elapsed time: 00:03:43  
Finished restore at 03-APR-15  
  
Starting recover at 03-APR-15  
using channel ORA_SBT_TAPE_1  
using channel ORA_SBT_TAPE_2  
using channel ORA_SBT_TAPE_3  
using channel ORA_DISK_1  
  
starting media recovery  
media recovery complete, elapsed time: 00:00:00  
  
Finished recover at 03-APR-15  
  
Statement processed  
  
RMAN> █
```

- 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;
```

```
[oracle@oraclelinux6 OPCWorkshop]$ sqlplus alpha/oracle@alphapdb
SQL*Plus: Release 12.1.0.2.0 Production on Fri Apr 3 12:55:59 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Apr 03 2015 12:24:47 -07:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> select * from mstars;

FIRST          LAST
-----          -----
Jimmy          Stewart
Katharine     Hepburn
Tom            Hanks
Anne           Hathaway

SQL> ■
```

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;

Section 4: Database Development

4.1: 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.

4.2: Objectives

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

4.3: 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.

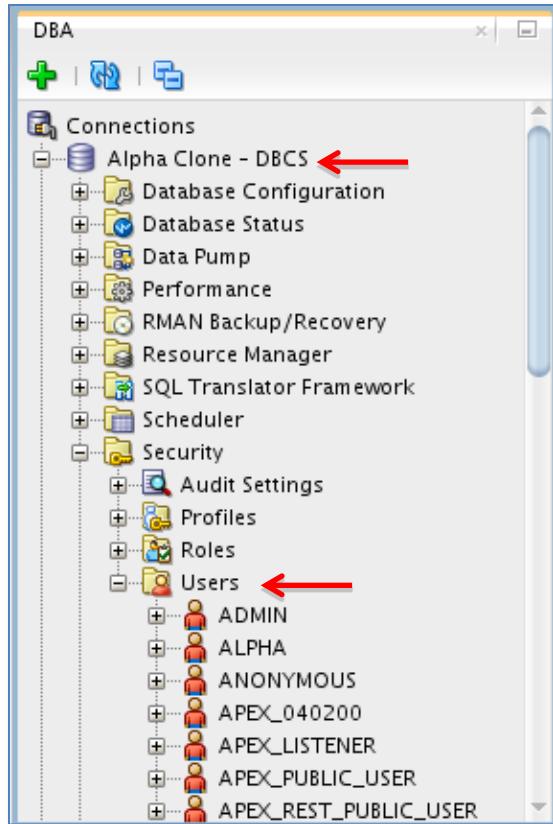
4.4: Alpha Office and APEX

4.4.1: APEX WORKSPACE ADMINISTRATION

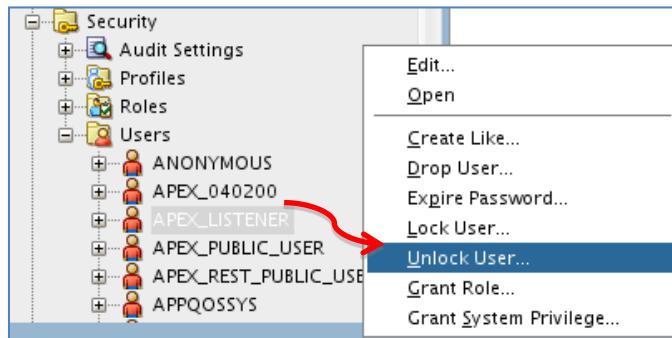
The Alpha Clone database contains an unused APEX configuration. As the first part of this lab we will complete the configuration of the cloned database APEX configuration.

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.*

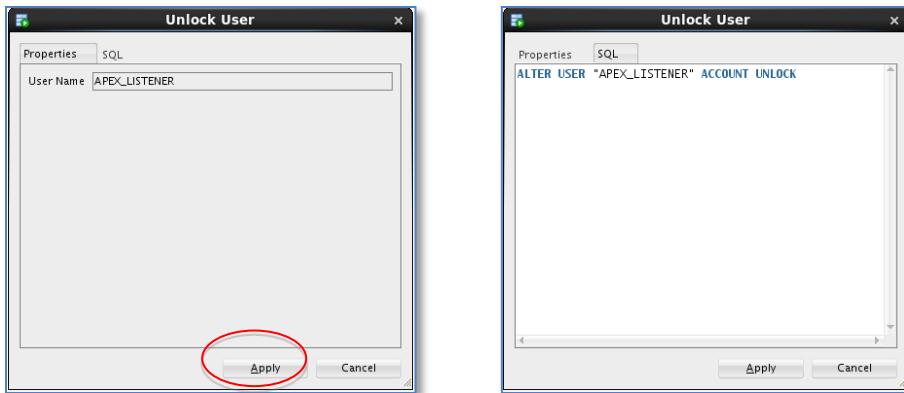
- ✓ Make sure the SSH tunnels you set up in lab 100 are still active in your terminal window, if not refer to lab 100 to set up the SSH tunnels.
- ✓ During the plug-in operation, many of the common objects in the pluggable database were evaluated by the database and some changes were made to the new 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.



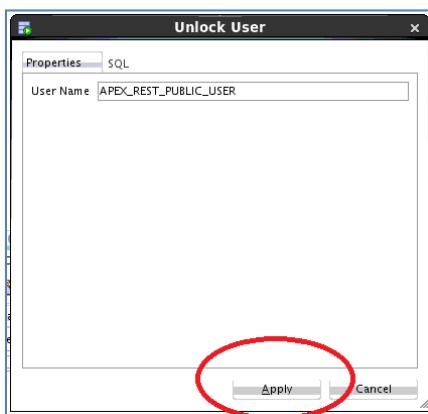
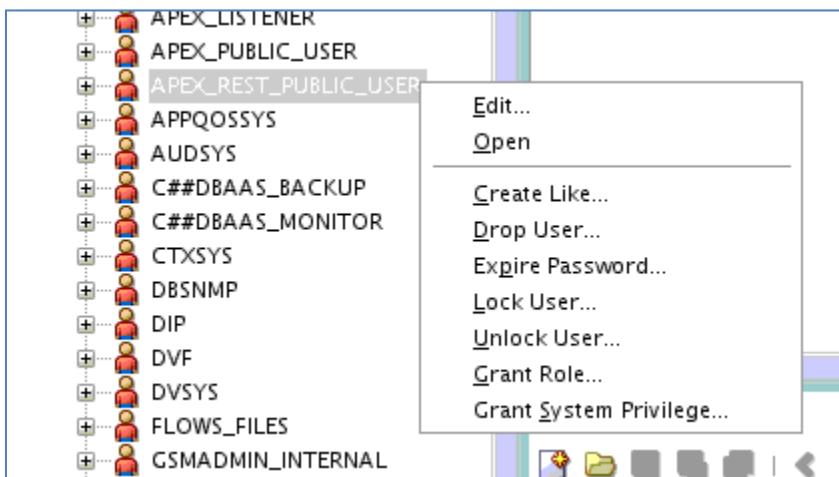
- Right-mouse on **APEX_LISTENER** and select **Unlock User...**



- 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**.

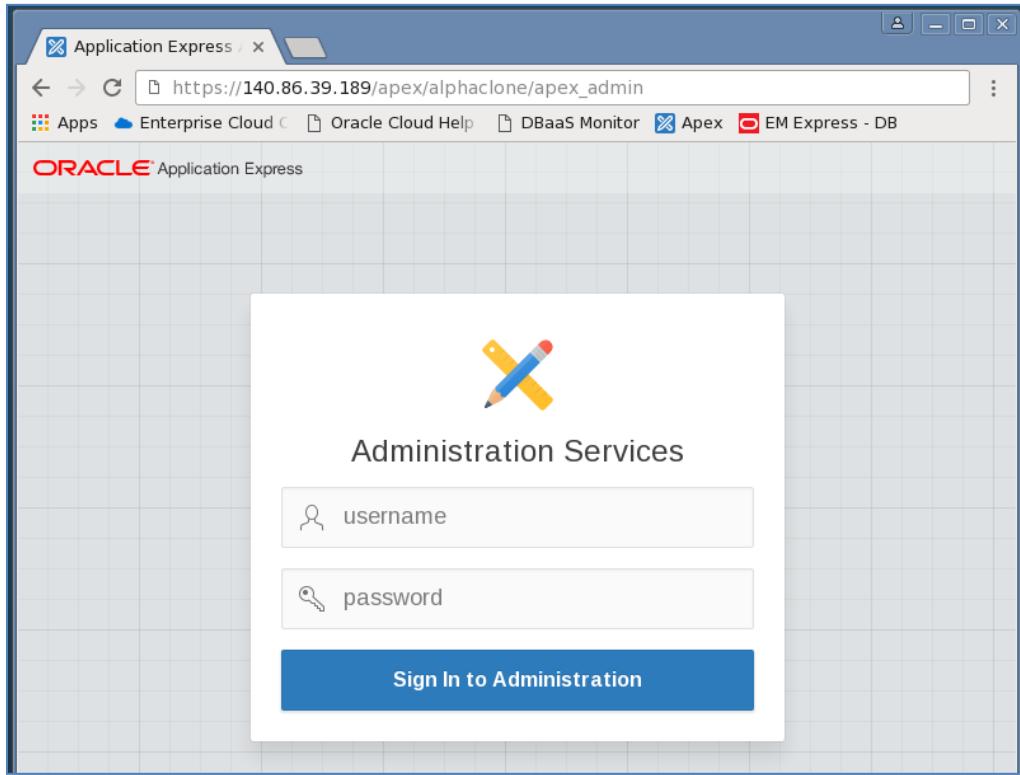


4.4.2: CREATE THE ALPHA OFFICE WORKSPACE

- ✓ 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.

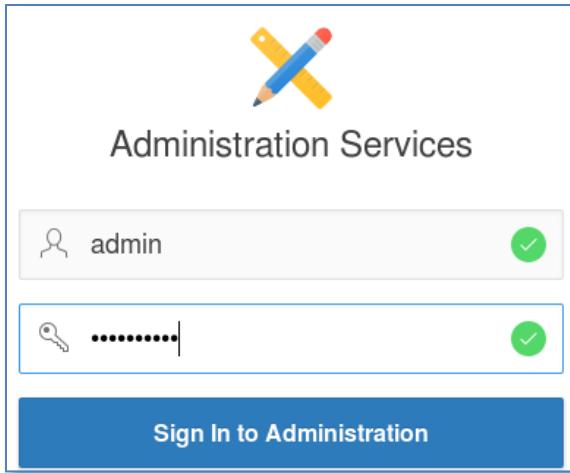
Note: Be sure to use the https protocol.

`https://<your-Public-IP>/apex/alphaclone/apex_admin`



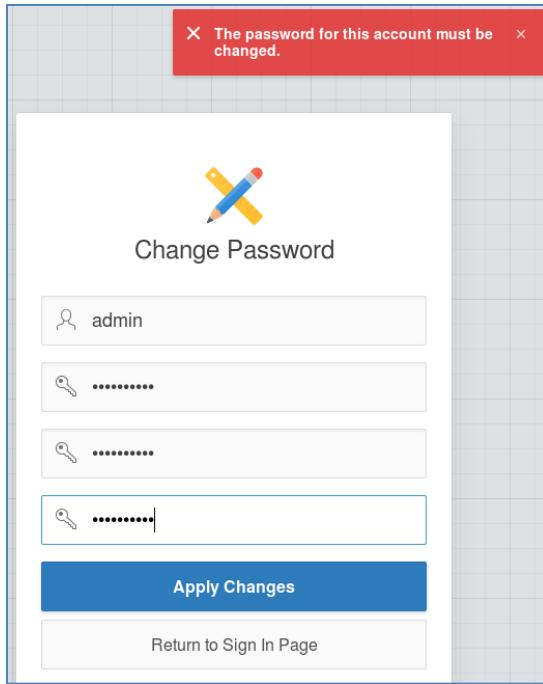
- After you've accepted the SSL certificate and see the APEX administration page, enter the following admin credentials and click the **Login to Administration** button:

Username:	admin
Password:	Alpha2014_

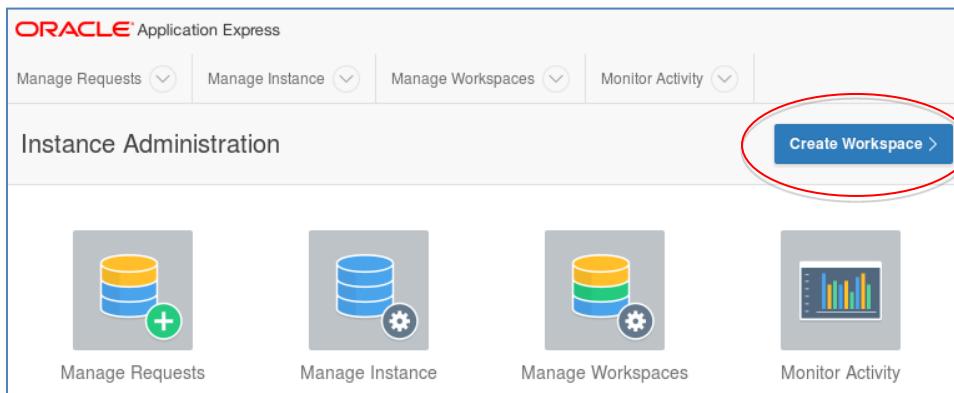


- You **may** be prompted to change the ADMIN user password, if not, skip to the next step. These credentials apply to the APEX objects local to the pluggable database. For convenience, we will enter the same password as the container database.
- Enter the following values and click the **Apply Changes**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

Identify Workspace

* Workspace Name ?

Workspace ID ?

Workspace Description ?

Tasks
Create Multiple Workspaces

Cancel Next

- 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

Identify Schema

Select whether or not the schema already exists. If the schema exists, select the schema from the list. If the schema does not exist, enter a name and password and choose the size of the associated tablespace to be created.

Re-use existing schema? ?

* Schema Name ?

* Schema Password ?

* Space Quota (MB) ?

< Cancel Next >

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

* Administrator Username: ADMIN

* Administrator Password: ?

First Name:

Last Name:

* Email: dummy@localhost.localdomain ?

< Cancel Next >

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

You have requested to provision a new Workspace.

Workspace Information:

Name	AlphaDev
Workspace ID	System Assigned
Description	--

Administrator Information:

User Name	ADMIN
E-mail	dummy@localhost.localdomain

Schema Information:

Reuse Existing Schema	Yes
Schema Name	ALPHA

< Cancel Create Workspace

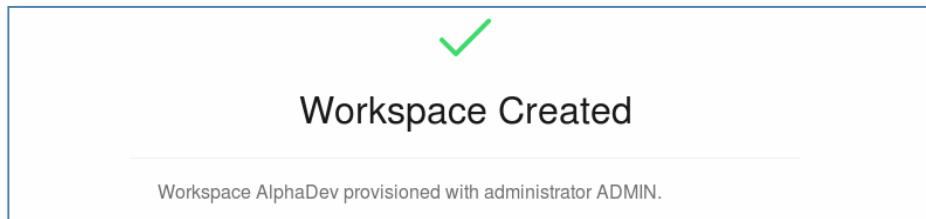
You have requested to provision a new Workspace.

Workspace Information:

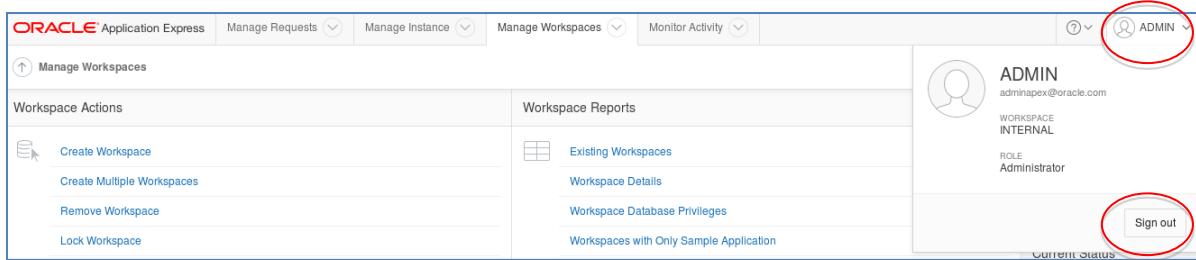
Name	Alpha Office Development
------	--------------------------

< Cancel Create Workspace

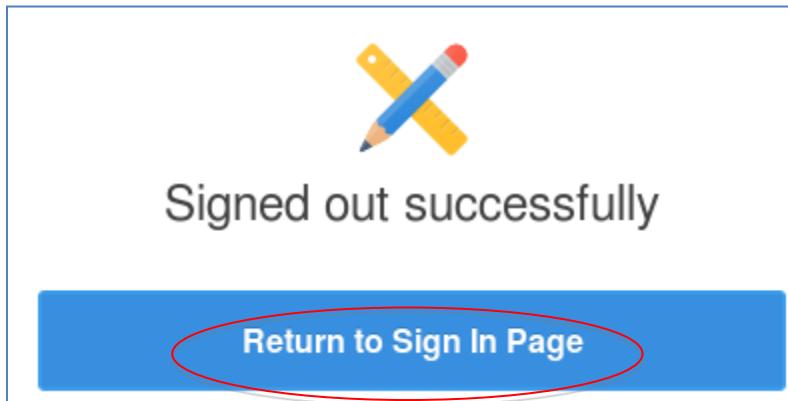
- APEX will display the ‘Workspace Created’ message
- Click **Done**



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



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

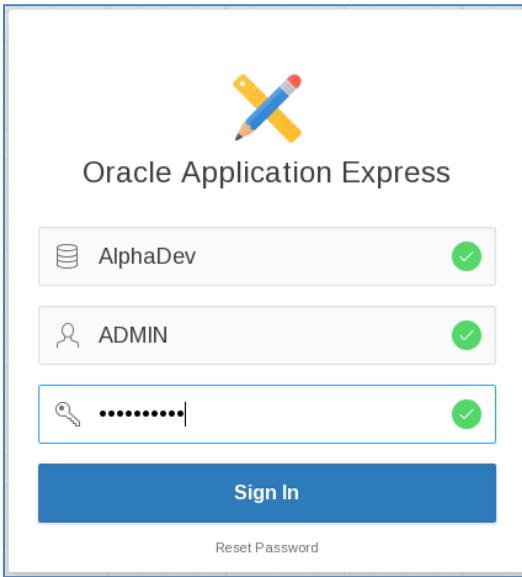


4.4.3: BUILD REST SERVICES

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

Workspace:	AlphaDev
Username:	ADMIN

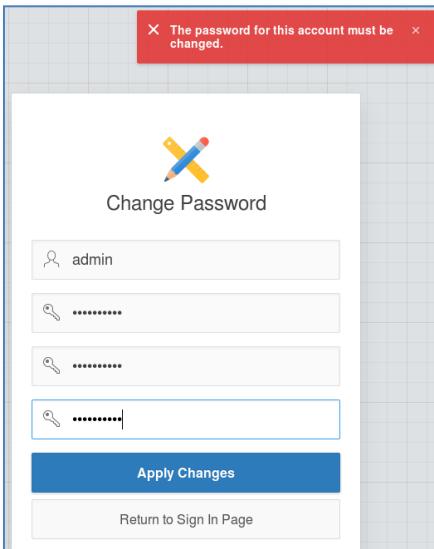
Password:	Alpha2014_
-----------	------------



The image shows the Oracle Application Express sign-in page. It features a logo of two crossed pencils at the top. Below the logo, the text "Oracle Application Express" is displayed. There are three input fields: the first contains "AlphaDev" with a green checkmark; the second contains "ADMIN" with a green checkmark; the third contains a masked password "*****" with a green checkmark. A large blue "Sign In" button is centered below the fields. At the bottom left, there is a "Reset Password" link.

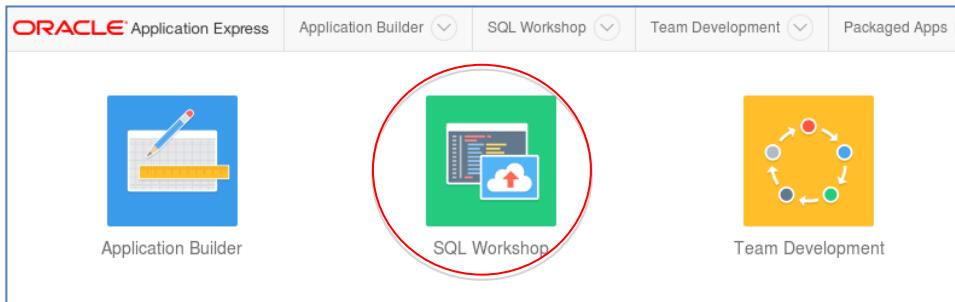
- You **may** be prompted to change your password. Enter the following values and click the **Apply Changes** button.

Enter Current Password	Alpha2014_
Enter New Password	Alpah2015!
Confirm New Password	Alpha2015!

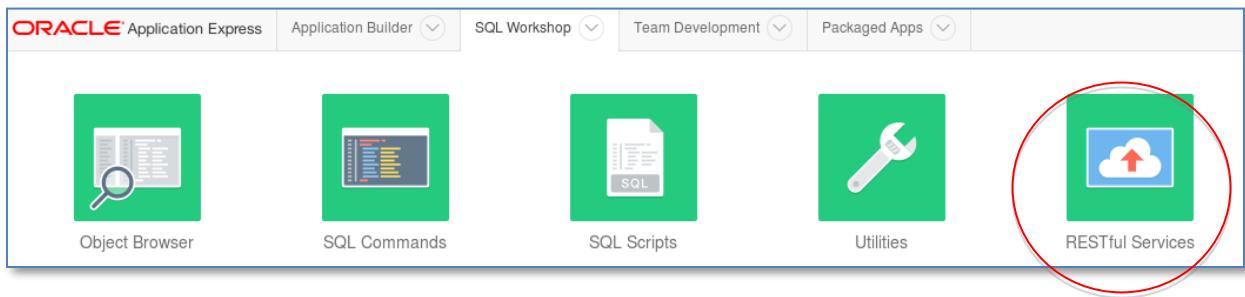


The image shows the "Change Password" page. At the top, a red banner displays the message "The password for this account must be changed." Below the banner, the text "Change Password" is centered. There are four input fields: the first contains "admin"; the second contains a masked password "*****"; the third contains a masked password "*****"; and the fourth contains a masked password "*****". Below these fields is a blue "Apply Changes" button. At the bottom, there is a link "Return to Sign In Page".

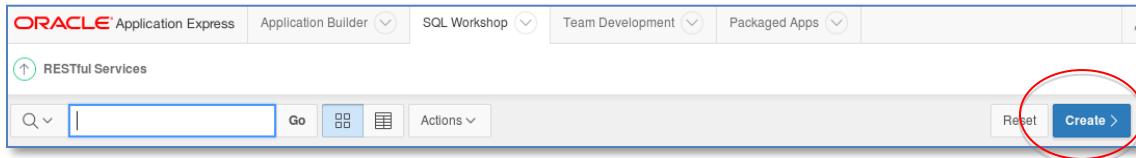
- Once you've logged in successfully, click the **SQL Workshop** button.



- 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

The screenshot shows a user interface for creating a RESTful service module. The top navigation bar includes a back arrow, the text 'RESTful Services', and 'Create RESTful Service Module'. On the right are 'Cancel' and 'Create Module' buttons. The interface is divided into three main sections, each highlighted with a red border:

- RESTful Services Module**: Contains fields for Name (required), URI Prefix, Origins Allowed, Status (set to Published), Pagination Size (set to 25), and Required Privilege (set to - Assign Privilege -).
- Resource Template**: Contains fields for Add a Resource Template, URI Template (required), Priority (set to 0), and Entity Tag (set to Secure HASH).
- Resource Handler**: Contains fields for Add a Resource Handler and Method (set to - Select Method -).

✓ 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/

RESTful Services Module

* Name	alpha.office
URI Prefix	alphaofc
Origins Allowed	
Status	Published
* Pagination Size	25
Required Privilege	- Assign Privilege -

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

URI Template	products/
--------------	-----------

Resource Template

Add a Resource Template	
* URI Template	products/
Priority	0
Entity Tag	Secure HASH

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

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

Resource Handler

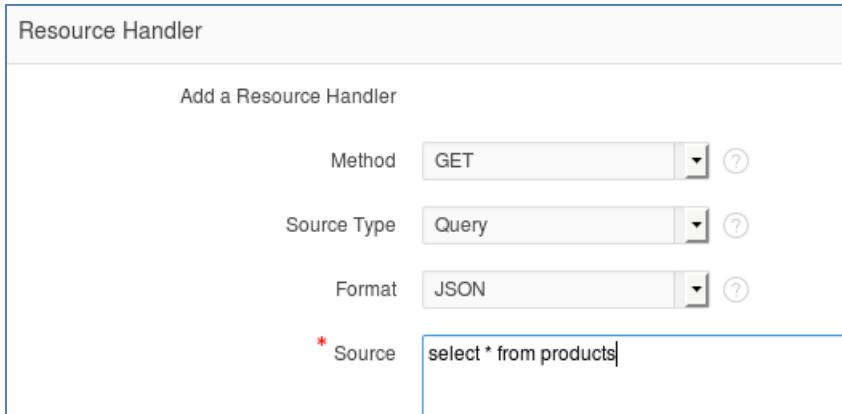
Add a Resource Handler

Method: GET

Source Type: Query

Format: JSON

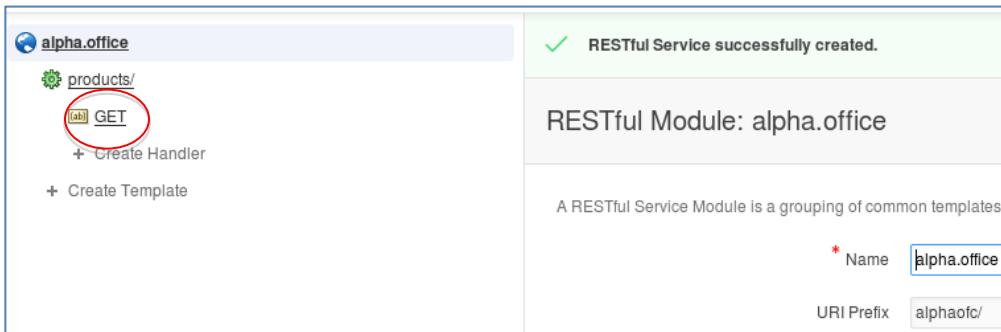
* Source: select * from products



- 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.



alpha.office

products/

(ab) GET

+ Create Handler

+ Create Template

✓ RESTful Service successfully created.

RESTful Module: alpha.office

A RESTful Service Module is a grouping of common templates

* Name: alpha.office

URI Prefix: alphaofc/

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

Resource Handler: GET

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: alphaofc/ (i)

URI Template: products/ (i)

Method: GET (i)

Source Type: Query (i)

Format: JSON (i)

Requires Secure Access: Yes (i)

Pagination Size: (i)

Source

* Source (i)

```
1 select * from products
```

Example

Test (circled in red) Set Bind Variables >

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

https://localhost/ords/pdb1/alphadev/alphaofc/products/

```
{"next": {"$ref": "https://localhost/ords/pdb1/alphadev/alphaofc/products/?page=1"}, "items": [{"product_id": 1039, "parent_category_id": 1002, "category_id": 1011, "product_name": "Crayola Original Markers - Broad Line, Classic Colors", "product_status": "AVAILABLE", "cost_price": 3.19, "list_price": 4.25, "min_price": 2.99, "warranty_period_months": 6, "external_url": "Images/OfficeSupplyProducts/Write/Write-Crayola.Markers.jpg", "created_by": "0", "creation_date": "2014-01-10T13:04:21Z", "last_updated_by": "0", "last_update_date": "2014-01-10T13:04:21Z", "object_version_id": 1}, {"product_id": 1040, "parent_category_id": 1002, "category_id": 1011, "product_name": "Expo Low Odor Chisel Tip Dry Erase Markers, 4 Colored Markers", "product_status": "AVAILABLE", "cost_price": 4.99, "list_price": 6.99, "min_price": 4.29, "warranty_period_months": 6, "external_url": "Images/OfficeSupplyProducts/Write/Write-Expo.Dry.Erase.jpg", "created_by": "0", "creation_date": "2014-01-10T13:04:21Z", "last_updated_by": "0", "last_update_date": "2014-01-10T13:04:21Z", "object_version_id": 1}, {"product_id": 1041, "parent_category_id": 1003, "category_id": 1013, "product_name": "Scotch Magic Tape Deluxe Dispenser Value Pack", "product_status": "AVAILABLE", "cost_price": 17.99, "list_price": 22.99, "min_price": 15.99, "warranty_period_months": 6, "external_url": "Images/OfficeSupplyProducts/Fasten/Fasten-Scotch.Tape.and.Dispenser.jpg", "created_by": "0", "creation_date": "2014-01-10T13:04:21Z", "last_updated_by": "0", "last_update_date": "2014-01-10T13:04:21Z", "object_version_id": 1}, {"product_id": 1042, "parent_category_id": 1003, "category_id": 1013, "product_name": "Scotch Gift Wrap Tape in Handheld Dispenser 3/4\" x 600\"", "product_status": "AVAILABLE", "cost_price": 6.29, "list_price": 8.99, "min_price": 4.99, "warranty_period_months": 6, "external_url": "Images/OfficeSupplyProducts/Fasten/Fasten-Scotch.Gift.Tape.jpg", "created_by": "0", "creation_date": "2014-01-10T13:04:21Z", "last_updated_by": "0", "last_update_date": "2014-01-10T13:04:21Z", "object_version_id": 1}, {"product_id": 1043, "parent_category_id": 1003, "category_id": 1013, "product_name": "Scotch Chameleon Tape Dispenser with Scotch Magic Tape", "product_status": "AVAILABLE", "cost_price": 9.25, "list_price": 13.99, "min_price": 6.99, "warranty_period_months": 6, "external_url": "Images/OfficeSupplyProducts/Fasten/Fasten-Scotch.Chameleon.jpg", "created_by": "0", "creation_date": "2014-01-10T13:04:21Z", "last_updated_by": "0", "last_update_date": "2014-01-10T13:04:21Z", "object_version_id": 1}
```

4.4.4: 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.

The screenshot shows the Oracle Application Express interface for creating a RESTful service. The path is RESTful Services > RESTful Service Module > Resources. Under the 'products/' resource, there is a 'GET' method entry. At the bottom of the list, there is a link labeled 'Create Template', which is circled in red.

- 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}
---------------	--------------

This is a screenshot of a modal dialog titled 'URI Template'. It contains fields for 'RESTful Service Module' (set to 'alpha.office'), 'URI Template' (set to 'product/{id}'), 'Priority' (set to '0'), and 'Entity Tag' (set to 'Secure HASH'). The 'Create' button at the top right is circled in red, and the 'URI Template' input field is also circled in red.

- 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.

Action Processed.

URI Template: product/{id}

RESTful Service Module: alpha.office

* URI Template product/{id}

- 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.

Resource Handler:

RESTful Service Module: alpha.ofc/ (i)

URI Template: product/{id} (i)

Method: GET (i)

Source Type: Query (i)

Format: JSON (i)

Requires Secure Access: Yes (i)

Pagination Size: (i)

Source

* Source (i)

```
1 select *
2   from products
3   where product_id = :id
```

- ✓ 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.

The screenshot shows a 'Test' section with a 'Test' button and a 'Set Bind Variables' button. Below the buttons is a text area containing instructions about testing the RESTful Service Handler.

To test the behavior of the RESTful Service Handler, click the Test button. If the RESTful Service Handler uses parameters, click the Set Bind Variables button to set test values for the parameters. Before testing, ensure that you have saved all changes to this page by clicking Apply Changes. For better results, ensure you have a JSON Viewer installed in your browser.

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

:ID	1020
-----	------

The screenshot shows a 'Source' configuration section with a query: 'select * from products where product_id = :id'. Below it is a 'Bind Variable' table with a row for ':ID' where the 'Value' is '1020'.

Bind Variable	Value
:ID	1020

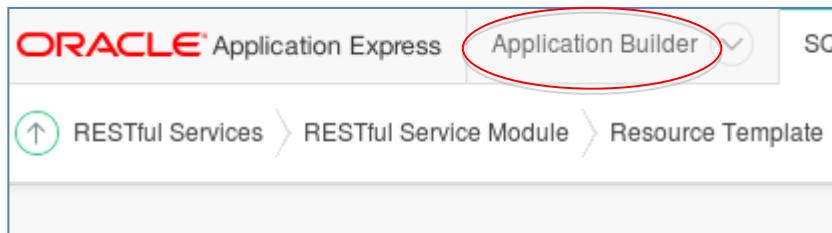
- ✓ In the new browser window, notice only the single product shows in the JSON object.
- **Close** this pop-up window.

The screenshot shows a browser window displaying a JSON object for product ID 1020. The JSON includes details like product name, price, and external URL.

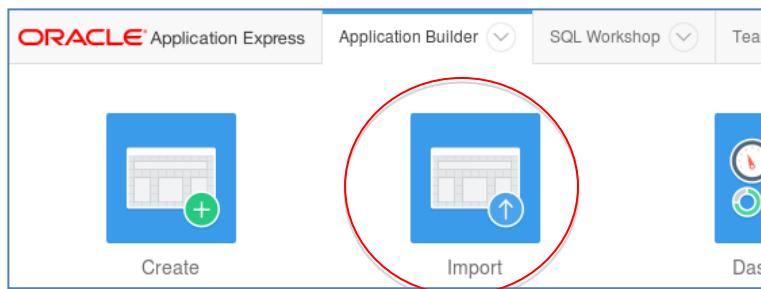
```
{"next":{"$ref":"https://localhost/ords/pdb1/alphadev/alphaofc/product/1020?page=1"},"items":[{"product_id":1020,"parent_category_id":1003,"category_id":1012,"product_name":"Office Depot® Brand Brass Fasteners, 1 1/2\" Length, Pack Of 60","product_status":"AVAILABLE","cost_price":2.49,"list_price":3.99,"min_price":1.52,"warranty_period_months":6,"external_url":"Images/OfficeSupplyProducts/Fasten/Fasten-RoundHead-60.jpg","created_by":"0","creation_date":"2014-01-10T13:04:21Z","last_updated_by":"0","last_update_date":"2014-01-10T13:04:21Z","object_version_id":1}]}  
https://localhost/ords/pdb1/alphadev/alphaofc/product/1020 - Google Chrome
```

4.4.5: 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.

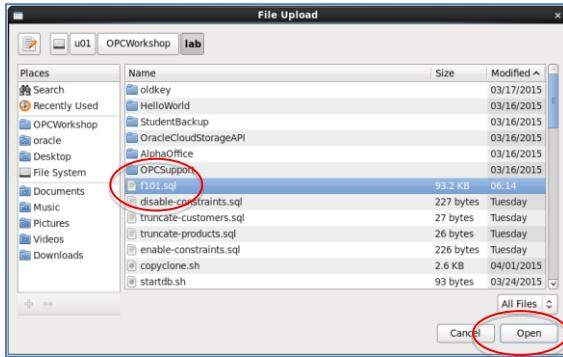
The dialog box title is "Import".
Text: "Select the file you wish to import to the export repository. Once imported, you can install your file.
If the imported file is a packaged application export, the installation wizard will allow you to run the packaged installation scripts after installing the application definition."
Form fields:

- * Import file: f107.sql
- * File Type: Database Application, Page or Component Export
 Websheet Application Export
 Plug-in
 Theme Export
 User Interface Defaults
 Team Development Feedback
 CSS Export [Deprecated]
 Image Export [Deprecated]
 File Export [Deprecated]
- File Character Set:

-
-

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

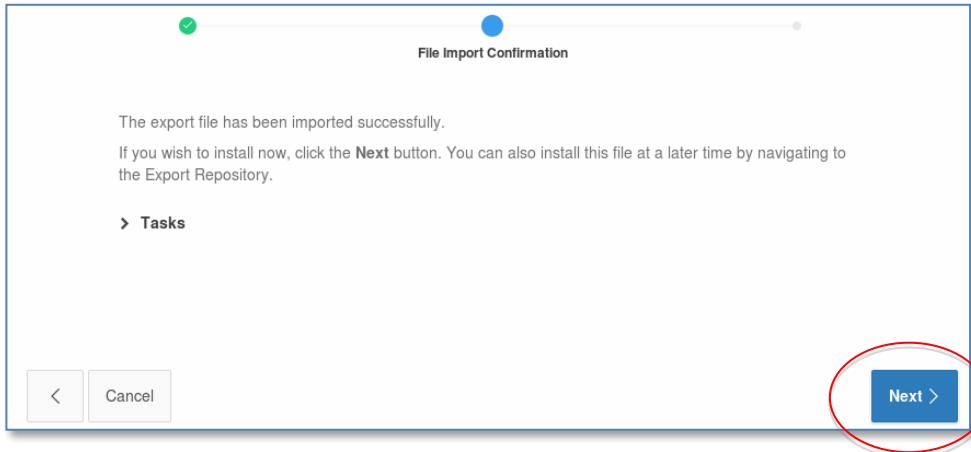
/u01/OPCWorkshop/lab/f101.sql



- 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**

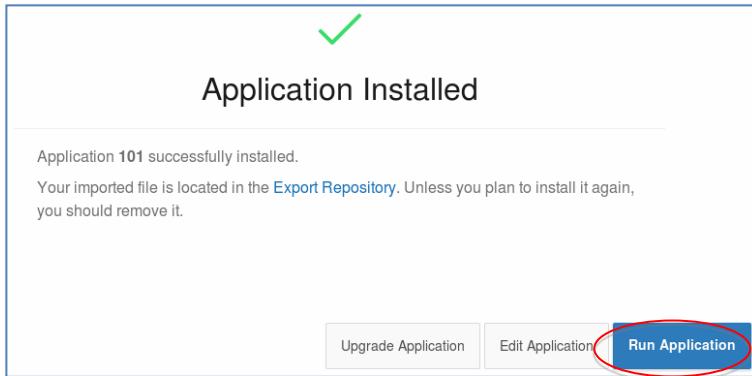
- Click **Install Application**.

The screenshot shows the 'Install Application' step of the APEX wizard. At the top, there are three green circular progress indicators followed by a blue 'Install' button. Below the indicators is a descriptive text block. Underneath are several configuration fields:

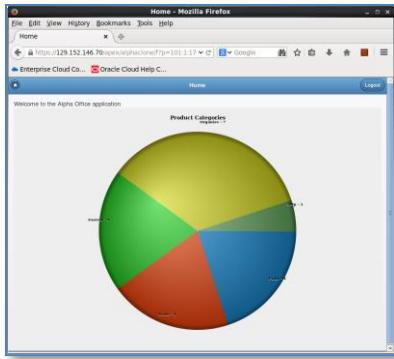
- Current Workspace: ALPHADEV
- Export File Workspace ID: 2222523363265404
- Export File Application ID: 101
- Export File Version: 2012.01.01
- Export File Parsing Schema: ALPHA
- Application Origin: This application was exported from another workspace.
- Parsing Schema: ALPHA
- Build Status: Run and Build Application
- Install As Application:
 - Auto Assign New Application ID
 - Reuse Application ID 101 From Export File
 - Change Application ID

A red oval highlights the 'Install As Application' section, specifically the radio button for 'Reuse Application ID 101 From Export File'. At the bottom left are 'Tasks' and navigation buttons ('<', 'Cancel'). At the bottom right is a large blue 'Install Application' button, which is also circled in red.

- APEX displays a success message for the import
- Click **Run Application**.



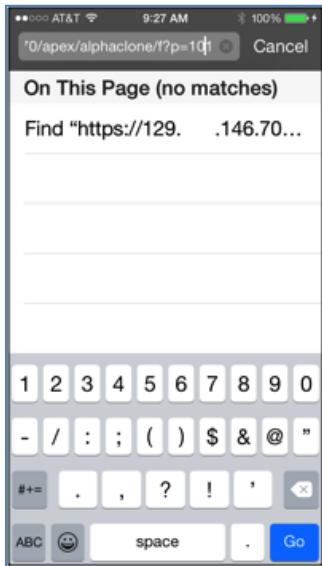
- ✓ APEX renders the first page of the mobile application in the browser – it might not look quite right since we are using a mobile template.



4.4.6: ACCESS THE ALPHA OFFICE MOBILE APPLICATION ON YOUR SMART DEVICE

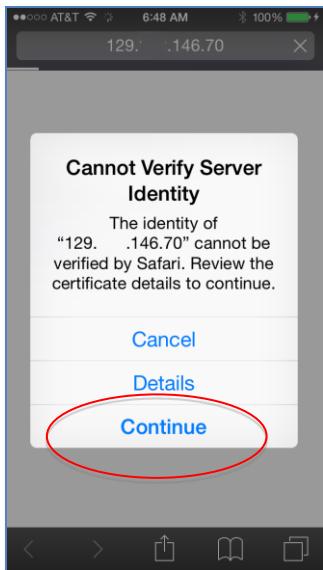
- ✓ Using any Internet connected smart phone or tablet we will access the mobile application using the port we opened earlier in the lab. This example is using an Apple iPhone 5s.
- Use your device's browser and navigate to the following URL:

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

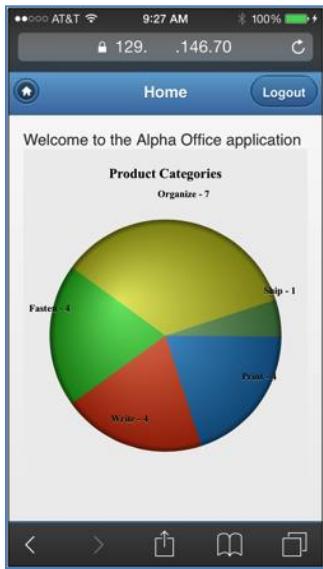


Note: It may be more convenient to email the link to your device.

- The browser should prompt you to accept the unknown certificate.
- Click or 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.