



*North American Technology Division
Solution Engineering Team*

200 Oracle Public Cloud Workshop

Database Cloud Service Workshop

Update Dec 4, 2017

Introduction

In this lab, you will explore some common use cases for moving your data from on-premises 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 your on-premises database (your Virtual Machine) to your cloud database. You will also use standard Oracle Data Pump tools to export a schema from the on-premises database, and then import that data to your cloud database in a new schema. The final exercise uses RMAN to move a tablespace to the cloud. These are only a few of the many options for moving data into and between Oracle databases. More choices for how to migrate Oracle Databases from on-premise to the cloud can be found in this [Migration Strategies Document](#).

This lab supports the following use cases:

- Migration of on-premise pluggable databases to a cloud based environment.
- Rapid creation of test or development pluggable database copies in the Cloud.

- Rapid replication of on-premise data for reporting and analytics.
- Rapid re-fresh of selected on-premise schemas for test and/or development activities and/or reporting and analytics.
- To log issues and view the Lab Guide source, go to the [github oracle](#) repository.

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.

Required Artifacts

- The following labs assume that the steps outlined in lab guide 100 have been completed.
- The SSH tunnels must be active in a terminal window.

Cloud Migration Using Pluggable Databases

Configure the Environment

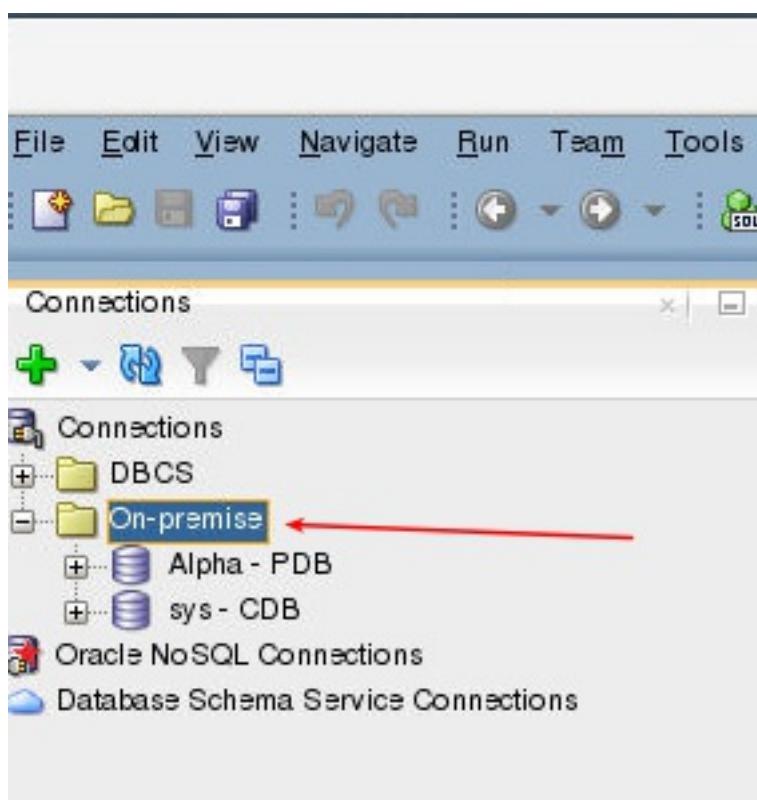
STEP 1: Open a **DBA Navigator** connection to the on-premise database

- From the VNC Session desktop, locate and double-click on the **SQL Developer** icon. **NOTE:** The first time SQL Developer is brought up; it

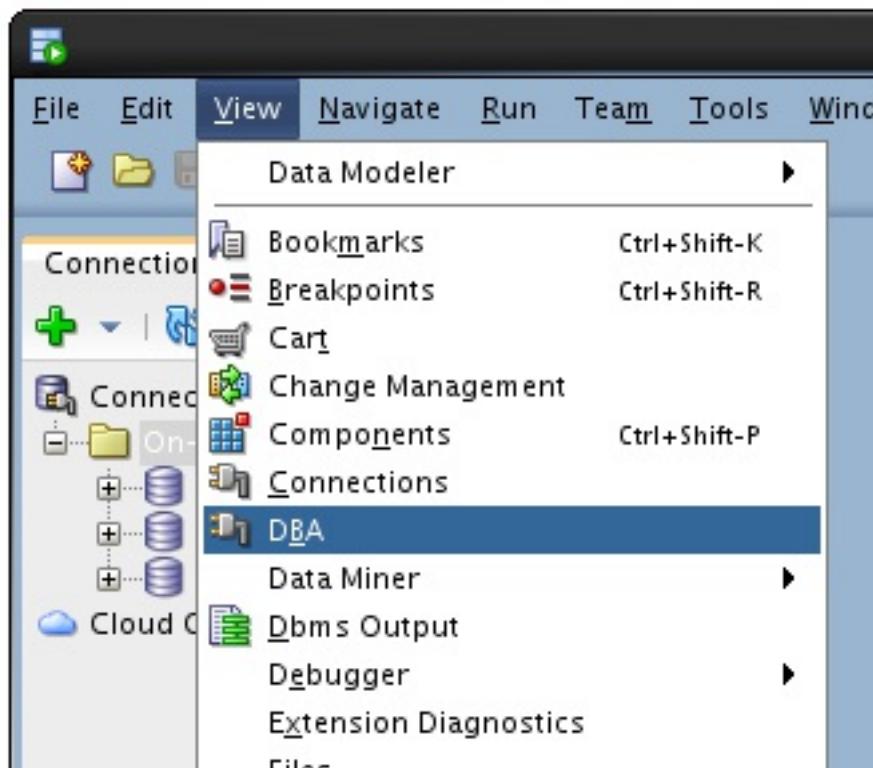
may take some time to instantiate.



- Double-click the **On-premises** folder to expand the list of database connections. Please note the pre-configured connections to the on-premises database.

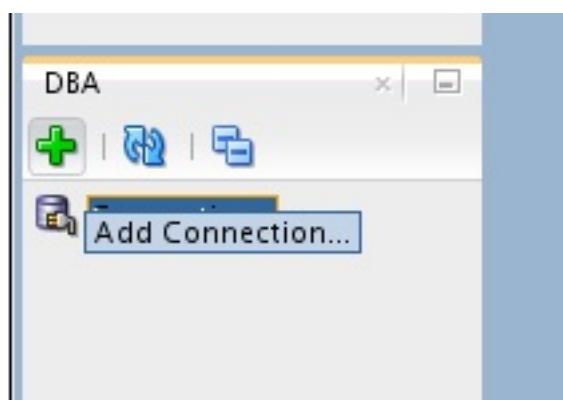


- Select the **View -> DBA** menu option from the top dropdown menu.



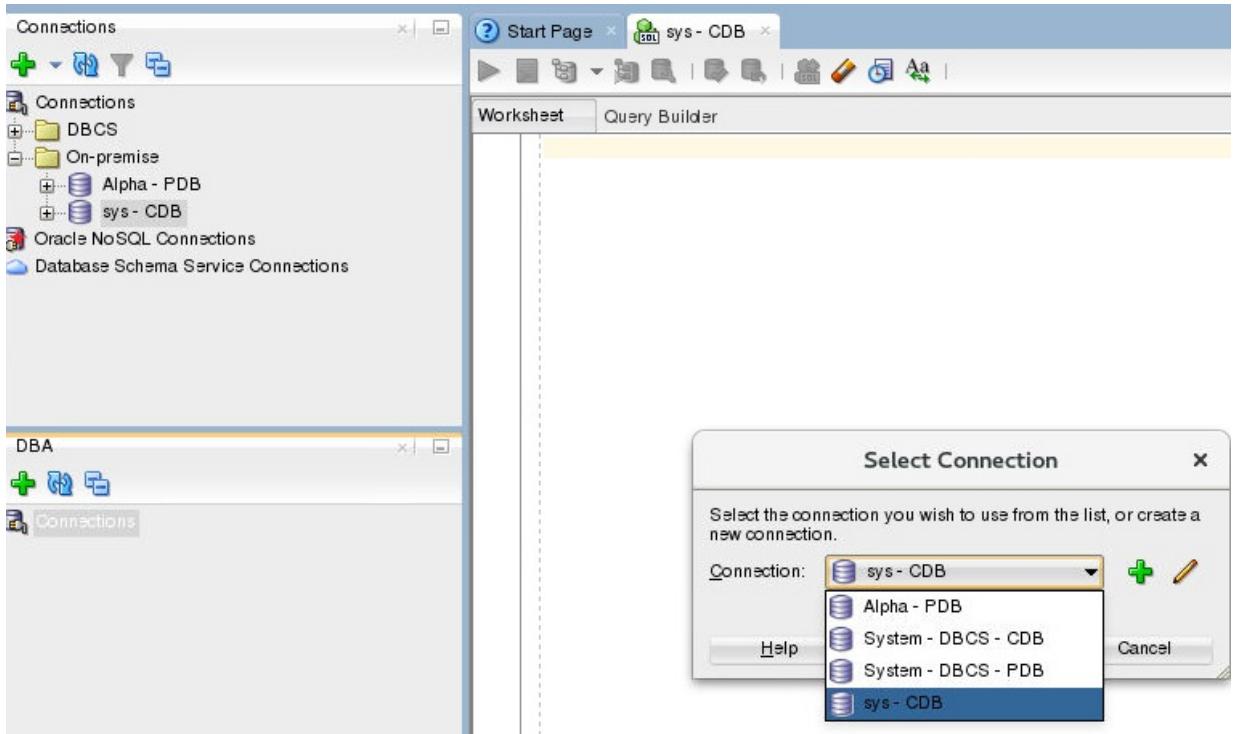
- On the DBA tab, click the green plus icon to create a new connection.

Note: you may also right-click on Connections and select Add Connection.

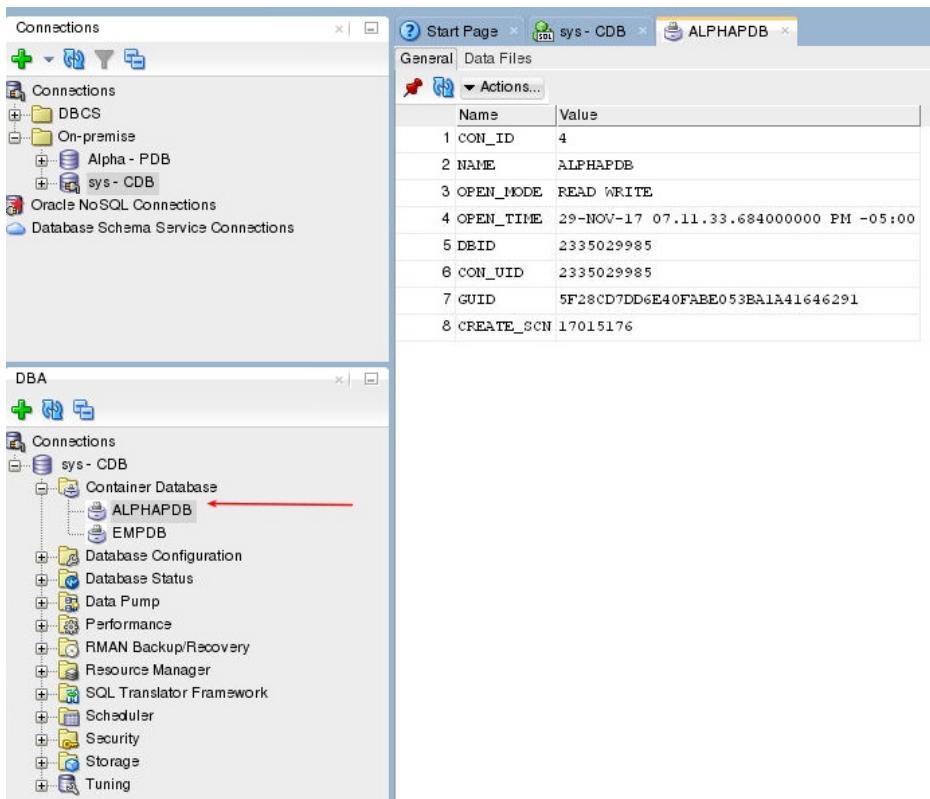


- Select the **sys - CDB** connection and click **OK**.

Note: **sys - CDB** is an “on-premises” database located on the Virtual Box Image.



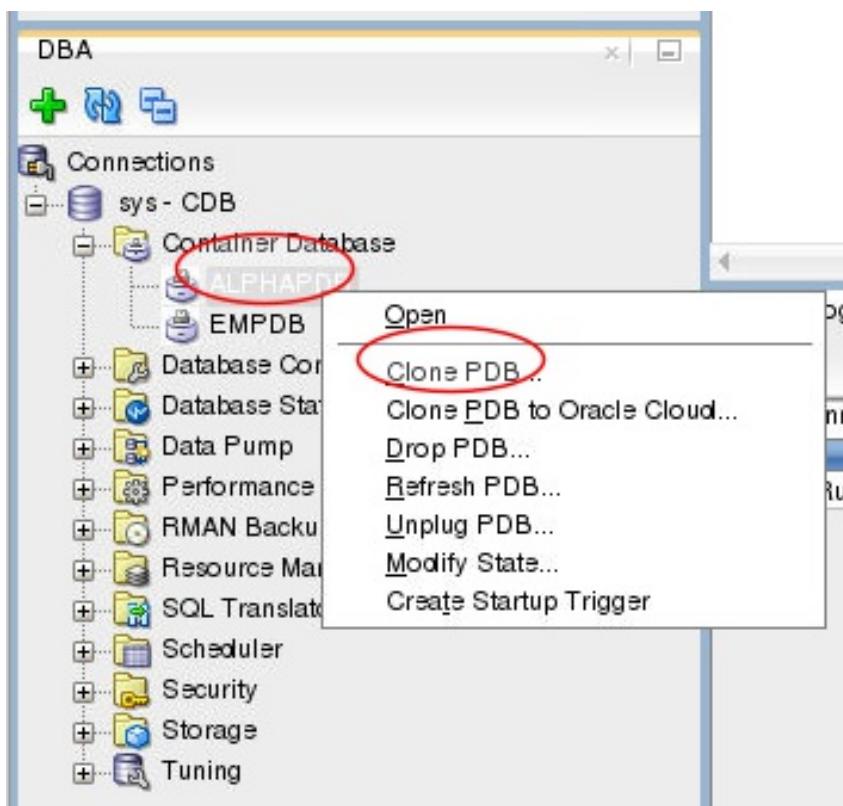
- Expand the **sys - CDB connect**, and then expand the **Container Database** tree item. Click on the **ALPHAPDB** pluggable database to show the details for the pluggable database.



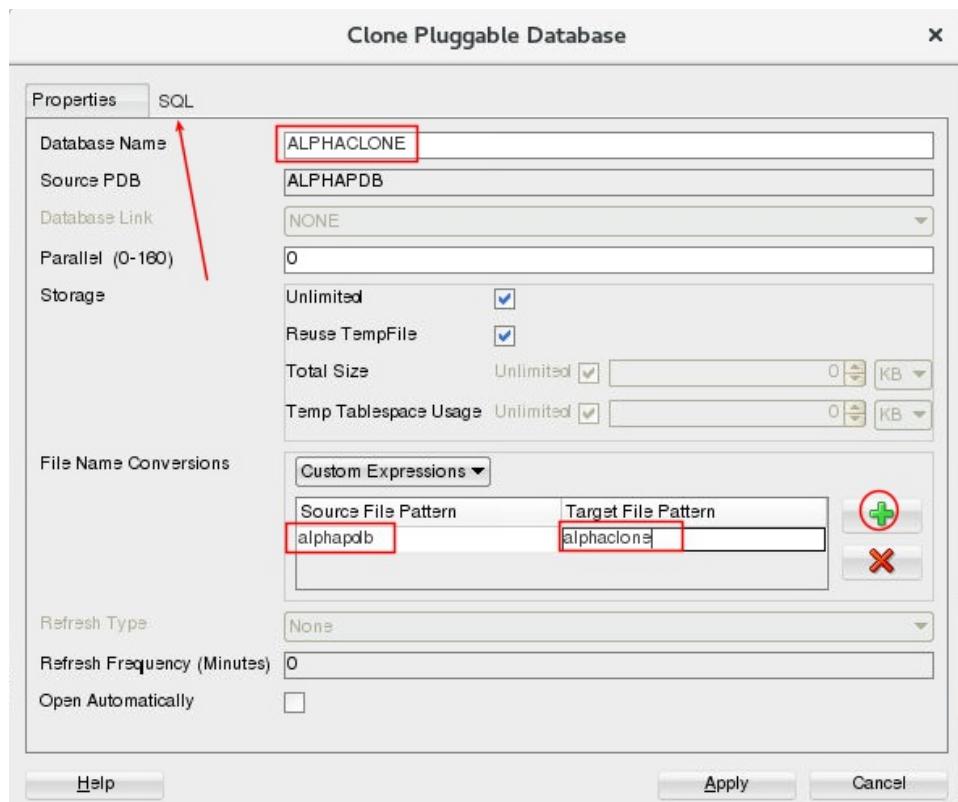
Clone the On-premise ALPHAPDB

STEP 2: Clone the ALPHAPDB

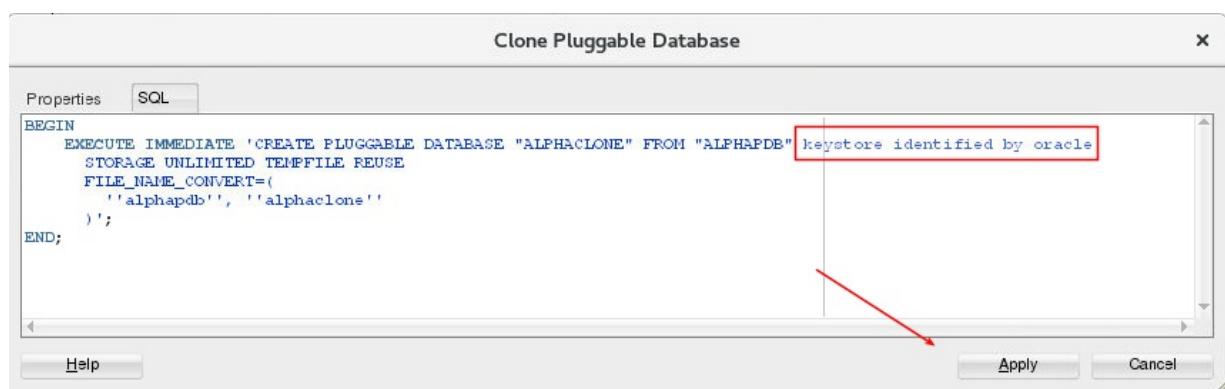
- In the DBA Navigator panel, right click on the ALPHAPDB pluggable database and select the Clone PDB... menu option.



- Enter the following
 - Database Name:** **ALPHACLONE**
 - File Name Conversions:** **Customer Expressions** (note by default the directory name is derived from the database name - we are overriding this)
 - Click the + on the right**
 - Source File Pattern:** **alphapdb**
 - Target File Pattern:** **alphaclone**
 - Then select the SQL table above



- Since we have implemented TDE (Transparent Data Encryption) we need to override the SQL. Select the SQL tab above and enter the following
 - **keystore identified by oracle**



- You should see the new PDB

The screenshot shows the Oracle Database Control interface. On the left, the DBA navigator displays a tree structure with 'Container Database' expanded, showing 'ALPHACLONE', 'ALPHAPDB', and 'EMPDB'. The 'ALPHACLONE' node is highlighted with a red box. On the right, the 'General' tab of the properties window for 'ALPHAPDB' is open, showing various database parameters. A message dialog box titled 'Successful' in the center says 'Successfully processed SQL command' with an 'OK' button. Below it, the 'Task Progress' and 'Open Connection (Failed)' sections are visible.

Name	Value
1 CON_ID	4
2 NAME	ALPHAPDB
3 OPEN_MODE	READ WRITE
4 OPEN_TIME	05-DEC-17 08.40.11.375000000 AM -05:00
5 DBID	2335029985
6 CON_UID	2335029985
7 GUID	5F28CD7DD6E40FABE053BA1A41646291
8 CREATE_SCN	17015176

- 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.

The screenshot shows the Oracle Database Control interface. The DBA navigator on the left shows the 'Container Database' section with 'ALPHACLONE' highlighted by a red circle. The properties window on the right is for 'ALPHACLONE', showing its configuration. The 'OPEN_MODE' parameter is circled in red in the table below.

Name	Value
1 CON_ID	6
2 NAME	ALPHACLONE
3 OPEN_MODE	MOUNTED
4 OPEN_TIME	05-DEC-17 10.08.56.783000000 AM -05:00
5 DBID	2655304485
6 CON_UID	2655304485
7 GUID	5F98EE36D1B1218CE053BA1A4164B77C
8 CREATE_SCN	20084730

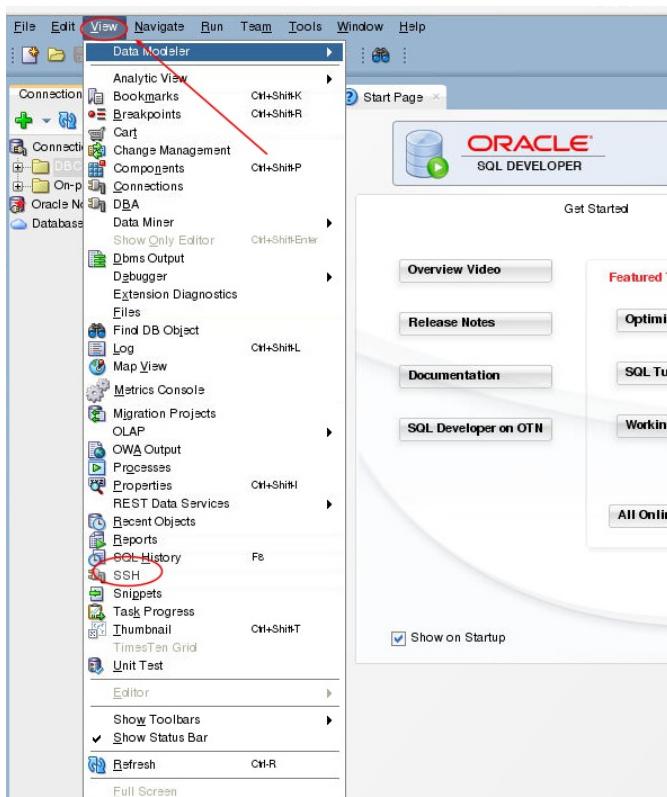
- Click on the Data Files tab for the ALPHACLONE to review the data files created during the cloning operation.

NAME	STATUS	ENABLED	BYTES	BLOCK_SIZE
1 /u01/app/oracle/oradata/orcl/alphaclone/sysaux01.dbf	ONLINE	READ WRITE	256901120	8192
2 /u01/app/oracle/oradata/orcl/alphaclone/system01.dbf	SYSTEM	READ WRITE	220200960	8192
3 /u01/app/oracle/oradata/orcl/alphaclone/temp01.dbf	ONLINE	READ WRITE	20971520	8192
4 /u01/app/oracle/oradata/orcl/alphaclone/undotbs01.dbf	ONLINE	READ WRITE	251658240	8192
5 /u01/app/oracle/oradata/orcl/alphaclone/users.dbf	ONLINE	READ WRITE	16973824	8192

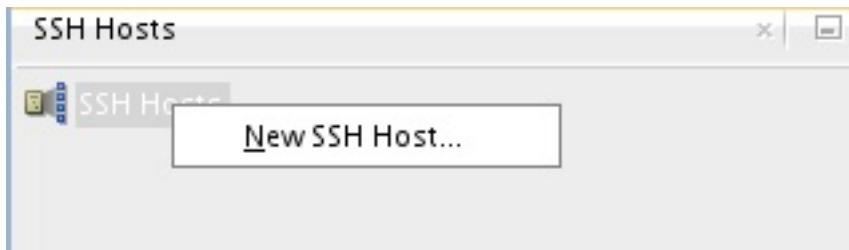
Clone the ALPHACLONE DB to the Cloud

STEP 3: Create SSH and SYS Database Cloud Connections

- First we need to setup a SSH host connection to the Database Cloud Service instance. 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**.

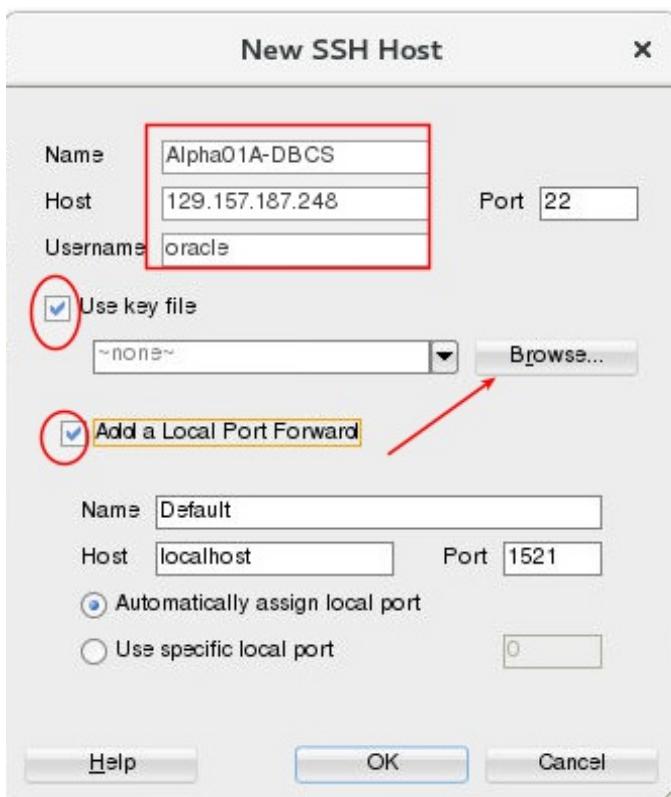


- We will now configure an SSH connection to our DBCS instance

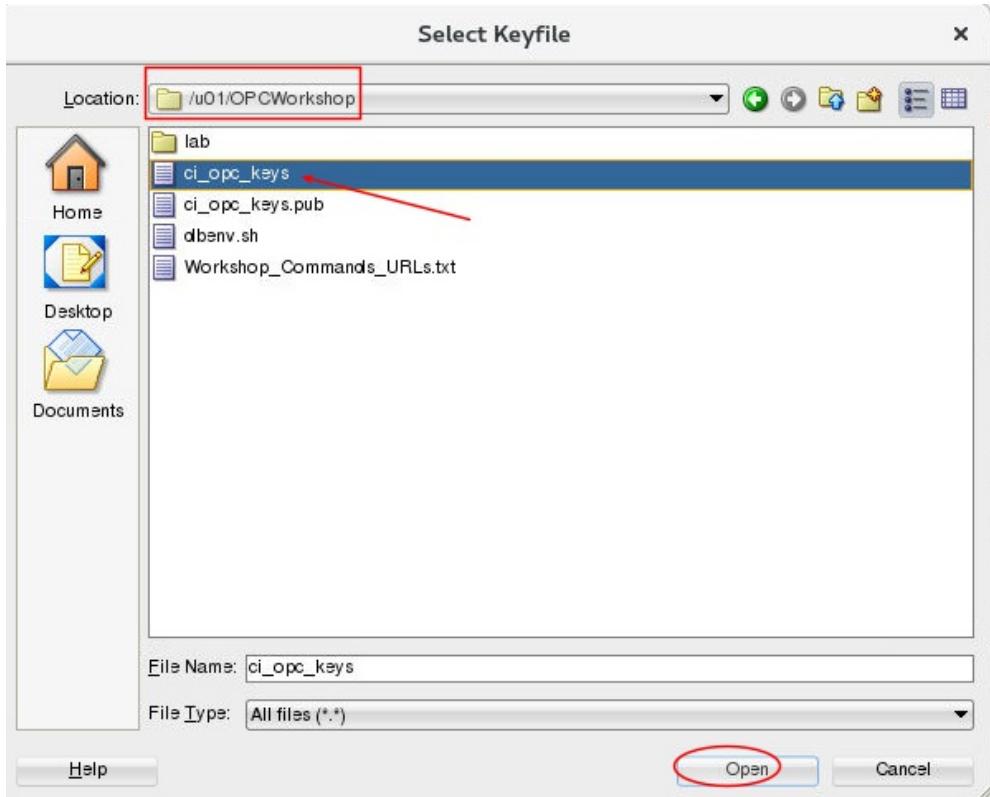
Name: Alpha01A-DBCS

Host: <Database Image public IP you obtained in lab 100>

Username: oracle



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



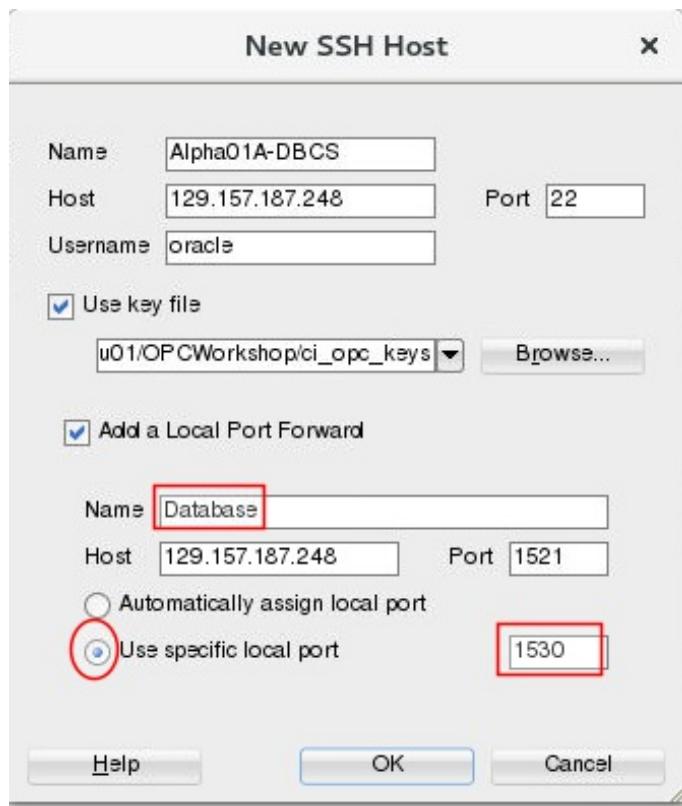
- Click **Add a Local Port Forward** and enter the following values:

Name: Database

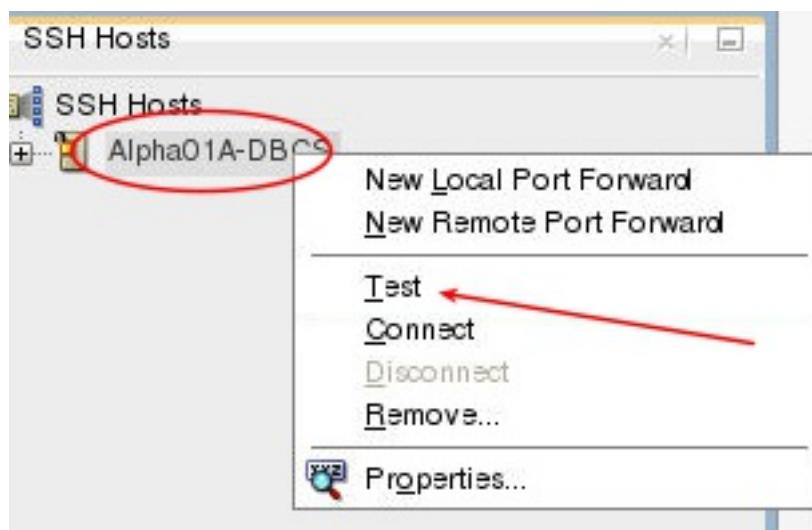
Host: <Database Image Public IP you obtained in lab 100>

- Select **Use specific local port** and enter **1530**

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



- Verify the configuration and click **OK**
- Right click on the SSH connection and test. You should see a message saying the connection was successful.





STEP 4: Create a SQL Developer connection to the Public Cloud database SYS schema

- Click the green plus sign in the **Connections** window to create a new connection; enter the following connection details:

Connection Name: sys - OPCDBCS

Username: sys

Password: Alpha2018_

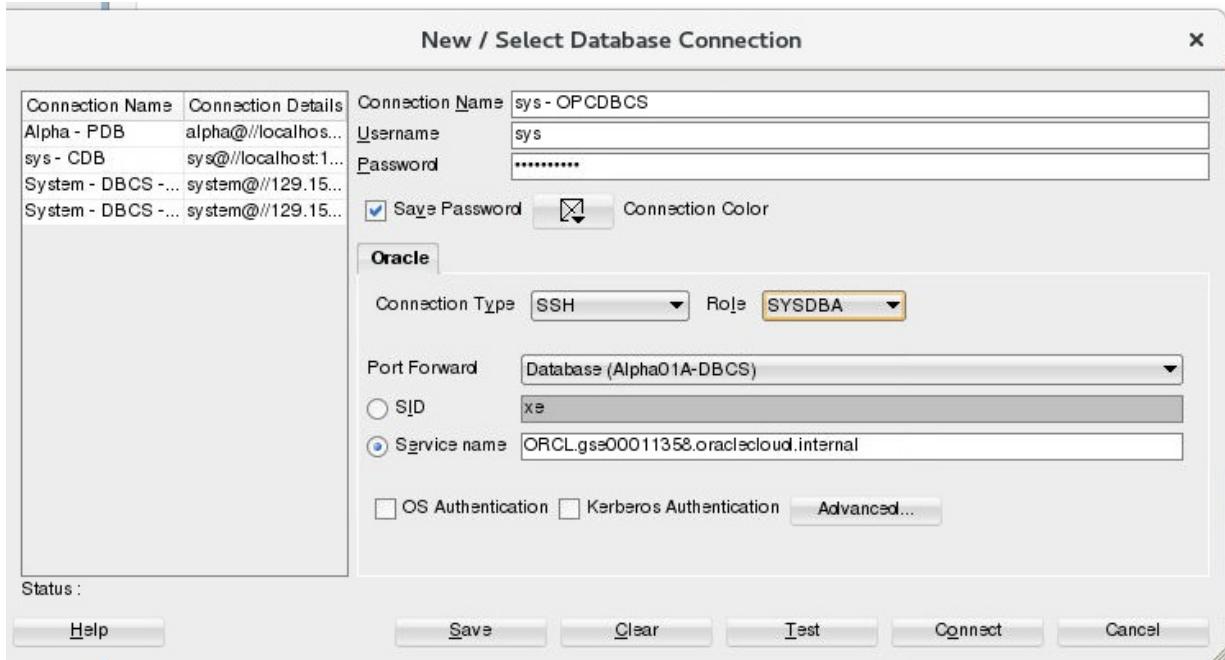
Check "Save Password"

Optionally select a color for the connection

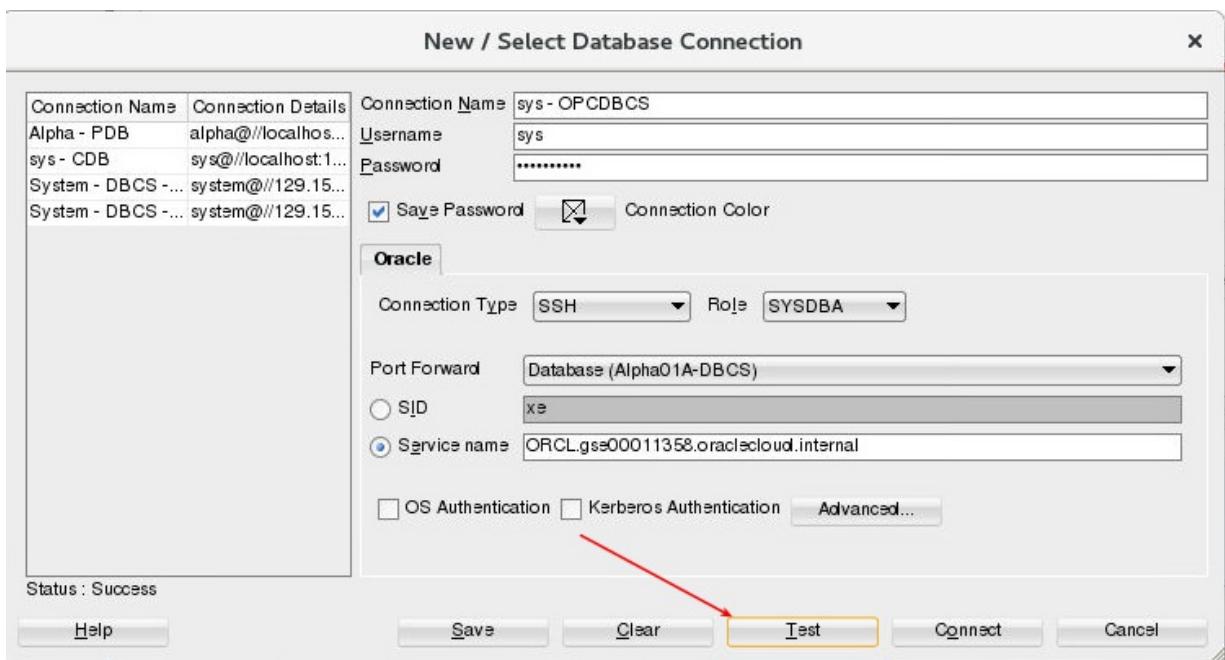
ConnectionType: SSH

Role: SYSDBA

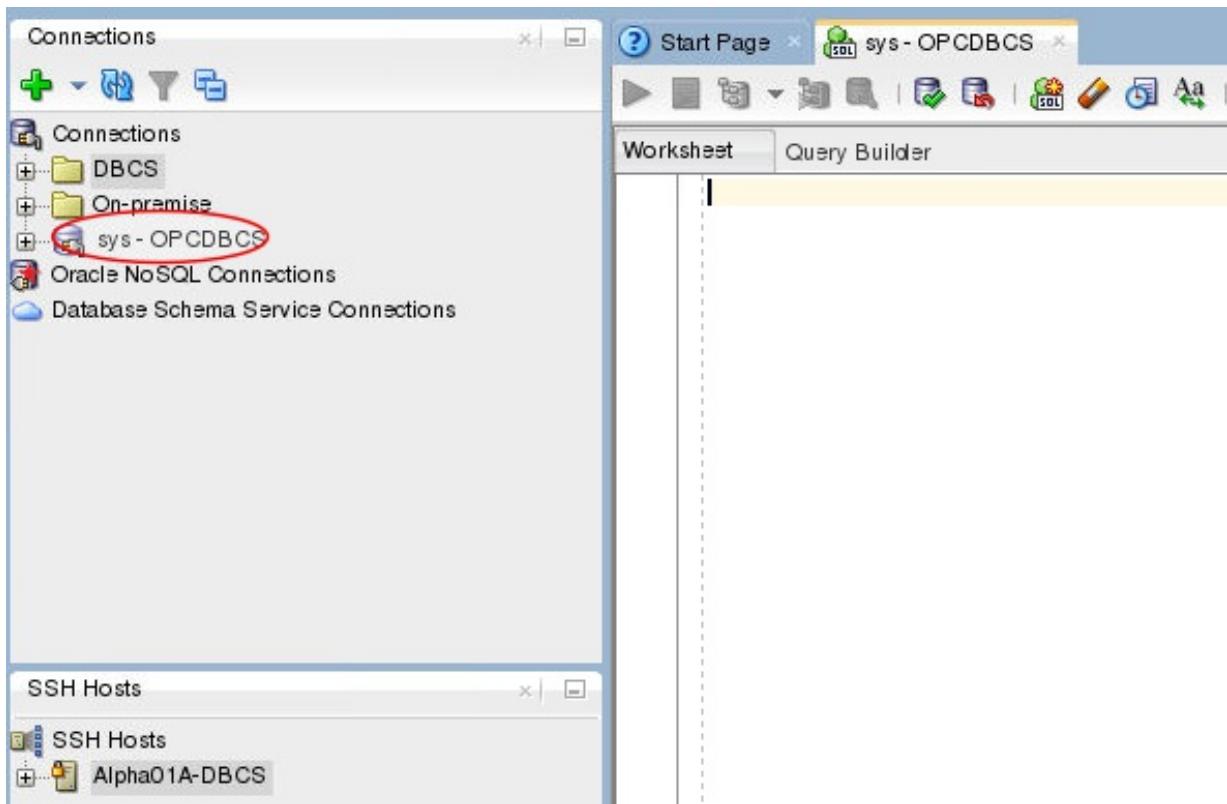
Service Name: ORCL.<Your ID Domain>.oraclecloud.internal



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

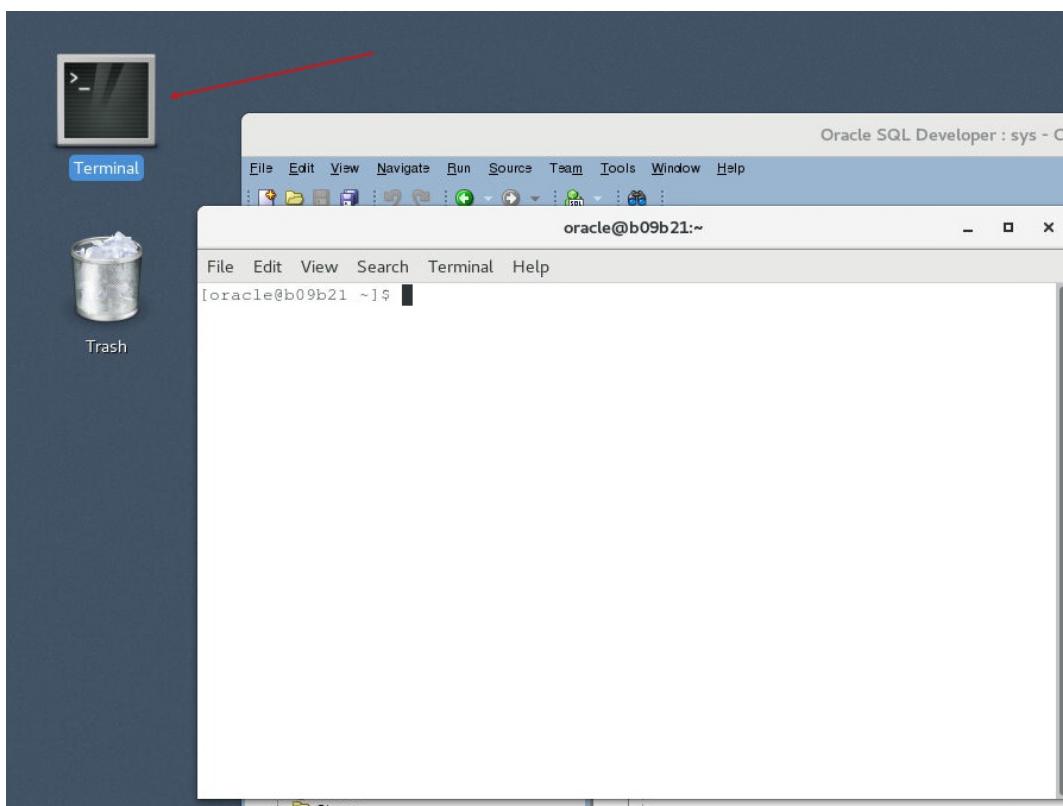


- Click **Connect** to save the connection information which opens a new SQL Worksheet.



STEP 5: Unplug ALPHACLONE

- Since TDE requires additional commands that are not included in the generation of SQL within SQL Developer we will do step 5 in a terminal window. Open a terminal window off the desktop.



- Remove the auto-open wallet. We will use the password wallet. Enter the following commands
 - source dbenv.sh
 - sqlplus sys/Alpha2018_ as sysdba
 - shutdown immediate
 - exit
 - mv /u01/app/oracle/product/12.2/wallet/cwallet.sso /u01/app/oracle/product/12.2
 - source dbenv.sh
 - sqlplus sys/Alpha2018_ as sysdba
 - startup



The screenshot shows a terminal window titled "oracle@b09b21:~". The session logs are as follows:

```

oracle@b09b21:~$ source dbenv.sh
[oracle@b09b21 ~]$ sqlplus sys/Oracle_123 as sysdba

SQL*Plus: Release 12.2.0.1.0 Production on Tue Dec 5 14:51:08 2017
Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64
bit Production
[oracle@b09b21 ~]$ mv /u01/app/oracle/product/12.2/wallet/cwallet.sso /u01/app/o
racle/product/12.2
[oracle@b09b21 ~]$ sqlplus sys/Oracle_123 as sysdba

SQL*Plus: Release 12.2.0.1.0 Production on Tue Dec 5 14:52:46 2017
Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 2516582400 bytes
Fixed Size          8795904 bytes
Variable Size       738199808 bytes
Database Buffers   1761607680 bytes
Redo Buffers        7979008 bytes
Database mounted.
Database opened.
SQL> 
```

- Enter the following commands
 - alter pluggable database all open;
 - administer key management set keystore open identified by oracle container=all;
 - alter session set container=alphaclone;
 - administer key management set keystore open force keystore

- identified by oracle;
- administer key management export encryption keys with secret "oracle" to '/u01/app/oracle/oradata/orcl/alphaclone/alphaclone.p12' identified by oracle;
- alter session set container=cdb\$root;
- alter pluggable database alphaclone close;
- alter pluggable database alphaclone unplug into '/u01/app/oracle/oradata/orcl/alphaclone/alphaclone.xml';

```

SQL> alter pluggable database all open;
Pluggable database altered.

SQL> administer key management set keystore open identified by oracle container=
all;
keystore altered.

SQL> alter session set container=alphaclone;
Session altered.

SQL> administer key management set keystore open force keystore identified by or
acle;
keystore altered.

SQL> administer key management export encryption keys with secret "oracle" to '/
u01/app/oracle/oradata/orcl/alphaclone/alphaclone.p12' identified by oracle;
keystore altered.

SQL> alter session set container=cdb$root;
Session altered.

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

SQL> alter pluggable database alphaclone unplug into '/u01/app/oracle/oradata/or
cl/alphaclone/alphaclone.xml';
Pluggable database altered.

SQL> ■

```

STEP 6: Copy the data files of new cloned on-premise PDB to the Oracle Cloud

- Exit out of SQL PLus and run the following commands.
 - exit;
 - cd /u01/app/oracle/oradata/orcl/alphaclone
 - tar -cvzf alphaclone.tar.gz *

```

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64
bit Production
[oracle@ol74-clientv22 ~]$ cd /u01/app/oracle/oradata/orcl/alphaclone
[oracle@ol74-clientv22 alphaclone]$ tar -cvzf alphaclone.tar.gz *
alphaclone.p12
alphaclone.xml
sysaux01.dbf
system01.dbf
temp01.dbf
undotbs01.dbf
users.dbf
[oracle@ol74-clientv22 alphaclone]$ 
```

- SSH into the Alpha01A-DBCS instance, create an alphaclone directory, and then copy the pluggable database to DBCS.
 - ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@<Alpha01A-DBCS IP>
 - mkdir /u02/app/oracle/oradata/ORCL/alphaclone
 - exit
 - scp -i /u01/OPCWorkshop/ci_opc_keys /u01/app/oracle/oradata/orcl/alphaclone/alphaclone.tar.gz oracle@<Alpha01A-DBCS IP>:/u02/app/oracle/oradata/ORCL/alphaclone

```

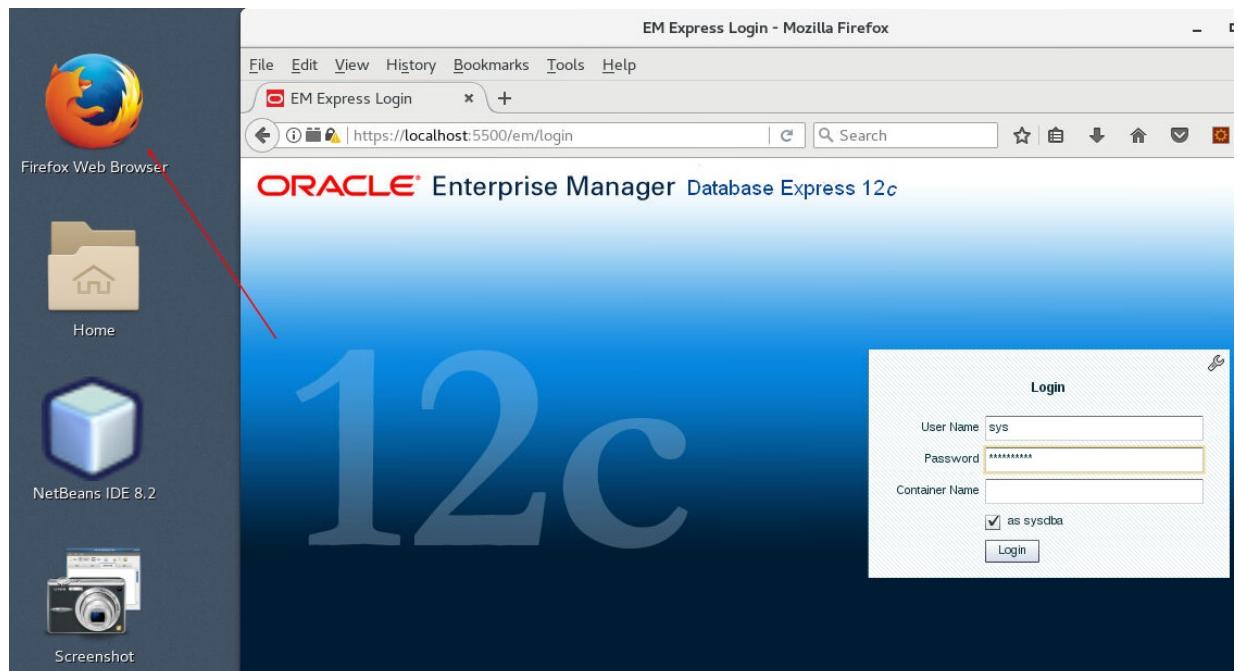
[oracle@ol74-clientv22 alphaclone]$ ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@1
29.157.187.121
[oracle@Alpha01A-DBCS ~]$ mkdir /u02/app/oracle/oradata/ORCL/alphaclone
[oracle@Alpha01A-DBCS ~]$ exit
logout
Connection to 129.157.187.121 closed.
[oracle@ol74-clientv22 alphaclone]$ scp -i /u01/OPCWorkshop/ci_opc_keys /u01/app
/oracle/oradata/orcl/alphaclone/alphaclone.tar.gz oracle@129.157.187.121:/u02/ap
p/oracle/oradata/ORCL/alphaclone
alphaclone.tar.gz                                         100%    72MB 313.0MB/s   00:00
[oracle@ol74-clientv22 alphaclone]$ 
```

- Untar the files
 - ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@<Alpha01A-DBCS IP>
 - cd /u02/app/oracle/oradata/ORCL/alphaclone
 - tar -xvf alphaclone.tar.gz

```
[oracle@ol74-clientv22 alphaclone]$ ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@129.157.187.121
[oracle@Alpha01A-DBCS ~]$ cd /u02/app/oracle/oradata/ORCL/alphaclone
[oracle@Alpha01A-DBCS alphaclone]$ tar -xvf alphaclone.tar.gz
alphaclone.p12
alphaclone.xml
sysaux01.dbf
system01.dbf
temp01.dbf
undotbs01.dbf
users.dbf
[oracle@Alpha01A-DBCS alphaclone]$
```

STEP 7: Plug the cloned on-premise database to Oracle Cloud Database

- Open Firefox on the Desktop and log into Enterprise Manager Express
 - <https://localhost:5500/em>
 - **User Name:** sys
 - **Password:** Alpha2018_
 - **Container Name:** leave blank
 - check 'as sysdba'



- Click on the CDB link.

The screenshot shows the Oracle Database Home page for the ORCL instance. The top navigation bar includes Configuration, Storage, Security, and Performance tabs. The main content area is divided into several sections:

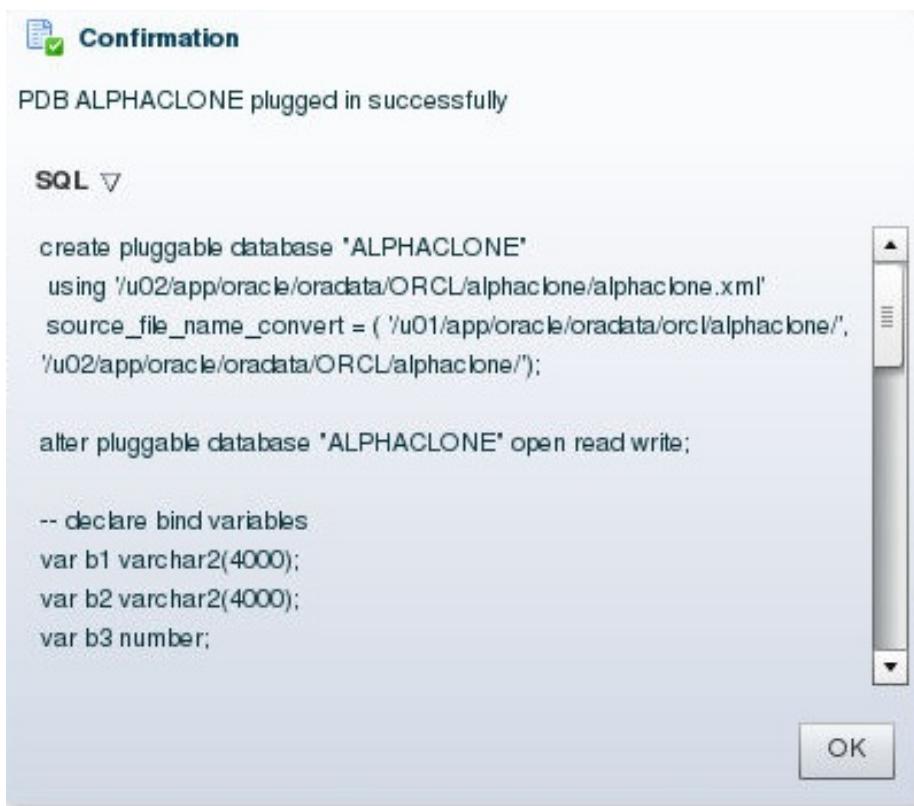
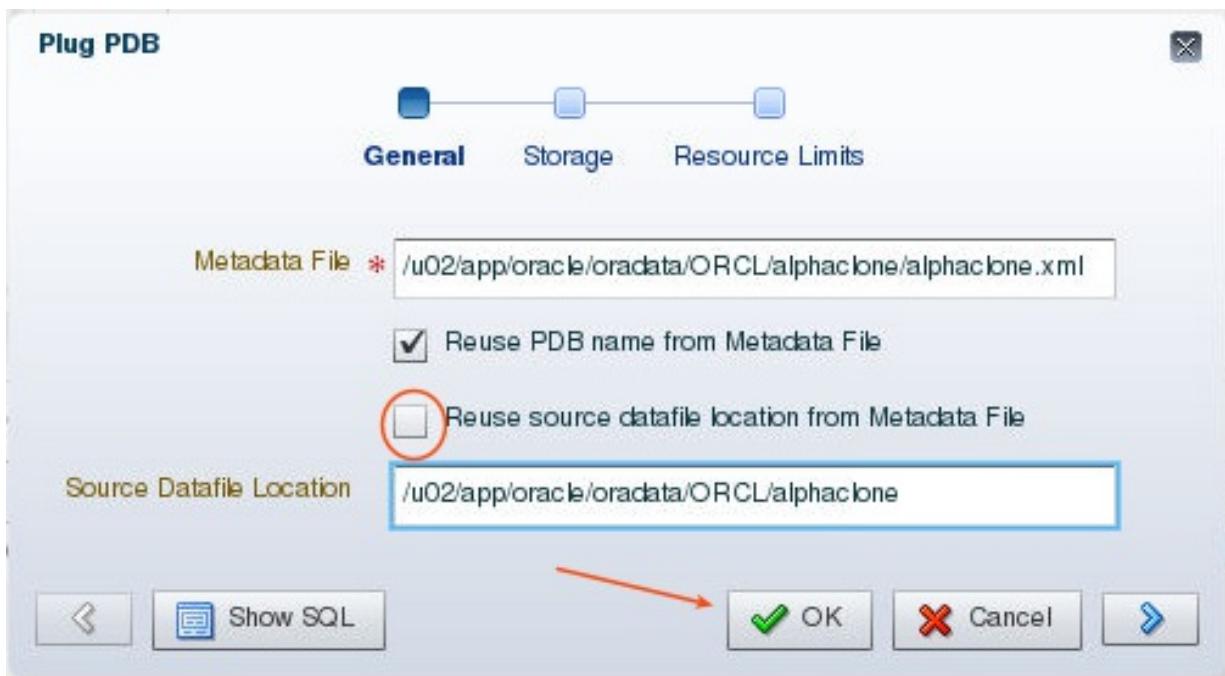
- Status:** Displays basic information like Up Time (1 day, 5 hours, 50 minutes), Type (Single instance (ORCL)), Version (12.2.0.1.0 Enterprise Edition), and Host Name (Alpha01A-DBCS). The "Type" entry is circled in red.
- Performance:** Shows a chart for Activity Class and Containers. The chart indicates "No data available".
- Resources:** Includes three charts: Host CPU (Host CPU usage), Active Sessions (Active Sessions count), and Memory (Memory usage).
- Incidents - Last 24 Hours:** A table showing no incidents.
- SQL Monitor - Last Hour (20 max):** A section showing SQL monitoring data.

- Click on Plug

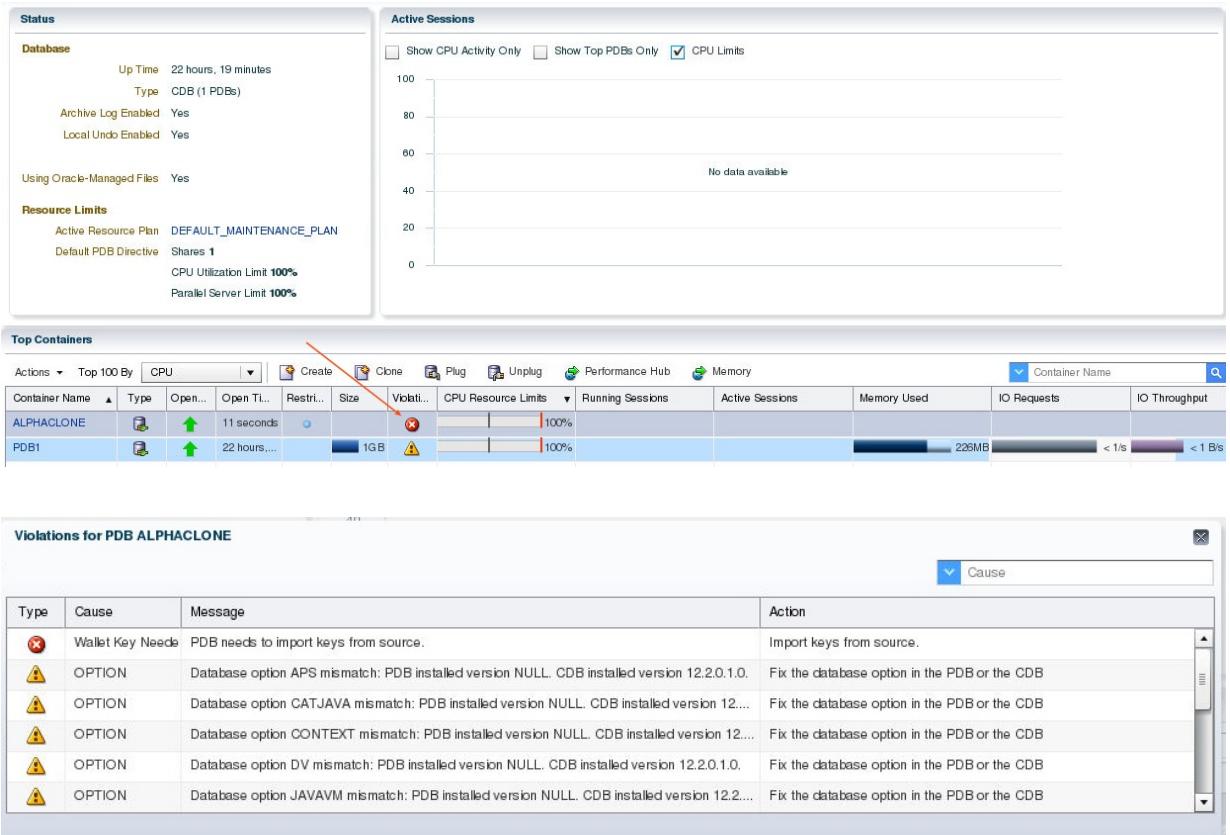
The screenshot shows the Oracle Container Management page. The top navigation bar includes Change Resource Plan and Configure Oracle-Managed Files. The main content area is divided into several sections:

- Status:** Displays Database information (Up Time, Type, Archive Log Enabled, Local Undo Enabled) and Resource Limits (Active Resource Plan, Default PDB Directive).
- Active Sessions:** A chart showing Active Sessions (No data available).
- Top Containers:** A table listing Top Containers. The "Plug" button for PDB1 is highlighted with a red arrow.

- Enter Meetadata file, uncheck the 'Reuse source datafile' and enter the Source Datafile Location
 - **Metadata File:**
`/u02/app/oracle/oradata/ORCL/alphaclone/alphaclone.xml`
 - **Source Datafile Location:**
`/u02/app/oracle/oradata/ORCL/alphaclone`



- Notice the pluggable database has a violation. Click on this.



- Cloud databases have TDE configured by default, but we need to import the alphadb key. Open a terminal window on the compute desktop and SSH into the Alpha01A-DBCS instance.
 - source dbenv.sh
 - ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@<Alpha01A-DBCS IP>
 - sqlplus sys/Alpha2018_ as sysdba
 - alter session set container = alphaclone;
 - select * from v\$encryption_wallet;

```

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> alter session set container = alphaclone;

Session altered.

SQL> select * from v$encryption_wallet;

WRL_TYPE
-----
WRL_PARAMETER
-----
STATUS          WALLET_TYPE      WALLET_OR_FULLY_BAC
-----          -----
CON_ID
-----
FILE
-----
OPEN_NO_MASTER_KEY    AUTOLOGIN     SINGLE      UNDEFINED
4

SQL> █

```

- Import keys

- administer key management import keys with secret "oracle" from '/u02/app/oracle/oradata/ORCL/alphaclone/alphaclone.p12' force keystore identified by Alpha2018_ with backup;

```

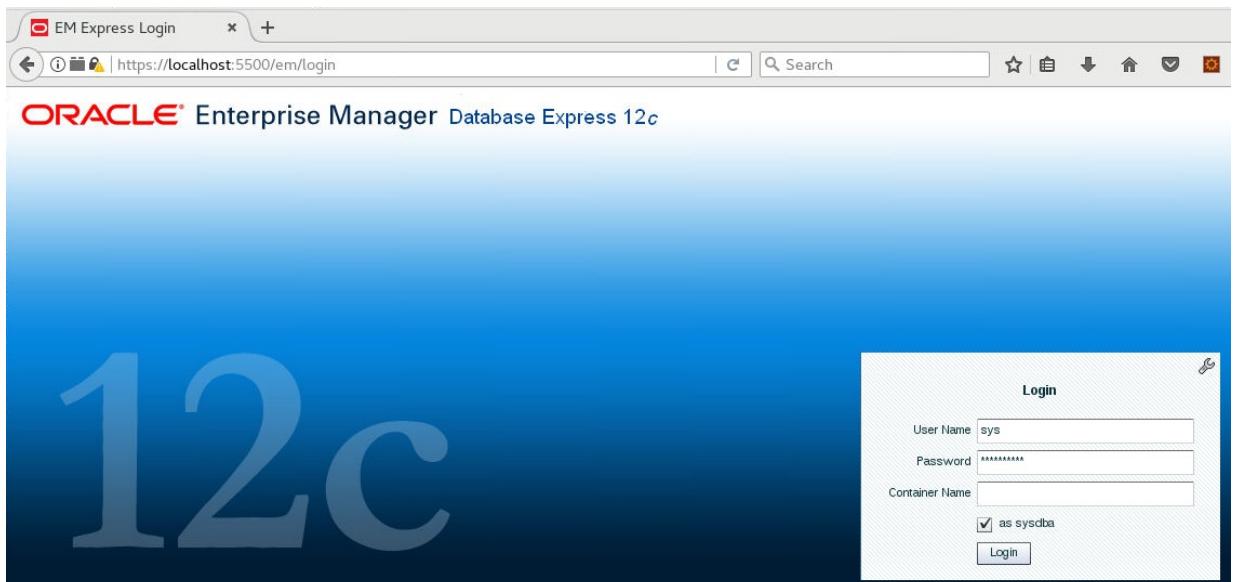
SQL> administer key management import keys with secret "oracle" from '/u02/app/oracle/oradata/ORCL/alphaclone/alphaclone.p12' force keystore identified by Alpha2018_
 _ with backup;
keystore altered.

SQL> █

```

- Review status of pluggable database in EM. Open (or go back to) Firefox and log into IM:

- <https://localhost:5500/em>
 - User:** sys
 - password:** sys password



- Navigate to containers.

The screenshot shows the Oracle Enterprise Manager Database Express 12c Database Home page. The top navigation bar includes links for Configuration, Storage, Security, and Performance. The main area is titled 'Database Home'. On the left, there's a 'Status' panel with various database statistics: Up Time (4 days, 13 hours, 57 minutes), Type (Single instance (ORCL)), CDB (2 PDBs) (highlighted with a red arrow pointing to the Resources section below), Version (12.2.0.1.0 Enterprise Edition), Database Name (ORCL), Instance Name (ORCL), Platform Name (Linux x86 64-bit), Host Name (Alpha01A-DBCS), Thread (1), and Archiver (Started). To the right is a 'Performance' panel with a chart showing activity levels across different activity classes. Below the Status panel is a 'Resources' section, which is partially visible at the bottom of the screenshot.

- Highlight the Alphadb pluggable database (don't click on the link) and select action close.

Top Containers

Actions ▾ Top 100 By

- Create
- Clone
- Remote Clone
- Relocate
- Refresh
- Plug
- Unplug
- Drop
- Set Storage Limits
- Set Resource Limits

Open/Close PDB

Save/Discard State ▾

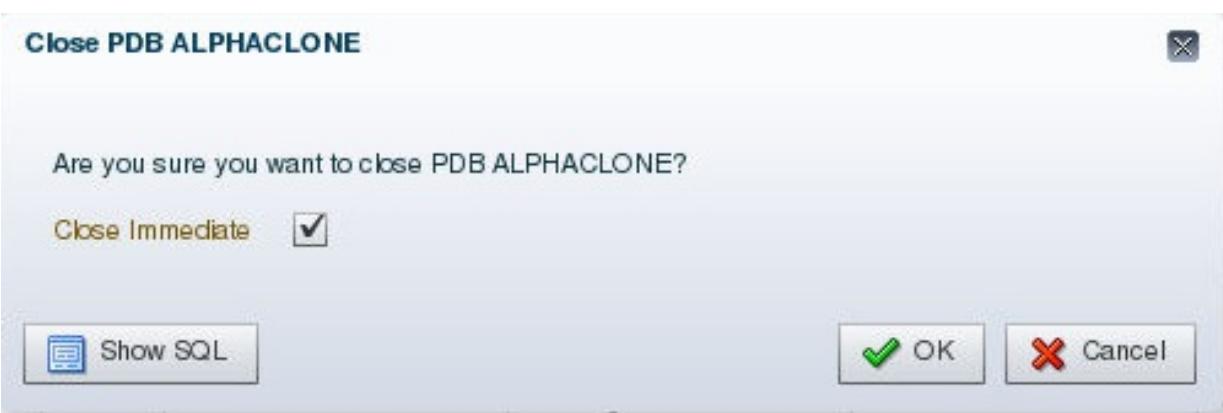
Open

Open All

Close

Close All

Open Time	Restri...	Size	Violati...	Active Sessions	Memory Used	IO Requests
4 days, 13 h...		731MB	X		87MB	
		1GB	!		374MB	



- Re-open alphadb pdb

Top Containers

Actions ▾ Top 100 By

- Create
- Clone
- Remote Clone
- Relocate
- Refresh
- Plug
- Unplug
- Drop
- Set Storage Limits
- Set Resource Limits

Open/Close PDB

Save/Discard State ▾

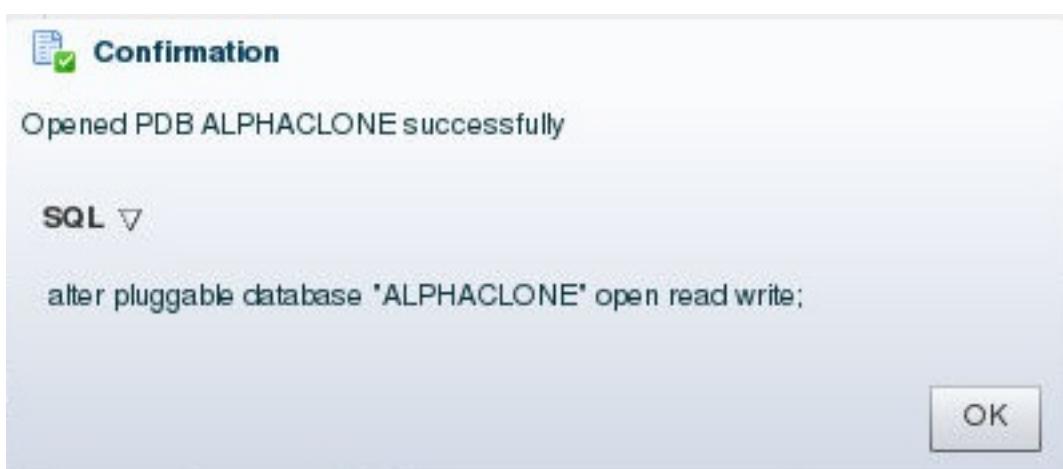
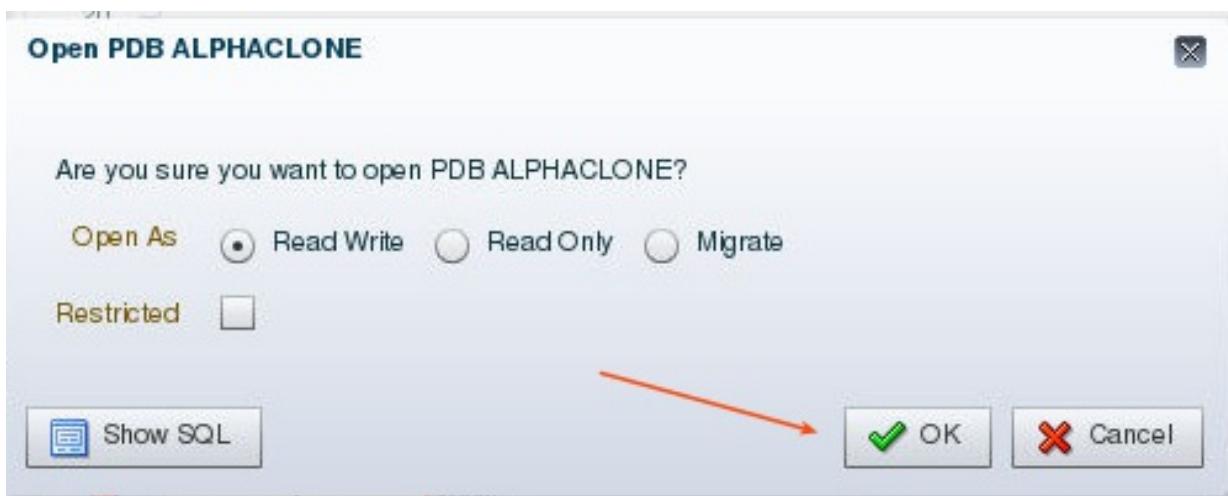
Open

Open All

Close

Close All

Open Time	Restri...	Size	Violati...	Active Sessions	Memory Used
4 days, 13 h...		731MB	X		87MB
		1GB	!		374MB



Top Containers													
Actions	Top 100 By	CPU	Create	Clone	Plug	Unplug	Performance Hub	Memory	Container Name	Running Sessions	Active Sessions	Memory Used	IO Requests
Container Name	Type	Open...	Open Ti...	Restri...	Size	Violati...	CPU Resource Limits	Running Sessions					
ALPHAclone		2 seconds				⚠️		100%					
PDB1		22 hours...			1GB	⚠️		100%				226MB	< 1/s

STEP 8: Create a SQL Developer connection to the Public Cloud database ALPHAPDB schema

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

Connection Name: `Alpha01A-DBCS`

Username: `alpha`

Password: `Alpha2018_`

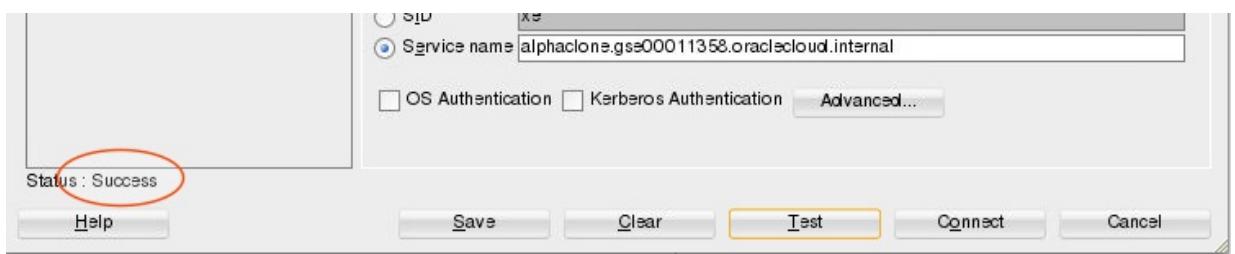
Check "Save Password"

Optionally select a color for the connection

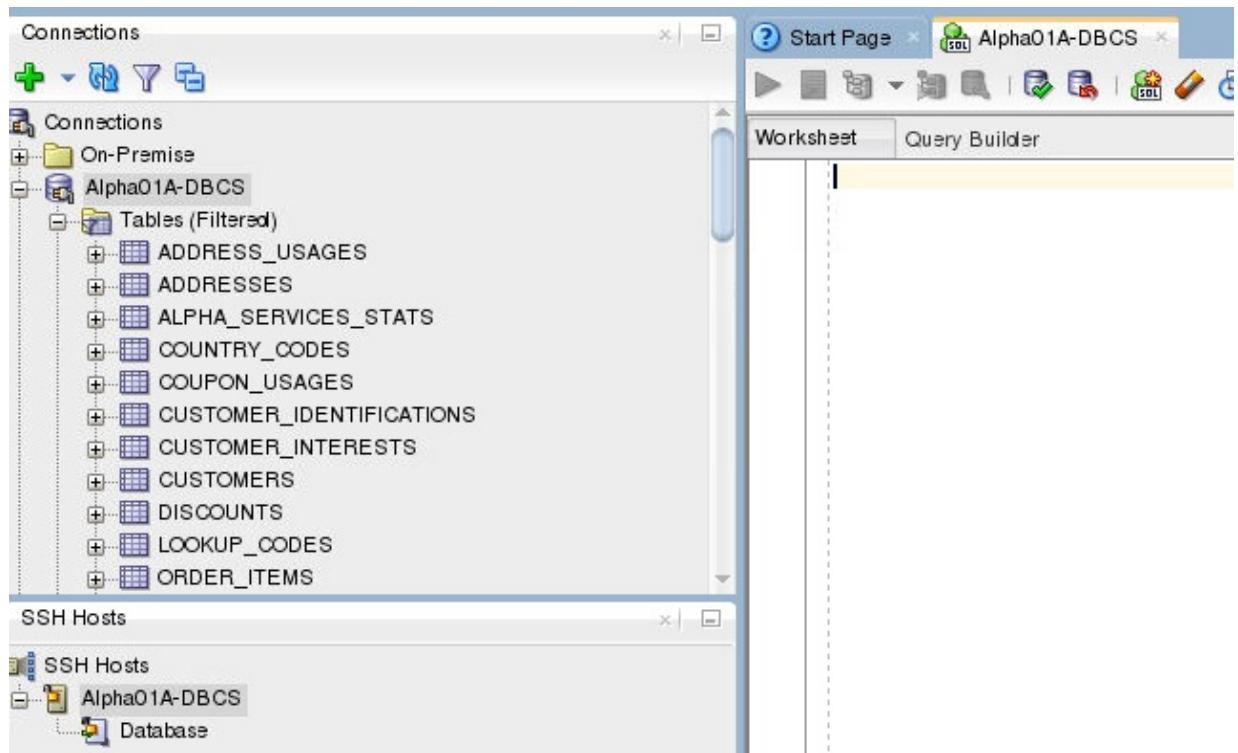
ConnectionType: `SSH`

Service Name: `alphaclone.<Your ID
Domain>.oraclecloud.internal`

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



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



Cloud Migration Using Data Pump: Schema Level

Export the Alpha Schema

STEP 9: Create directory for datapump export

- Open a terminal window and run the following commands
 - source dbenv.sh
 - sqlplus alpha/Alpha2018_@alphapdb
 - create directory oracle as '/home/oracle';
 - exit;

```
[oracle@ol74-clientv22 ~]$ sqlplus alpha/Alpha2018_@alphapdb
SQL*Plus: Release 12.2.0.1.0 Production on Sat Dec 16 16:43:42 2017
Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Sat Dec 16 2017 16:42:30 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> create directory oracle as '/home/oracle';

Directory created.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64
bit Production
[oracle@ol74-clientv22 ~]$
```

STEP 10: Run datapump export

- Exit out of sqlplus and export the data
 - **exit** if you are still in sqlplus, otherwise open a new command window and enter source dbenv.sh
 - **expdp alpha/Alpha2018_@alphapdb directory=oracle dumpfile=alpha.dmp compression=all**
 - Note that directory tmp was created previously for you and is mapped to /tmp. Also we are compressing the export file on the fly to reduce the size for when we scp the file to DBCS.

```

[oracle@ol74-clientv22 ~]$ sqlplus /nolog
SQL*Plus: Release 12.1.0.2.0 Production on Sat Dec 16 16:46:45 2017

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, Oracle Label Security, Oracle Data Mining,
Oracle R and Oracle Machine Learning in-database capabilities
Version 12.1.0.2.0

SQL> expdp system/alpha dumpfile=alpha.dmp compression=all
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROACT_SCHEMA
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/FUNCTION/FUNCTION
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/FUNCTION/ALTER_FUNCTION
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/TRIGGER
. . exported "ALPHA"."SALES_ORDER_HEADER"          130.8 KB  21456 rows
. . exported "ALPHA"."SALES_ORDER_LINE_ITEM"      71.01 KB  21456 rows
. . exported "ALPHA"."PRODUCTS"                   9.507 KB   57 rows
. . exported "ALPHA"."COUNTRY_CODES"             8.281 KB  244 rows
. . exported "ALPHA"."ADDRESSES"                9.125 KB  119 rows
. . exported "ALPHA"."CUSTOMERS"                 7.171 KB   16 rows
. . exported "ALPHA"."ORDERS"                   6.281 KB   3 rows
. . exported "ALPHA"."LOOKUP_CODES"              6.75 KB   56 rows
. . exported "ALPHA"."PAYMENT_OPTIONS"           6.257 KB   7 rows
. . exported "ALPHA"."ALPHA_SERVICES_STATS"       6.156 KB  33 rows
. . exported "ALPHA"."CUSTOMER_IDENTIFICATIONS"  5.984 KB   6 rows
. . exported "ALPHA"."ADDRESS_USAGES"            5.843 KB  25 rows
. . exported "ALPHA"."PRODUCT_CATEGORIES"         5.968 KB  20 rows
. . exported "ALPHA"."DISCOUNTS"                 5.867 KB   7 rows
. . exported "ALPHA"."ORDER_ITEMS"               5.617 KB   4 rows
. . exported "ALPHA"."COUPON_USAGES"             5.507 KB   3 rows
. . exported "ALPHA"."PRODUCT_STOCK_LEVELS"        5.453 KB  26 rows
. . exported "ALPHA"."CUSTOMER_INTERESTS"          5.453 KB   6 rows
. . exported "ALPHA"."PRODUCT_IMAGES"             0 KB   0 rows
Master table "ALPHA"."SYS_EXPORT_SCHEMA_01" successfully loaded/unloaded
*****
Dump file set for ALPHA.SYS_EXPORT_SCHEMA_01 is:
/home/oracle/alpha.dmp
Job "ALPHA"."SYS_EXPORT_SCHEMA_01" successfully completed at Sat Dec 16 16:46:45
2017 elapsed 0 00:01:03
[oracle@ol74-clientv22 ~]$ 

```

STEP 11: Copy the export Data Pump file to the server

- Use the following secure copy (**scp**) command to transfer the Data Pump export to the DBCS server using your Database Service's Public IP address identified in Lab 100:
 - **scp -i /u01/OPCWorkshop/ci_opc_keys /home/oracle/alpha.dmp oracle@{your public IP}:**

```

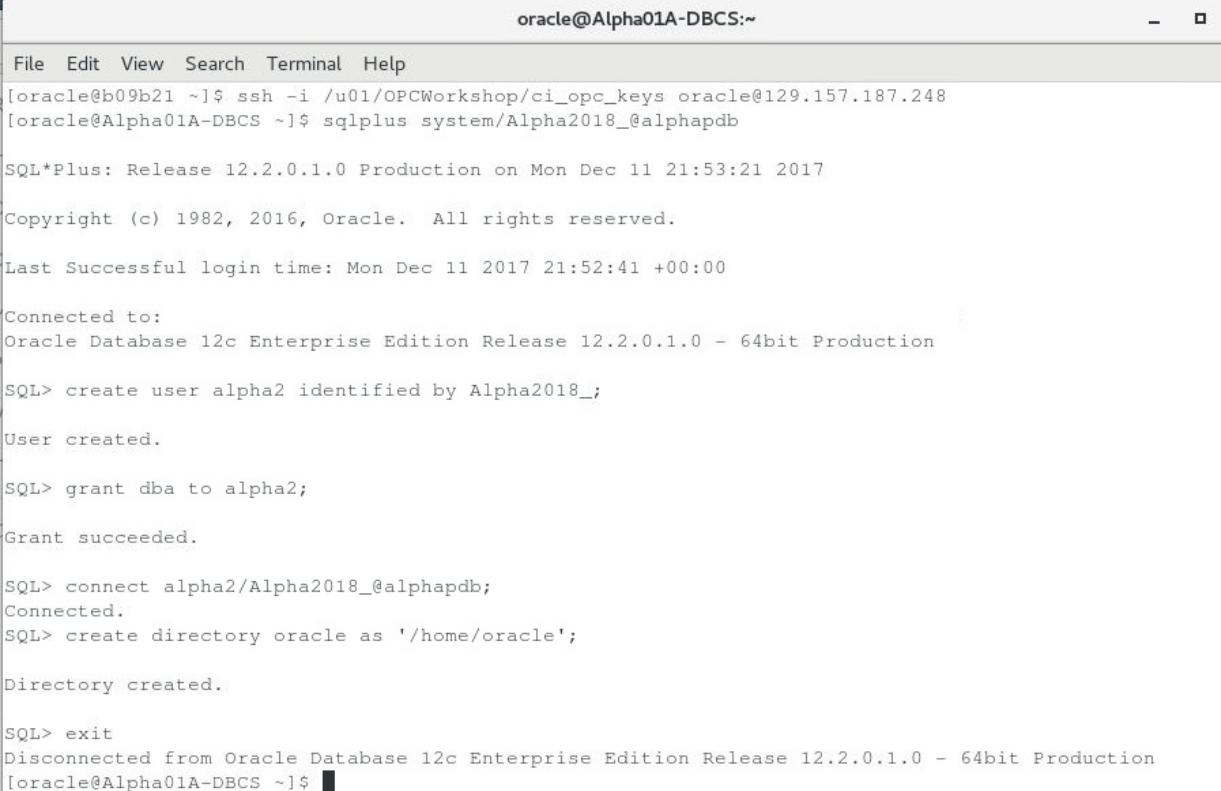
[oracle@ol74-clientv22 ~]$ scp -i /u01/OPCWorkshop/ci_opc_keys /home/oracle/alpha.dmp oracle@129.157.187.121:.
alpha.dmp                                         100%   668KB 116.1MB/s   00:00
[oracle@ol74-clientv22 ~]$ 

```

STEP 12: Create a new schema to hold a copy of the data

- Open a new terminal window (or use the current one) and enter the following commands that will SSH to the Alpha01A-DBCS instance, create an Oracle directory, and import the data.
 - **ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@{your public IP} -- log into your remote DBCS instance**

- `sqlplus system/Alpha2018_@alphapdb;` -- log into system in the alphapdb
- `create user alpha2 identified by Alpha2018_;` -- create schema alpha2 (alpha already has the data from the previous lab)
- `grant dba to alpha2;`
- `connect alpha2/Alpha2018_@alphapdb;` -- connect to alpha2 so we can create the oracle directory
- `create directory oracle as '/home/oracle';`
- `exit`



The screenshot shows a terminal window titled "oracle@Alpha01A-DBCS:~". The window contains the following text:

```

oracle@Alpha01A-DBCS:~ - □
File Edit View Search Terminal Help
[oracle@b09b21 ~]$ ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@129.157.187.248
[oracle@Alpha01A-DBCS ~]$ sqlplus system/Alpha2018_@alphapdb

SQL*Plus: Release 12.2.0.1.0 Production on Mon Dec 11 21:53:21 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Mon Dec 11 2017 21:52:41 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> create user alpha2 identified by Alpha2018_;

User created.

SQL> grant dba to alpha2;

Grant succeeded.

SQL> connect alpha2/Alpha2018_@alphapdb;
Connected.
SQL> create directory oracle as '/home/oracle';

Directory created.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
[oracle@Alpha01A-DBCS ~]$ ●

```

STEP 13: Import the data

- Import the data. Run the following command in your terminal window.
 - `impdp alpha2/Alpha2018_@alphapdb directory=oracle dumpfile=alpha.dmp remap_schema=alpha:alpha2`

```

Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
  . . imported "ALPHA2"."SALES_ORDER_HEADER"          129.3 KB  21456 rows
  . . imported "ALPHA2"."SALES_ORDER_LINE_ITEM"      69.58 KB  21456 rows
  . . imported "ALPHA2"."EXPORT_JOB_SQLDEV_82"       155.5 KB   1467 rows
  . . imported "ALPHA2"."PRODUCTS"                   9.515 KB    57 rows
  . . imported "ALPHA2"."COUNTRY_CODES"              8.273 KB   244 rows
  . . imported "ALPHA2"."ADDRESSES"                 9.125 KB   119 rows
  . . imported "ALPHA2"."CUSTOMERS"                  7.164 KB    16 rows
  . . imported "ALPHA2"."ORDERS"                     6.281 KB     3 rows
  . . imported "ALPHA2"."LOOKUP_CODES"               6.742 KB    56 rows
  . . imported "ALPHA2"."PAYMENT_OPTIONS"            6.273 KB     7 rows
  . . imported "ALPHA2"."ALPHA_SERVICES_STATS"       6.140 KB   33 rows
  . . imported "ALPHA2"."CUSTOMER_IDENTIFICATIONS"   5.992 KB    6 rows
  . . imported "ALPHA2"."ADDRESS_USAGES"              5.851 KB   25 rows
  . . imported "ALPHA2"."PRODUCT_CATEGORIES"          6 KB      20 rows
  . . imported "ALPHA2"."DISCOUNTS"                  5.773 KB     7 rows
  . . imported "ALPHA2"."ORDER_ITEMS"                5.664 KB    4 rows
  . . imported "ALPHA2"."COUPON_USAGES"              5.531 KB   3 rows
  . . imported "ALPHA2"."PRODUCT_STOCK_LEVELS"        5.468 KB   26 rows
  . . imported "ALPHA2"."CUSTOMER_INTERESTS"          5.484 KB    6 rows
  . . imported "ALPHA2"."MSTARS"                     4.875 KB    4 rows
  . . imported "ALPHA2"."PRODUCT_IMAGES"              0 KB      0 rows
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/FUNCTION/FUNCTION
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/FUNCTION/ALTER_FUNCTION
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/TRIGGER
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
ORA-39082: Object type FUNCTION:"ALPHA2"."CALCTPS" created with compilation warnings

Job "ALPHA2"."SYS_IMPORT_FULL_01" completed with 2 error(s) at Mon Dec 11 22:02:19 2017 elapsed 0 00:0
0:40
● [oracle@Alpha01A-DBCS ~]$

```

Cloud Migration Using Data Pump: Tablespace Level

While datapump provides a very fast multi-threaded technique to move data quickly between Oracle Databases, it was not designed for very large volumes of data. Transportable Tablespaces enable a database file copy technique that does not require an exported copy of the data, but instead allow you to copy the in-place data files to target, which using Datapump to capture the metadata only.

Export the euro Schema

We will be exporting a GG (GoldenGate) tablespace, since there already is a users tablespace in the target database.

STEP 14: Open euro in read/write mode and export the metadata

- Open a terminal window and create an Oracle export directory and put

the tablespace in read only mode.

- source dbenv.sh
- sqlplus system/Alpha2018_@pdb1
- create directory oracle as '/home/oracle';
- alter tablespace gg read only;

```
[oracle@ol74-clientv22 ~]$ source dbenv.sh
[oracle@ol74-clientv22 ~]$ sqlplus system/Alpha2018_@pdb1

SQL*Plus: Release 12.2.0.1.0 Production on Sat Dec 16 18:43:20 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Sat Dec 16 2017 18:42:16 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> create directory oracle as '/home/oracle';

Directory created.

SQL> alter tablespace gg read only;

Tablespace altered.

SQL> □
```

- Export the data. Enter the following commands in the terminal window.
 - exit
 - expdp system/Alpha2018_@pdb1 directory=oracle dumpfile=ggtbs.dmp transport_tablespaces=gg exclude=statistics encryption_password=oracle logfile=full_tts_export.log

```

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production
Starting "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01": system/********@pdb1 directory
=oracle dumpfile=ggtbs.dmp transport tablespaces=gg exclude=statistics encryption_password=***** logfile=full_tts_export.log
Processing object type TRANSPORTABLE_EXPORT/INDEX_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/CONSTRAINT/CONSTRAINT
Master table "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully loaded/unloaded
*****
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /home/oracle/ggtbs.dmp
*****
Datafiles required for transportable tablespace GG:
  /u01/app/oracle/oradata/orcl/pdb1/gg.dbf
Job "SYSTEM"."SYS_EXPORT_TRANSPORTABLE_01" successfully completed at Sat Dec 16
18:56:19 2017 elapsed 0 00:00:31

[oracle@ol74-clientv22 ~]$ □

```

STEP 15: Copy the datafiles to the target DBCS instance

- In the terminal window enter the following:
 - `scp -i /u01/OPCWorkshop/ci_opc_keys /home/oracle/ggtbs.dmp oracle@<DBCS IP>: -- metadata`
 - `scp -i /u01/OPCWorkshop/ci_opc_keys /u01/app/oracle/oradata/orcl/pdb1/gg.dbf oracle@<DBCS IP>: -- datafiles`

```

[oracle@ol74-clientv22 ~]$ scp -i /u01/OPCWorkshop/ci_opc_keys /home/oracle/ggtbs.dmp oracle@129.157.187.121:.
ggtbs.dmp                                         100%   600KB 149.8MB/s   00:00
[oracle@ol74-clientv22 ~]$ scp -i /u01/OPCWorkshop/ci_opc_keys /u01/app/oracle/oradata/orcl/pdb1/gg.dbf oracle@129.157.187.121:.
gg.dbf                                           100%    10MB 171.0MB/s   00:00
[oracle@ol74-clientv22 ~]$ □

```

STEP 16: Import the tablespace into the target DBCS instance

We will be importing the data into the pdb1 instance. First we'll need to drop the users tablespace since it already exists, then create an Oracle tmp directory, and then run the import.

- SSH to the target DBCS instance, log in, and create euro user (must

already exist in the target).

- ssh -i /u01/OPCWorkshop/ci_opc_keys oracle@<DBCS IP>
- sqlplus system/Alpha2018_@pdb1
- create user euro identified by Alpha2018_;
- grant dba to euro;
- create directory oracle as '/home/oracle';

```
[oracle@Alpha01A-DBCS ~]$ sqlplus system/Alpha2018_@pdb1
SQL*Plus: Release 12.2.0.1.0 Production on Sun Dec 17 00:01:36 2017
Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Sat Dec 16 2017 21:58:34 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> create user euro identified by Alpha2018_;
User created.

SQL> grant dba to euro;
Grant succeeded.

SQL> create directory oracle as '/home/oracle';
Directory created.

SQL> █
```

- Exit from sqlplus and copy the ggtbs.dbf datafile to the pdb1 directory.

- exit
- cp /home/oracle/gg.dbf /u02/app/oracle/oradata/ORCL/PDB1

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64
bit Production
[oracle@Alpha01A-DBCS ~]$ cp /home/oracle/gg.dbf /u02/app/oracle/oradata/ORCL/PD
B1
[oracle@Alpha01A-DBCS ~]$ █
```

- Import the data (note you will see some errors due to existence of some objects).
- impdp system/Alpha2018_@pdb1 directory=oracle

```
dumpfile=ggtbs.dmp logfile=full_tts_imp.log  
encryption_password=oracle  
transport_datafiles='/u02/app/oracle/oradata/ORCL/PDB1/gg.dbf'
```

```
[oracle@Alpha01A-DBCS ~]$ impdp system/Alpha2018_@pdb1 directory=oracle dumpfile=ggtbs.dmp logfile=full_tts_imp.log encryption_password=oracle transport_datafiles='/u02/app/oracle/oradata/ORCL/PDB1/gg.dbf'  
  
Import: Release 12.2.0.1.0 - Production on Sun Dec 17 00:07:30 2017  
  
Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit  
Production  
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully loaded/unloaded  
Source time zone is -04:00 and target time zone is +00:00.  
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01": system/********@pdb1 directory=  
=oracle dumpfile=ggtbs.dmp logfile=full_tts_imp.log encryption_password=*****  
transport_datafiles=/u02/app/oracle/oradata/ORCL/PDB1/gg.dbf  
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK  
Processing object type TRANSPORTABLE_EXPORT/TABLE  
Processing object type TRANSPORTABLE_EXPORT/CONSTRAINT/CONSTRAINT  
Processing object type TRANSPORTABLE_EXPORT/INDEX_STATISTICS  
Processing object type TRANSPORTABLE_EXPORT/TABLE_STATISTICS  
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK  
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Sun Dec 17  
00:07:49 2017 elapsed 0 00:00:12  
  
[oracle@Alpha01A-DBCS ~]$ ]
```

- Confirm tablespace and contents exist by querying the Oracle dictionary.
Log into sqlplus and run the following query.
 - `sqlplus system/Alpha2018_@pdb1`
 - `select tablespace_name, count(*) from dba_tables group by
tablespace_name;`

```
[oracle@Alpha01A-DBCS ~]$ sqlplus euro/Alpha2018_@pdb1
SQL*Plus: Release 12.2.0.1.0 Production on Sun Dec 17 00:09:47 2017
Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> select table_name from user_tables;

TABLE_NAME
-----
ORDER_DETAILS
CATEGORIES
CATEGORIES_DESCRIPTION
PRODUCTS
PRODUCTS_DESCRIPTION
PRODUCTS_TO_CATEGORIES
CUSTOMERS
NEXT_CUST
CUSTOMERS_INFO
ORDERS
ORDERS_PRODUCTS

TABLE_NAME
-----
ORDERS_STATUS_HISTORY
ORDERS_TOTAL
NEXT_ORDER
CUSTOMERS_LKUP

15 rows selected.

SQL> select distinct tablespace_name from user_tables;

TABLESPACE_NAME
-----
GG

SQL> █
```

Cloud Migration Using Database Links (Table Level)

Occasionally you just want to copy one or more tables from one database to another, and the easiest/quickest way to do that is to use a database link. A database link connects two databases with sqlnet allowing you to reference remote tables in your local database. This is good for table data, but other object types such as stored procedures, etc. cannot be replicated as easily.

Create a Database Link

STEP 17: Create Database Link on the local system

- Since we need to use tunnels to communicate with the remote DBCS instance when using ports other than 22 (which is open) we need to ensure our tunnels are still open (from lab 100). In a terminal window enter the following

- sudo su
- ps -ef|grep DBCS

```
[oracle@b09b21 ~]$ sudo su
[root@b09b21 oracle]# ps -ef|grep dbcs
root      27060 27025  0 10:36 pts/1    00:00 grep --color=auto dbcs
```

-- if this is all you see then the tunnels are not open. If you see the tunnels then no need for the next step

- If you do need to re-establish the tunnels enter the following, and do NOT close the terminal window, just minimize it.
 - /u01/OPCWorkshop/lab/setssh.sh DBONLY -- enter the DBCS IP address when prompted
- Re-test tunnels - you should see this.

```
[root@b09b21 oracle]# ps -ef|grep DBCS
root      3147      1  0 Dec11 ?        00:00:00 sudo ssh -t -t -F myssh AlphaDBCS
root      3150     3147  0 Dec11 ?        00:00:00 ssh -t -t -F myssh AlphaDBCS
root     27633  27580  0 10:41 pts/1    00:00:00 zenity --info --text=Tunnels have been successfully created on...\n
      DBCS: 129.157.187.248\n  DO NOT CLOSE THE TERMINAL WINDOW
root     27812  27728  0 10:42 pts/2    00:00:00 grep --color=auto dbcs
[root@b09b21 oracle]#
```

- Log into SQLPlus in and create the database link. Be sure to update this command with your Identity Domain.
 - source dbenv.sh
 - sqlplus alpha/Alpha2018_@alphapdb;
 - create database link alpha_dbcs.oracledemo.com connect to alpha identified by Alpha2018_ using '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1530))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=alphapdb.<IDENTITY DOMAIN>.oraclecloud.internal)))';
 - select sysdate from dual@alpha_dbcs.oracledemo.com;

```

[oracle@b09b21 ~]$ source dbenv.sh
[oracle@b09b21 ~]$ sqlplus alpha/Alpha2018_@alphapdb

SQL*Plus: Release 12.2.0.1.0 Production on Tue Dec 12 11:57:55 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Tue Dec 12 2017 11:56:16 -05:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> create database link alpha_dbcs.oracledemo.com connect to alpha identified
by Alpha2018_ using '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=localhost) (PORT=1
530)) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=alphapdb.gse00011358.oraclecl
oud.internal))';
Database link created.

SQL> select sysdate from dual@alpha_dbcs.oracledemo.com;

SYSDATE
-----
12-DEC-17

SQL> □

```

- Copy a table from the remote DBCS to the local instance. Note that you cannot create a remote table (DDL operation) on the remote server but you can do an insert operation. First need alter the local users tablespace and make it read write.
 - **create table mstars_local as select * from mstars@alpha_dbcs.oracledemo.com;**
 - **insert into mstars@alpha_dbcs.oracledemo.com select * from mstars;**
 - **commit;**

```
SQL> alter tablespace users read write;
Tablespace altered.

SQL> create table mstars_local as select * from mstars@alpha_dbcs.oracledemo.com
;

Table created.

SQL> insert into mstars@alpha_dbcs.oracledemo.com select * from mstars;
4 rows created.

SQL> commit;

Commit complete.

SQL> █
```