



**Hardware and Software**  
**Engineered to Work Together**

# Oracle Database Cloud for Oracle DBAs

Activity Guide

D96069GC10

Edition 1.0 | September 2016 | D97403

Learn more from Oracle University at [oracle.com/education/](http://oracle.com/education/)

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

## **Disclaimer**

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle. The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

## **Restricted Rights Notice**

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

### **U.S. GOVERNMENT RIGHTS**

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

## **Trademark Notice**

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

## **Authors**

Dominique Jeunot, Donna K. Keesling, James L. Spiller

## **Technical Contributors and Reviewers**

Jean-François Verrier

## **Editors**

Arijit Ghosh, Vijayalakshmi Narasimhan

## **Graphic Designer**

Maheshwari Krishnamurthy

## **Publisher**

Sumesh Koshy

# Table of Contents

1. Practicing Database Deployment	4
1.1 Practice 1-1 Walking Through Oracle Cloud	5
1.2 Practice 2-1 Creating a Database Deployment	6
1.3 Practice 3-1 Navigating Through Consoles	10
1.4 Practice 3-2 Connecting to the Database Deployment Compute Node	12
1.5 Practice 3-3 Adding Compute Node Users	14
1.6 Practice 3-4 Managing Database Deployment Database Users and Privileges	17
1.7 Practice 3-5 Scaling Up Storage	20
1.8 Practice 3-6 Deleting a Database Deployment	23
1.9 Practice 4-1 Backing Up Database Deployment	25
1.10 Practice 4-2 Patching your Database Deployment	39
1.11 Practice 4-3 Recovering a Database Deployment	41
1.12 Practice 5-1 Opening Ports to Compute Node	42
1.13 Practice 5-2 Using Security Applications	44
1.14 Practice 6-1 Authenticating to Your Compute Node and Database	46
1.15 Practice 6-2 Accessing the Database via DBaaS Monitor Console	51
1.16 Practice 7-1 Protecting Data at Rest Using Encryption	52
1.17 Practice 7-2 Checking Data Protection in Transit	54
1.18 Practice 7-3 Cleaning Up Audit Files	56
1.19 Practice 8-1 Monitoring the Database Deployment	59
1.20 Practice 8-2 Using REST APIs	60
1.21 Practice 8-3 Creating a Linked Clone (Optional)	63
2. Practicing Migration	66
2.1 Practice 2-1 Using Data Pump Conventional Export/Import	67
2.2 Practice 3-1 Unplugging/Plugging a PDB	69
2.3 Practice 5-1 Using RMAN Cross-Platform Transportable Tablespace Backup Sets	72
2.4 Practice 6-1 Using SQL Developer and INSERT Statements to Migrate Selected Objects	75
3. Practicing Performance	77
3.1 Practice 3-1 Scaling Up the Database Deployment	78

# Practicing Database Deployment

- Practice 1-1 Walking Through Oracle Cloud
- Practice 2-1 Creating a Database Deployment
- Practice 3-1 Navigating Through Consoles
- Practice 3-2 Connecting to the Database Deployment Compute Node
- Practice 3-3 Adding Compute Node Users
- Practice 3-4 Managing Database Deployment Database Users and Privileges
- Practice 3-5 Scaling Up Storage
- Practice 3-6 Deleting a Database Deployment
- Practice 4-1 Backing Up Database Deployment
- Practice 4-2 Patching your Database Deployment
- Practice 4-3 Recovering a Database Deployment
- Practice 5-1 Opening Ports to Compute Node
- Practice 5-2 Using Security Applications
- Practice 6-1 Authenticating to Your Compute Node and Database
- Practice 6-2 Accessing the Database via DBaaS Monitor Console
- Practice 7-1 Protecting Data at Rest Using Encryption
- Practice 7-2 Checking Data Protection in Transit
- Practice 7-3 Cleaning Up Audit Files
- Practice 8-1 Monitoring the Database Deployment
- Practice 8-2 Using REST APIs
- Practice 8-3 Creating a Linked Clone (Optional)

## Practice 1-1 Walking Through Oracle Cloud

### Overview

In this practice, you will find the database offerings and for each of them the database cloud pricing table and possible associated options.

### Tasks

1. Use the web address ([URL](#)) provided by the instructor to walk through the Oracle Cloud offerings.
2. Click the Platform tab from the Home page.
3. Click the [Database](#) link under the Data Management heading.
4. Scroll up to the top of the page and click the Pricing tab.

For each offering:

- For Oracle database, you have different Oracle editions provided.
- For Oracle database, you can choose different options and packs associated.
- Billing is per hour or per month.

## Practice 2-1 Creating a Database Deployment

### Overview

In this practice, you will create your own database deployment, considering the following attributes.

- The database deployment must be created with an Oracle Database 12c environment installed with the In-Memory Database option, named `<student_initials>DBCS<no_of_minutes_on_your_clock_right_now>` with a pre-created database named `MYORCL` and a PDB named `MYPDB1`, with the same administrator password for the `SYS` and `SYSTEM` database users (The password is provided by your instructor). For the purpose of ease, the database deployment in all practices will be referred to as `MYDBCS`.
- The service level required is "Oracle Database Cloud Service" and not "Oracle Database Cloud Service - Virtual Image" because you want the database instance be created for you.
- You choose a monthly billing frequency.
- You need **1 CPU** and 15 GB RAM, 30 GB for the databases.
- You do not want to store your database backups on an Oracle Storage Cloud container.

### Assumptions

In the training environment, you have an Oracle Cloud account assigned. All information related to your Oracle Cloud account is provided by the instructor.

Record those items here:

Domain ID: \_\_\_\_\_

User name: \_\_\_\_\_

Password: \_\_\_\_\_

At this point, there is no deployment database.

### Tasks

1. Sign in to Oracle Cloud using a web address ([Oracle Cloud Sign-in](#)) provided by your Oracle Cloud account administrator, the instructor. First enter the identity domain, and then the user name and password. For details on accessing the console, see the documentation on [Accessing the Database Cloud Service Console](#). You access the My Services - Dashboard console.
2. Among the services ('list' in old format - 'tile' in new format), click the "Oracle Database Cloud Service" or "Database".
3. Click the Open Service console.
4. Click Services tab.
5. Click Create Service to start the database deployment process.
6. Then fill in the fields on each page of the database deployment creation until you confirm its creation. Refer to the instructions listed in the overview of the practice and to the summary of the last step to ensure you created the appropriate database deployment.

If you need details on each step of the creation process, follow [Creating a Database Deployment](#).

**Note:** If you want an environment installed with the In-Memory Database option, clicking Details next to

Enterprise Edition in one of the steps displays the Pricing tab to get the information about which Service you will need.

**Note:** If you asked for system generated keys, you may have downloaded the keys. Thus, a zip file named `sshkeybundle.zip` containing the `publicKey` and `privateKey` files in `open_ssh` format is generated, both files respectively containing the public key and private key. For security reasons, move the zip file to `~/ .ssh`, and then unzip. Change the permissions on the `privateKey` file to owner access only with `chmod 600 privateKey`.

If you plan to use PuTTY to connect to your compute node, use PuTTY Key Generator to load the system generated private key file and save the file in PuTTY's own format.


**Note:** If you prefer to create the key pair manually, create the secure shell (SSH) private/public key pair before step 4, with copies of the private and public key files on your local computer. You will point to the SSH public key when you create the database deployment.

For details on generating an SSH private/public key pair, refer to the [Generating a Secure Shell \(SSH\) Private/Public Key Pair](#) section of the Using Oracle Database Cloud Service (Database as a Service) documentation.


**Note:** If the key files are generated under Windows, the last step of the documentation explanation suggests that you export the private key if you or others are going to use an SSH client that requires the OpenSSH format for private keys (such as the `ssh` utility on Linux).

If you paste in the public key value, make sure the value does not contain line breaks or end with a line break.

7. Before confirming the database deployment, you should get the following summary:
  - a. Usable Database Storage set to: 25GB
  - b. **Important** Record the Administrative password: \_\_\_\_\_




**Service Level:** Oracle Database Cloud Service  
**Billing Frequency:** Monthly  
**Software Release:** Oracle Database 12c Release 1  
**Software Edition:** Enterprise Edition - Extreme Performance  
**Service Name:** MYDBCS  
**Description:**  
**Shape:** OC1M - 1 OCPU, 15 GB RAM  
**Timezone:** (UTC) Coordinated Universal Time(UTC)  
**Key:** ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCyI5JQKYp...  
**Usable Database Storage:** 50  
**Total Data File Storage:** 116  
**DB Name (SID):** MYORCL  
**PDB Name:** MYPDB1  
**Character Set:** AL32UTF8 - Unicode Universal character set UTF-8 form 32-bit  
**National Character Set:** AL16UTF16 - Unicode UTF-16 Universal character set  
**Standby Database with Data Guard:** No  
**Disaster Recovery:** No  
**Include "Demos" PDB:** No  
**Include GoldenGate:** No  
**Database Clustering with RAC:** No




**Backup Destination:** None

The example shows the summary of a database deployment creation using the SSH key pair wizard. The key pair was not created before the service creation. In this case, the public key value appears partially in the summary.






**Service Level:** Oracle Database Cloud Service  
**Billing Frequency:** Monthly  
**Software Release:** Oracle Database 12c Release 1  
**Software Edition:** Enterprise Edition - Extreme Performance  
**Service Name:** MYDBCS  
**Description:**  
**Shape:** OC1M - 1 OCPU, 15 GB RAM  
**Timezone:** (UTC) Coordinated Universal Time(UTC)  
**Key:** DJ.pub  
**Usable Database Storage:** 50  
**Total Data File Storage:** 116  
**DB Name (SID):** MYORCL  
**PDB Name:** MYPDB1  
**Character Set:** AL32UTF8 - Unicode Universal character set UTF-8 form 32-bit  
**National Character Set:** AL16UTF16 - Unicode UTF-16 Universal character set  
**Standby Database with Data Guard:** No  
**Disaster Recovery:** No  
**Include "Demos" PDB:** No  
**Include GoldenGate:** No  
**Database Clustering with RAC:** No  
 **Backup Destination:** None

The example shows the summary of a database deployment creation for which the key pair was manually created before the service creation. In this case, you named the files containing the values of the keys. The summary displays the name of the file containing the public key uploaded to the compute node of the database deployment.

**Remark :** Because Oracle Cloud is constantly evolving, new fields may appear that are not visible in the summary above.

The database deployment is in progress and then completed:



**MYDBCS**  
**Status:** In Progress  
**Version:** 12.1.0.2  
**Edition:** Enterprise Edition - Extreme Performance  
**Submitted On:** May 25, 2016 11:05:02 AM UTC  
**OCPU:** 1  
**Memory:** 15 GB  
**Storage:**

## Practice 3-1 Navigating Through Consoles

### Overview

In this practice, you will familiarize yourself with the Cloud consoles.

### Tasks

1. Access the My Services - Dashboard console.
2. Among the list of services, set the Oracle Database Cloud Service, the Oracle Database Backup Service, and the Oracle Compute Cloud Service as your favorite subscriptions by clicking the star icon on each service.

Read [Exploring the My Services Dashboard](#) to understand all the details provided by the console.

3. View details about the database deployments in the domain. Go to the Oracle Database Cloud Service.

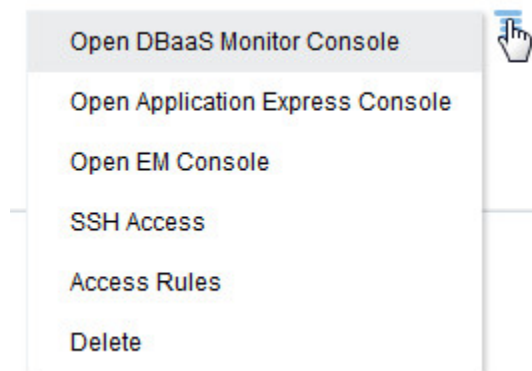
For details on accessing the console, see the documentation on [Accessing the Database Cloud Service Console](#).

For all details provided, read [Oracle Database Cloud Service Services Page](#).

From the



menu icon of your deployment database name, you get the list of consoles that you can open. You also get the Delete option to delete your deployment database.



4. View details about your own deployment database. Among the list of deployment database services, click your deployment database name. Click the "show more" link in the brackets adjacent to the Additional Information heading to get more details about the database deployment.

Which important information do you get now that was not displayed in the previous step?

- The identity domain name
- The public IP address
- The Oracle database instance name
- The Oracle PDB name if the Oracle database is a CDB. If the Oracle database were a non-CDB, it would display the Oracle database name.
- The SQL\*Net port used to access the Oracle CDB and PDB
- The full connect string to the Oracle database instance
- The subscription type (monthly or hourly)

5. Access the Compute Cloud Service console. Click Dashboard to get back to the My Services - Dashboard console.

Then among the services displayed, select the Oracle Compute Cloud Service

Read [Accessing Oracle Compute Cloud Service Using the Web Console](#).

**Question:** Which actions can you complete with the console?

**Answer:**

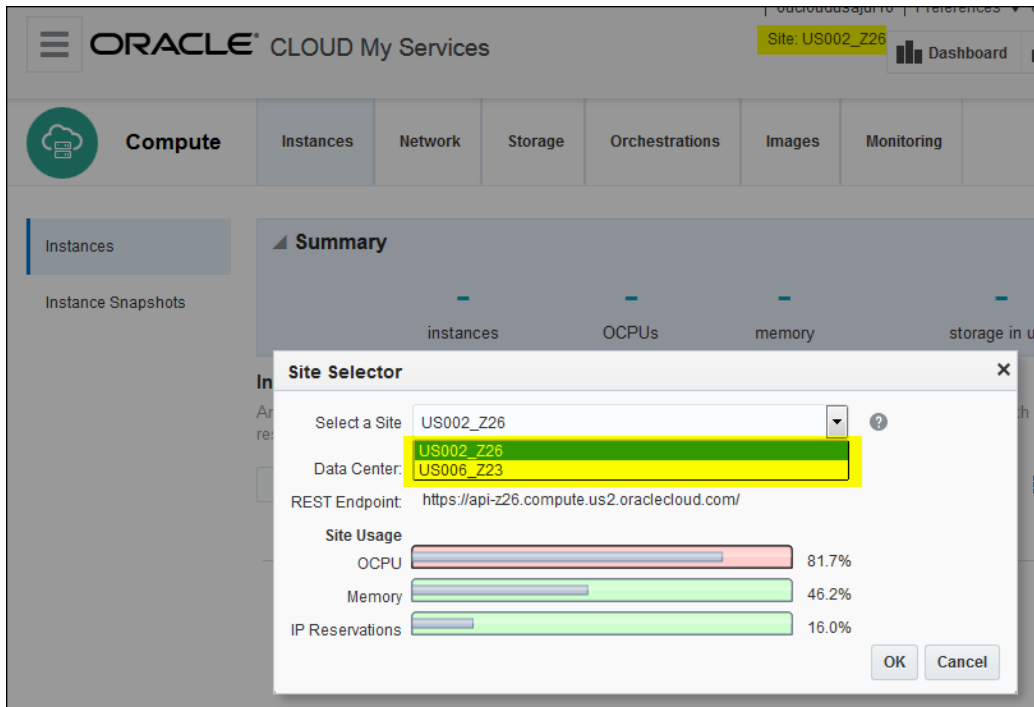
- Rapidly provision virtual machines on Oracle Cloud with all the necessary storage and networking resources
- Manage and scale your virtual machine topology in the cloud easily

6. Click Open Service Console.

**Question:** Which information do you get now that was not displayed in previous steps?

**Answer:** The private IP address.

**Note:** You may not see the information related to all instances, network, storage and other metadata in tabs. This depends on the site selector visible in the upper right of the compute console.



**Question2:** What do you get when you click the Storage tab?

**Answer2:** The five storage volumes attached to your deployment database:

- boot for the boot image of the VM
- bits for the files of the VM
- data for the data files of the Oracle database
- fra for the flash recovery area of the Oracle database
- redo for the redo log files of the Oracle database

**Question3:** Which action can you perform from the Storage tab in the Compute Cloud Service console?

**Answer3:**

- View details about each storage volume: click a storage volume to get the size of a particular volume.
- Create a new storage volume
- Detach a storage volume that you would have added

## Practice 3-2 Connecting to the Database Deployment Compute Node

### Overview

In this practice, you will configure a connection each for the `opc` user and the `oracle` user.

### Setup

If you are working in an Oracle Classroom, certain modifications must be made.

Edit the file `~/.ssh/config` to reflect the IP address of DB instance from OPC:

```
Host <IP address of DB instance from OPC>
ProxyCommand nc -X connect -x ges-proxy.us.oracle.com:80 %h %p
GSSAPIAuthentication no
```

### Tasks

1. Retrieve the IP Address of the compute node assigned to your database deployment. Read [Viewing Detailed Information for a Database Deployment](#).

If you signed out, sign in to Oracle Cloud using your Cloud user account. First enter the identity domain, and then the user name and password.

You access the My Services - Dashboard console.

2. Among the list of services, click the Oracle Database Cloud Service.
3. Click the Open Service Console.
4. Click the name of your database deployment. The public IP Address is displayed.

If you are using Windows, use PuTTY to create a new session where the host name is the public IP Address of the compute node, the Connection Data uses `opc` as the Auto-Login username, and the Connection SSH Auth uses the private key file generated in the first practice (Practice Creating a Database Deployment). Read [Connecting to a Compute Node Using the PuTTY Program on Windows](#).

If you are using Linux, use `ssh` to connect to the database deployment compute node. Read [Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms](#).

This operation opens the session to the compute node. You are now connected to the compute node as `opc`.

```
$ ssh -i your_private_key_file opc@your_compute_node_IP_Address
[opc@MYDBCS ~]$
```

**Question:** Did the `opc` user provide any password to log in to the compute node?

**Answer:** No. The `opc` user authentication is completed with the SSH private/public keys, the SSH private key file that pairs with the public key used during the database deployment creation process.

5. Perform the same operation to create a new session as `oracle` to the compute node. You are now connected to the compute node

```
as oracle.
```

```
[opc@MYDBCS ~]$ exit
```

```
$ ssh -i your_private_key_file oracle@your_compute_node_IP_Address
```

```
[oracle@MYDBCS ~]$
```

## Practice 3-3 Adding Compute Node Users

### Overview

When the database deployment was created, three Linux users were created on the compute node associated with database deployment: `opc`, `oracle`, and `root`.

- `oracle` authorized to log in to the database deployment compute node but not authorized to run root commands
- `opc` authorized to log in to the database deployment compute node and to run root commands
- `root` not authorized to log in to the database deployment compute node

In this practice, you will create an additional compute node user to your database deployment who will be able to perform standard OS operations, such as installing and running applications. The user will be authorized to log in to the database deployment compute node but not authorized to run root commands. You will see how to change the permissions of the new user to allow him to run root commands. The compute node user is called `user1`.

### Tasks

1. Open the session to the compute node as `opc`. You are now connected to the compute node.

```
[opc@MYDBCS ~]$
```

2. Create the new Linux user `user1` with his home directory `/home/user1/.ssh`.

```
[opc@MYDBCS opc]$ sudo -s
[root@MYDBCS opc]# useradd user1
[root@MYDBCS opc]# mkdir /home/user1/.ssh
```

3. Copy the SSH public key value to the authorization file.
  - a. Open a text editor.
  - b. Open the local SSH public key file.
  - c. Copy the value to the new user's `/home/user1/.ssh/authorized_keys` file.

```
[root@MYDBCS opc]# echo "ssh-rsa AAAAB3NzaC1..." > /home/user1/.ssh/authorized_keys
```

4. Edit the `/etc/ssh/sshd_config` file. Find the `AllowUsers` line and add the user `user1` name.
5. Set the ownership of the new user's home directory files and appropriate permission on the `.ssh` directory.

```
[root@MYDBCS opc]# chown -R user1:user1 /home/user1/.ssh
[root@MYDBCS opc]# chmod -R 700 /home/user1/.ssh
```

6. Restart the SSH daemon on your instance and exit the session.

```
[root@MYDBCS opc]# /sbin/service sshd restart
[root@MYDBCS opc]# exit
[opc@MYDBCS ~] $ exit
```

7. Test if you can log in as `user1`.

- a. If you are using Windows, use PuTTY to create a new session where the host name is the IP Address of the compute node, the Connection Data uses `user1` as the Auto-Login username, and the Connection SSH Auth uses the private key file generated in the first practice (Practice Creating a Database Deployment). Read [Connecting to a Compute Node Using the PuTTY Program on Windows](#).
- b. If you are using Linux, use `ssh` to configure connections to the database deployment compute node. Read [Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms](#).

8. Open the session to the compute node. You are now connected to the compute node.

```
[user1@MYDBCS ~]$
```

9. If you are using Windows, save the session as `Cloud_user1`.

**Question:** Can `user1` perform `opc` operations?

**Answer:** No. See the `opc` attempt.

```
[user1@MYDBCS ~]$ sudo -s
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for user1:
```

10. To allow the new user to run root commands, relog in as `opc` to edit the `/etc/sudoers` file with `visudo` to add a new line as follows:

```
%user1    ALL=(ALL)  NOPASSWD: ALL
```

**Note:** Names beginning with a "%" indicate group names. The `user1` group can execute any command as any user on any host. `NOPASSWD` is a "tag" that means no password will be requested. If `user1` has a different group removing the % allows the user `user1` to use `sudo`.

11. Reconnect as the new user and verify that `user1` can run root commands.

```
[user1@MYDBCS ~]$
```

**Question:** Can `user1` perform `opc` operations?

**Answer:** Yes.

```
$ sudo -s
```

```
[root@MYDBCS user1]#
```

**Question2:** Do these changes made to your database deployment persist?

**Answer2:** *If for any reason, the compute node is automatically reprovisioned by Oracle, then any changes made on that compute node will be lost.*



## Practice 3-4 Managing Database Deployment Database Users and Privileges

### Overview

In this practice, you will connect to the compute node of your MYDBCS database deployment, check that the pre-created Oracle database instance MYORCL is running and that the pre-created Oracle database holds a PDB named MYPDB1. Connect to the Oracle database instance as the SYS user and check the pre-created users.

### Tasks

1. Using either PuTTY or `ssh`, log on to the compute node of your database deployment MYDBCS using the `Cloud_oracle` session.
2. Check that the pre-created Oracle database instance MYORCL is running.

```
[oracle@MYDBCS ~]$ pgrep -lf smon
```

```
23741 ora_smon_MYORCL
```

```
[oracle@MYDBCS ~]$
```

3. Connect to the Oracle database instance as SYSDBA to verify that the pre-created Oracle database holds a PDB named MYPDB1.

```
[oracle@MYDBCS ~]$ sqlplus / AS SYSDBA
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 31 10:35:55 2016
```

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

```
Connected to:
```

```
Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0 - 64bit Production
```

```
With the Partitioning, Oracle Label Security, OLAP, Advanced Analytics  
and Real Application Testing options
```

```
Connected.
```

```
SQL> SHOW pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	MYPDB1	READ WRITE	NO

```
SQL> EXIT
```

```
Disconnected from Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0
```

- 64bit Production

With the Partitioning, Oracle Label Security, OLAP, Advanced Analytics  
and Real Application Testing options

```
[oracle@MYDBCS ~]$
```

**Question:** What is different in the banner of the database deployment database from an on-premises database instance?

**Answer:** The banner displays the Cloud Edition defined at the database deployment creation: Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0

4. Connect to MYPDB1 and find the users that do not exist in an on-premises database.

```
[oracle@MYDBCS ~]$ sqlplus system@MYPDB1
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 31 10:38:38 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
Enter password:
```

```
Last Successful login time: Tue May 31 2016 10:48:00 +00:00
```

```
Connected to:
```

```
Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0 - 64bit Production
With the Partitioning, Oracle Label Security, OLAP, Advanced Analytics
and Real Application Testing options
```

```
SQL>
```

**Question:** Which password did you use to connect to the PDB?

**Answer:** The password defined during the database deployment creation.

```
SQL> COL username format A40
```

```
SQL> SELECT username, con_id FROM cdb_users ORDER BY 1;
```

USERNAME	CON_ID
ANONYMOUS	3
APEX_050000	3

```

APEX_LISTENER                                3
APEX_PUBLIC_USER                             3
APEX_REST_PUBLIC_USER                        3
...
C##DBAAS_BACKUP                              3
C##DBAAS_MONITOR                             3
...
SCOTT                                         3
..
43 rows selected.

```

SQL>

The users that do not exist in an on-premises database are APEX\_XXXX users, and C##DBAAS\_BACKUP and C##DBAAS\_MONITOR users.

- A database deployment includes Oracle Application Express, which you manage using the Oracle Application Express administration console. If you are new to Oracle Application Express, see its [Overview](#) and [Getting Started](#) pages on Oracle Technology Network to learn about its features and get started using it.
- C##DBAAS\_BACKUP is the common user updating the status in a database table that is then used for showing Automated backup status on UI and to send messages to user when Automated backups fail.
- C##DBAAS\_MONITOR is the common user used from Database Cloud Service monitoring application, which users can use to monitor and manage the database.

**Question:** Will users, privileges and roles management be different in the pre-created database than in an on-premises database?

**Answer:** The common users and common roles are created in the CDB root the same way as it is in an on-premises CDB root and the local users and local roles are created in the `MY_PDB1` PDB the same way as it is in any on-premises PDB. Privileges are granted commonly or locally in the CDB root or PDB the same way as it is in an on-premises CDB root or PDB.

## Practice 3-5 Scaling Up Storage

### Overview

In this practice, you will scale up the storage capacity of the pre-created Oracle database instance `MYORCL`. The CDB should get 10 GB more to store application data.

### Tasks

1. Open the Oracle Database Cloud Service console.
2. Click the `MYDBCS` link.

From the



menu icon, select Scale Up/Down.

In the dialog set Additional Storage to 10GB

Select an option from the Add Storage To pulldown:

**Question :** Which storage scale up options do you see?

**Answer :**

**Create New Storage Volume :** adds a new storage volume to the database deployment and mounts it as the next available `/u0n` mount point.

**Extend Data Storage Volume :** adds the storage volume to the existing Linux LVM disk group (or Oracle ASM disk group on deployments that use Oracle Real Application Clusters) for database data storage.

**Extend Backup Storage Volume :** adds the storage volume to the existing Linux LVM disk group (or Oracle ASM disk group on deployments that use Oracle Real Application Clusters) for backup and FRA storage.


Read [Scaling Up the Storage for a Database Deployment](#) .

**Question2 :** Which storage scale up option will you choose to increase the application storage volume for the database?

**Answer 2 :** Extend Data Storage Volume is the appropriate solution.

The scaling operation begins. The deployment is in Maintenance status and unavailable while the scaling operation is in progress. The deployment is shut down and rebooted during this operation.

### Nodes





**MYDBCS**  
Public IP: 129.144.21.179

SQL \*Net Port: 1521  
SID: MYORCL  
PDB Name: MYPDB1

OCPUs: 1  
Memory: 15 GB  
Storage: 127 GB

### Activity

Activity Summary	
 <b>Scale Up/Down Scheduled</b>	<b>Start Time:</b> May 31, 2016 4:33:49 PM UTC <b>End Time:</b>



**MYDBCS**  
**Status:** Maintenance  
Version: 12.1.0.2  
Edition: Enterprise Edition - Extreme Performance

Submitted On: May 25, 2016 11:05:02 AM UTC

OCPUs: 1  
Memory: 15 GB  
Storage: 127 GB

When the operation is completed, the maintenance status period should display:

#### Overview

1  
Node

Last scale up/down succeeded

#### Administration

0  
Patches available

**Nodes**

**OCPUs**

**Memory**

**Storage**


1

1

15 GB

137 GB

### Nodes



**MYDBCS**  
Public IP: 129.144.21.179

SQL \*Net Port: 1521  
SID: MYORCL  
PDB Name: MYPDB1


OCPUs: 1  
Memory: 15 GB  
**Storage: 137 GB**

**Question3 :** What do you observe?

**Answer3 :** The storage capacity increased since the operation completed.

1. Show the Activity region.

### Activity







Activity Summary	
 <b>Scale Up/Down Completed</b>	<b>Start Time:</b> May 31, 2016 4:33:49 PM UTC <b>End Time:</b> May 31, 2016 4:55:59 PM UTC

2. Switch to the Oracle Compute Cloud Service console to view details about the storage extension of your database deployment.
3. Click the Open Service console.

4. Click View from the



menu icon for your database deployment. The storage volume added is 10 GB.

Storage Volumes				
	Name	Description	Status	Size
	MYDBCS/db_1/vm-1/boot	boot	Online	30 GB
	MYDBCS/db_1/vm-1/bits	bits	Online	30 GB
	MYDBCS/db_1/vm-1/fra	fra	Online	7 GB
	MYDBCS/db_1/vm-1/redo	redo	Online	10 GB
	MYDBCS/db_1/vm-1/data	data	Online	50 GB
	MYDBCS/db_1/vm-1/add_s...		Online	10 GB

**Remark:**

Observe that the Oracle Compute Cloud console allows you to create a storage volume with the Create Storage Volume button. The created volume is not attached to a specific database deployment. You must then attach the storage volume to a database deployment and then mount the storage volume on the database deployment.

Read both topics from [Attaching a Storage Volume to an Instance](#).

## Practice 3-6 Deleting a Database Deployment

### Overview

In this practice, you will delete your own database deployment because you realize that you forgot to ask for automatic backup.

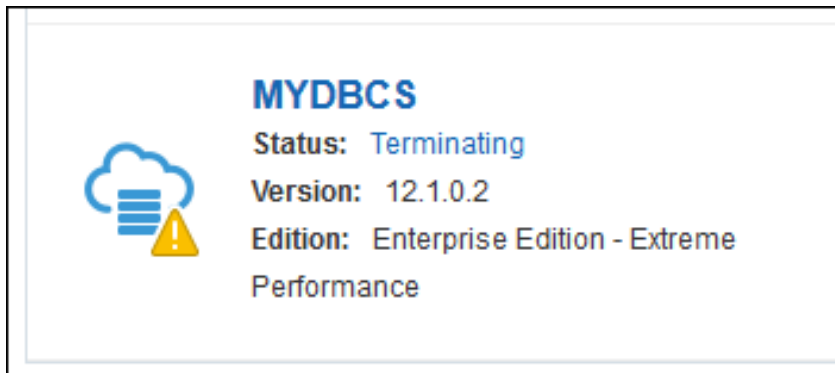
### Tasks

1. From the Oracle Database Cloud Service console, select the Delete option from the

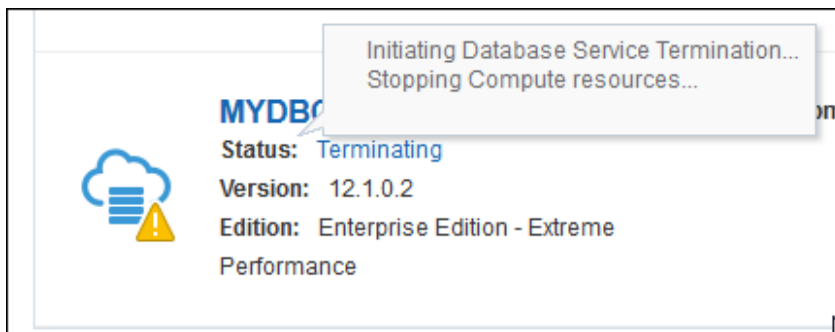


menu icon next to your database deployment.

2. Confirm the service deletion.




3. Click the Terminating Status to observe the database deployment deletion progress.



4. From the Oracle Compute Cloud Service, you can see that the storage volumes are already detached from the deleted database deployment.
5. Back to the Oracle Database Cloud Service console, you can see that the database deployment does not show up anymore and the delete history thumbnail displays the name of the database deployment deleted.

**Instance create and delete history**

Range Last 24 hours ☐ Show only failed attempts

 MYDBCS

**Delete Succeeded On:** Jun 1, 2016 4:21:51 PM UTC  
[Details](#)



## Practice 4-1 Backing Up Database Deployment

### Overview

In this practice, you will re-create your database deployment as you did in Practice 2-1 *Creating a Database Deployment*, with the same characteristics except that the Oracle Database 12c environment installed does not require the In-Memory Database option and will use a smaller memory footprint like 7.5 GB and only 25 GB of storage for the database (this is the default).

This time, you will ensure that backups are created automatically and stored on local storage on the compute nodes associated with the database deployment.

### Tasks

From the Oracle Database Cloud Service, click the Open Service console.

#### First part:

1. Click Create Service to start the database deployment process.
2. Then fill the fields on each page of the database deployment creation until you need to fill up the *Backup and Recovery Configuration* section.

**Question:** Why, in the Backup and Recovery Configuration section, don't you find the "Local Storage Only" destination?

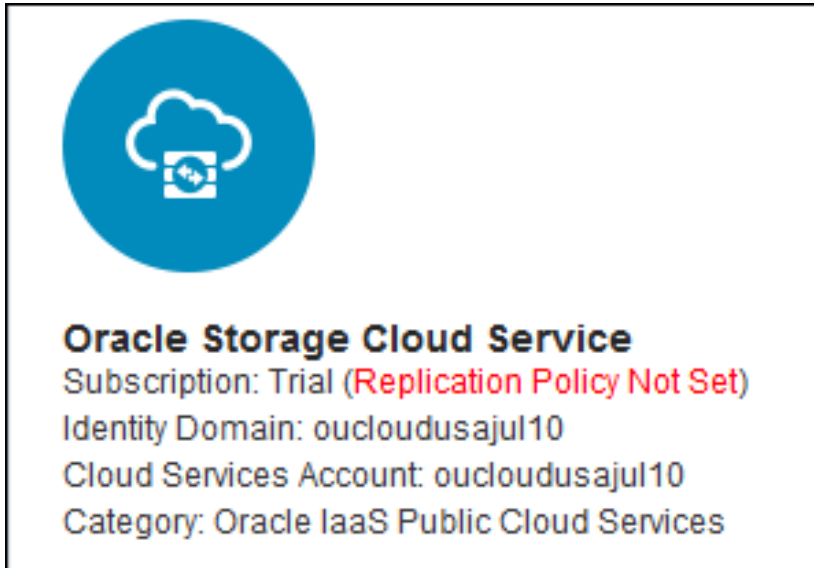
**Answer:** This destination is not available when you use the Create Database Cloud Service wizard to create a database deployment. To create a deployment that uses the Local Storage Only destination, you must use the `oracle-dbcs-cli` utility. Read [The Data File for the oracle-dbcs-cli create Subcommand](#).

3. You ask for backups created automatically and stored both on local compute node storage and on an Oracle Storage Cloud Service container.

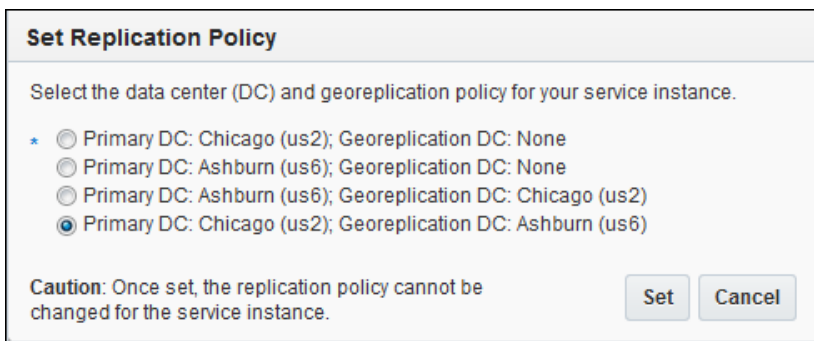
**Question:** What is requested when you ask for "Both Cloud Storage and Local Storage" destination?

**Answer:** This destination requires a Cloud Storage Container for the backups.

Check first in the Storage Cloud Service detail page, You may see the red message: Replication Policy Not Set. This message appears when your domain spans multiple sites (Check in the Compute Service console). For the details of replication see [Selecting a Replication Policy for Oracle Storage Cloud Service](#)



If this is the case, set the replication. If this is not the case, go to the next step. For this practice, Replication may be set to None.



## Second part:

1. Create the Cloud storage container
    - The Cloud Storage container can be the name of an existing Oracle Storage Cloud Service container.
    - If not already created, you can easily create the container by selecting the "Create Cloud Storage Container" checkbox provided during the database deployment process in the "Backup and Recovery Configuration" section, or use the manual `curl` command explained below.
      - **If you create the container by selecting the "Create Cloud Storage Container" checkbox** provided during the database deployment process in the "Backup and Recovery Configuration" section, provide the following information:
        - The storage container name structured as `https|http://<datacenter_code>/<version>/<schema name>/<container name>`.
- To find the `<datacenter_code>/<version>/<schema name>`, go to the Oracle Database Backup Cloud service and retrieve the REST endpoint field value.

**Oracle Database Backup Service****Additional Information**

<b>Plan:</b> <a href="#">Oracle Database Backup Service</a>	<b>CSI Number:</b> Not available
<b>Service Start Date:</b> 30-Aug-2016	<b>Data Center:</b> EMEA Commercial 2 - Amsterdam
<b>Service End Date:</b> 30-Aug-2017	<b>Version:</b> 16.1.0.0.0
<b>Subscription ID:</b> 540449037	<b>Status:</b> Active
<b>Service Instance ID:</b> 540449183	<b>REST Endpoint:</b> <a href="https://em2.storage.oraclecloud.com/v1/Storage-ouopc021/JLScontainer">https://em2.storage.oraclecloud.com/v1/Storage-ouopc021/JLScontainer</a>
<b>Customer Account:</b> OU La France (FR)	

Append to this value the name of the container you want to create for your backups.

**Warning** : Ensure you provide the right `datacenter_code`. If not, the database deployment gets created but with no storage container available for your backups. Any further backup will fail.

Example: If the REST end point is `https://em2.storage.oraclecloud.com` and if I want to create `JLScontainer`, in the identity domain `ouopc021`. I enter the following value:  
`https://em2.storage.oraclecloud.com/v1/Storage-ouopc021/JLScontainer`

- The name of the owner of the container : the user name of your Cloud account
  - The password of the owner of the container: the password of your Cloud account
- **If you manually create the storage container using the `curl` commands**, follow the steps below.  
 On Linux, the `curl` command is provided. On Windows, download the `curl` executable and use it from the DOS Command Prompt.  
 Read [Creating Containers](#) documentation and more precisely for the practice [Oracle Storage Cloud Service: Creating Containers Using the REST API](#).
- a. If your client is Linux, you can use the following interactive script to create the storage container (this is the easiest way):

```
#!/bin/bash
## DESCRIPTION: Oracle Storage Cloud Service: Creating
Containers Using the REST API
## DOCUMENTATION: http://www.oracle.com/webfolder/technetwo
rk/tutorials/obe/cloud/objectstorage/creating\_containers\_RE
ST\_API/creating\_containers\_REST\_API.html

## KNOWN ISSUES: Does not work with the following identity
domains:
## - oucloudusa8
## AUTHOR: Sebastien Colas (sebastien.colas@oracle.com)

echo -n 'Your identity domain :'
read identity
echo -n 'Your username :'
read username
echo -n 'Your password :'
read password
echo -n 'Container name you want to create :'
read container
echo -n 'Your REST Endpoint domain :'
read RESTEndpoint
```

```

echo

curl -v -X GET -H "X-Storage-User:
Storage- $\{identity\}$ : $\{username\}$ " -H "X-Storage-Pass:
 $\{password\}$ " https:// $\{RESTEndpoint\}$ .storage.oraclecloud.co
m/auth/v1.0 2> auth.txt

storage=`cat auth.txt | grep X-Storage-Url auth.txt | cut
-f 3 -d ' ' | col -bp`
auth=`cat auth.txt | grep X-Auth-Token | cut -f 3 -d ' '`

echo "X-Storage-Url :  $\{storage\}$ "
echo "X-Auth-Token :  $\{auth\}$ "

curl -v -X PUT -H "X-Auth-Token:  $\{auth\}$ "
 $\{storage\}$ / $\{container\}$  2> create.txt
ret=`cat create.txt | grep 'HTTP/1.1 201 Created'`

echo

if [ -n "$ret" ]
then
    echo "Container  $\{container\}$  successfully created"
    echo "Cloud Storage Container:  $\{container\}$ "
else
    echo "an error occurred please see the log files"
fi

```

**or**

You can manually complete the following steps:

- i. Oracle Storage Cloud Service requires authentication for any operation against the service instance. Request an authentication token by running the following `curl` command where you changed the `myIdentityDomain`, the name, the password:

```

curl -v -k -X GET -H "X-Storage-User: Storage-myIdentityDomain:jack.jones@example.com" -H "X-Storage-Pass: ft7)Dvjo" https://<RESTEndpoint>.storage.oraclecloud.com/auth/v1.0

```

```

[djeunot@myserver ~]$ curl -v -k -X GET -H
"X-Storage-User: Storage-< myIdentityDomain > :<username>" -H "X-Storage-Pass: <password>" https:// < RESTEndpoint> .storage.oraclecloud.com/auth/v1.0

```

\* About to connect() to < RESTEndpoint> .storage.orac

```

lecloud.com port 443 (#0)
* Trying 100.100.100.1... connected
* Connected to < RESTEndpoint> .storage.oraclecloud.com (129.152.172.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CPath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*   subject: CN=*.storage.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*   start date: Aug 07 00:00:00 2015 GMT
*   expire date: Oct 05 23:59:59 2016 GMT
* common name: *.storage.oraclecloud.com
*   issuer: CN=VeriSign Class 3 Secure Server CA - G3,OU=Terms of use at https://www.verisign.com/rpa(c)10,OU=VeriSign Trust Network,O="VeriSign, Inc.",C=US
> GET /auth/v1.0 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: < RESTEndpoint> .storage.oraclecloud.com
> Accept: */*
> X-Storage-User: Storage- < myIdentityDomain> :<user name>
>
< HTTP/1.1 200 OK
< date: 1464968827818
< X-Auth-Token :
AUTH_tk08670719c5fd4a0216f11522fb44dd17
< X-Storage-Token:
AUTH_tk08670719c5fd4a0216f11522fb44dd17
< X-Storage-Url : https://storage.us2.oraclecloud.com/v1/Storage- < myIdentityDomain>
< Content-Length: 0
< Server: Oracle-Storage-Cloud-Service
<
* Connection #0 to host < RESTEndpoint> .storage.oraclecloud.com left intact
* Closing connection #0
[djeunot@myserver ~]$

```

- b. Retrieve the values of the X-Storage-Url header and the X-Auth-Token header. Note that the authentication token expires after 30 minutes, after which

you should request a fresh token.

- c. Create the container named `<student_initials>Container<nb_of_seconds_on_your_clock_right_now>` .

```
curl -v -X PUT -H "X-Auth-Token: X-Auth-Token_value " X-Storage-Url_value/ <student_initials>Container<nb_of_seconds_on_your_clock_right_now>
```

```
[djeunot@ myserver ~]$ curl -v -X PUT -H "X-Auth-Token: AUTH_tk08670719c5fd4a0216f11522fb44dd17" https://storage.us2.oraclecloud.com/v1/Storage- < myIdentityDomain> /<studentContainer>
```

```
* getaddrinfo(3) failed for X-Storage-Url:80
* Couldn't resolve host 'X-Storage-Url'
* Closing connection #0
* About to connect() to storage.us2.oraclecloud.com port 443 (#0)
*   Trying 100.100.100.3... connected
* Connected to storage.us2.oraclecloud.com (129.152.172.3) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*   subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*   start date: Dec 08 00:00:00 2015 GMT
*   expire date: Jan 06 23:59:59 2017 GMT
*   common name: *.us2.oraclecloud.com
*   issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust Network,O=Symantec Corporation,C=US
> PUT /v1/Storage- < myIdentityDomain> / <studentContainer> HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu)
libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage.us2.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk08670719c5fd4a0216f11522fb44dd17
>
< HTTP/1.1 201 Created
< X-Trans-Id: txc729f6b2fd074b699b209-005751a875ga
< Date: Fri, 03 Jun 2016 15:55:33 GMT
< Connection: keep-alive
```

```
< X-Last-Modified-Timestamp: 1464969333.50324
< Content-Type: text/html; charset=UTF-8
< Content-Length: 0
< Server: Oracle-Storage-Cloud-Service
<
* Connection #0 to host storage.us2.oraclecloud.com left
intact
* Closing connection #0
[djeunot@myserver ~]
```

- Verify the existence of the container. Use the interactive script:



```
#!/bin/bash

## DESCRIPTION: Oracle Storage Cloud Service: List Containers Using the REST API
## DOCUMENTATION: http://docs.oracle.com/cloud/latest/storagecs\_common/SSAPI/Status%20Codes.html

## KNOWN ISSUES: Does not work with the following identity domains:
## - oucloudusa8
## AUTHOR: Sebastien Colas (sebastien.colas@oracle.com)

echo -n 'Your identity domain : '
read identity
echo -n 'Your username : '
read username
echo -n 'Your password : '
read password
echo -n 'Your REST Endpoint domain : '
read RESTEndpoint

echo

curl -v -X GET -H "X-Storage-User: Storage-${identity}:${username}" -H
"X-Storage-Pass: ${password}" https://${RESTEndpoint}.storage.oraclecloud.com/auth/v
1.0 2> auth.txt

storage=`cat auth.txt | grep X-Storage-Url auth.txt | cut -f 3 -d ' ' | col -bp`
auth=`cat auth.txt | grep X-Auth-Token | cut -f 3 -d ' '`

echo "X-Storage-Url : ${storage}"
echo "X-Auth-Token : ${auth}"
echo

curl -v -X GET -H "X-Auth-Token: ${auth}" ${storage} > containers.txt 2> list.txt

echo "Your Storage containers :"
cat containers.txt ;echo
```

o r


Use the following manual curl command:

```
curl -v -X GET -H "X-Auth-Token: X-Auth-Token_value" X-Storage-Url_value/<student_in
itials>Container<nb_of_seconds_on_your_clock_right_now>
```


```
[djeunot@myserver ~]$ curl -v -X GET -H "X-Auth-Token: AUTH_tk08670719c5fd4a0216f115
22fb44dd17" https://storage.us2.oraclecloud.com/v1/Storage- < myIdentityDomain> / <s
tudentContainer>
* About to connect() to storage.us2.oraclecloud.com port 443 (#0)
*   Trying 100.100.100.3... connected
* Connected to storage.us2.oraclecloud.com (129.152.172.3) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
*   CAspath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Dec 08 00:00:00 2015 GMT
*       expire date: Jan 06 23:59:59 2017 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> GET /v1/Storage- < myIdentityDomain> / <studentContainer> HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic
ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: storage.us2.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk08670719c5fd4a0216f11522fb44dd17
>
< HTTP/1.1 204 No Content
< X-Container-Object-Count: 0
< X-Container-Write: dbpml8.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1464969333.52014
< X-Container-Read: < myIdentityDomain> .Storage.Storage_ReadOnlyGroup,dbpml8.Storag
e.Storage_ReadWriteGroup
< X-Container-Bytes-Used: 0
< X-Trans-Id: tx4292e700f34e4cea98619-005751a899ga
< Date: Fri, 03 Jun 2016 15:56:09 GMT
< Connection: keep-alive
< X-Storage-Class: Standard
< X-Container-Meta-Policy-Georeplication: us2
< X-Last-Modified-Timestamp: 1464969333.50324
< Content-Type: text/html; charset=UTF-8
< Server: Oracle-Storage-Cloud-Service
<
* Connection #0 to host storage.us2.oraclecloud.com left intact
* Closing connection #0
```

```
[djeunot@myserver~]$
```

- Before confirming the database deployment, you should get the following summary:



**Service Level:** Oracle Database Cloud Service  
**Billing Frequency:** Monthly  
**Software Release:** Oracle Database 12c Release 1  
**Software Edition:** Enterprise Edition  
**Service Name:** MYDBCS  
**Description:**  
**Shape:** OC3 - 1 OCPU, 7.5 GB RAM  
**Timezone:** (UTC) Coordinated Universal Time(UTC)  
**Key:** ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCrqPIOMY9...  
**Usable Database Storage:** 25  
**Total Data File Storage:** 88.5  
**DB Name (SID):** MYORCL  
**PDB Name:** MYPDB1  
**Character Set:** AL32UTF8 - Unicode Universal character set UTF-8 form 32-bit  
**National Character Set:** AL16UTF16 - Unicode UTF-16 Universal character set  
**Standby Database with Data Guard:** No  
**Disaster Recovery:** No  
**Include "Demos" PDB:** No  
**Include GoldenGate:** No

**Backup Destination:** Both Cloud Storage and Local Storage  
**Username:** dominique.jeunot@oracle.com  
**Cloud Storage** **Container:** https://dbtestcs4.storage.oraclecloud.com/v1/Storage-dbtestcs4/myDJcontainer

- Once the database deployment is complete , perform an on-demand backup by using the Cloud Database Service console. Read [Creating an On-Demand Backup by Using the Oracle Database Cloud Service Console](#) .
- Log in to the database deployment compute node as `oracle` to verify the backup.

- Display the content of the database files configuration file (`/home/oracle/bkup/<DBNAME>/dbcfg.spec` ) and the system files configuration file ( `/home/oracle/bkup/ <DBNAME> /oscfg.spec` ) to know which files are automatically backed up.
- Check that the database files have been backed up using RMAN.  

```
RMAN> list backup;
```

**Question:** Have the files mentioned in the configuration files (`/home/oracle/bkup/<DBNAME>/dbcfg.spec` and `/home/oracle/bkup/ <DBNAME> /oscfg.spec` ) been backed up during the on-demand backup ?

**Answer:** No. These files are backed up during the automated scheduled backups defined in the

*crontab. Because the automated backup are scheduled to run every night, files mentioned in the configuration files will be backed up during the next night. Verify tomorrow.*

- Tomorrow, check that important system files have been backed up. Follow the directories in your DBCS deployment the dates will be different, so directory names filenames with date stamps, will vary from the examples

```
[oracle@MYDBCS]$ cd /u03/app/oracle/fast_recovery_area/MYORCL
[oracle@MYDBCS MYORCL]$ ls
1DC10E160C914070E053920D6A0A55BD  archivelog  control02.ctl
onlinelog
2C42B21F5C9B1640E053E649C40AE9A4  autobackup  datafile  oscfgfiles
es
345D5540529B225EE05366EBC40AB991  backupset   ohcfgfiles
[oracle@MYDBCS MYORCL]$ cd oscfgfiles
[oracle@MYDBCS oscfgfiles]$ ls -l
total 4
4 drwxrwxrwx 2 root root 4096 Jun  3 11:44 2016_06_03
[oracle@MYDBCS oscfgfiles]$ cd 2016_06_03/
[oracle@MYDBCS 2016_06_03]$ ls
oscfgfiles_20160603_1144.tar.gz
[oracle@MYDBCS 2016_06_03]$ tar tzvf
oscfgfiles_20160603_1144.tar.gz
```

- Get the list of files contained in /u03/app/oracle/fast\_recovery\_area/MYORCL/ohcfgfiles/<TimeStamp>/ohcfgfiles\_XXX.tar.gz file. Use tar tvzf.

If, when connecting to the compute node, you encounter an OBKUP error indicating that backups to cloud storage could not complete due to the error KBHS-00715 HTTP CONNECTION ERROR, or any other error, you'll have to find the log files related to the backup operations.

- Find the location of the backup log files by reading the content of the /var/opt/oracle/cleandb/cleandblogs.pl script:

```
[oracle@MYDBCS log]$ cat
/var/opt/oracle/cleandb/cleandblogs.pl|grep -i bkup

# 10/15/15 - on 15.0.0.0 for bkup logs dont use dbname
"obkupLogRetention" => 30,};
# Process obkup logs
$log->logprt("Processing obkup logs.\n");
PurgeLogFiles("/home/oracle/bkup/${ORACLE_SID}/log/obkup*.log",
$CleanLogs->{obkupLogRetention});
$log->logprt("Completed obkup log maintenance.\n");
```

1.1.1 10/13/2015 4:01:14 PM : added glassfish and  
obkup support, configuration value validations  
[oracle@MYDBCS log]\$

- List the backup log files.

```
[oracle@MYDBCS log]$ ls -ltr /home/oracle/bkup/<DBNAME>/log
Total 68
-rwx----- 1 oracle dba      28768 Jun 13 23:26
obkup2016-06-13_23:20:08.log
-rwx----- 1 oracle dba      23977 Jun 14 23:22
obkup2016-06-14_23:20:07.log
-rw-r--r-- 1 oracle oinstall 10898 Jun 16 05:01
obkup2016-06-15_23:20:07.log
lrwxrwxrwx 1 oracle oinstall   50 Jun 15 23:20 obkup.log ->
/home/oracle/bkup/ <DBNAME>/ log/obkup2016-06-15_23:20:07.log
```

- You can now read the content of the log file and find out the cause of a possible error during backup.

```
[oracle@MYDBCS log]$ more /home/oracle/bkup/<DBNAME>/log /obkup
2016-06-15_23:20:07.log
```

```
OBKUP:: Oracle database state is up and running .
catalog mode no
##### PERIODICAL BACKUP STARTING
#####
logfile:
/home/oracle/bkup/MYORCL/log/obkup2016-06-15_23:20:07.log
OBKUP:: Performing backup to local storage (primary backup)
OBKUP:: Executing rman instructions
Connection string:
/u01/app/oracle/product/12.1.0/dbhome_1/bin/rman msgno target
/
-> set encryption on;
...
      RMAN-03090: Starting recover at 15-JUN-16
      RMAN-00571:
=====
      RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS
=====
      RMAN-00571:
=====
      RMAN-03002: failure of recover command at 06/16/2016
02:10:34
      ORA-19554: error allocating device, device type:
SBT_TAPE, device name:
```

```
ORA-27023: skgfsbi: media manager protocol error
ORA-19511: non RMAN, but media manager or vendor
specific failure, error text:
KBHS-01405: Automatic retry wait time limit 539 reached
KBHS-00715: HTTP error occurred 'client-error'
KBHS-00703: unable to connect to HTTP server
dbpm18.storage.oraclecloud.com; received ORA-12535
KBHS-00718: operation failed, retry possible
...
```

**Question:** *Does the backup encrypt data?*

**Answer:** *Yes. Encryption is automatic.*

There is an available fix for this KBHS-00715 HTTP CONNECTION ERROR.  
Read [Backups to cloud storage fail with error KBHS-00715 HTTP CONNECTION ERROR](#). You will apply the fix in the next practice.

## Practice 4-2 Patching your Database Deployment

### Overview

When you connect to a service instance or view the backup log, a message from OBKUP MAY appear after some time. It indicates that backups to cloud storage could not complete due to the error KBHS-00715 HTTP CONNECTION ERROR.

There is an available patch to fix the issue.

If you do not encounter the error, go to the next practice.

In this practice, you will apply the patch to your database deployment.

### Setup

If you are in an Oracle Classroom and have not done so already, make the following modifications.

Edit file `~/.ssh/config` with the following content to reflect the [IP address of DB instance from OPC](#).

```
Host <IP address of DB instance from OPC>
ProxyCommand nc -X connect -x ges-proxy.us.oracle.com:80 %h %p
GSSAPIAuthentication no
```

### Tasks

1. Log in to the compute node as `opc`.

```
[opc@MYDBCS ~]$
```

2. Start a root-user command shell.

```
[opc@MYDBCS opc]$ sudo -s
```

3. Download the patching script.

```
[root@MYDBCS opc]# wget https://storage.us2.oraclecloud.com/v1/dbcsswlibp-usoracle29538/dbaas_patch/dbcs_libopc_patch
--2016-06-17 09:28:04-- https://storage.us2.oraclecloud.com/v1/dbcsswlibp-usoracle29538/dbaas_patch/dbcs_libopc_patch
Resolving storage.us2.oraclecloud.com... ...
Connecting to storage.us2.oraclecloud.com|... .. connected.
HTTP request sent, awaiting response... 200 OK
Length: 3391 (3.3K) [application/octet-stream]
Saving to: "dbcs_libopc_patch.1"

100%[=====>] 3,391      --.-K/s   in 0s

2016-06-17 09:28:05 (24.8 MB/s) - "dbcs_libopc_patch.1" saved [3391/3391]

[root@MYDBCS opc]#
```

4. Set the file permissions on the script to make it executable.

```
[root@MYDBCS opc]# chmod +x dbcs_libopc_patch
[root@MYDBCS opc]#
```

**Question:** What do you observe in the patch script?

**Answer:** There is no `dbaascli dbpatchm` command. This is a patch that does not need to relink the kernel.

5. Execute the patching script.

```
[root@MYDBCS opc]# ./dbcs_libopc_patch

INFO : Starting Patch of dbcs libopc bug 22493500
INFO : Logfile for this run is : /var/opt/oracle/log/opcpatch/opcpatch_2016-06-17_09:42:21.log
INFO : Downloading https://storage.us2.oraclecloud.com/v1/odbsdev-usoracleinter76305/oracle-opc-downloads/opc_linux64.zip to /home/oracle/tmp/libopc.zip
INFO : Unzipping /home/oracle/tmp/libopc.zip
INFO : Backing up libopc from /u01/app/oracle/product/12.1.0/dbhome_1
INFO : Copying new libopc.so to /u01/app/oracle/product/12.1.0/dbhome_1/lib
INFO : Successfully patched /u01/app/oracle/product/12.1.0/dbhome_1 with latest version of libopc.so
INFO : Copying new /home/oracle/tmp/libopc.zip to bkup assistant
INFO : Successfully patched bkup asst with latest version of libopc zip
INFO : Patch for dbcs libopc bug 22493500 Completed Successfully
[root@MYDBCS opc]#
```

6. Exit the root-user command shell and disconnect from the compute node.

```
[root@MYDBCS opc]# exit

exit

[opc@MYDBCS ~]$
```



## Practice 4-3 Recovering a Database Deployment

### Overview

In this practice, you will recover your database deployment database after a datafile loss in the MYORCL CDB. You will use the Database Cloud Service console.

### Tasks

1. Log in to your database deployment compute node as `oracle`. Remove one of the datafiles of your MYPDB1 PDB like `/u02/app/oracle/oradata/MYORCL/MYPDB1/MYPDB1_users01.dbf`.

2. Create a table on the `USERS` tablespace:

```
[oracle@MYDBCS MYORCL]$ sqlplus system/<adminPassword>@MYPDB1
...

SQL> CREATE TABLE mytab (c NUMBER) TABLESPACE users;

CREATE TABLE mytab (c NUMBER) TABLESPACE users
*

ERROR at line 1:

ORA-01116: error in opening database file 10

ORA-01110: data file 10:

'/u02/app/oracle/oradata/MYORCL/MYPDB1/MYPDB1_users01.dbf'

ORA-27041: unable to open file

Linux-x86_64 Error: 2: No such file or directory

Additional information: 3

SQL> EXIT
```

3. From the Oracle Database Cloud Service, click the Open Service console and get to your database deployment. Read [Restoring from the Most Recent Backup](#) to know how to proceed.
4. From the database deployment database, click the Administration tab to view Backup/recovery Details.
5. Click Recover from the Latest backup as this is not an incomplete recovery to be executed but a complete recovery.
6. This can take a couple of minutes before the console shows that the recovery is completed. The recovery is fast. Check that you can create a table on the `USERS` tablespace in your MYPDB1 PDB.

## Practice 5-1 Opening Ports to Compute Node

### Overview

In this practice, you will enable the security rule that allows you or others to connect to your database deployment MYDBCS via EM Express.

You will see that the possible methods used to get access to your database deployment via any tool like EM Express can allow only you to get access to your database deployment or can allow other users to access your database deployment.

### Tasks

The first way is to create a tunnel for port forwarding using `ssh`.

1. On your Linux or Windows desktop, create an SSH tunnel to use EM Express port (5500) on the compute node of your database deployment. Read either [Creating an SSH Tunnel Using the ssh Utility on Linux](#) or [Creating an SSH Tunnel Using the PuTTY Program on Windows](#).  
  

```
$ ssh -i MYDBCSKey -L 5500:<your_VM_IP_Address>:5500 oracle@<your_VM_IP_Address>
[oracle@MYDBCS ~]$
```
2. With Firefox on your Linux or Windows desktop, launch `https://localhost:5500/em`
3. EM Express is started and as you are used to, connect to your database deployment CDB using either the `SYSTEM` or the `SYS` user.
4. Ask your neighbor if he or she can access your database deployment CDB with EM Express with `https://<your_VM_IP_Address>:5500/em`. This will fail.
5. Close the `ssh` tunnel by quitting your Linux desktop session where the SSH tunnel was kept opened.

The second way is to open the port to use EM Express and let anyone access your database deployment CDB via this port.

1. Find the security application constituted of a protocol-type and port mapping, defined and used in security rules to control traffic to/from instances via EM Express. Read [About Security Applications](#). You find the `MYDBCS/db_1/ora_dbexpress` security application for port 5500 for your database deployment.
2. Find the security rule that uses this security application to access your database deployment CDB via EM Express. Enable the `ora_p2_dbexpress` security rule that gives access via EM Express to your database deployment CDB. Read [Enabling Access to a Compute Node Port](#).
3. Check that you can launch `https://<your_VM_IP_Address>:5500/em` and access your database deployment CDB. (For Windows machines on VPN, make sure you are outside VPN).
4. Ask your neighbor to access your database deployment CDB with EM Express using the same URL `https://<your_VM_IP_Address>:5500/em`. The attempt is successful.

Use `https://IP_address_VM_neighbor:5500/em`. The attempt is successful if your neighbor performed the same steps on his or her database deployment.

**Question:** What is the source of the `ora_p2_dbexpress` security rule?

**Answer:** The hosts from which traffic is allowed are `public-internet`.

**Question2:** What is the destination of the `ora_p2_dbexpress` security rule?

**Answer2:** The Security list to which traffic is allowed. This is the `MYDBCS/db_1/ora_db` security list for your deployment.

Therefore, any host can access this destination, provided that he or she knows the database deployment password to access to.

5. Reset the `ora_p2_dbexpress` security rule to the default `DISABLED` status. Ensure that your neighbor completes the same operation.

## Practice 5-2 Using Security Applications

### Overview

In this practice, you will configure network settings so that access via EM Express (or any other tool like DBaaS Monitor, APEX) to your database deployment can use VNC Viewer. For this purpose, you will have to create a new security application.

### Tasks

1. Log in to your Compute Cloud Service console.
2. In the previous practice, you observed that either only your localhost could access your database deployment via an `ssh` tunneling forwarding the EM Express port, or that all hosts could access your database deployment when you enabled the security rule responsible for EM Express. You want a solution that is more subtle.

- a. Create a Security Application. Read [Creating a Security Application](#).

```
Name = VNCViewer
Port Type = TCP
Port Range Start = 5901
Port Range End = 5902
```

- b. Create the `VNC_for_MySelf` security rule that allows a VNCViewer client to connect to your VNC Server on your compute node. Read [Creating a Security Rule](#).

- Source (*Security IP list*) = `Public_Internet`
- Security Application = `VNCViewer`
- Destination (*Security List*) = `MYDBCS/db_1/ora_db`

3. Start VNC server on your compute node on port 5901 as user `oracle`. Make sure you are outside VPN.

```
vncserver -kill :1
vncserver
```

4. **Note:** the security rule created in step 2, may not work in the Oracle classroom due to firewall restrictions. From the Oracle classroom machine you will have to create an `ssh` tunnel as follows:

```
ssh -i <private_key_file> -L 6501:<public_ip>:5901 oracle@<public_ip>
```

SSH tunnels can also be created using Putty on windows machines, see [Creating an SSH Tunnel Using the PuTTY Program on Windows](#).

5. Immediately after VNC server is started, use your VNCViewer to connect to your compute node with the following in the `vcnviewer` connection dialog:

- a. If using the tunnel created in step4 `localhost:6501`
- b. If not in the oracle classroom use `<public_ip>:1`

6. In the VNC session, make sure you disable screen locking timeouts: click System>Preferences>ScreenSaver
  - a. In the Screensaver Preferences dialog, de-select Lock screen when screensaver is active. Set the slider to as long as possible (2hrs). Click Close.
7. From your VNCViewer, start a browser window and connect to EM Express using the following URL: `https://<your_VM_IP_Address>:5500/em`.

## Practice 6-1 Authenticating to Your Compute Node and Database

### Overview

In this practice, you will connect to the compute node of your database deployment `MYDBCS`. You will check that to run the `root` commands, the OS user must be logged as the `opc` user and perform a `sudo -s` command. You will also check that the environment variables such as `ORACLE_SID` and `ORACLE_HOME` are set to those defined by default during the database deployment creation for the `oracle` and `opc` default users.

### Tasks

- Using PuTTY `Cloud_oracle` session or `ssh`, log on to the compute node of your database deployment `MYDBCS`.
- Check that environment variables such as `ORACLE_SID` and `ORACLE_HOME` are set to those defined by default during the database deployment creation.

```
[oracle@MYDBCS ~]$ env | grep ORA
ORACLE_UNQNAME=MYORCL
ORACLE_SID=MYORCL
ORACLE_HOSTNAME=MYDBCS.compute-<your_domain>
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
[oracle@MYDBCS ~]$
```

- Try to execute the `bkup_api` utility to check the current backup status of your database deployment.

```
[oracle@MYDBCS ~] /var/opt/oracle/bkup_api/bkup_api status
API::ERROR Api requires root rights or sudoer
[oracle@MYDBCS ~]$ sudo -s
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for oracle:
```

**Question:** *Is oracle a sudoer?*

**Answer:** *No, only opc is.*

- d. Using PuTTY or `ssh`, log on to the compute node of your database deployment `MYDBCS` using the PuTTY session `Cloud_opc` or `ssh`.

```
[opc@MYDBCS ~]$ /var/opt/oracle/bkup_api/bkup_api status
API::ERROR Api requires root rights or sudoer

[opc@MYDBCS ~]$ sudo -s

[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api status
DBaaS Backup API V1.5 @2015 Single Instance
-> Action : status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
API::ERROR Unknown action: status
Please choose an action to perform
Available actions:
-> bkup_start, bkup_chgcfg, bkup_chkcfg
-> bkup_stop
-> reval_start
-> recover_start
-> update_wallet
-> --uuid ='your request'
-> bkup_status      (Compatibility mode v1)
-> get_tag          (Compatibility mode v1)
-> recover_status   (Compatibility mode v1)
-> --rargs          (Compatibility mode v1)
-> recover_list     (Compatibility mode v1)

[root@MYDBCS opc]#
[root@MYDBCS opc]#
[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Single Instance
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
* Current backup settings:
  catalog: no
```

```

type: diskoss

* Last registered Bkup: 06-05 02:18 API::15851:: Starting dbaas
backup process

* Bkup state: running

*****

* API History: API steps

API:: NEW PROCESS 15851

API:: Starting dbaas backup process

*****

* Backup steps

-> API:: Oracle database state is up and running

-> API:: Performing backup to local storage (primary backup)

-> API:: Executing rman instructions

*

* RETURN CODE:0

#####

[root@MYDBCS opc]#
[root@MYDBCS opc]#
[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog

* Current backup settings:

* Last registered Bkup: 07-25 01:12 API::3637:: Starting dbaas
backup process

* Bkup state: finished

*****

* API History: API steps

API:: NEW PROCESS 15851

API:: Starting dbaas backup process

API:: Your new dbaas backup tag is TAG20160725T011425

```



```

API:: BKUP COMPLETE YOUR BKUP TAG TAG20160725T011425
*****
* Backup steps
-> API:: Oracle database state is up and running
-> API:: DB instance: MYORCL
-> API:: Determining if the filesystem is not full
-> API:: ..... OK
-> API:: Performing backup to local storage (primary backup)
-> API:: Executing rman instructions
-> API:: ..... OK
-> API:: Backup to local storage is completed
-> API:: Clean MOTD.
-> API:: Performing backup to cloud storage (secondary backup)
-> API:: Executing rman instructions
-> API:: .....OK
-> API:: Backup to cloud storage is completed
-> API:: Clean MOTD.
-> API:: Validating the backup repository .....
-> API::      All backup pieces are ok
-> API:: Starting backup of config files
-> API:: at time: 2016-07-25:01:19:24
-> API:: Determining the oracle database id
-> API::  DBID: 338429947
-> API:: Creating directories to store config files
-> API:: Determining the oracle database id
-> API::  DBID: 338429947
-> API:: Compressing config files into tar files
-> API:: .... OK
-> API:: Uploading config files to cloud storage
-> API:: Completed at time: 2016-07-25:01:19:27
-> API:: at time: 2016-07-25:01:19:27
-> API:: Config files backup ended successfully

```

```
-> API:: Clean MOTD.  
-> API:: All requested tasks are completed  
*  
* RETURN CODE:0  
#####  
[root@MYDBCS opc]# exit  
exit  
[opc@MYDBCS ~]$
```

## Practice 6-2 Accessing the Database via DBaaS Monitor Console

### Overview

In this practice, you will launch the DBaaS Monitor console to later monitor your database deployment `MYDBCS` in another practice. To launch EM Express, you used two methods. In this practice, you will verify that Oracle DBaaS Monitor is accessible via HTTPS port (443). For information about unblocking a port, see [Enabling Access to a Compute Node Port](#).

### Tasks

1. Open the Oracle Database Cloud Service console.

2. From the



menu for the database deployment, select Access Rules. If you encounter issues accessing rules management from this console, use the Compute Cloud Service console.

**Question:** Do you get the whole list of security or access rules of all database deployments of the domain as it was the case in the Compute Cloud Service console?

**Answer:** No. The Database Cloud Service console displays only the security rules created for your database deployment.

**Question2:** Are the names of the security rules the same as in the Compute Cloud Service console?

**Answer2:** No. As you can see only the rules for your database deployment, the names are limited. For example, `ora_p2_dbexpress` corresponds to `MYDBCS/db_1/ora_dbexpress` and `ora_p2_httpssl` corresponds to `MYDBCS/db_1/ora_httpssl`.

3. Locate the `ora_p2_httpssl` access rule, which controls access to port 443, the port used for HTTPS connections, including Oracle DBaaS Monitor.

4. From the



menu for the located rule, check that it is Enabled. The given port on the compute node is opened to the public internet.

5. Use `https://public_IP_Address_your_VM/dbaas_monitor` to connect to the DBaaS Monitor console.
6. When prompted for a user name and password, enter `dbaas_monitor` as the user name and the password specified when the database deployment was created.

**Remark:** Monitoring the database deployment is covered in another practice.

## Practice 7-1 Protecting Data at Rest Using Encryption

### Overview

In this practice, you will create the new `TESTPDB` PDB in `MYORCL` CDB with DBaaS Monitor console and then create the `TEST` tablespace in the PDB. Check that the `TEST` tablespace uses encryption by default.

### Tasks

1. Log in to the DBaaS Monitor console.
2. From Database > Manage, create the new `TESTPDB` PDB. While creating `TESTPDB` be sure to check the Create TDE key and provide the admin password (this is the administrative password you provided when you created the DCBS service). The PDB creation requires the CDB keystore password to create the PDB's own master encryption key, which is stored in a single keystore used by all containers. Use the password defined at the database deployment creation. View the SQL generated behind the scene. Read [Creating and Activating a Master Encryption Key for a PDB](#).
3. Use the `Cloud_oracle` session to log in to `MYDBCS` compute node.
4. Use SQL\*Plus to connect to the `TESTPDB` PDB as the `SYSTEM` user. If you do not know which connect string to use, from the DBaaS Monitor console, in the menu for your PDB, click `Connection Details`.

5. Create the `TEST` tablespace.

```
CREATE TABLESPACE TEST;
```

6. Verify that the `TEST` tablespace uses encryption.

```
SQL> SELECT tablespace_name, encrypted FROM dba_tablespaces;
```

TABLESPACE_NAME	ENC
-----	----
SYSTEM	NO
SYSAUX	NO
TEMP	NO
TEST	YES

```
SQL>
```

**Question:** Do all tablespaces use encryption?

**Answer:** No. Only user-defined tablespaces use TDE encryption.

7. Verify the value of the parameter responsible for this default behavior.

```
SQL> SHOW PARAMETER encrypt_new_tablespaces
```

NAME	TYPE	VALUE
-----	-----	-----
encrypt_new_tablespaces	string	CLOUD_ONLY

SQL>

8. Find the TDE wallet of the database deployment database.

```
[oracle@MYDBCS ~]$ grep ENCRYPTION_WALLET $ORACLE_HOME/network/admin/sqlnet.ora
ENCRYPTION_WALLET_LOCATION =
(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/MYORCL/tde_wal
let)))

[oracle@MYDBCS ~]$ cd /u01/app/oracle/admin/MYORCL/

[oracle@MYDBCS MYORCL]$ ls
adump  db_wallet  dpdump  opc_wallet  pfile  tde_wallet  xdb_wallet

[oracle@MYDBCS MYORCL]$ cd tde_wallet

[oracle@MYDBCS tde_wallet]$ ls -l

total 36

-rw----- 1 oracle oinstall 8925 Jun 20 15:27 cwallet.sso
-rw----- 1 oracle oinstall 2400 Jun 20 15:19 ewallet_2016062015190656.p12
-rw----- 1 oracle oinstall 5328 Jun 20 15:27
ewallet_2016062015275500_tde_dbaas_bkup.p12
-rw----- 1 oracle oinstall 8880 Jun 20 15:27 ewallet.p12

[oracle@MYDBCS tde_wallet]
```

## Practice 7-2 Checking Data Protection in Transit

### Overview

In this practice, you will check that data in transit over the network is protected. Encryption of network data prevents unauthorized parties to view data as it passes over the network.

Read [Using Network Encryption and Integrity](#).

You will also check that integrity algorithms protect against data modification and illegitimate replay.

### Tasks

1. Log in to MYDBCS compute node as `oracle`.
2. Check that the following parameter settings in `sqlnet.ora` are:

```
[oracle@MYDBCS admin]$ cat $ORACLE_HOME/network/admin/sqlnet.ora

SQLNET.ENCRYPTION_SERVER = required

SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA1)

SQLNET.CRYPTO_CHECKSUM_SERVER = required

ENCRYPTION_WALLET_LOCATION =
(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/MYORCL/tde_wallet)))

SQLNET.ENCRYPTION_TYPES_SERVER = (AES256, AES192, AES128)

NAMES.DIRECTORY_PATH = (TNSNAMES, EZCONNECT)

SQLNET.WALLET_OVERRIDE = FALSE

SQLNET.EXPIRE_TIME = 10

SSL_VERSION = 1.0

WALLET_LOCATION =
(SOURCE=(METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/MYORCL/db_wallet))(METHOD=FILE))
```

3. You can also perform the verification by connecting to your Oracle database and examining the network service banner entries associated with each connection.
  - a. Use SQL\*Plus to connect to the MYORCL CDB as the `SYSTEM` user. If you do not know which connect string to use, from the DBaaS Monitor console, in the menu for your PDB, click `Connection Details`.
  - b. Display the content of the following view.

```
SQL> SELECT network_service_banner
```

```

FROM    v$session_connect_info

WHERE   sid in (select distinct sid from v$mystat);  2      3

```

NETWORK\_SERVICE\_BANNER

```

-----
---

Oracle Bequeath NT Protocol Adapter for Linux: Version 12.1.0.2.0 -
Production

Authentication service for Linux: Version 12.1.0.2.0 - Production

Encryption service for Linux: Version 12.1.0.2.0 - Production

AES256 Encryption service adapter for Linux: Version 12.1.0.2.0 - Production

Crypto-checksumming service for Linux: Version 12.1.0.2.0 - Production

SHA1 Crypto-checksumming service adapter for Linux: Version 12.1.0.2.0 -
Production

```

6 rows selected.

SQL>

## Practice 7-3 Cleaning Up Audit Files

### Overview

In this practice, you will check that audit files are automatically purged every day. You will change the retention period for the audit files to 2 days and you will clean up those files manually without waiting for the next automatic execution.

### Tasks

1. Log in to MYDBCS compute node as oracle.
2. Read the content of the `/var/opt/oracle/cleandb/cleandblogs.pl` script:

```
[oracle@MYDBCS admin]$ cat /var/opt/oracle/cleandb/cleandblogs.pl |grep -i aud

"AuditRetentionDB" => 1,
"select 'adumpDest:'||value from v\${parameter} where name = 'audit_file_dest'";
$log->logprt("Processing audit logs...\n");
PurgeLogFiles("${results{'adumpDest'}/*.aud", $CleanLogs->{AuditRetentionDB});
PurgeLogFiles("${ORACLE_HOME}/rdbms/audit/*.aud", $CleanLogs->{AuditRetentionDB});

[oracle@MYDBCS admin]$
```

3. Verify that audit files exist in the following directories:

- a. The `audit_file_dest` directory, `/u01/app/oracle/admin/MYORCL/adump`

```
[oracle@MYDBCS admin]$ ls /u01/app/oracle/admin/MYORCL/adump

MYORCL_j000_13266_20160606013017264430143795.aud
MYORCL_j000_1335_20160606220008338595143795.aud
...
```

- b. The `$ORACLE_HOME/rdbms/audit` directory

```
[oracle@MYDBCS admin]$ ls $ORACLE_HOME/rdbms/audit/*.aud

/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/audit/DBUA2801326_ora_8578_
20160603162802924191143795.aud
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/audit/DBUA2801326_ora_8578_
20160603162807666372143795.aud
...
```

4. Edit the content of the `/var/opt/oracle/cleandb/cleandblogs.cfg` configuration file. Change the retention period for the audit files. You should get the following result after editing.

```
[oracle@MYDBCS admin]$ more /var/opt/oracle/cleandb/cleandblogs.cfg

#
```



```
# cleandblogs.cfg
#
# all values are in days
# ##
AlertRetention=14
ListenerRetention=14
AuditRetentionDB=2
CoreRetention=7
TraceRetention=7
shorttpRetention=7
longpRetention=30
LogDirRetention=14
gfLogRetention=14
obkupLogRetention=30
[oracle@MYDBCS admin]$
```

5. Launch the script to purge all audit files. Be aware that the script also removes all other log and diagnostics files according to the `/var/opt/oracle/cleandb/cleandblogs.cfg` configuration file used.

```
[oracle@MYDBCS admin]$ cd /var/opt/oracle/cleandb
[oracle@MYDBCS admin]$ ./cleandblogs.pl
```

Log file is `/var/opt/oracle/log/cleandblogs/cleandblogs_2016-06-08_08:56:40.log`

Parameter configuration file is `/var/opt/oracle/cleandb//cleandblogs.cfg`

Initializing program configuration

ADR purge started

...

Processing audit logs...

**Removing files like `/u01/app/oracle/admin/MYORCL/adump/*.aud` older than 2 days.**

**Removing files like `/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/audit/*.aud` older than 2 days.**

Completed maintenance activities on instance MYORCL.

Processing GlassFish server log.

Removing files like

```
/u01/app/oracle/product/glassfish3/glassfish/domains/domain1/logs/server_*.log  
older than 14 days.
```

```
Completed GlassFish log maintenance.
```

```
Processing obkup logs.
```

```
Removing files like /home/oracle/bkup/log/obkup*.log older than 30 days.
```

```
Completed obkup log maintenance.
```

```
Removing files like /var/opt/oracle/log/cleandblogs/* older than 14 days.
```

```
Job Completed. RC=(0) 2 Elapsed Seconds, 2 Seconds
```

```
[oracle@MYDBCS cleandb]$
```

## Practice 8-1 Monitoring the Database Deployment

### Overview

In this practice, you will monitor your database deployment using DBaaS Monitor.

### Tasks

1. Launch DBaaS Monitor using `https://IP_address_your_VM/dbaas_monitor`.
2. Explore the home page.
3. Close your PDB. Read [Administer Pluggable Databases](#).
4. Reopen your PDB.
5. Find out the Connection Details for MYPDB1.
6. Back to the Home page, explore the Online Database Storage.
7. Before finding the list of segments stored in the `USERS` tablespace in your PDB, create a table in the `USERS` tablespace in your PDB with `SQL*Plus`.
8. Back to the Home page, explore the Listener status.
9. Examine the percent of OS Memory used by various processes.
10. Check the percent used of the disks in your deployment. Are any close to being full?. (Overall OS Storage)

## Practice 8-2 Using REST APIs

### Overview

In this practice, you will retrieve the characteristics of your database deployment using [the appropriate REST API](#).

### Tasks

1. Log in the compute node of the database deployment as `opc` and verify that the Oracle REST Data Services (ORDS) is started.

```
[opc@MYDBCS ~]$ sudo -s
[root@MYDBCS opc]# /u01/app/oracle/product/ords/ords status
INFO: Obtaining Oracle REST Data Services status...
INFO: Oracle REST Data Services is already running with PID 5569
[root@MYDBCS opc]# exit
```

2. Obtain the REST Endpoint for your deployment from the my services dashboard. How to determine your REST server is covered in the [URL Structure](#) topic of the *REST API for Oracle Database Cloud - Database as a Service* publication. Navigate to the Service Details:Oracle Database Cloud Service page. The REST Endpoint for this practice accessing the Database Cloud Service is listed in the REST Endpoint field.

3. Use `curl` as follows either from a Linux or Windows desktop:

```
[djeunot@myserver ~]$ curl -v -X GET -u <your_username>:<your_password> -H "X-ID-TENANT-NAME:<your_domain>" \
https://<REST_Endpoint>/paas/service/dbcs/api/v1.1/instances/<your_domain>/<your_database_deploym
ent>

* About to connect() to <REST_Endpoint> port 443 (#0)
*   Trying 160.34.0.107... connected
* Connected to <REST_Endpoint> (160.34.0.107) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
*   CPath: none
* SSL connection using TLS_DHE_RSA_WITH_AES_256_CBC_SHA
* Server certificate:
*   subject: CN=*.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*   start date: Jul 02 00:00:00 2015 GMT
*   expire date: Aug 30 23:59:59 2016 GMT
*   common name: *.oraclecloud.com
*   issuer: CN=VeriSign Class 3 Secure Server CA - G3,OU=Terms of use at https://www.verisign
.com/rpa (c)10,OU=VeriSign Trust Network,O="VeriSign, Inc.",C=US
* Server auth using Basic with user '<username>'
> GET /paas/service/dbcs/api/v1.1/instances/<domain>/<deployment> HTTP/1.1
> Authorization: Basic ZG9taW5pcXVlLmpldW5vdEBvcnFjbGUuY29tOkFoYm9uMTk5NW==
```

```

> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: <REST_Endpoint>
> Accept: */*
> X-ID-TENANT-NAME:<domain>
>
< HTTP/1.1 200 OK
< Date: Tue, 05 Jul 2016 17:35:35 GMT
< Server: Oracle-Application-Server-11g
< Content-Length: 1539
< X-ORACLE-DMS-ECID: 005DiRXOMqn6uHFpR0H7id0003kQ0001^0
< X-ORACLE-DMS-ECID: 005DiRXOMqn6uHFpR0H7id0003kQ0001^0
< X-Frame-Options: DENY
< Service-URI: https://<REST_Endpoint>:443/paas/service/dbcs/api/v1.1/instances/<domain>/<deployment>
< Vary: Accept-Encoding,User-Agent
< Content-Language: en
< Content-Type: application/json
<
{
  "service_name": "<deployment>",
  "version": "12.1.0.2",
  "status": "Running",
  "description": "<deployment>",
  "identity_domain": "<domain>",
  "creation_time": "Mon Jun 20 14:58:35 UTC 2016",
  "last_modified_time": "Mon Jun 20 14:58:35 UTC 2016",
  "created_by": "<username>",
  "sm_plugin_version": "16.2.5-125",
  "backup_supported_version": "16.2.3",
  "service_uri": "https://<REST_Endpoint>:443/paas/service/dbcs/api/v1.1/instances/<domain>/<deployment>",
  "num_nodes": 1,
  "level": "PAAS",
  "edition": "EE",
  "shape": "oc4",
  "subscriptionType": "MONTHLY",
  "creation_job_id": "5258807",
  "num_ip_reservations": 1,
  "backup_destination": "BOTH",
  "cloud_storage_container": "Storage-<domain>/<Container>",
  "failover_database": false,
  "rac_database": false,
  "scale_job_id": "5540636",
  "current_version": "12.1.0.2.160119",
  "sid": "MYORCL",
  "pdbName": "MYPDB1",
  "listenerPort": 1521,

```

```

"timezone": "UTC",
"em_url": "https://129.144.21.73:5500/em",
"connect_descriptor": "<deployment>:1521/MYPDB1.<domain>.oraclecloud.internal",
"connect_descriptor_with_public_ip": "129.144.21.73:1521/MYPDB1.<domain>.oraclecloud.internal"
,
"apex_url": "https://129.144.21.73/apex/mypdb1/",
"glassfish_url": "",
"dbaasmonitor_url": "https://129.144.21.73/dbaas_monitor",
"charset": "AL32UTF8",
"ncharset": "AL16UTF16",
"is_clone": false,
"clone_supported_version": "16.3.1",
"compute_site_name": "US006_Z25",
"jaas_instances_using_service": ""
* Connection #0 to host <REST_Endpoint> left intact
* Closing connection #0
}[djeunot@myserver ~]$

```

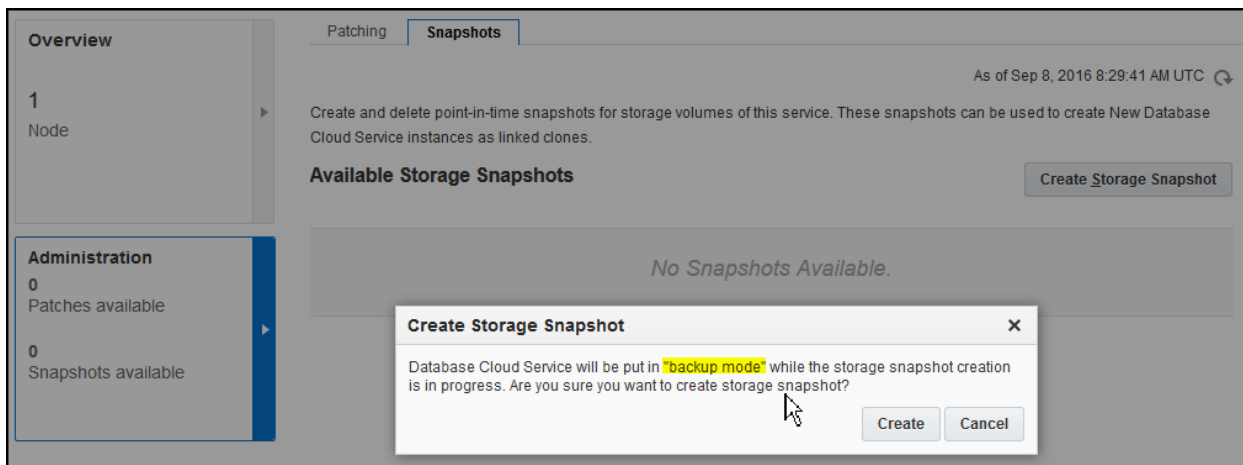
## Practice 8-3 Creating a Linked Clone (Optional)

### Overview

In this practice, you will create a linked clone deployment from your own database deployment after making some data updates in the `MYPDB1` PDB. You will check data in the PDB of the linked clone, and finally make updates in the PDB of the linked clone.

### Tasks

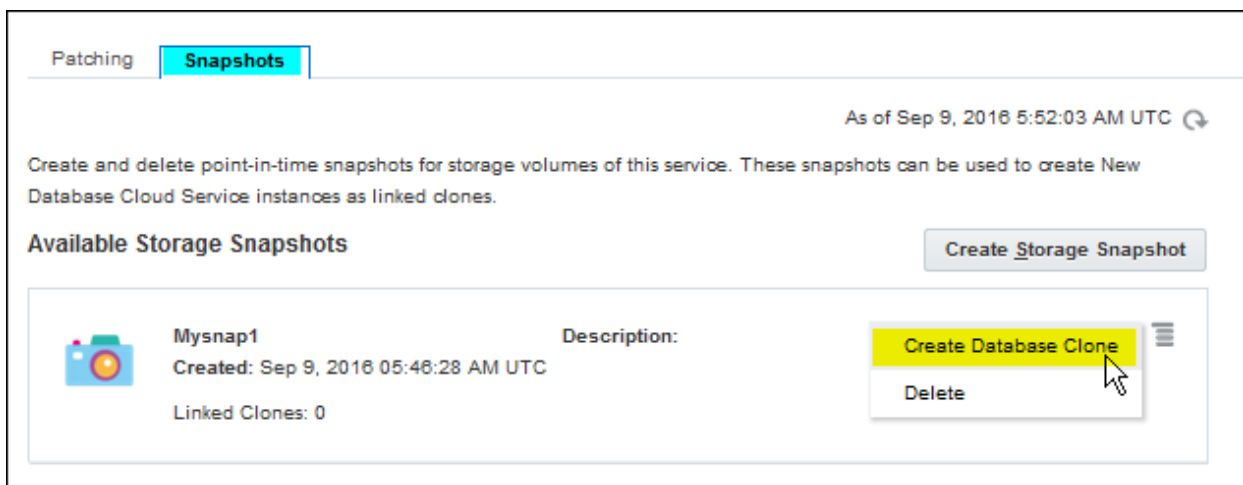
1. Access `MYPDB1` PDB in the `MYDBCS` deployment, create the `SYSTEM.TEST` table in `MYPDB1`, insert a row and commit.
2. Create a snapshot from your database deployment. If you need details on each step of the snapshot creation process, follow [Creating a Snapshot](#).



Be aware that during the snapshot creation, the database is unavailable.

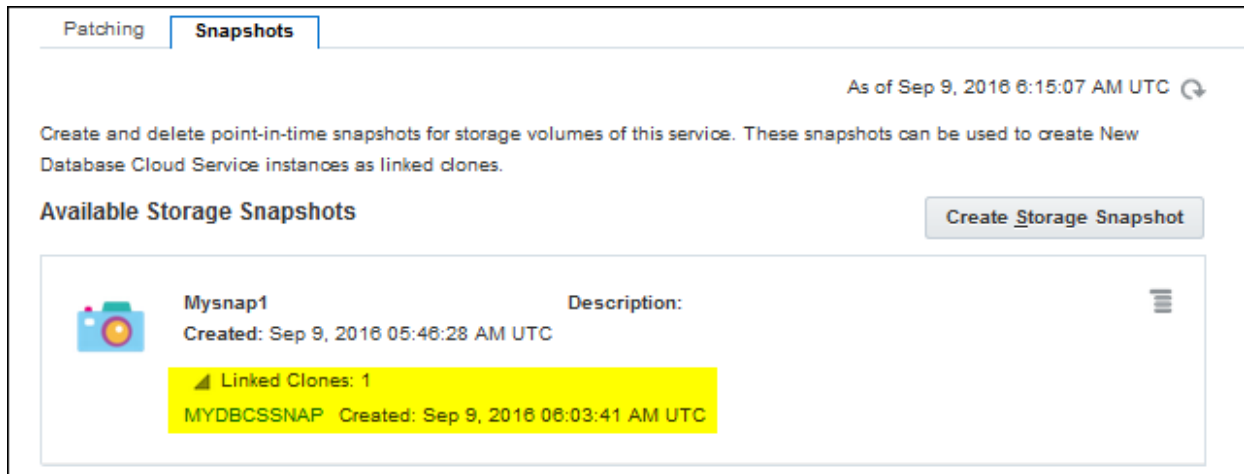
The storage used for the snapshot is equivalent to the storage used by the database deployment.

3. Create a linked clone from the snapshot. If you need details on each step of the linked clone creation process, follow [Creating a Database Deployment from a Snapshot](#).



The storage used for the linked clone is equivalent to the storage used by the database deployment.

- List linked clone database deployments. If you need details on how to find the list, follow [Listing Linked-Clone Database Deployments Created from a Snapshot](#).



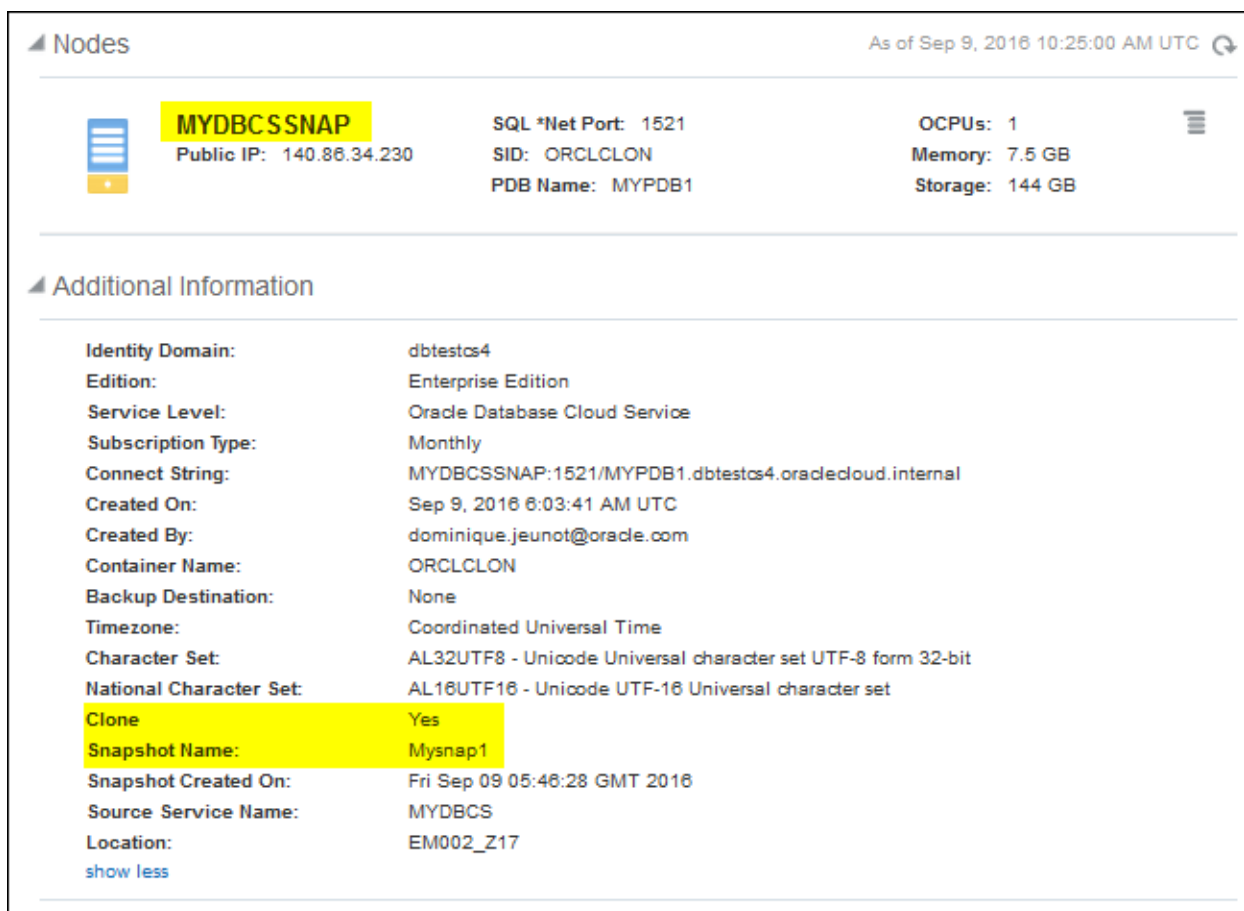
Patching | **Snapshots**

As of Sep 9, 2016 6:15:07 AM UTC

Create and delete point-in-time snapshots for storage volumes of this service. These snapshots can be used to create New Database Cloud Service instances as linked clones.

**Available Storage Snapshots** [Create Storage Snapshot](#)

Icon	Name	Description
	Mysnap1	Created: Sep 9, 2016 05:46:28 AM UTC
<b>Linked Clones: 1</b> MYDBCSSNAP Created: Sep 9, 2016 06:03:41 AM UTC		



Nodes

As of Sep 9, 2016 10:25:00 AM UTC

Icon	Name	SQL *Net Port	OCPU	Memory	Storage
	MYDBCSSNAP	1521	1	7.5 GB	144 GB
	Public IP: 140.86.34.230	SID: ORCLCLON			
		PDB Name: MYPDB1			

**Additional Information**

Identity Domain:	dbtestcs4
Edition:	Enterprise Edition
Service Level:	Oracle Database Cloud Service
Subscription Type:	Monthly
Connect String:	MYDBCSSNAP:1521/MYPDB1.dbtestcs4.oraclecloud.internal
Created On:	Sep 9, 2016 6:03:41 AM UTC
Created By:	dominique.jeunot@oracle.com
Container Name:	ORCLCLON
Backup Destination:	None
Timezone:	Coordinated Universal Time
Character Set:	AL32UTF8 - Unicode Universal character set UTF-8 form 32-bit
National Character Set:	AL16UTF16 - Unicode UTF-16 Universal character set
<b>Clone</b>	<b>Yes</b>
<b>Snapshot Name:</b>	<b>Mysnap1</b>
Snapshot Created On:	Fri Sep 09 05:46:28 GMT 2016
Source Service Name:	MYDBCS
Location:	EM002_Z17

[show less](#)

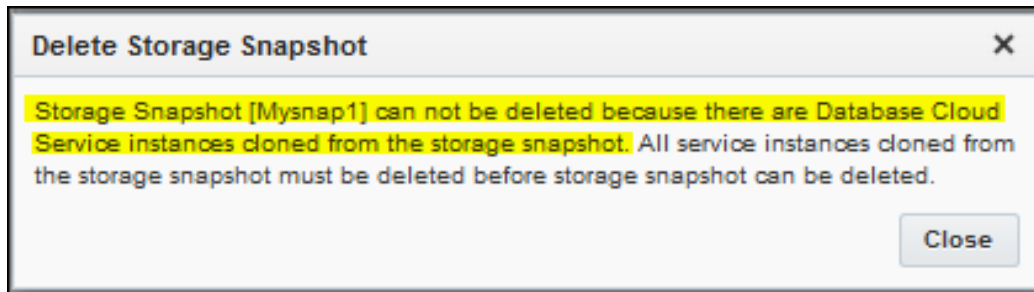
- Check if the `SYSTEM.TEST` table with the row exists in the linked clone database.



**Question:** If you create a table in the PDB of the linked clone deployment, insert a row and commit, would the table be visible in MYP DB1 in the MYDBCS deployment?

**Answer:** No. The new database in the linked clone is fully independent from the snapshot.

6. Remove the snapshot to release resources. If you need details on each step of the snapshot removal, follow [Deleting a Snapshot](#).



This means that you have to delete the linked clone created in step 3 first and then you'll be able to delete the snapshot.

Be aware that during the snapshot deletion, the database is unavailable.

## Practicing Migration

- Practice 2-1 Using Data Pump Conventional Export/Import
- Practice 3-1 Unplugging/Plugging a PDB
- Practice 5-1 Using RMAN Cross-Platform Transportable Tablespace Backup Sets
- Practice 6-1 Using SQL Developer and INSERT Statements to Migrate Selected Objects

## Practice 2-1 Using Data Pump Conventional Export/Import

### Overview

This practice includes the tasks required to migrate a schema from an on-premises Oracle database to an Oracle Database Cloud Service database.

This practice illustrates a schema mode export and import. The same general procedure applies for a full database, tablespace, or table export and import.

In advance of this course, the following tasks were completed to create an export dump file that you will use in the import operation:

1. On the on-premises database host, create an operating system directory to use for the on-premises database export files.
2. On the on-premises database host, invoke SQL\*Plus and log in to the on-premises database as the `SYSTEM` user.
3. Create a directory object in the on-premises database to reference the operating system directory as shown in this example:  

```
SQL> CREATE DIRECTORY dp_for_cloud AS
'/u01/app/oracle/admin/orcl/dpdump/for_cloud';
```
4. Exit from SQL\*Plus.
5. On the on-premises database host, invoke Data Pump Export as the `SYSTEM` user or another user with the `DATA_PUMP_EXP_FULL_DATABASE` role and export the on-premises schemas as shown in this example:  

```
$ expdp system SCHEMAS=fsowner DIRECTORY=dp_for_cloud
```

### Tasks

You will now use the pre-created export dump file and perform an import into your Database Cloud Service database.

1. Log in to the compute node of your database deployment as the `oracle` user.
2. On the compute node of your database deployment, create the directories required for this practice.
  - a. Create a directory named **from\_onprem** in `/u01/app/oracle/admin/MYORCL/dpdump` for the dump file.  

```
$ mkdir -p /u01/app/oracle/admin/MYORCL/dpdump/from_onprem
```
  - b. Create a directory named `/home/oracle/labs/migrate/datapump`  

```
$ mkdir -p /home/oracle/labs/migrate/datapump
```
3. The Data Pump export dump file is staged on your classroom machine. You can use the SCP utility to transfer it to the compute node.
  - a. Open a terminal window on your desktop.
  - b. Use the SCP utility to transfer the dump file to the Database Enterprise Cloud Service compute node.  

```
$ scp -i private_key_file \
/home/oracle/labs/migrate/datapump/expdat.dmp \
oracle@IP_address_DBaaS_VM:/u01/app/oracle/admin/MYORCL/dpdump/from_onprem
```
4. Script files are also staged on your classroom machine. Use the SCP utility to transfer the files to the Database Cloud Service compute node.  

```
$ scp -i private_key_file /home/oracle/labs/migrate/datapump/*.sql oracle@IP_address_DBaaS_VM:/ho
```

```
me/oracle/labs/migrate/datapump
```

5. Return to your Database Cloud Service compute node session, invoke SQL\*Plus, and log in to MYPDB1 as the SYSTEM user.

```
$ sqlplus system/password@MYPDB1
```

6. Create a directory object in the MYPDB1 PDB.

```
SQL> CREATE DIRECTORY dp_from_onprem AS
'/u01/app/oracle/admin/MYORCL/dpdump/from_onprem';
```

7. Execute the create\_fsdata.sql and create\_fsindex.sql scripts to create the tablespaces for the objects that will be imported.

```
SQL> @$HOME/labs/migrate/datapump/create_fsdata.sql
SQL> @$HOME/labs/migrate/datapump/create_fsindex.sql
```

8. Exit from SQL\*Plus.

```
SQL> exit
```

9. On the compute node of your database deployment, invoke Data Pump Import and connect to MYPDB1. Import the data into the database.

```
$ impdp system@MYPDB1 SCHEMAS=fsowner DIRECTORY=dp_from_onprem
```

10. You can verify that the import created tables and inserted data by querying FSOWNER.CUSTOMERS.

- a. Invoke SQL\*Plus and log in to MYPDB1 as the SYSTEM user.

```
$ sqlplus system/password@MYPDB1
```

- b. Query FSOWNER.CUSTOMERS.

```
SQL> select count(*) from fsowner.customers;
```

- c. Exit from SQL\*Plus.

```
SQL> exit
```

11. After verifying that the data has been imported successfully, you can delete the expdat.dmp file from its location on the Database Cloud compute node.

## Practice 3-1 Unplugging/Plugging a PDB

### Overview

This practice includes the tasks required to migrate a PDB from an on-premises Oracle database to an Oracle Database Cloud Service database.

Before the class started, the following tasks were completed to create the files that will be used to plug the PDB into the Database Cloud Service database:

1. On the on-premises database host, invoke SQL\*Plus and close the on-premises PDB.  

```
SQL> ALTER PLUGGABLE DATABASE pdb2prem CLOSE;
```
2. On the on-premises database host, execute the ALTER PLUGGABLE DATABASE UNPLUG command to generate an XML file containing the list of datafiles that will be plugged in to the database on Database Cloud Service.  

```
SQL> ALTER PLUGGABLE DATABASE pdb2prem UNPLUG INTO '/tmp/pdb2prem.xml';
```

### Tasks

You will now use the pre-created XML file and datafiles from the PDB to plug the PDB into your Database Cloud Service database.

1. Open a session to your Database Enterprise Cloud Service compute node as the `oracle` user.
2. On the Database Enterprise Cloud Service compute node, create a directory for the data files.  

```
$ mkdir /u02/app/oracle/oradata/MYORCL/PDB2
```
3. The XML file and datafiles are staged on your classroom machine. You can use the SCP utility to transfer the files to the compute node.
  - a. Open a terminal window on your desktop.
  - b. Use the SCP utility to transfer the data files and the XML file from your classroom machine to the Database Cloud Service compute node.

```
$ scp -i private_key_file \
/home/oracle/labs/migrate/plug/*.dbf \
oracle@IP_address_DBaaS_VM:/u02/app/oracle/oradata/MYORCL/PDB2
```

```
$ scp -i private_key_file /home/oracle/labs/migrate/plug/pdb2prem.xml \
oracle@IP_address_DBaaS_VM:/tmp/pdb2.xml
```

4. In your Database Cloud Service compute node session, invoke SQL\*Plus and execute the CREATE PLUGGABLE DATABASE command to plug the database into the CDB.
  - a. Invoke SQL\*Plus and log in to the database using the SYSDBA role.  

```
$ sqlplus / as sysdba
```
  - b. Plug in the database by executing the CREATE PLUGGABLE DATABASE command.

```
CREATE PLUGGABLE DATABASE pdb2 USING '/tmp/pdb2.xml'
SOURCE_FILE_NAME_CONVERT =
('/u01/app/oracle/oradata/orclprem/orcl/pdb2prem',
'/u02/app/oracle/oradata/MYORCL/PDB2')
NOCOPY TEMPFILE REUSE;
```

```
SQL> CREATE PLUGGABLE DATABASE pdb2 USING '/tmp/pdb2.xml'
2  SOURCE_FILE_NAME_CONVERT =
3  ('/u01/app/oracle/oradata/orclprem/orcl/pdb2prem',
4  '/u02/app/oracle/oradata/MYORCL/PDB2')
5  NOCOPY TEMPFILE REUSE;
```

5. On the Database as a Service compute node, open the new PDB.

- a. Execute the ALTER PLUGGABLE DATABASE OPEN command.

```
SQL> alter pluggable database pdb2 open;
```

- b. If you receive a warning message when opening the PDB, investigate the status of the PDB by performing the following query:

```
SQL> SELECT name, open_mode, restricted FROM v$pdb;
```

**QUESTION:** What does a YES in the RESTRICTED column mean?

**ANSWER:** Only users that have been granted the RESTRICTED SESSION privilege can connect to the PDB.

- c. Investigate incompatibilities between the PDB and the CDB by performing the following query:

```
SELECT time, type, error_number, message, status, action
FROM pdb_plug_in_violations
WHERE name = 'PDB2';
```

**QUESTION:** Which issues do you need to address so that the PDB can be opened? How can you address those issues?

**ANSWER:** Issues with a value of PENDING in the STATUS column need to be resolved. The ACTION column indicates what action needs to be taken.

Some of the major issues are the datapatch patch to be applied on the PDBs and APEX component to be deinstalled (\$ORACLE\_HOME/apex/apxremov.sql) and reinstalled (\$ORACLE\_HOME/apex/apxins.sql) in the CDB so that the mismatch between the CDB version and PDB version is resolved.

**Note:** You will not resolve any PDB incompatibility issues during this course.

6. Add a service name for the new PDB to the tnsnames.ora file.

- a. Query v\$SERVICES to obtain the service name of the new PDB.

```
SQL> SELECT name FROM v$services WHERE pdb= 'PDB2' ;
```

- b. Edit the \$ORACLE\_HOME/network/admin/tnsnames.ora file to add the service name entry.

```
PDB2 =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = your_instance.your_host)(PORT =
1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = PDB_service_name)
  )
)
```

7. Connect to the PDB using the new service name you defined.

```
SQL> connect system/password@PDB2
```

8. Exit from SQL\*Plus.

## Practice 5-1 Using RMAN Cross-Platform Transportable Tablespace Backup Sets

### Overview

This practice includes the tasks required to migrate tablespaces in an Oracle Database PDB to an Oracle Database Cloud Service database. In this practice you will migrate the `FSDATA` and `FSINDEX` tablespaces.

Before this class started, the following tasks were completed to create the backup set and the dump file used for this migration:

1. On the on-premises database host, set all tablespaces that will be transported (the transportable set) to `READ ONLY` mode as shown in this example:  

```
SQL> ALTER TABLESPACE fsindex READ ONLY;
SQL> ALTER TABLESPACE fsdata READ ONLY;
```
2. On the on-premises database host, invoke RMAN and use the `BACKUP` command with the `TO PLATFORM` or `FOR TRANSPORT` clause and the `DATAPUMP` clause to create a backup set for cross-platform transport as shown in this example:  

```
RMAN> BACKUP FOR TRANSPORT
2> FORMAT '/u01/app/oracle/admin/orcl/rman_transdest/fs_tbs.bck'
3> TABLESPACE fsdata,fsindex
4> DATAPUMP FORMAT '/u01/app/oracle/admin/orcl/rman_transdest/fs_tbs.dmp';
```

### Tasks

You will now use the pre-created backup set and dump file to migrate the tablespaces to your Database Cloud Service database.

1. Open a session to your Database Cloud Service compute node as the `oracle` user.
2. On the Database Cloud Service compute node, create a directory for the backup set and dump file.  

```
$ mkdir /tmp/from_onprem
```
3. The backup set and export dump file are staged on your classroom machine. You can use the SCP utility to transfer the files to the compute node.
  - a. Open a terminal window on your desktop.
  - b. Use the SCP utility to transfer the backup set and export dump file from your classroom machine to the Database Cloud Service compute node.  

```
$ scp -i private_key_file \
/home/oracle/labs/migrate/rman/fs_tbs.bck \
oracle@IP_address_DBaaS_VM:/tmp/from_onprem

$ scp -i private_key_file \
/home/oracle/labs/migrate/rman/fs_tbs.dmp \
oracle@IP_address_DBaaS_VM:/tmp/from_onprem
```
4. On the Database Cloud Service compute node create the user that owns the objects that will be imported.
  - a. Create the required directory on the DBCS compute node. In a terminal session connected to the compute node as the `oracle` user create the `/home/oracle/labs/migrate/rman` directory.



- ```
$ mkdir -p /home/oracle/labs/migrate/rman
```
- b. Script files are also staged on your classroom machine. Use the SCP utility to transfer the script files to the Database Cloud Service compute node.
 

```
$ scp -i private_key_file /home/oracle/labs/migrate/rman/create_fsowner.sql oracle@IP_address_DBaaS_VM:/home/oracle/labs/migrate/rman
```
  - c. Invoke SQL\*Plus and log in to MYPDB1 as the SYSTEM user.
 

```
$ sqlplus system/password@MYPDB1
```
  - d. Execute the create\_fsowner.sql script to create the FSOWNER user.
 

```
SQL> @$HOME/labs/migrate/rman/create_fsowner.sql
```
  - e. Exit from SQL\*Plus.
 

```
SQL> exit
```
5. In preparation for the next step of this practice, drop the FSDATA and FSINDEX tablespaces that you created in an earlier practice. If you did not create these tablespaces in an earlier practice, you can skip this step.
    - a. Invoke SQL\*Plus and log in to MYPDB1 as the SYSTEM user.
 

```
$ sqlplus system/password@MYPDB1
```
    - b. Execute the DROP TABLESPACE command for the FSINDEX tablespace.
 

```
SQL> DROP TABLESPACE fsindex INCLUDING CONTENTS AND DATAFILES;
```
    - c. Execute the DROP TABLESPACE command for the FSDATA tablespace.
 

```
SQL> DROP TABLESPACE fsdata INCLUDING CONTENTS AND DATAFILES;
```
    - d. Exit from SQL\*Plus.
 

```
SQL> exit
```
  6. On the Database Cloud Service compute node, invoke RMAN and use the RESTORE command with the `ignoreFileSpec` subclause to restore the cross-platform backup.
    - a. Create a directory for the Data Pump dump file.
 

```
$ mkdir /tmp/datapump
```
    - b. Invoke RMAN and log in to MYPDB1 as a user that has been granted the SYSDBA or SYSBACKUP privilege. Use SYS if you do not want to create another user.
 

```
$ rman target SYS/password@MYPDB1
```
    - c. Execute the RESTORE command.
 

```
RMAN> RUN
{
  ALLOCATE CHANNEL CHANNEL1 DEVICE TYPE DISK;
  RESTORE FOREIGN TABLESPACE fsdata,fsindex TO NEW
    FROM BACKUPSET '/tmp/from_onprem/fs_tbs.bck'
    DUMP FILE DATAPUMP DESTINATION '/tmp/datapump'
    FROM BACKUPSET '/tmp/from_onprem/fs_tbs.dmp';
}
```
    - d. Exit from RMAN.
 

```
RMAN> exit
```
  7. On the Database Cloud Service compute node, set the tablespaces to READWRITE mode.
    - a. Invoke SQL\*Plus and log in to MYPDB1 as the SYSTEM user.
 

```
$ sqlplus system/password@MYPDB1
```
    - b. Set the FSDATA and FSINDEX tablespaces to READWRITE.
 

```
SQL> ALTER TABLESPACE fsdata READ WRITE;
SQL> ALTER TABLESPACE fsindex READ WRITE;
```
  8. You can verify the data by querying FSOWNER.CUSTOMERS.

- a. Query `FSOWNER.CUSTOMERS`.

```
SQL> select count(*) from fsowner.customers;
```

- b. Exit from SQL\*Plus.

```
SQL> exit
```

9. After verifying that the data has been imported successfully, you can delete the backup set files that were transported from the classroom machine.

## Practice 6-1 Using SQL Developer and INSERT Statements to Migrate Selected Objects

### Overview

In this practice SQL Developer is used to create a cart containing selected objects to be loaded into an Oracle Database 12c database on Oracle Database Cloud Service. You will use SQL `INSERT` statements generated by SQL Developer to create the objects and then load the data into your cloud database.

Before this class started, the following tasks were completed to create the file used in this practice:

1. Use SQL Developer to create a connection to the PDB as the `FSOWNER` user.
2. Click View -> Cart to open a new cart.
3. Expand tables, select the `FSOWNER.CUSTOMERS` table, and click Add to Cart.
4. Select Data to include the table data.
5. Click the Export Cart icon.
6. Select "insert" in the Format menu and click Apply.

### Tasks

You will now use the pre-created file to create a table and load data into your Database Cloud Service database.

1. If you have not already done so, enable the `ora_p2_dblistener` access rule to unblock port 1521 for SQL Developer.
2. If you are using the Oracle classroom machine, create an SSH tunnel to port 1521 on your DBCS deployment.
  - a. Use the command:
 

```
$ ssh -i <privatekey> -L 6501:<public_IP>:1521 oracle@<public_IP>
```
  - b. Leave this session open as long as you are connecting to your DBCS database with SQL Developer
3. In SQL Developer, create a new connection to `MYPDB1`.
  - a. Right-click **Connections** and select **New Connection**.
  - b. Fill in the fields. You can use `SYS` or `SYSTEM` as the user. If using ssh tunneling in the Oracle classroom, provide `localhost` in the Hostname field and the full PDB service name in the Service name field. See the Connect string shown by expanding Additional Information section of the DBCS deployment Detail on the Service console page. The Connect string is in the format `<DBCS_Name>:1521/<PDB Service Name>`. If using some machine outside the Oracle classroom, use the Connect string, including the `<DBCS_Name>` and port.
  - c. Click Connect to establish the connection.
4. In preparation for the next step, drop the `FSOWNER.CUSTOMERS` table that you created in an earlier practice. If you do not have a `FSOWNER.CUSTOMERS` table in `MYPDB1`, you can skip this step.
  - a. In a SQL Developer Worksheet, execute the `DROP TABLE` command to drop the `FSOWNER.CUSTOMERS` table.
 

```
DROP TABLE fsowner.customers;
```
5. Use the SQL Developer script to create the `FSOWNER.CUSTOMERS` table and load data into it. Make sure you are outside VPN.
  - a. In the menu click **File** and then click **Open**.
  - b. Navigate to the `~/labs/migrate/sqldev` folder, select the `export_customers.sql` file and click **Open**.
  - c. The script appears in the Worksheet. Click the **Run Script icon** at the top of the worksheet.
  - d. In the Select Connection box, select `MYPDB1` and click **OK**.
  - e. In the Script Output window you can see the creation of the table and insertion of rows.
6. Optionally, you can use SQL\*Plus to verify the creation of the table and insertion of rows.
  - a. Invoke SQL\*Plus and log in to `MYPDB1` as the `SYSTEM` user.
 

```
$ sqlplus system/password@MYPDB1
```

- b. Execute a query against the `FSOWNER.CUSTOMERS` table.

```
SQL> SELECT count(*) FROM fsowner.customers;
```

- c. Exit from SQL\*Plus.

```
SQL> exit
```

# Practicing Performance

- Practice 3-1 Scaling Up the Database Deployment

## Practice 3-1 Scaling Up the Database Deployment

### Overview

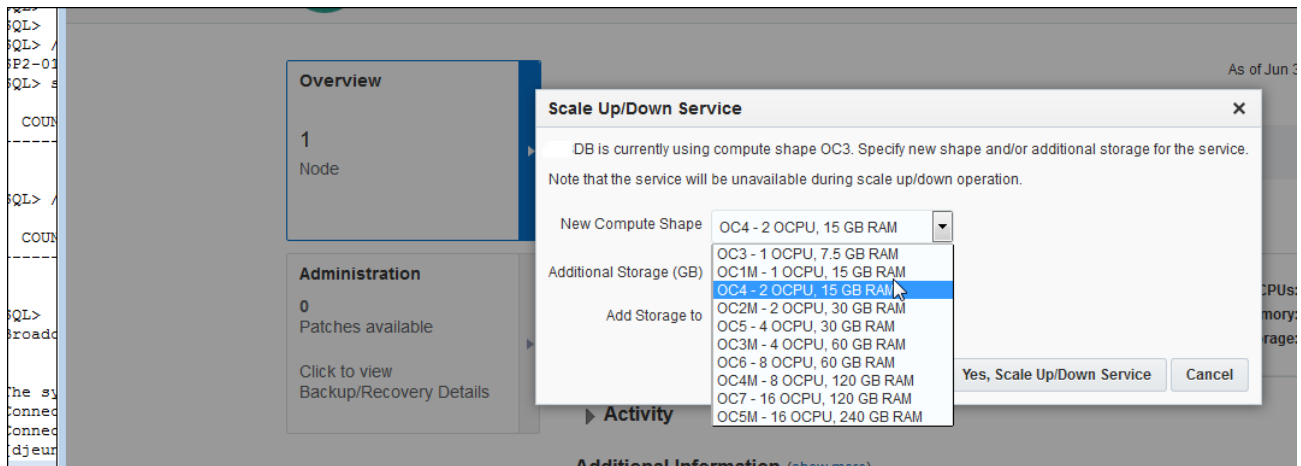
In this practice, you will scale up your database deployment, considering the following requirement.

- You need two CPUs and 15 GB RAM.

### Tasks

- Connect to the compute node of your database deployment.
- With SQL\*Plus, connect to one PDB of the CDB of your database deployment.
- Execute any SELECT statement.
- In the Database Cloud Service, open the service console and follow the steps described in [Scaling the Compute Shape for a Database Deployment](#).

After having chosen the right combination of CPU/RAM resources and confirmed the scale up operation,



the database instance and compute node are shut down.


```
SQL> /

COUNT(*)
-----
         3

SQL>
Broadcast message from root@MYDBCS
      (unknown) at 16:26 ...

The system is going down for power off NOW!
Connection to 129.144.21.73 closed by remote host.
Connection to 129.144.21.73 closed.
[diyunot@den01esm ~]$
```

- You are informed that the database deployment is under maintenance and therefore not available.

**MYDBCS**  
Status: Maintenance  
Version: 12.1.0.2  
Edition: Enterprise Edition


Submitted On: Jun 20, 2016 2:58:35 PM UTC

**OCPUs: 2**  
**Memory: 15 GB**  
**Storage: 137 GB**

☰

6. After about 10 minutes the scale up will be complete. This is visible in the Activity section.

**Activity**

| Activity Summary |                                                                                                                  |                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| ▶                |  <b>Scale-Up/Down Completed</b> | <b>Start Time:</b> Jul 6, 2016 6:31:58 PM UTC<br><b>End Time:</b> Jul 6, 2016 6:40:40 PM UTC |