Step by Step guides to master

Oracle Database Cloud Service – DBaaS

http://www.kamranagayev.com

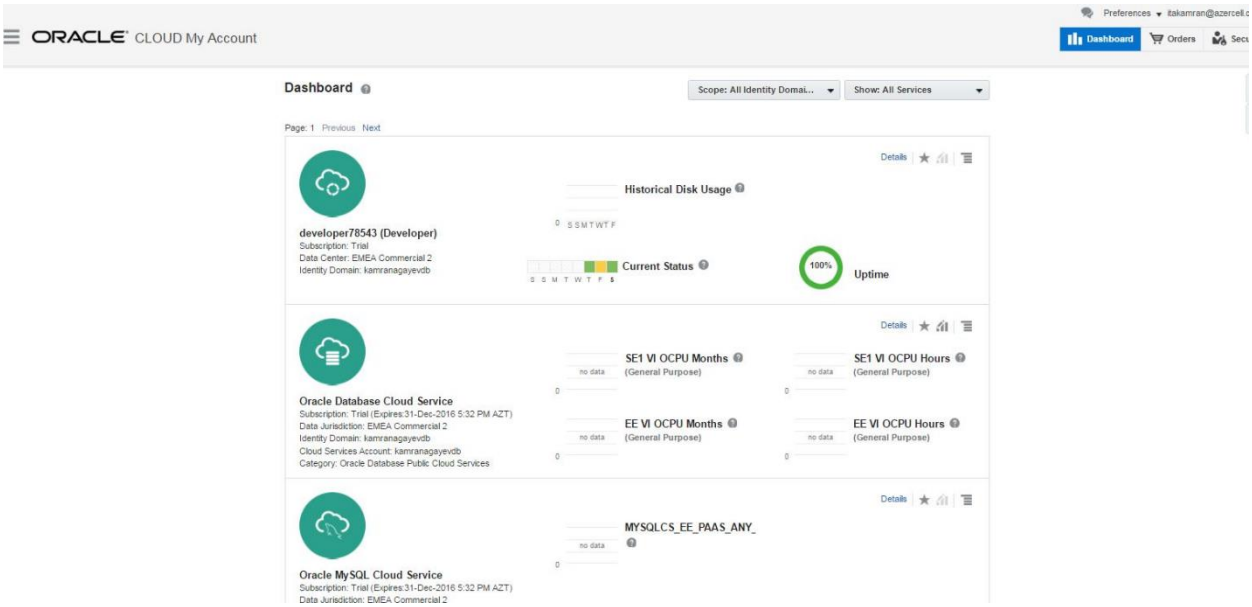
http://www.ocmguide.com

Step by step guide to create an Oracle Database in the Cloud

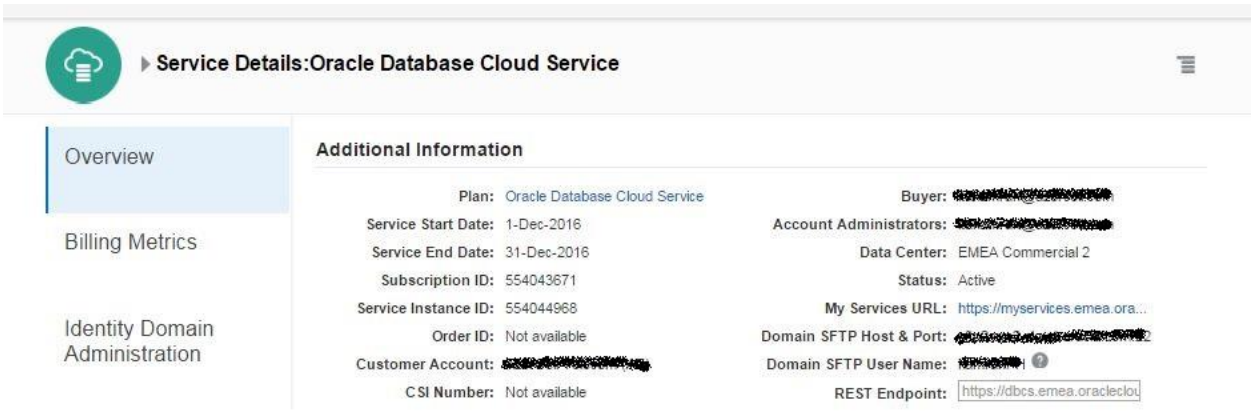
In this blog post I will share the steps to create an Oracle Database in Cloud. We will create a database service, create a second database in the same machine in a silent mode, access to the cloud machine with SSH and monitor the database with OEM.

First of all, make sure you have an Oracle account. Open oracle.com and click on Register link to get a free oracle account. Next, open cloud.oracle.com, select “Compute” from Infrastructure menu and click “Try It” button to get a free trial account for 1 month. Next, login with your oracle account and register for cloud account. Make sure you don’t apply with generic email addresses like hotmail, gmail and etc. Instead, use your company address.

After successfully registering you will get an email with your credential information. Click on the link specified in the email, provide username/password and login to your cloud account. Click Oracle Database Cloud Service link.

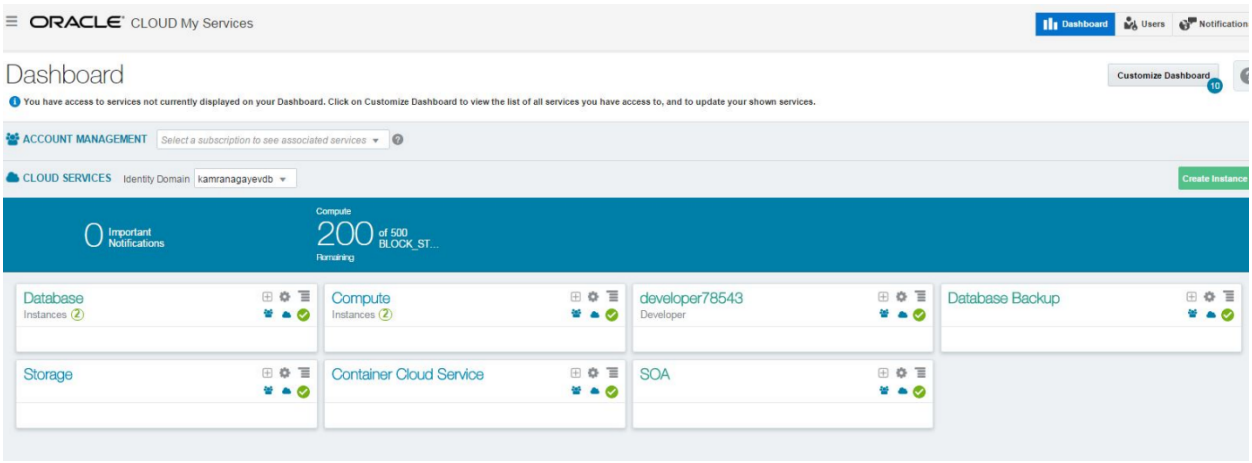


Click on the link under “My Service URL” to access list of cloud services you have.

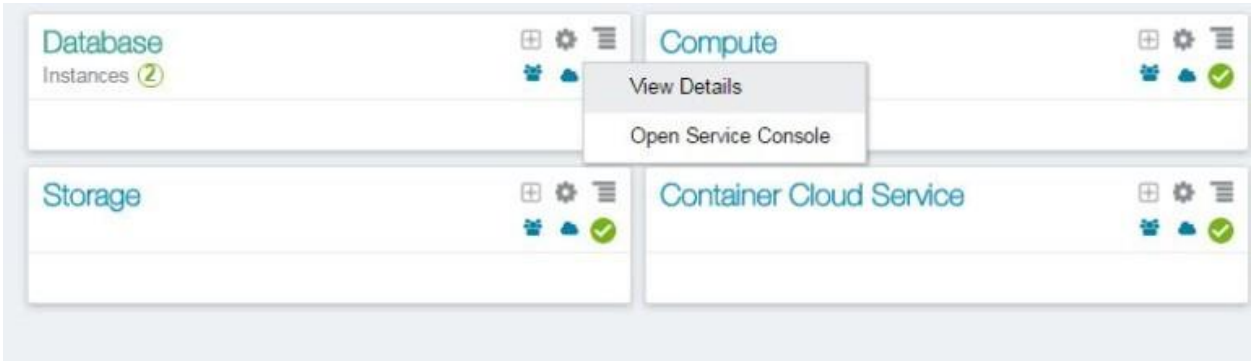


If you haven't signed in yet, sign in page will appear. Use the username and password provided in the email you got during the registration and login

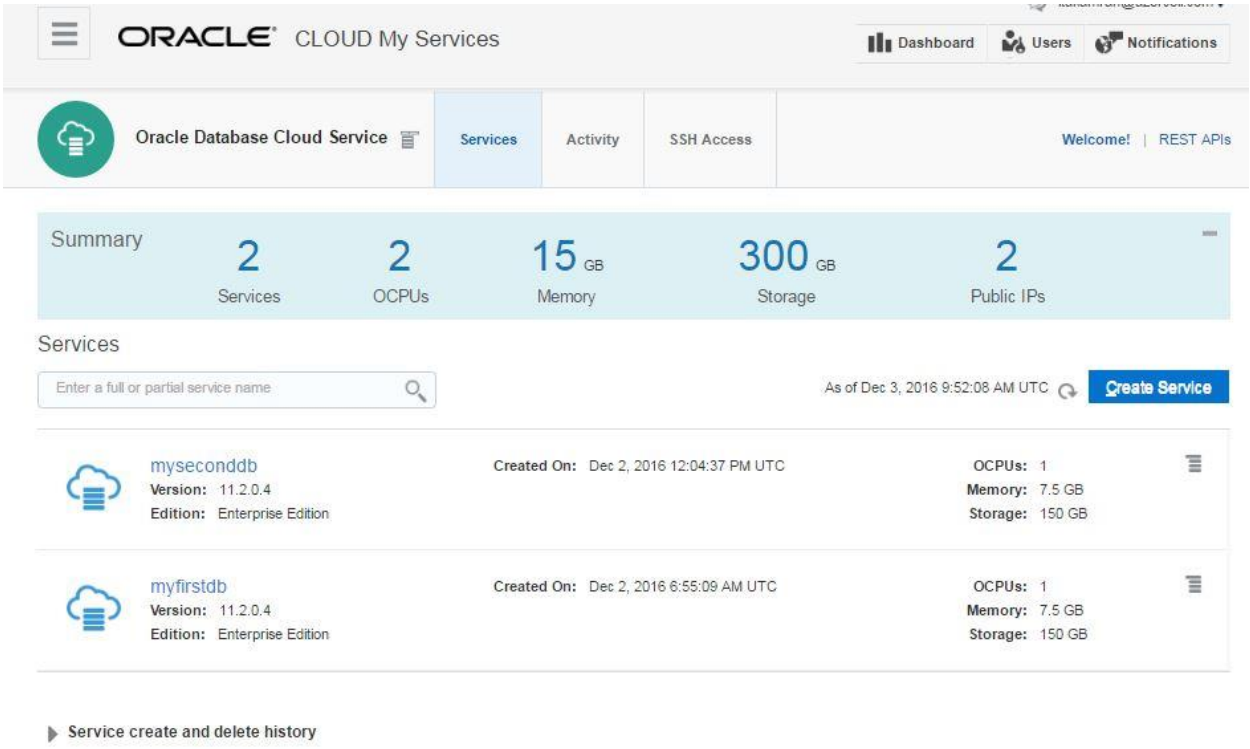
Next, Oracle Cloud Services dashboard will appear.



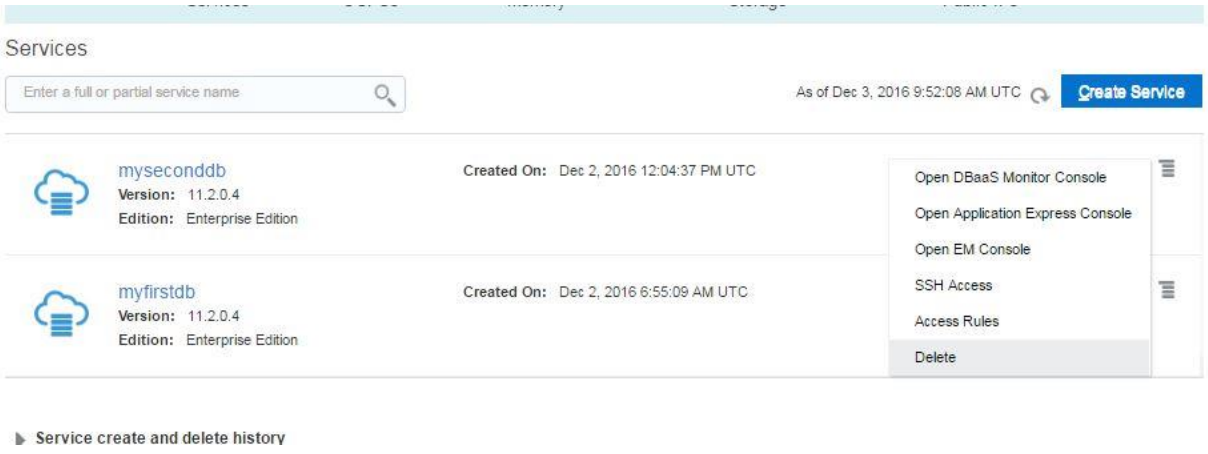
To create and manage database instances, click on the “Open Service Console” menu of the Database tab as shown below:



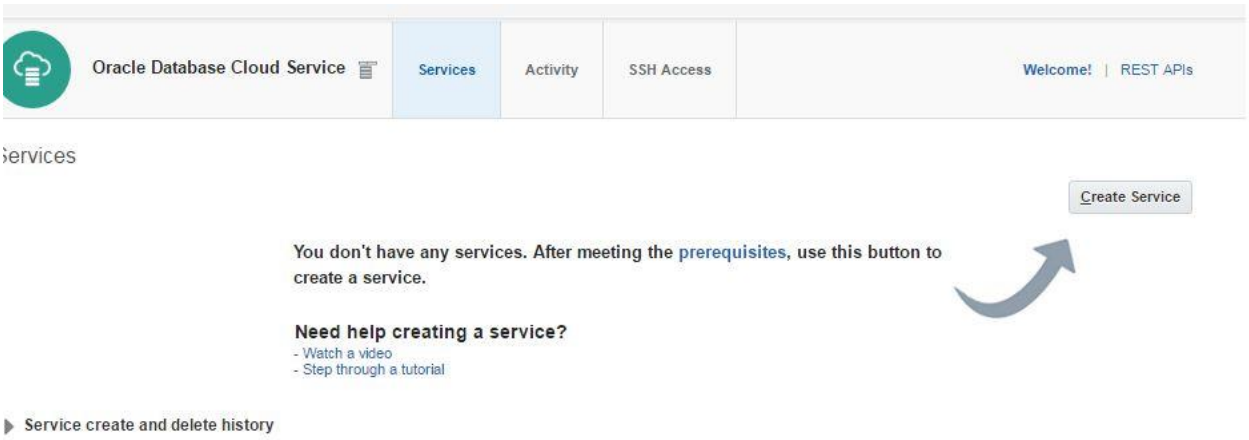
In this page you will get list of database services. Here, I have two database services (myfirstdb, myseconddb). I have 7.5G memory and 150gb of storage provided for each service.



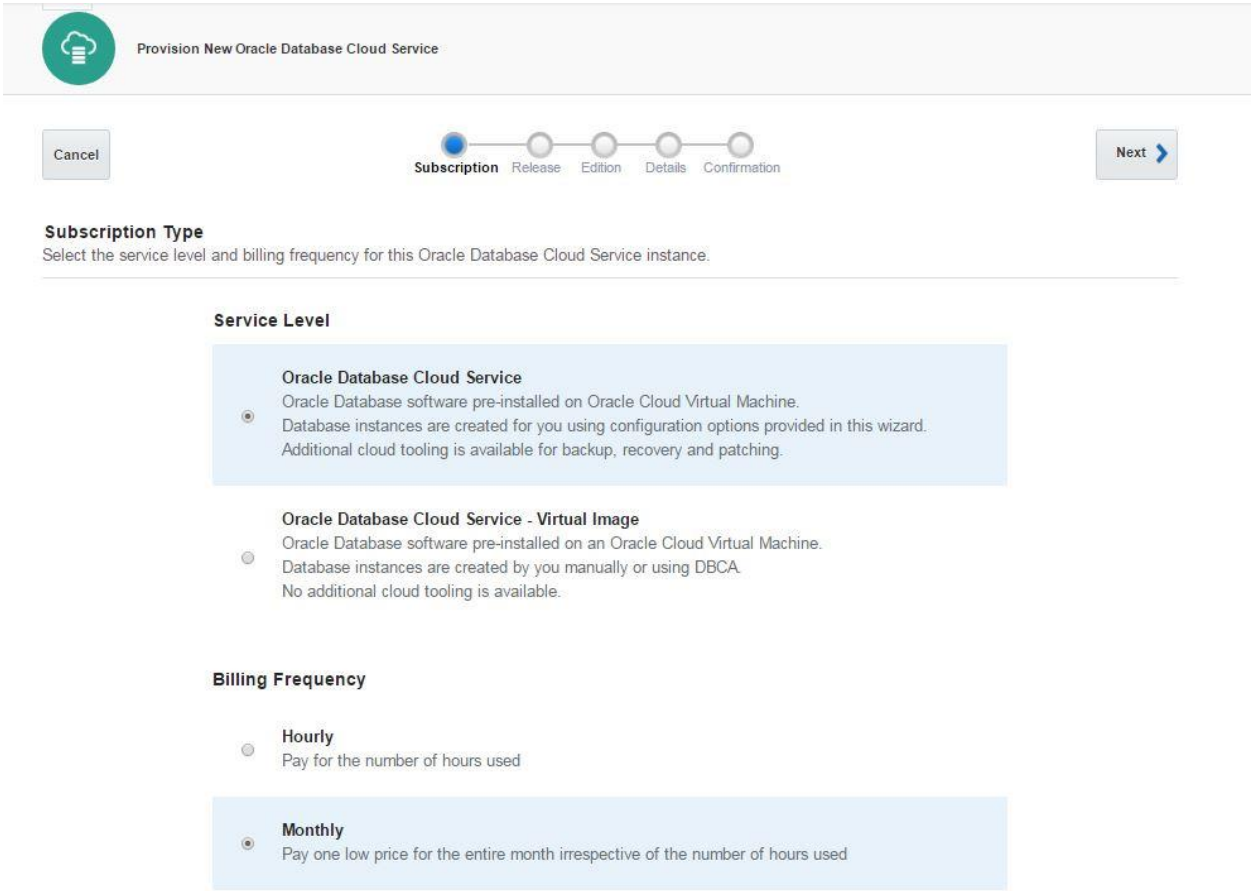
Click Delete on the drop down menu for each service and delete both services.



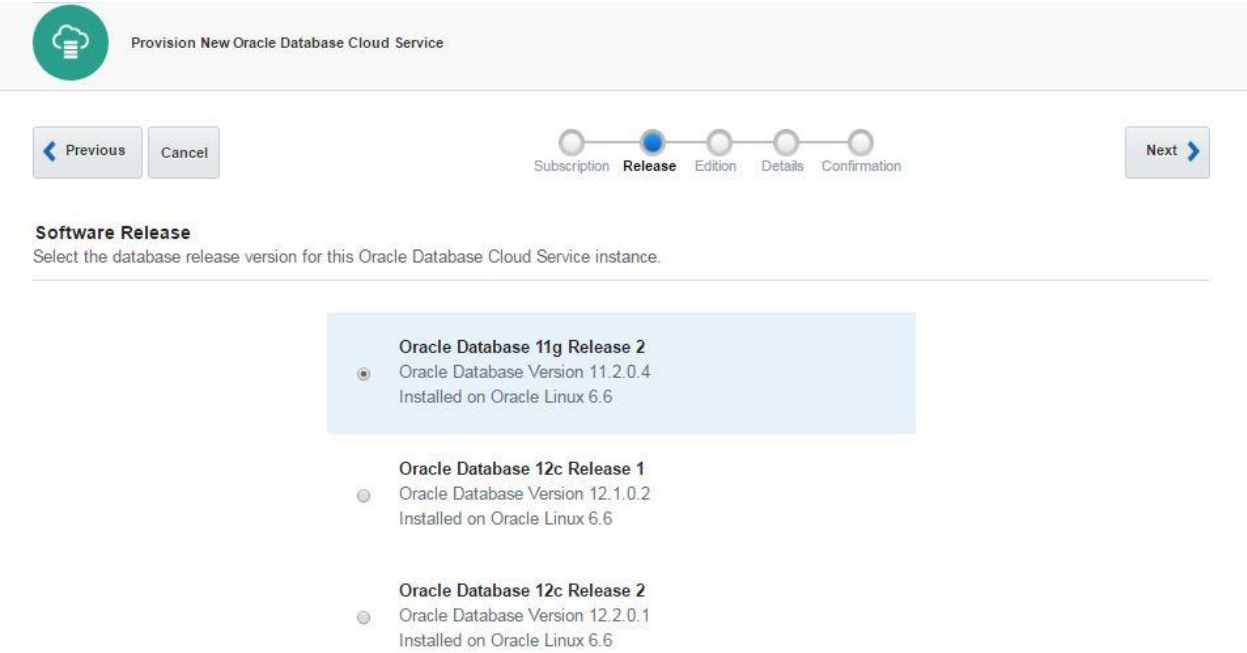
After successfully deleting available instances, click on “Create Service” button to create a new database instance:



Select “Oracle Database Cloud Service” option to create a database using a wizard. For the billing frequency you have two options: Hourly and Monthly. It doesn’t make sense when you use a trial account. So select any of them and click Next.



For the software release you have 3 options: 11.2, 12.1 and 12.2. Choose any of them and click Next.



As a software edition, you have 4 options:

Standard Edition

Enterprise Edition

Enterprise Edition – High Performance

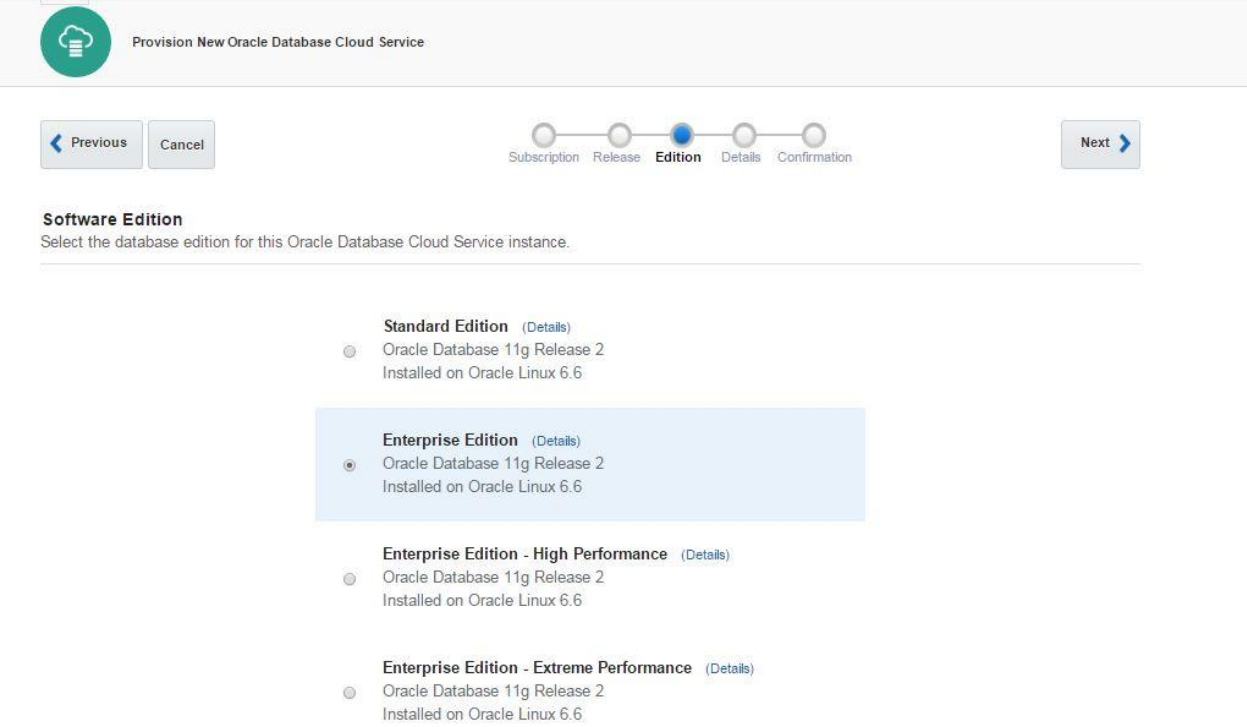
Enterprise Edition – Extreme Performance

To get more information and features that each edition provides, check the following documentation:

Home / Cloud / Oracle Database Cloud Service/ Using Oracle Database Cloud Service/ About Database Cloud Service Database Deployments/ Oracle Database Software Package

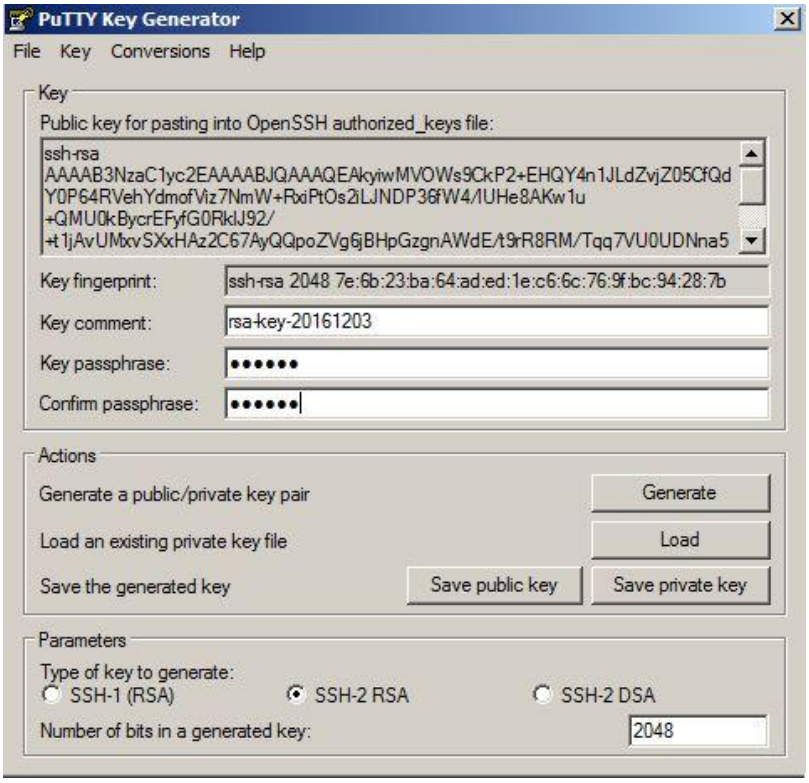
https://docs.oracle.com/cloud/latest/dbcs_dbaas/CSDBI/GUID-660363B8-0E2F-4A4F-A9BD-70A43F332A16.htm#GUID-6C0B1E17-98A5-4522-A3B9-36EFA05E64F9

Select “Enterprise Edition” option and click Next

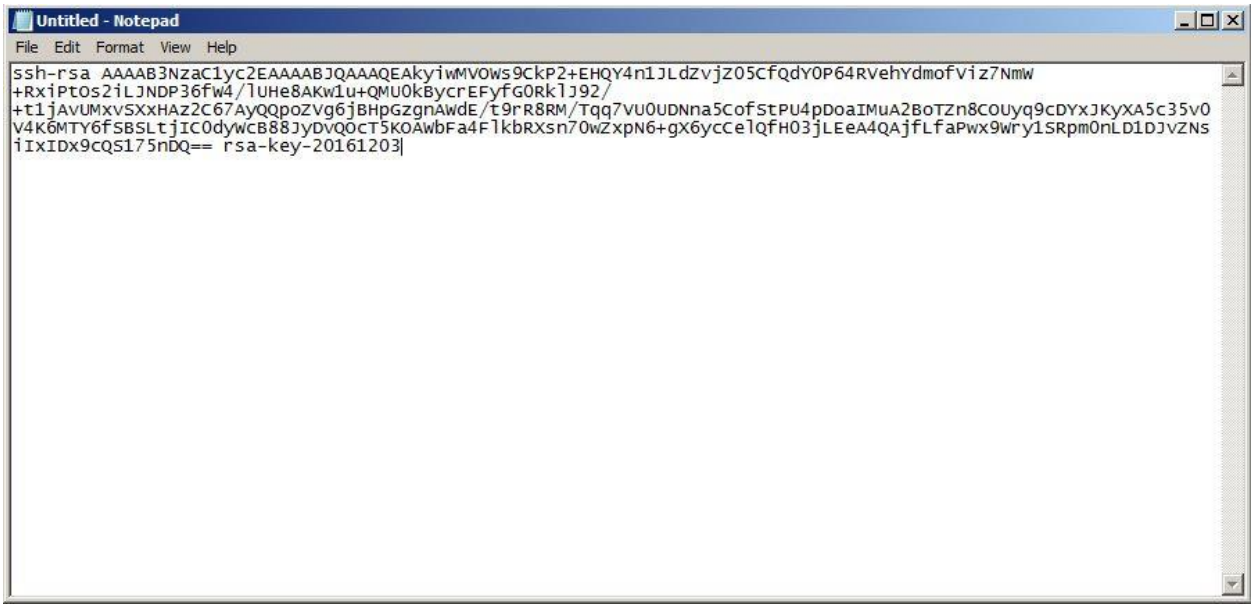


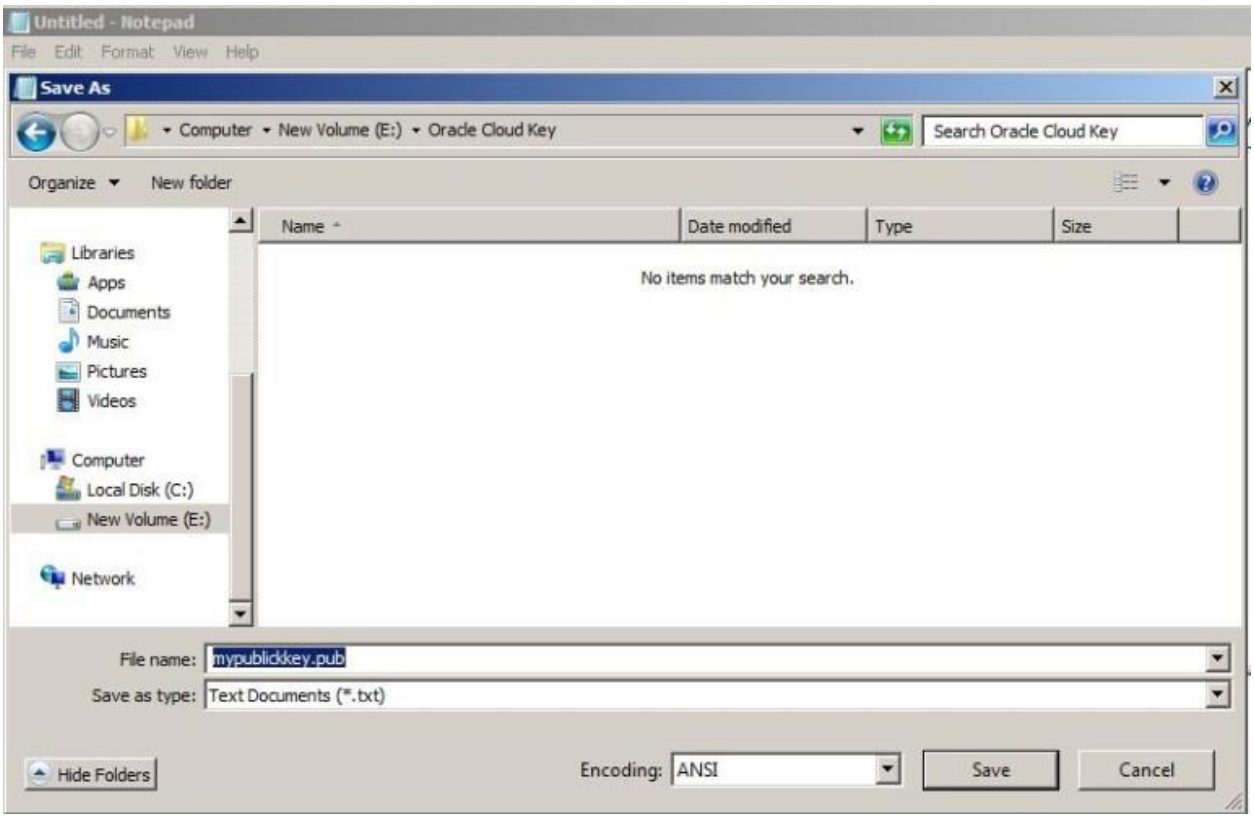
In the next page you will be asked to provide a SSH Public Key for the cloud service. For this, use Putty Key Generator executable to generate a SSH public key. Click on Generate button, move your mouse over the blank are to generate some action.

The SSH key will be generated as follows. Provide a password and click on “Save private key” button to save this key as a private key

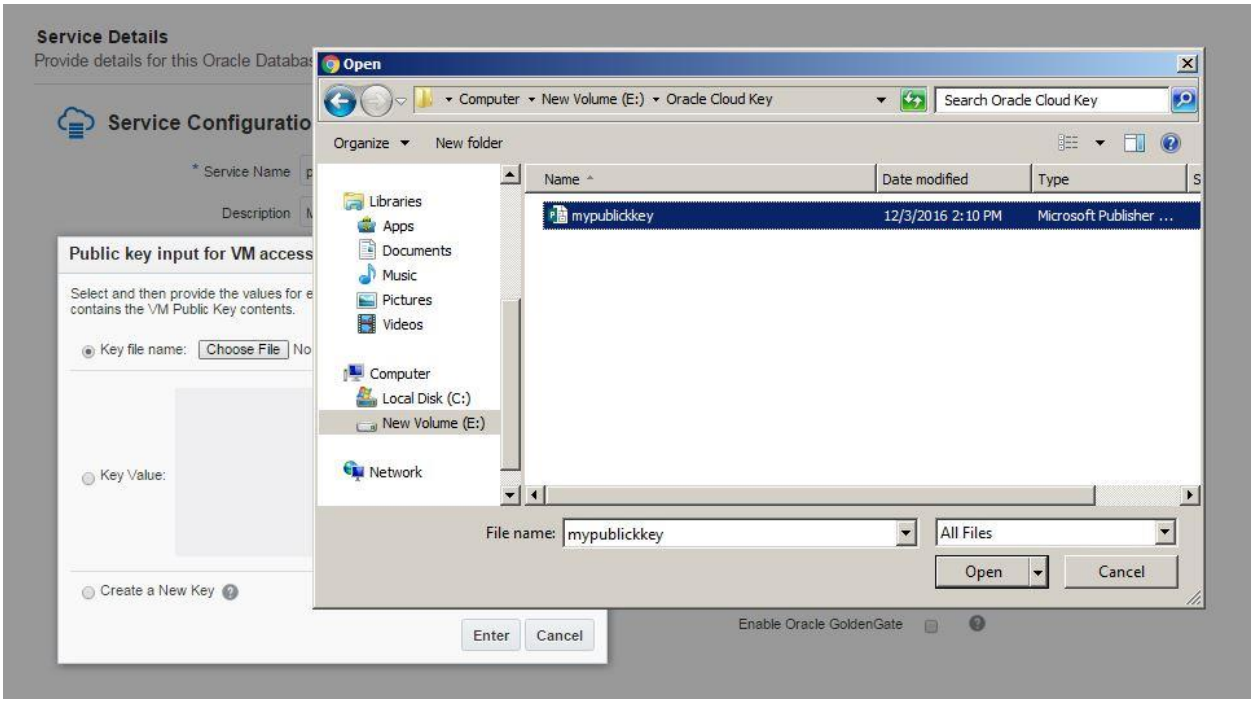


To save a key as a public key, copy the text and save it.

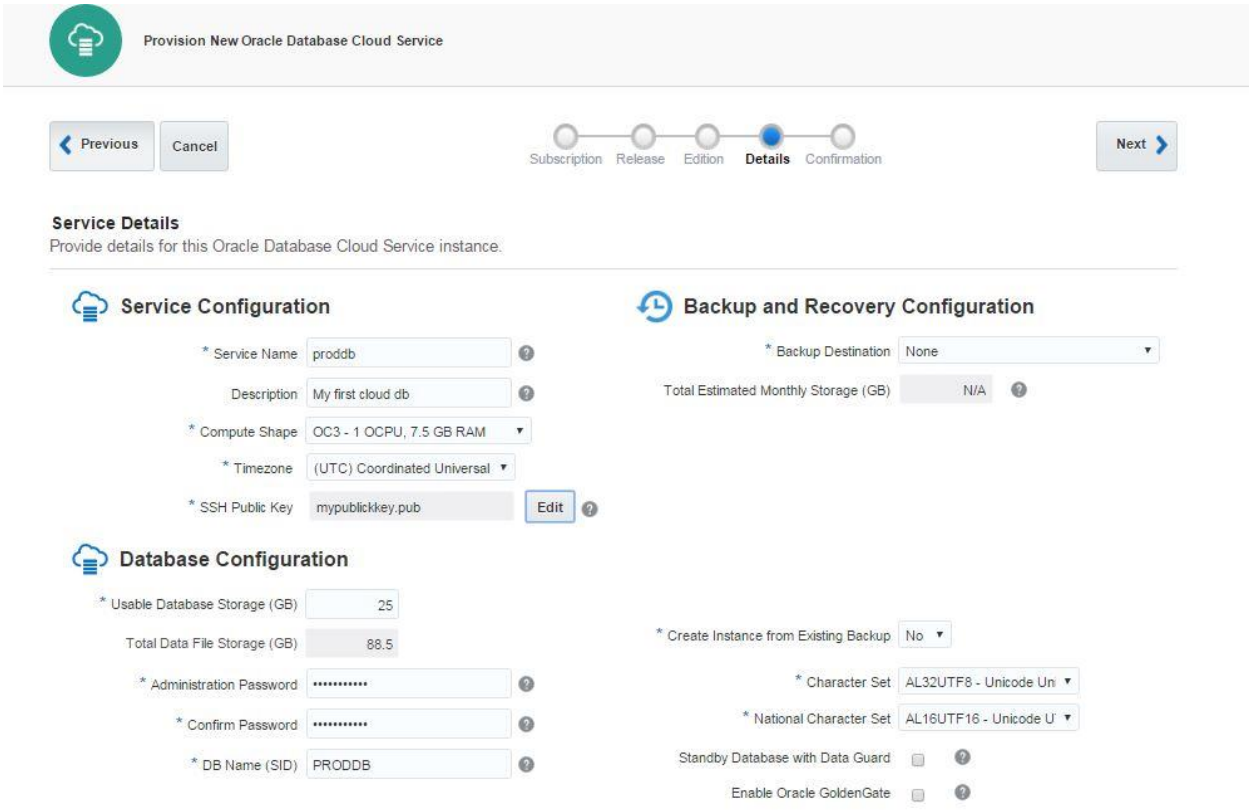




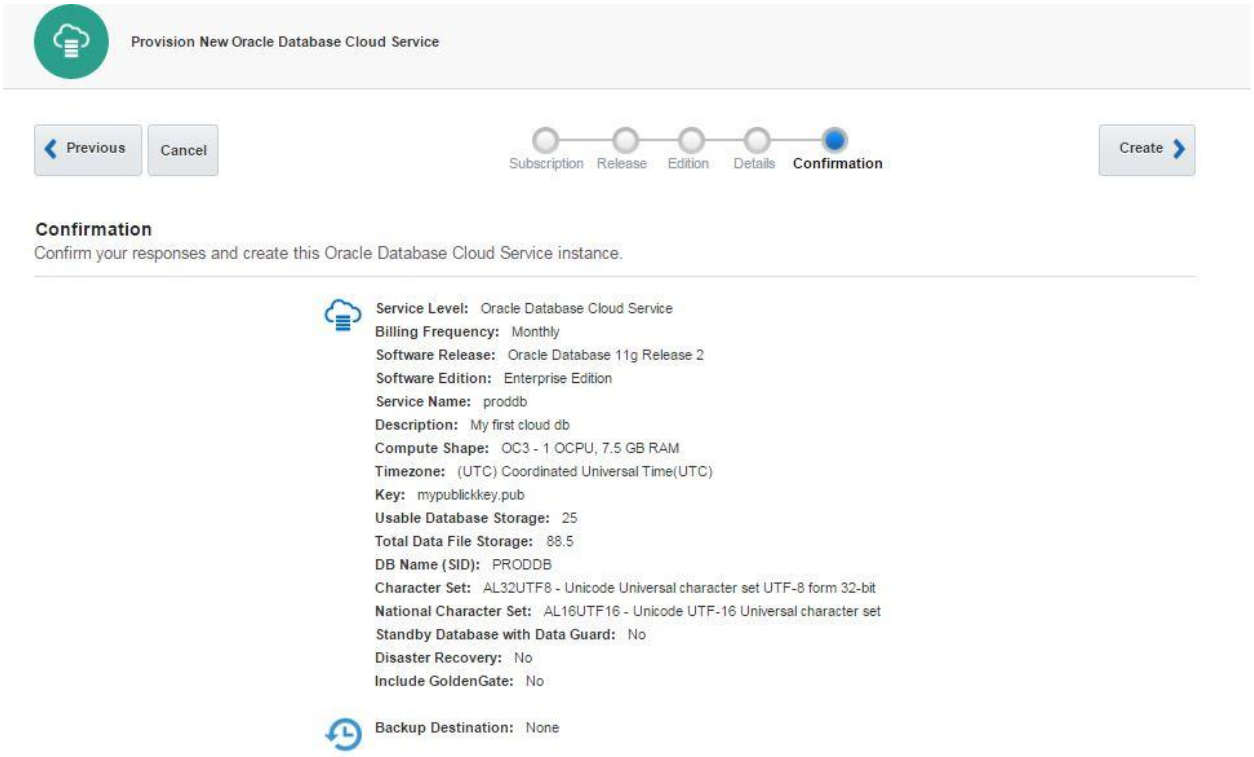
Provide this file for the “SSH Public Key” field of the database service creation wizard.



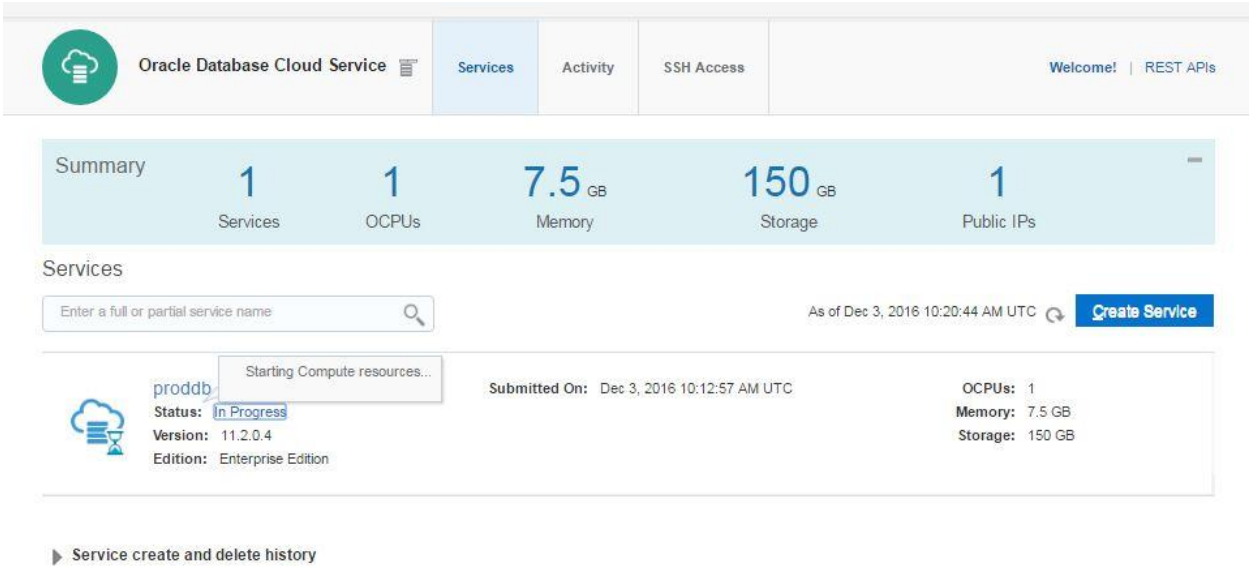
Provide the service name, database storage, select necessary compute shape and click Next.



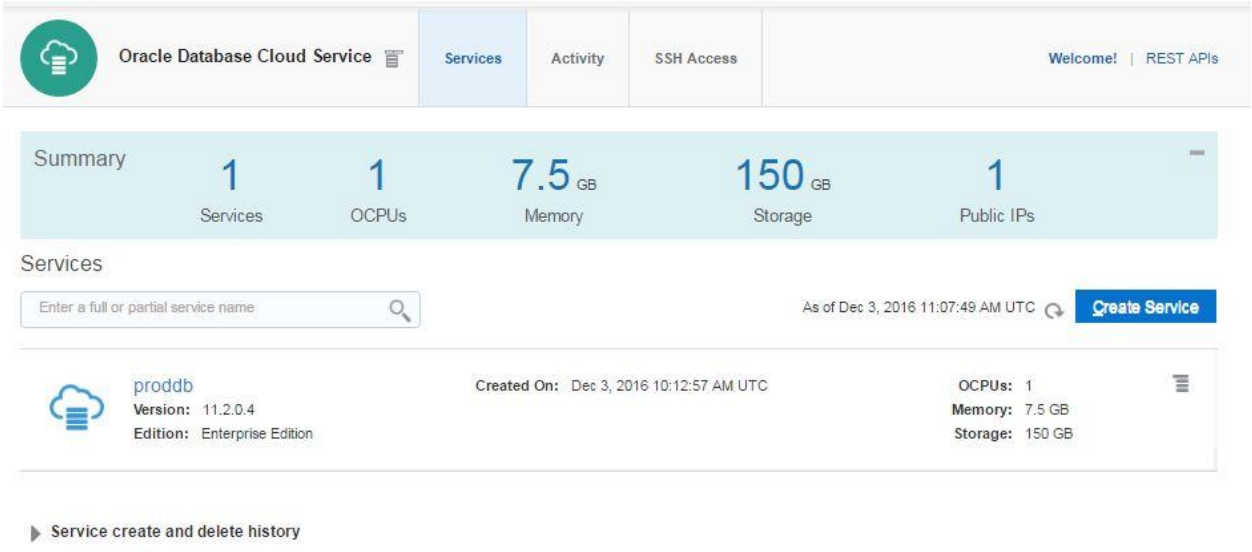
Review the information and click Create button to create a service



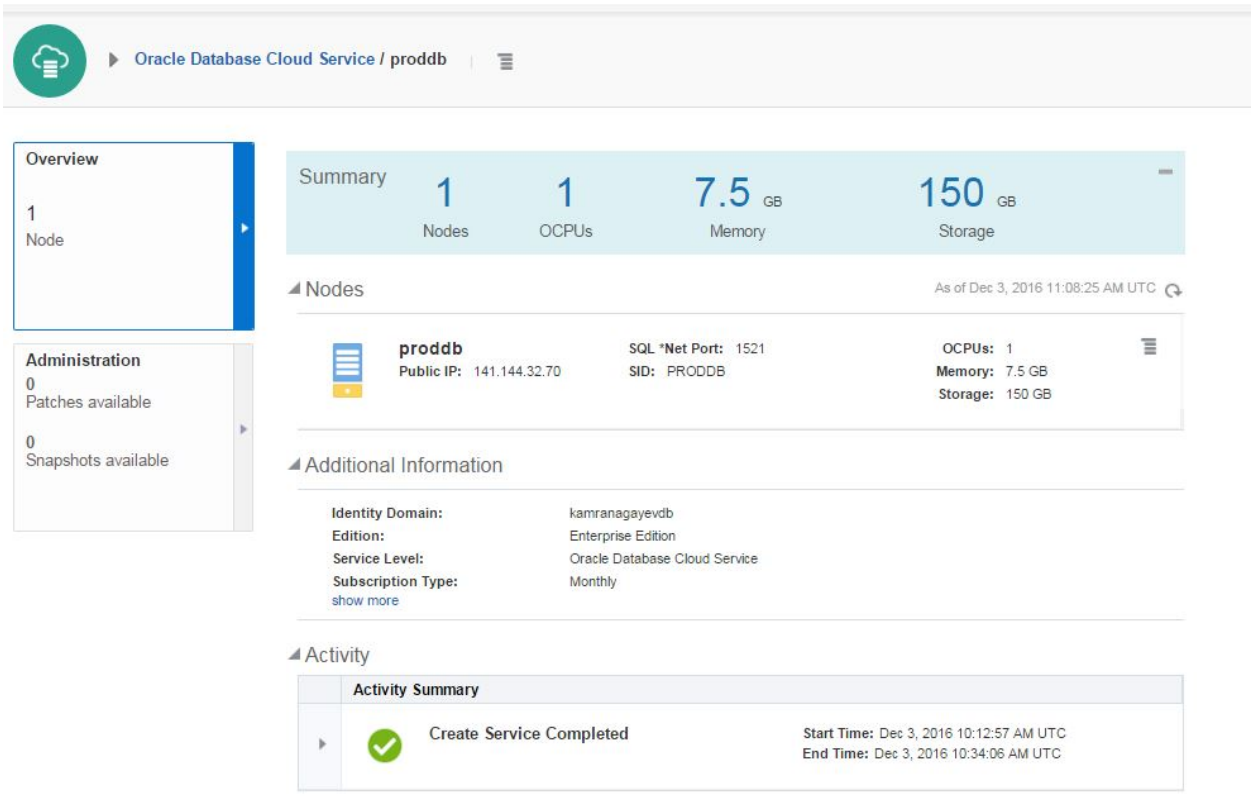
Click “In Progress” link to check the service creation progress.



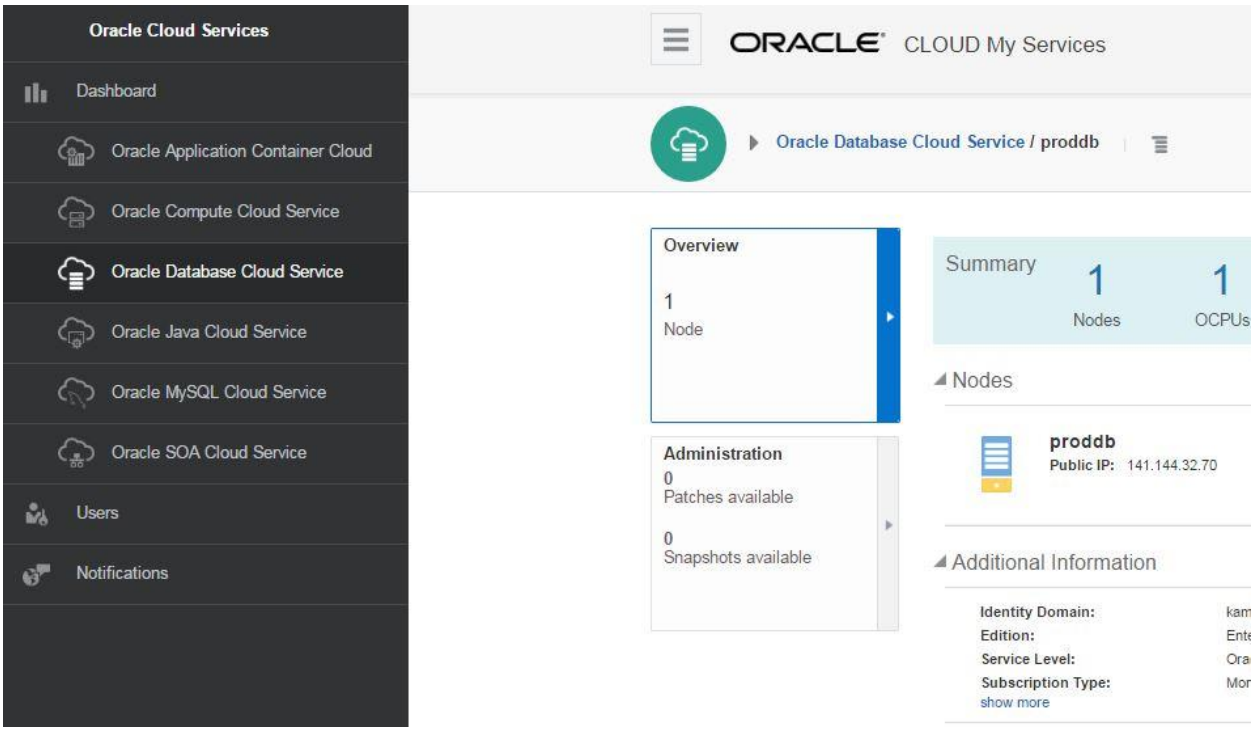
After a while, the service will be created successfully.



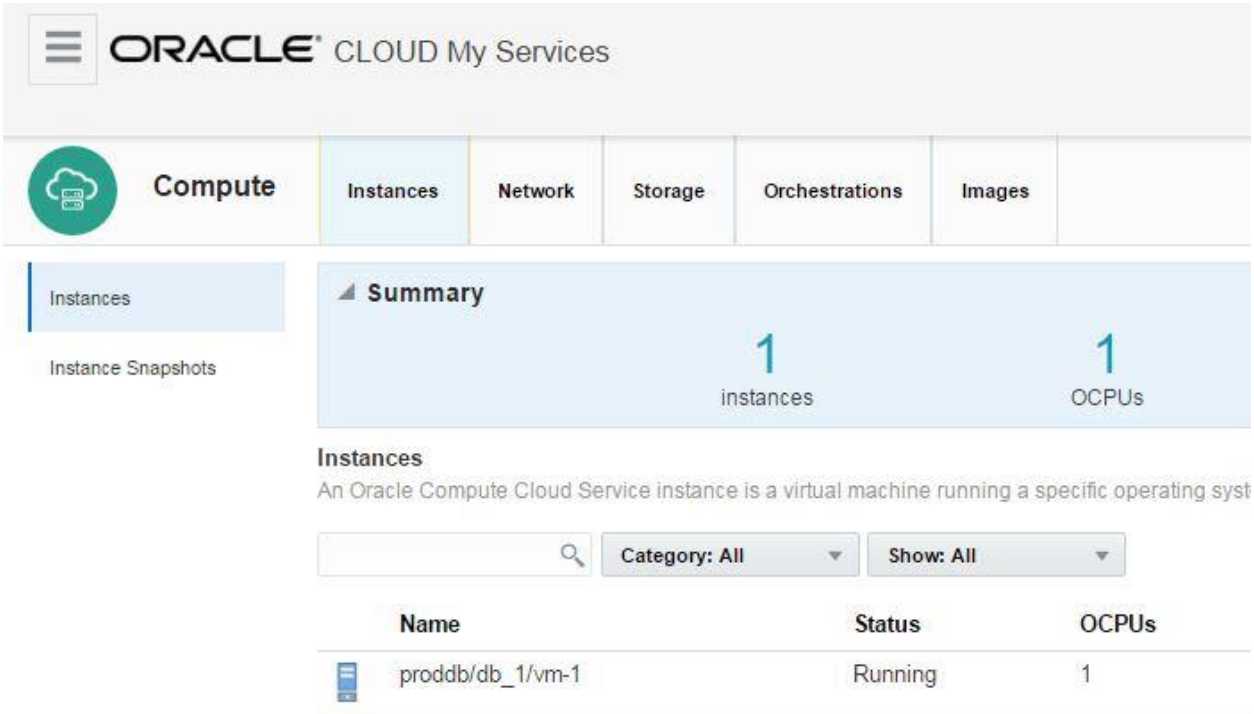
Click on proddb link to open the service.



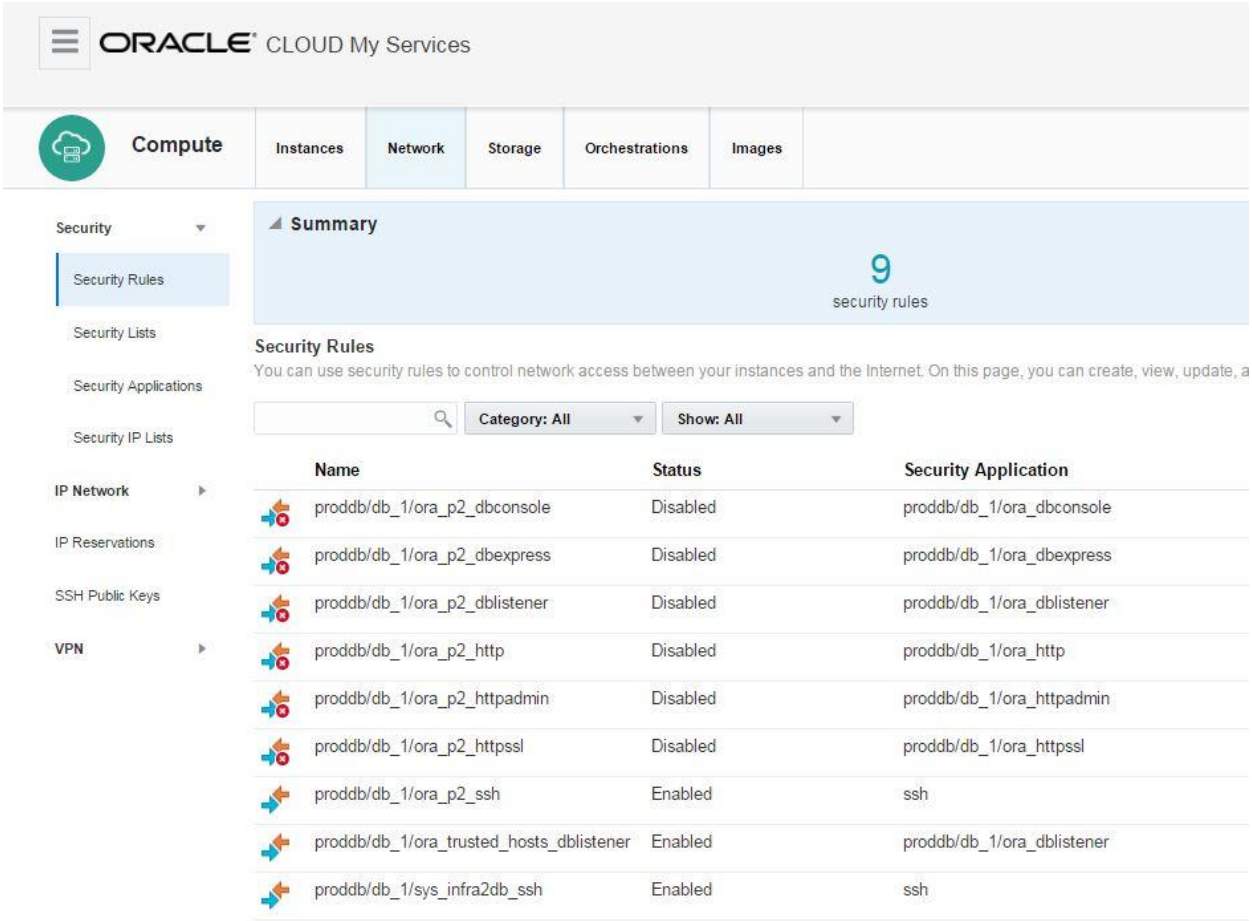
Before trying to connect to the database in the cloud from outside, you should enable dblistener security rule. Open “Oracle Database Cloud Service” dashboard.



Click on Network tab

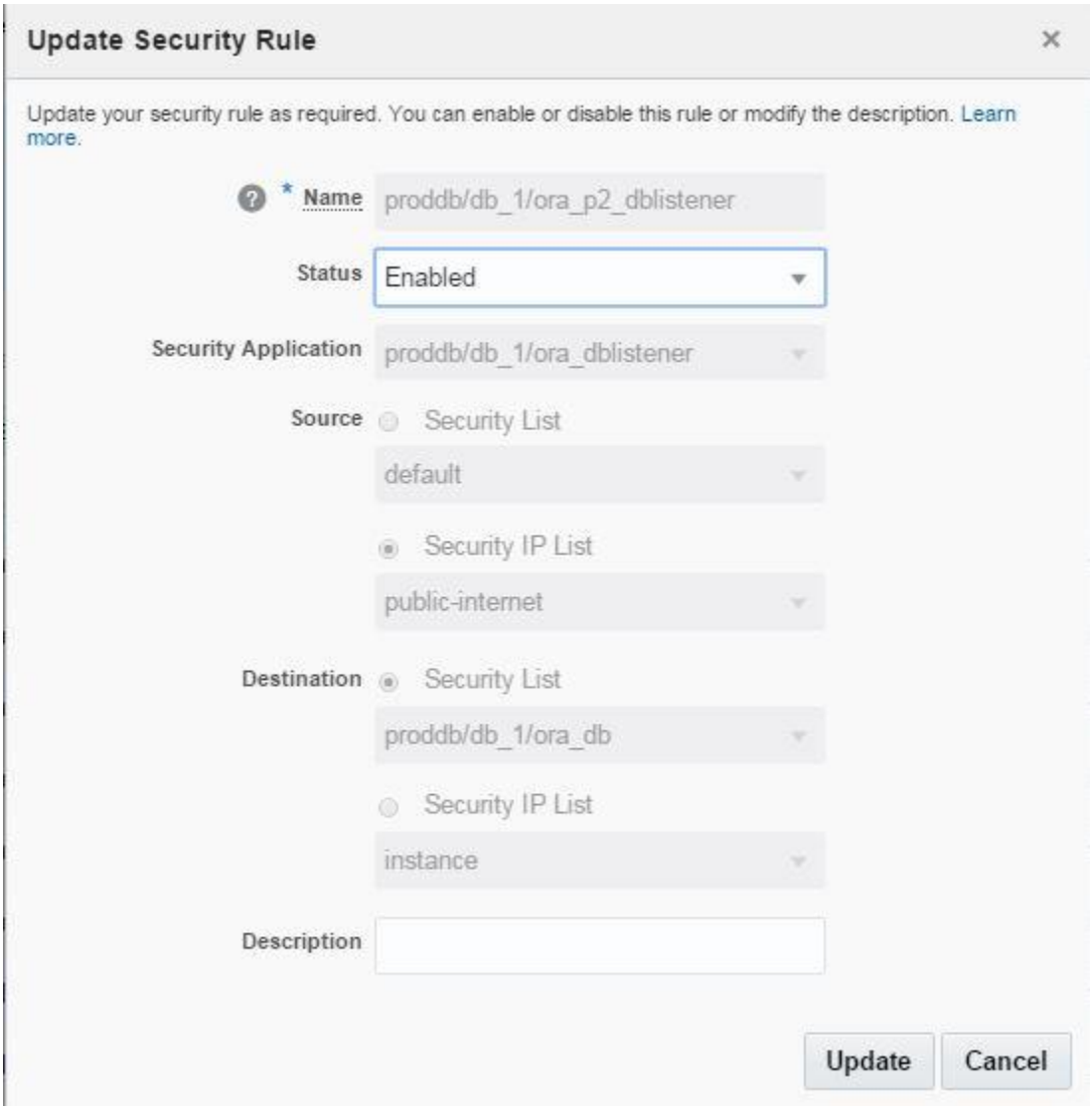


When you click on Network tab, you will get list of security roles.



Click on the menu icon for ora_p2_dblistener role and select Update.

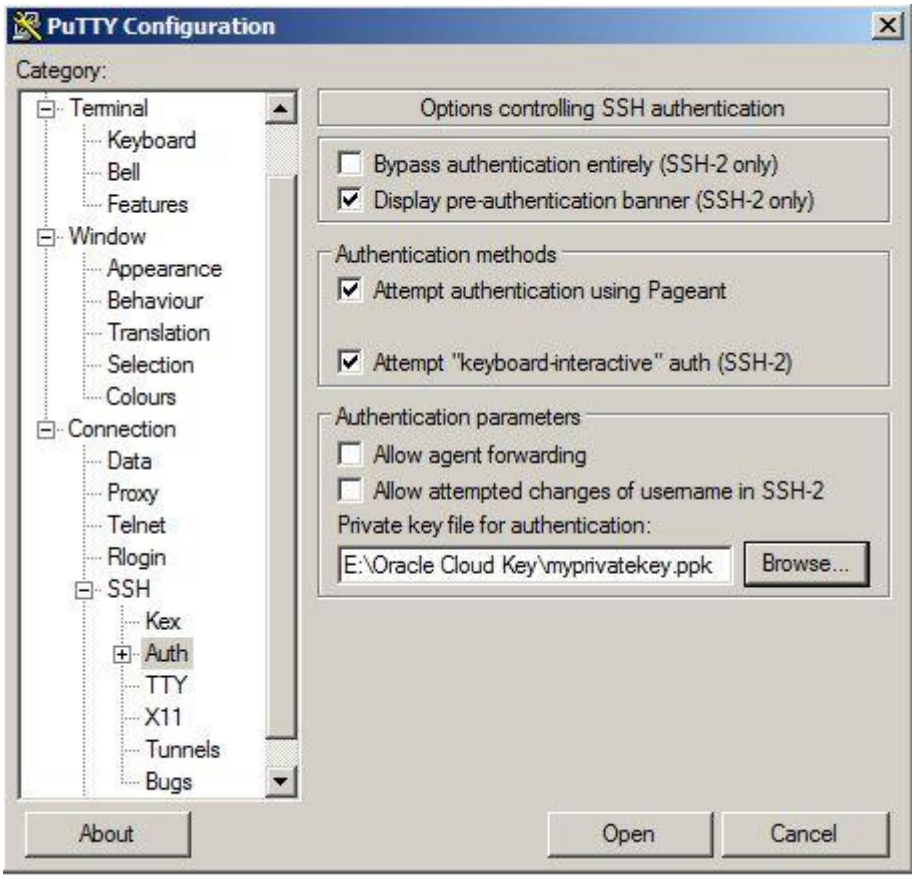
Enable the status of this security role



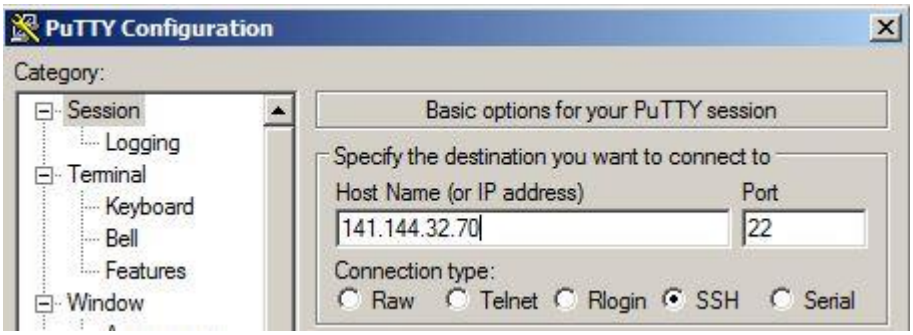
Enable the ora_p2_dbconsole security role to get access to OEM.

Name	Status
proddb/db_1/ora_p2_dbconsole	Enabled
proddb/db_1/ora_p2_dbexpress	Disabled
proddb/db_1/ora_p2_dblistener	Enabled

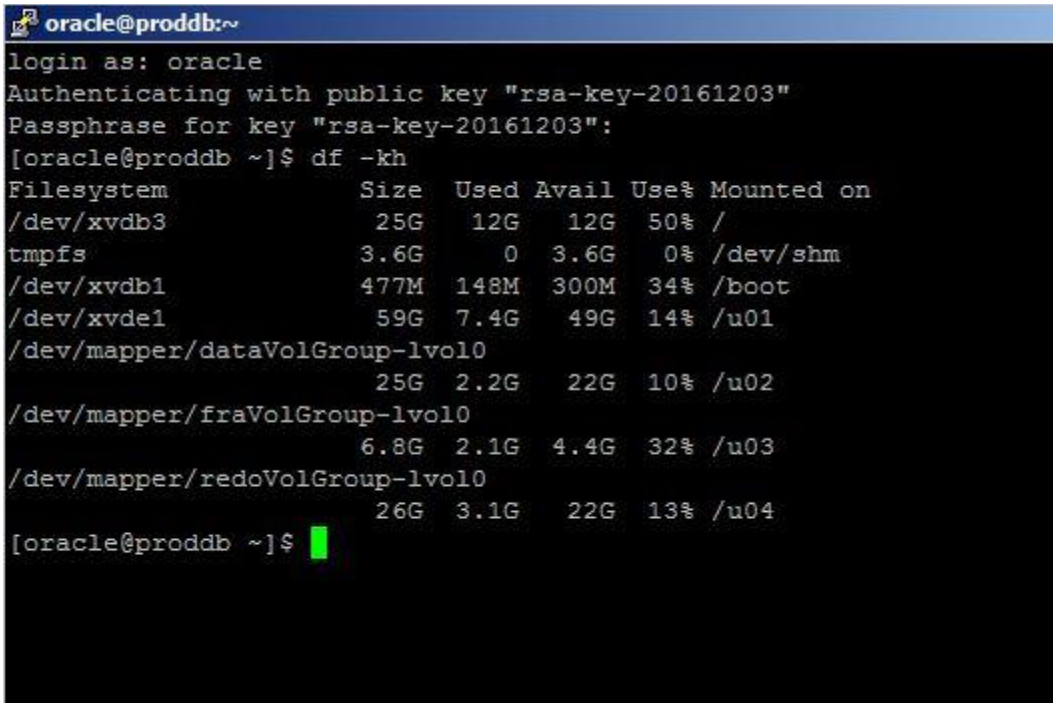
Now, let's connect to the database from SSH. Provide the private key that was saved above

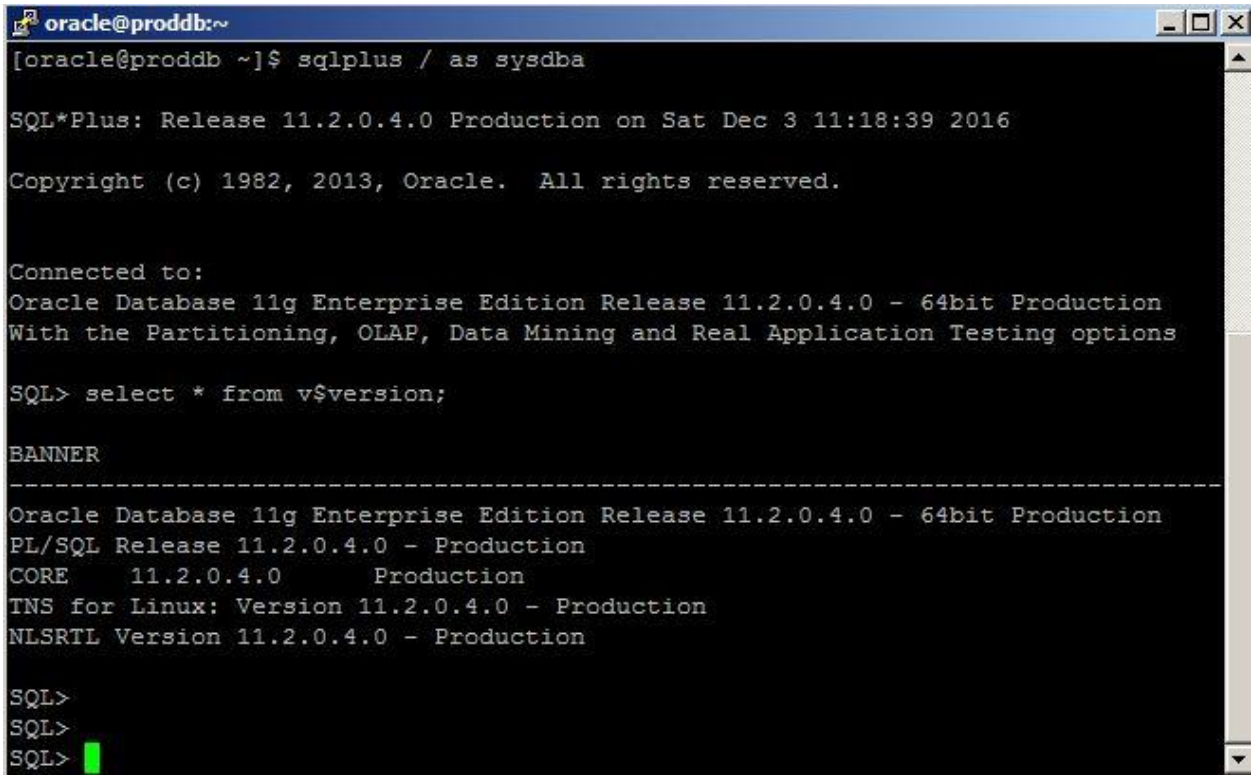


Provide the IP of the virtual machine that is provided in the main page of the database cloud service and click Open.



Provide username as “oracle” and password that was provided when generating a private key using PuTTy Key Generator tool and login to the server where the database is running. Connect to SQL*Plus and run SQL commands:

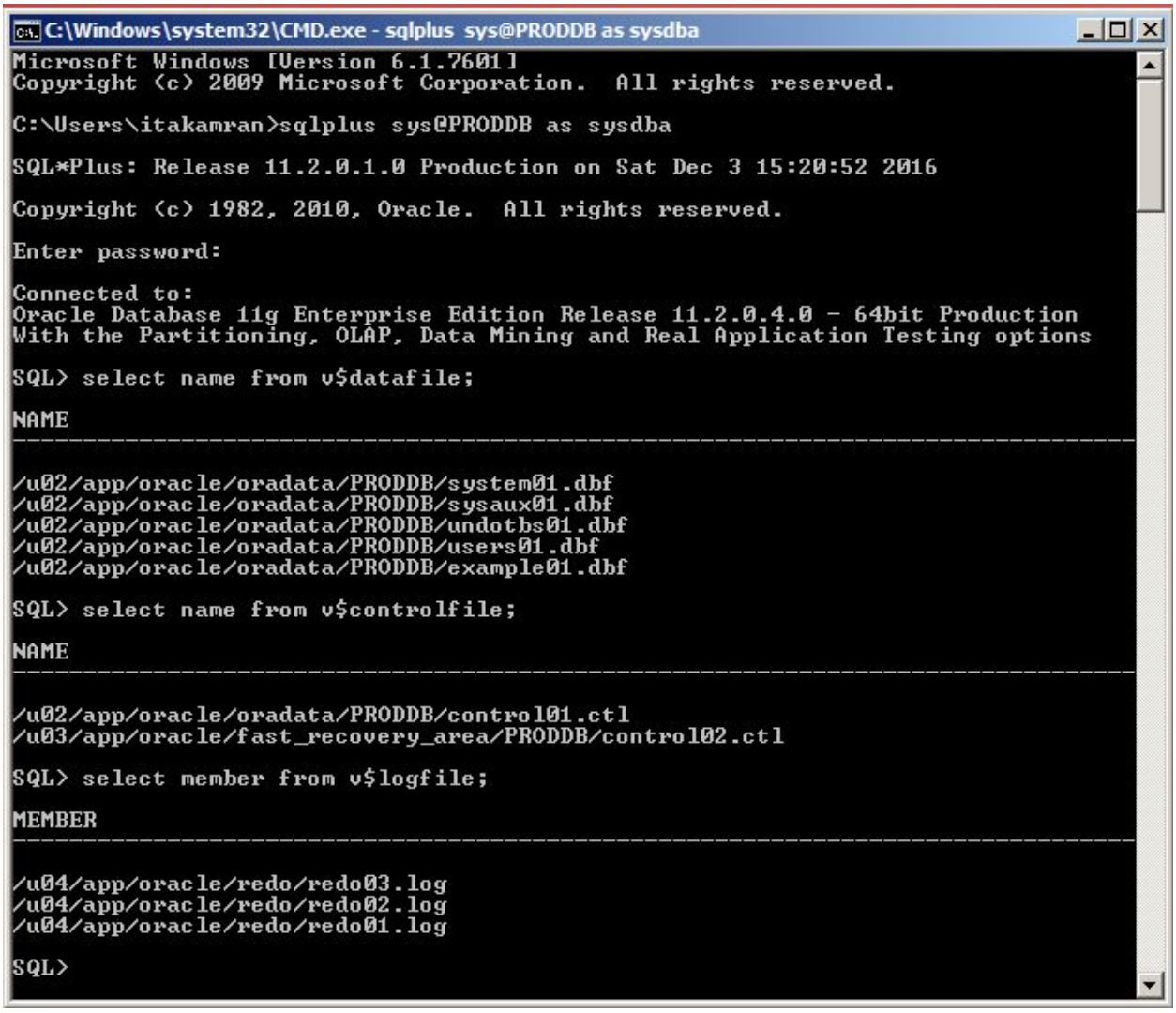

oracle@proddb:~
login as: oracle
Authenticating with public key "rsa-key-20161203"
Passphrase for key "rsa-key-20161203":
[oracle@proddb ~]$ df -kh
Filesystem Size Used Avail Use% Mounted on
/dev/xvdb3 25G 12G 12G 50% /
tmpfs 3.6G 0 3.6G 0% /dev/shm
/dev/xvdb1 477M 148M 300M 34% /boot
/dev/xvde1 59G 7.4G 49G 14% /u01
/dev/mapper/dataVolGroup-lvol0
25G 2.2G 22G 10% /u02
/dev/mapper/fraVolGroup-lvol0
6.8G 2.1G 4.4G 32% /u03
/dev/mapper/redoVolGroup-lvol0
26G 3.1G 22G 13% /u04
[oracle@proddb ~]$


oracle@proddb:~
[oracle@proddb ~]$ sqlplus / as sysdba
SQL*Plus: Release 11.2.0.4.0 Production on Sat Dec 3 11:18:39 2016
Copyright (c) 1982, 2013, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> select * from v$version;
BANNER
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
PL/SQL Release 11.2.0.4.0 - Production
CORE 11.2.0.4.0 Production
TNS for Linux: Version 11.2.0.4.0 - Production
NLSRTL Version 11.2.0.4.0 - Production
SQL>
SQL>
SQL>

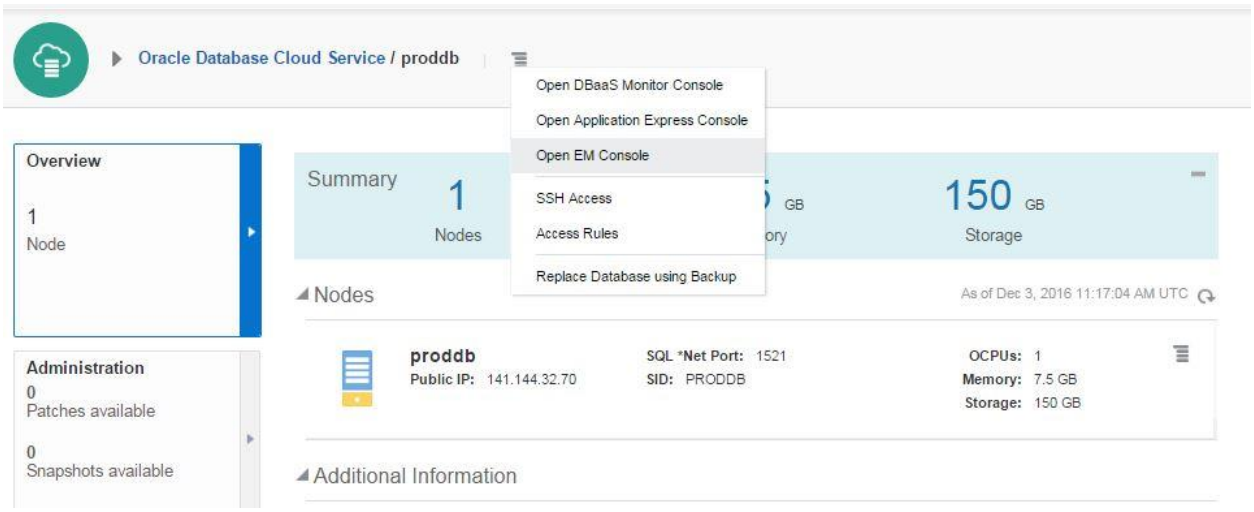
No open tnsnames.ora file and add the following entry:

```
tnsnames.ora
PRODDB =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = 141.144.32.70)(PORT = 1521))
(CONNECT_DATA =
(SID = PRODDB)
)
)
```

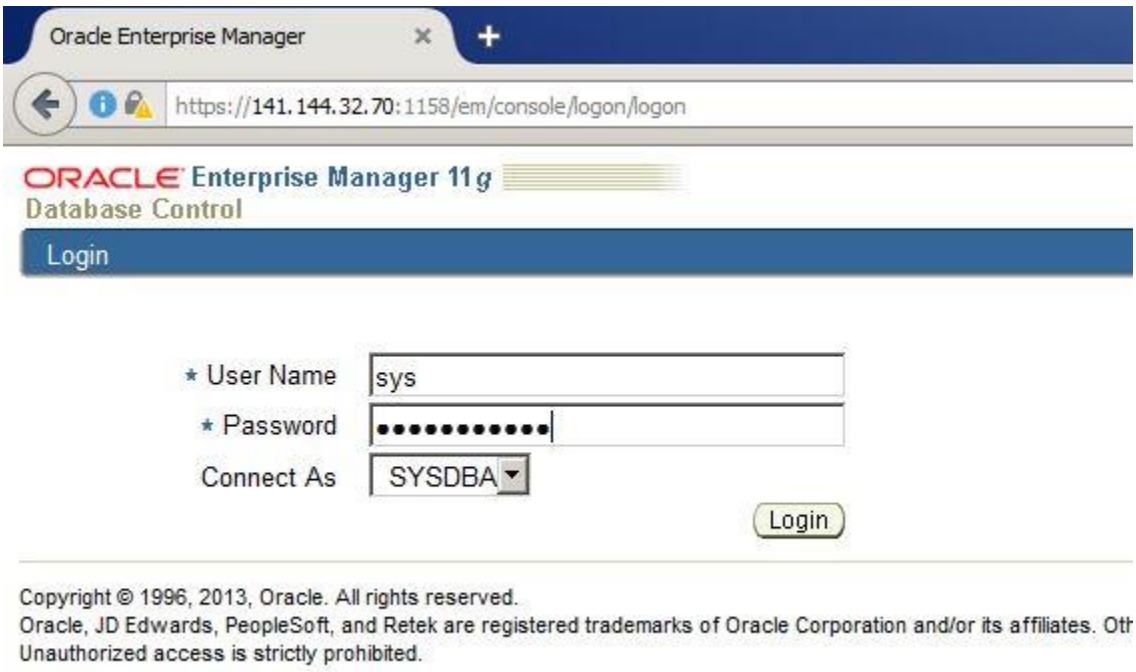
Open a command prompt, login to the database in the cloud and run SQL commands:

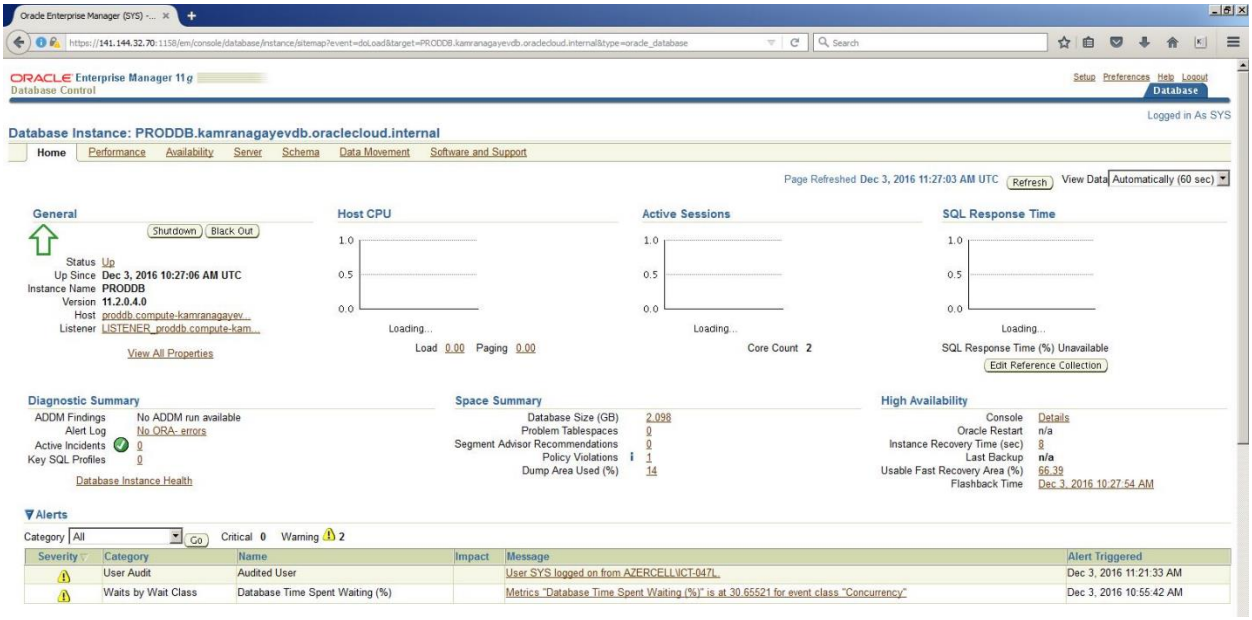


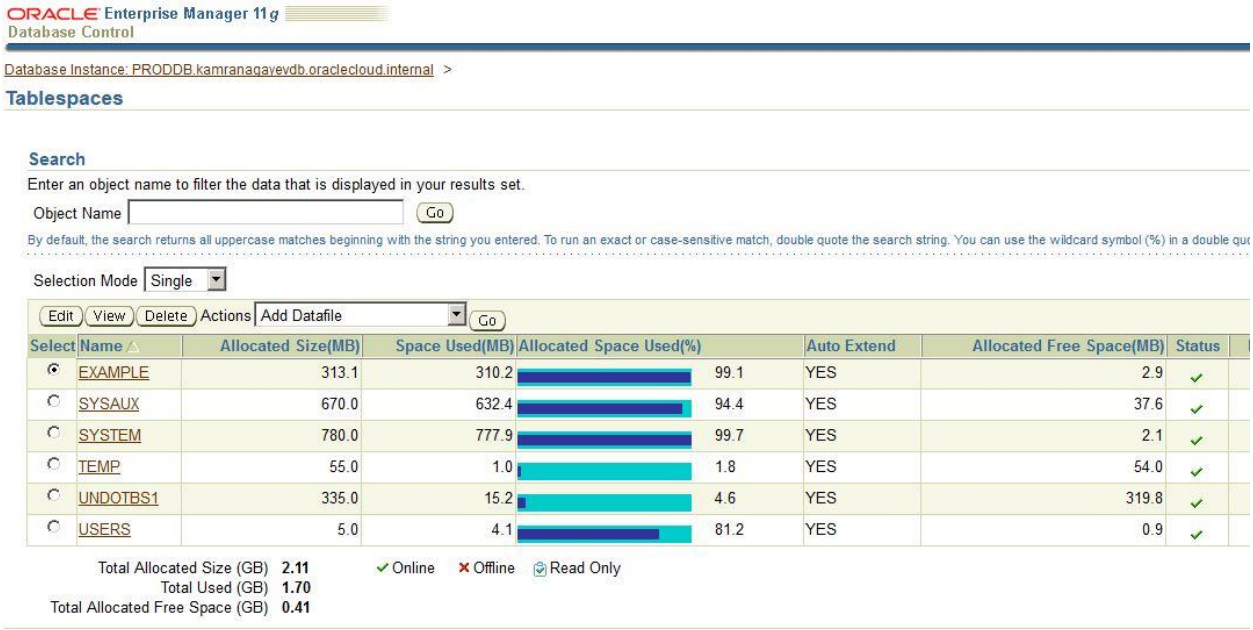
To open an OEM, click on the menu icon on the Database Cloud Service home page and select “Open EM Console”



Provide the username and password and login







After creating and configuring a database using a wizard, I decided to create a new database in a silent mode in the same machine as follows:

```
[oracle@proddb dbhome_1]$ dbca -silent -createdatabase -gdbname mydb -
templatename
/u01/app/oracle/product/11.2.0/dbhome_1/assistants/dbca/templates/Gene
ral_Purpose.dbc -sid mydb -syspassword oracle -systempassword oracle -
emConfiguration none -datafileDestination /u02/app/oracle/oradata/mydb
-memoryPercentage 40


Copying database files

1% complete

3% complete

11% complete

18% complete

26% complete

37% complete

Creating and starting Oracle instance

40% complete

45% complete

50% complete

55% complete

56% complete

60% complete

62% complete

Completing Database Creation

66% complete

70% complete

73% complete

74% complete

75% complete

76% complete

77% complete

88% complete

99% complete
```

```
100% complete

Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/mydb/mydb.log"
for further details.

[oracle@proddb dbhome_1]$
```

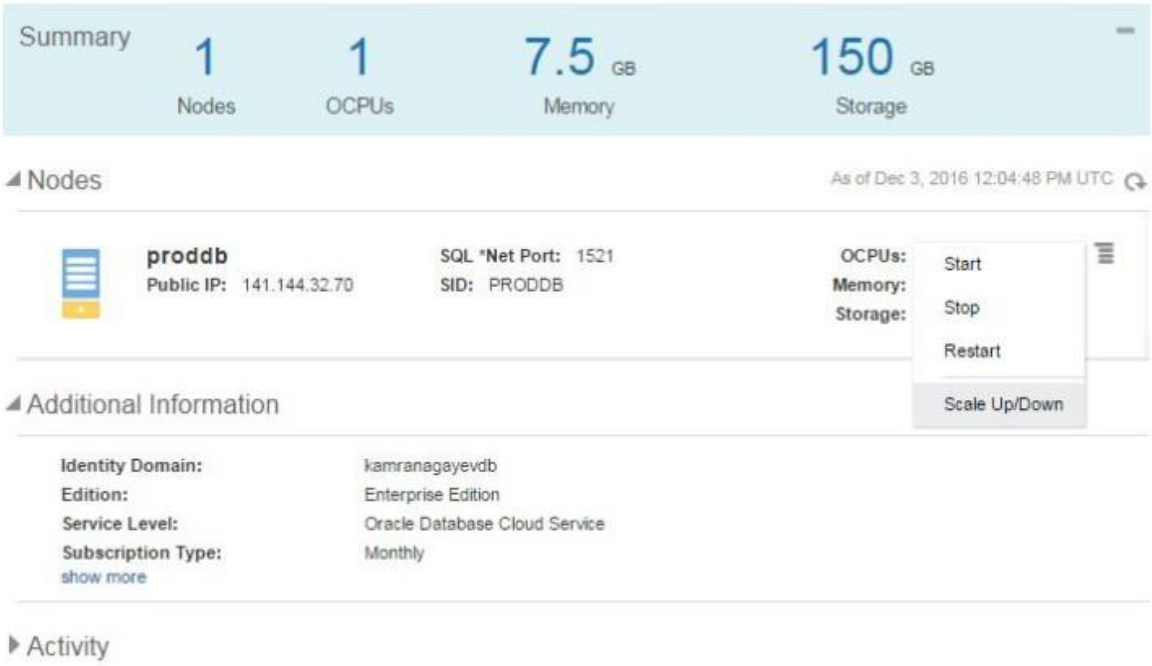
Check if the database is up and running and connect to it:

```
[oracle@proddb dbhome_1]$ ps -ef | grep smon
oracle    7040     1  0 11:40 ?        00:00:00 ora_smon_mydb
oracle    7226  2625  0 11:40 pts/1    00:00:00 grep smon
oracle   11837     1  0 10:30 ?        00:00:00 ora_smon_PRODDB
[oracle@proddb dbhome_1]$ export ORACLE_SID=mydb
[oracle@proddb dbhome_1]$ sqlplus / as sysdba
SQL>
```

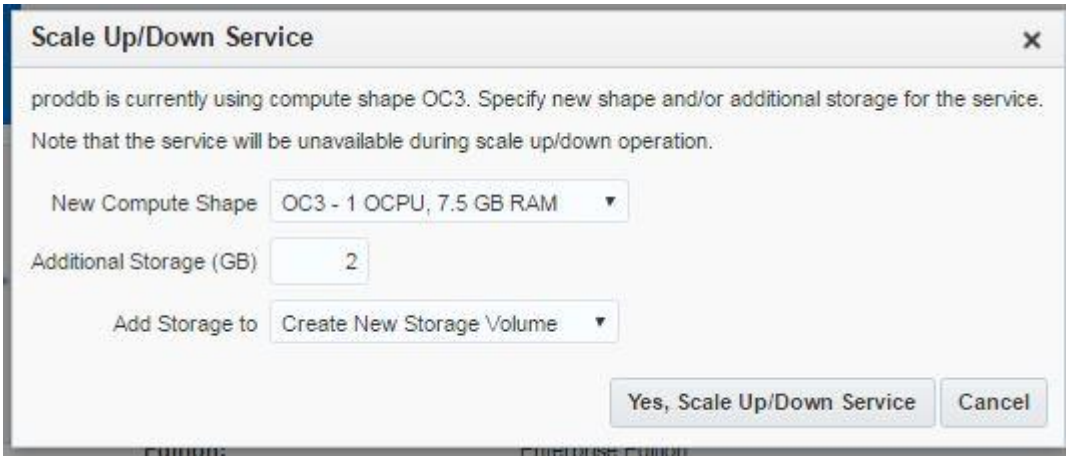
Run free command to check the free space of the machine. We have 2 databases running on this machine, so we have only 1g free memory.

```
[oracle@proddb dbhome_1]$ free
total       used       free     shared    buffers     cached
Mem:       7397060    6295684    1101376     266948      66356    1364664
-/+ buffers/cache:    4864664    2532396
Swap:      4194300      29916    4164384
[oracle@proddb dbhome_1]$
```

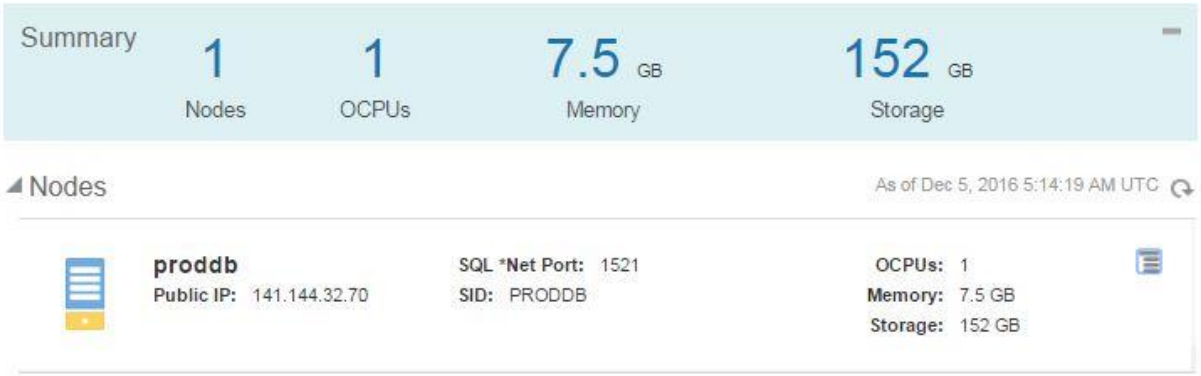
After having a database service with the specific parameters, you can change the parameters anytime. Let's add 2gb free space to the machine. Switch to the home page of the database cloud service, click on the menu icon and choose "Scale Up/Down" link



Provide size of the new space and click “Yes, Scale Up/Down Service” button



The host will reboot and the required space will be added :

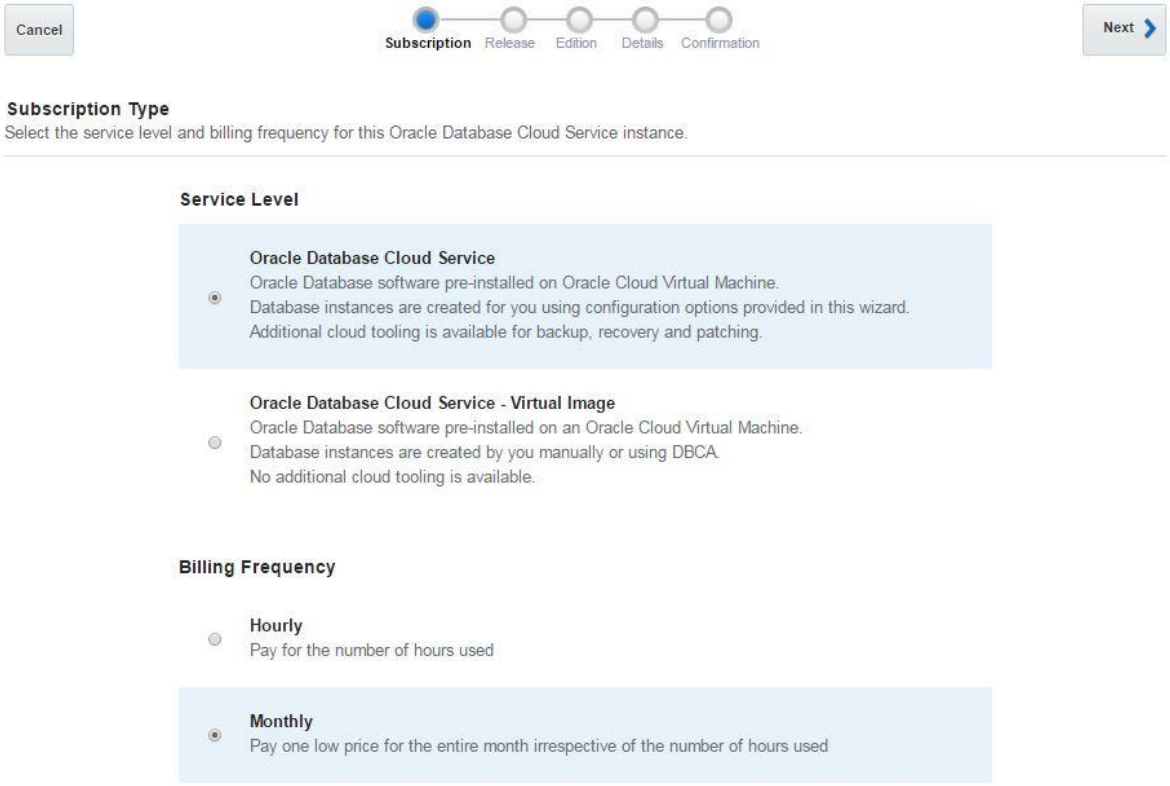


The new space is mounted to the new mount point (u05):

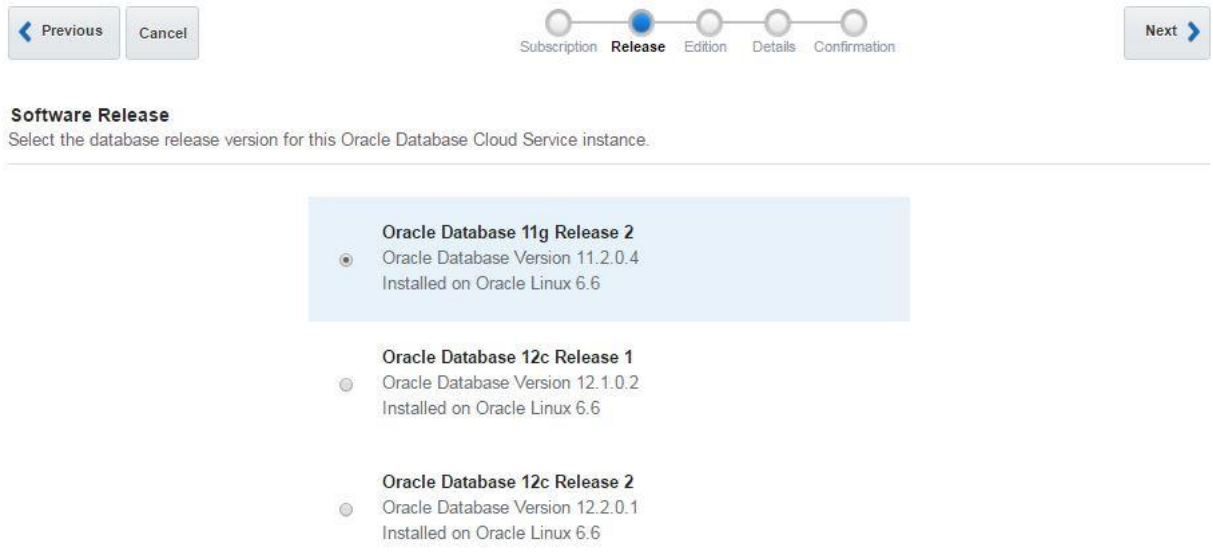
```
[oracle@proddb ~]$ df -kh
Filesystem            Size  Used Avail Use% Mounted on
/dev/xvdb3             25G   12G   12G  51% /
tmpfs                 3.6G     0  3.6G   0% /dev/shm
/dev/xvdb1            477M  148M  300M  34% /boot
/dev/xvde1             59G  7.4G   49G  14% /u01
/dev/mapper/dataVolGroup-lvol0
25G  3.8G   20G  17% /u02
/dev/mapper/fraVolGroup-lvol0
6.8G  2.1G  4.4G  32% /u03
/dev/mapper/redoVolGroup-lvol0
26G  3.1G   22G  13% /u04
/dev/xvdg1            2.0G  3.1M  1.9G   1% /u05
[oracle@proddb ~]$
```

Step by step guide – create a primary and standby database in the Cloud!

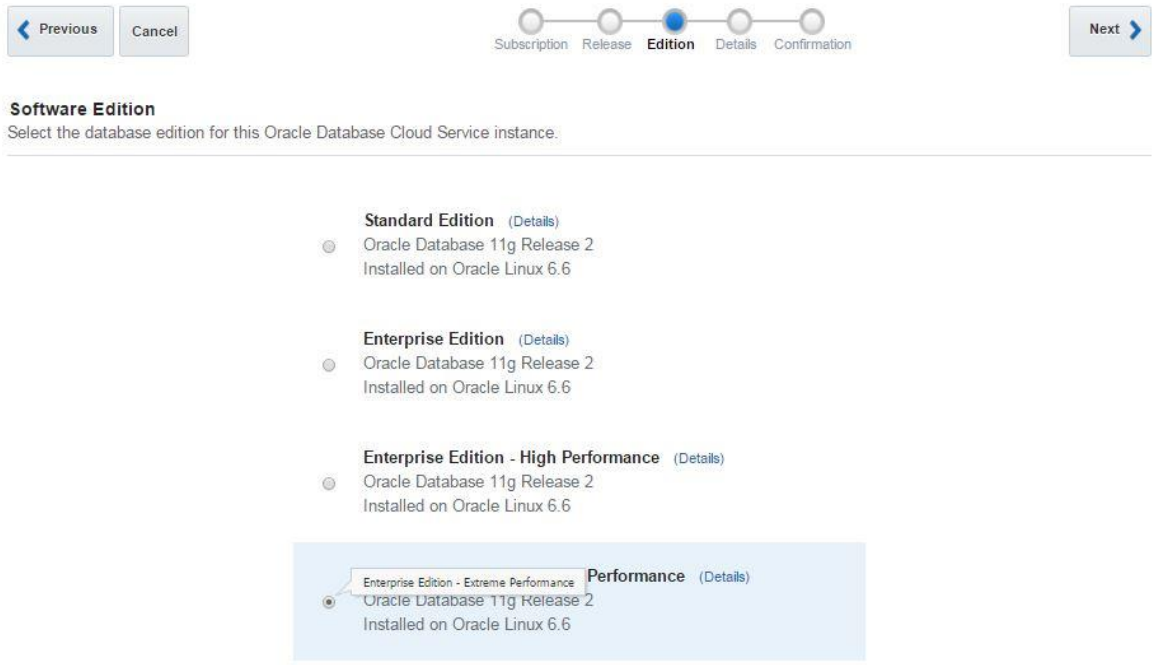
In this guide, I will show you how to create a primary and standby database in the cloud. Login to your cloud account, switch to the Oracle Database Cloud Service page and create a new service. Select “Oracle Database Cloud Service” as a subscription type and click Next.



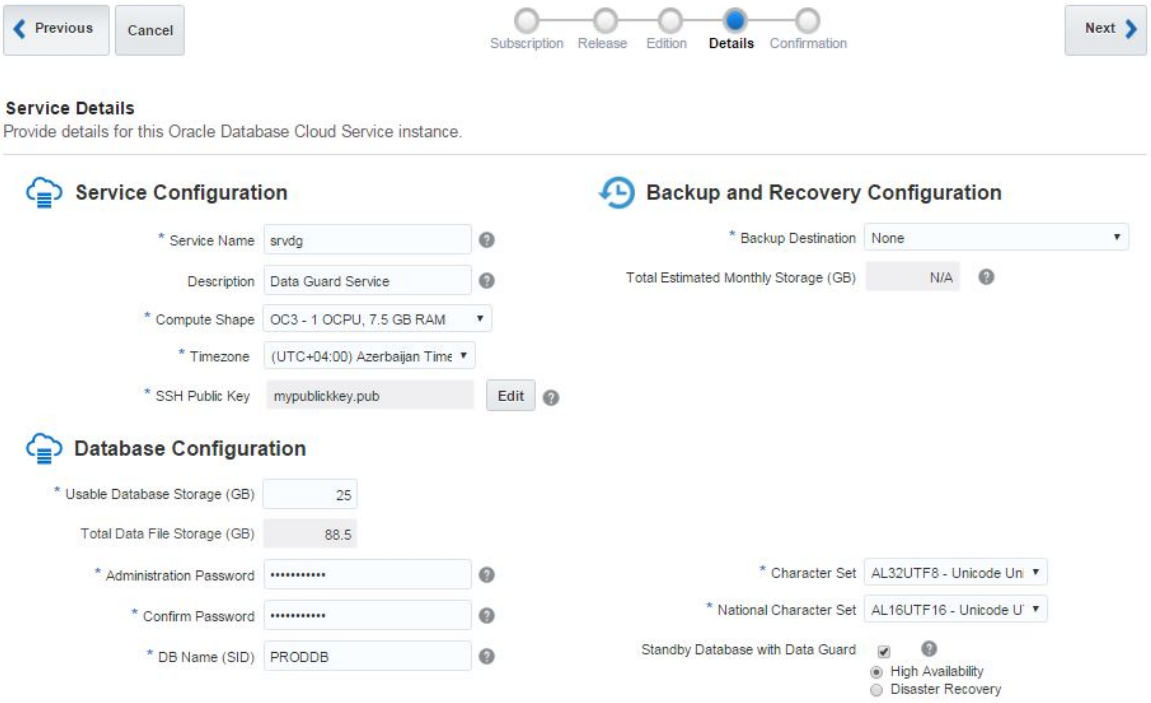
Select a database release and click Next.



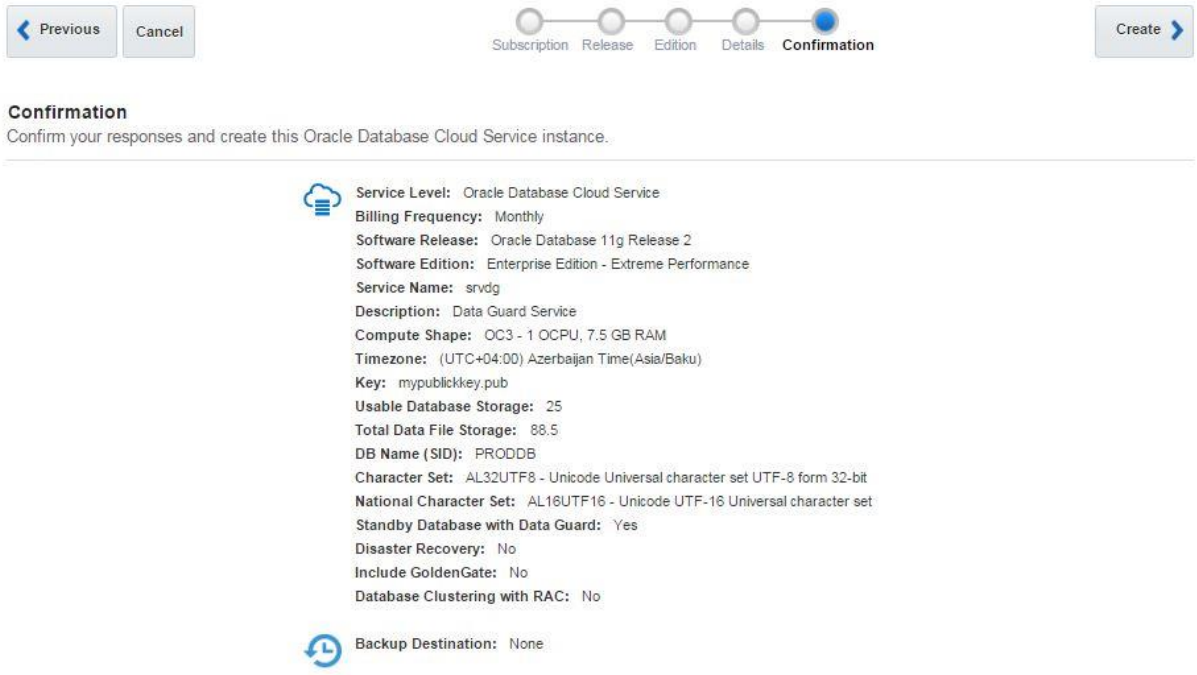
Select “Enterprise Edition – Extreme Performance” as a Software Edition and click Next



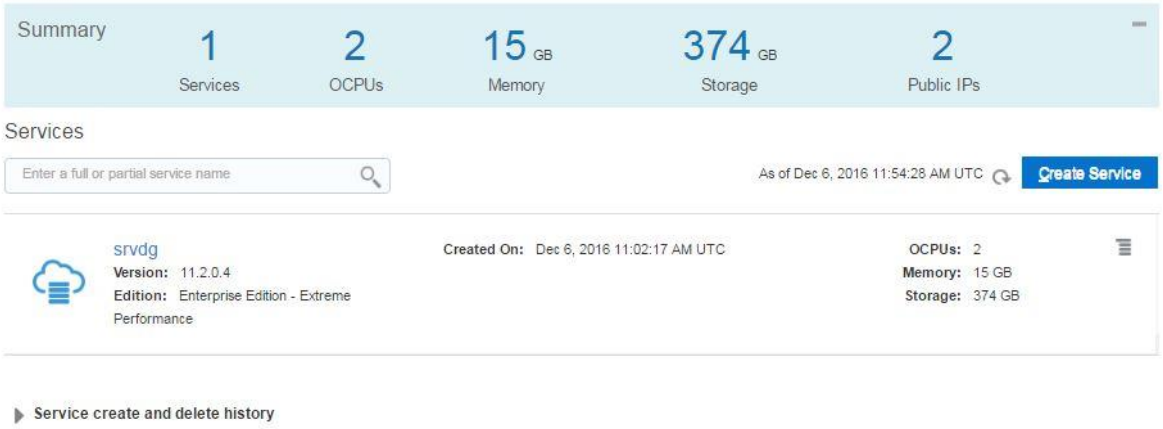
Provide the service name, SSH public key and select “Standby Database with Data Guard” option, select “High Availability” and click Next



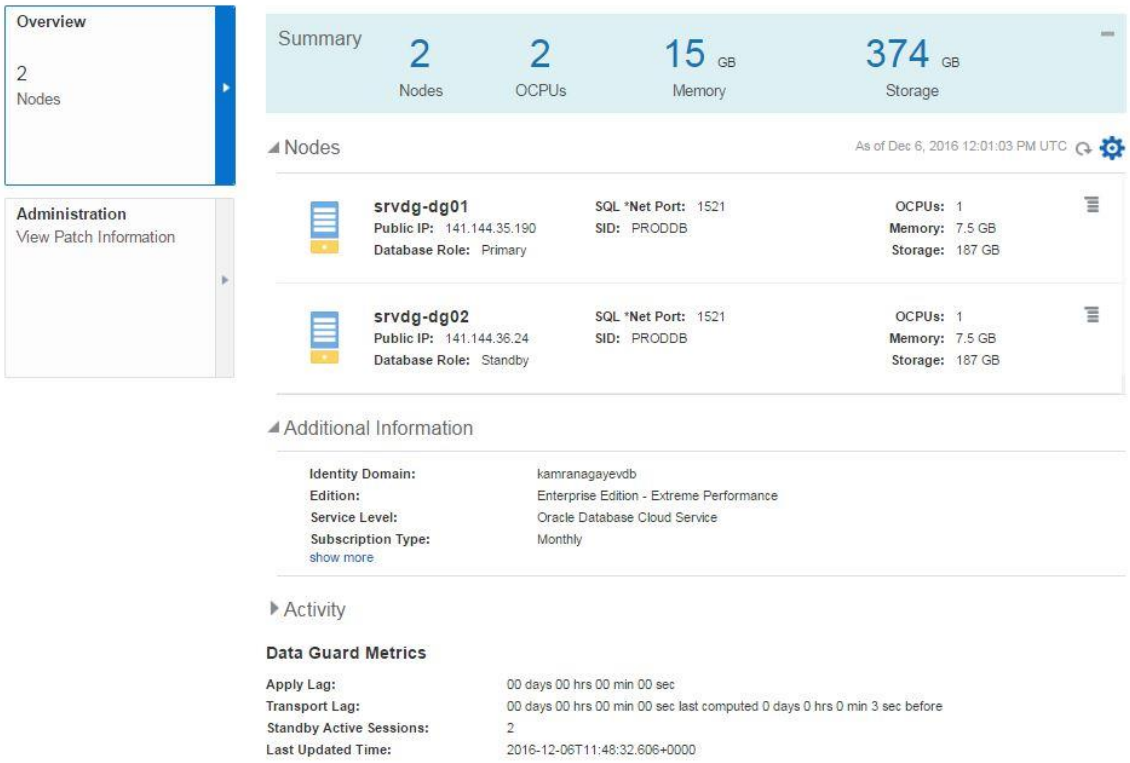
Review the configuration settings and click Create to create a primary and a standby database



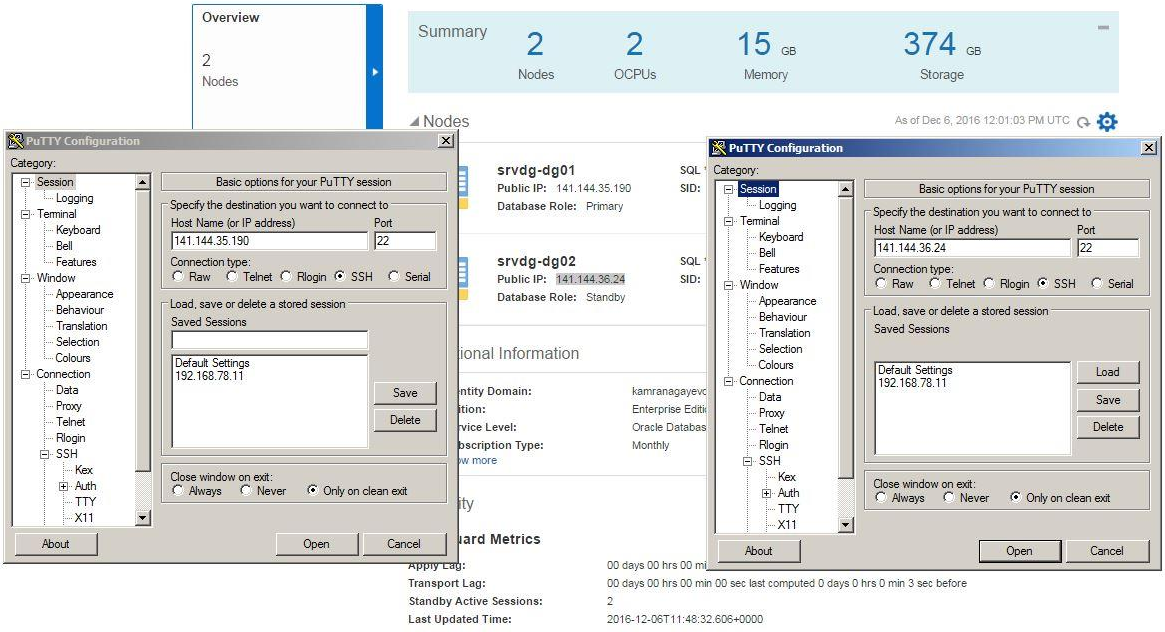
After a few minutes the primary and standby database will be created successfully



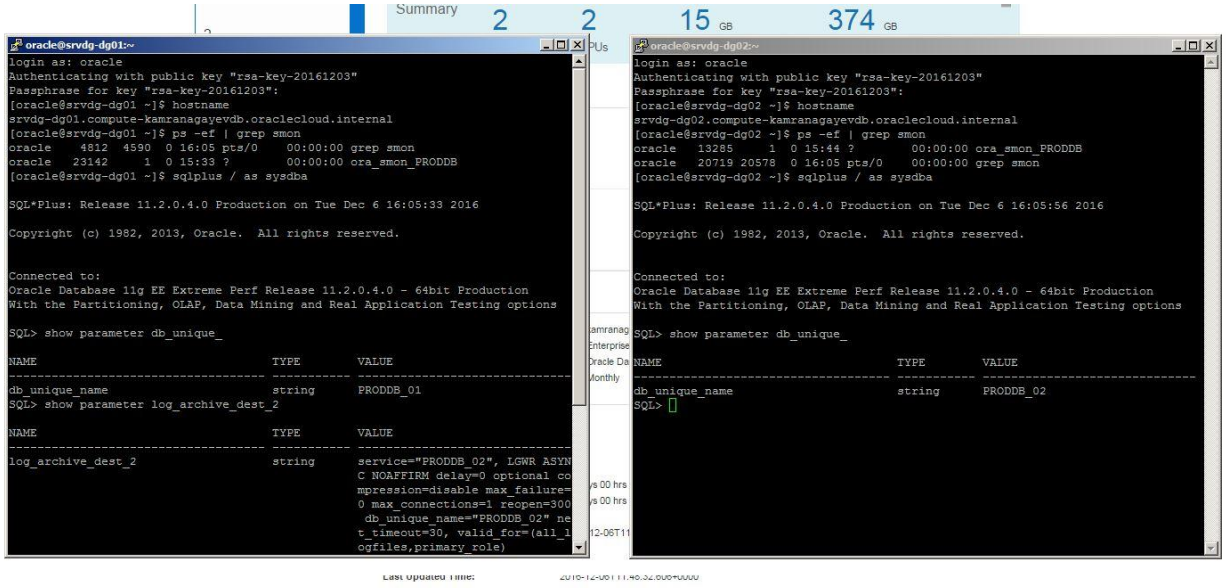
Click on the service name (srvdg) to open the home page of both databases



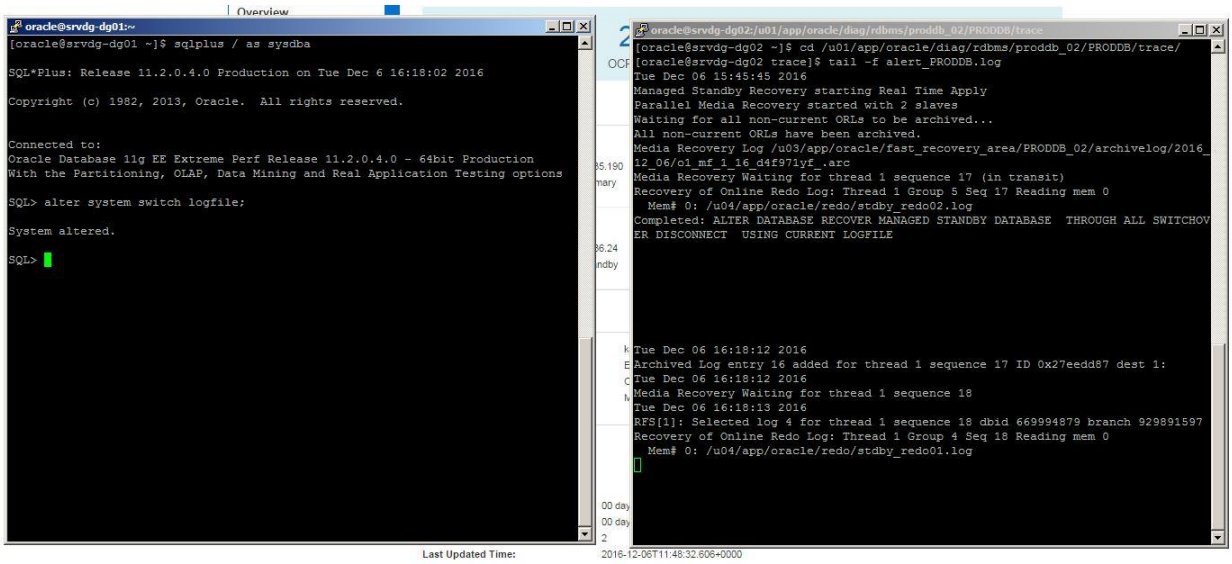
Now open two different Putty executables and connect to the both machines



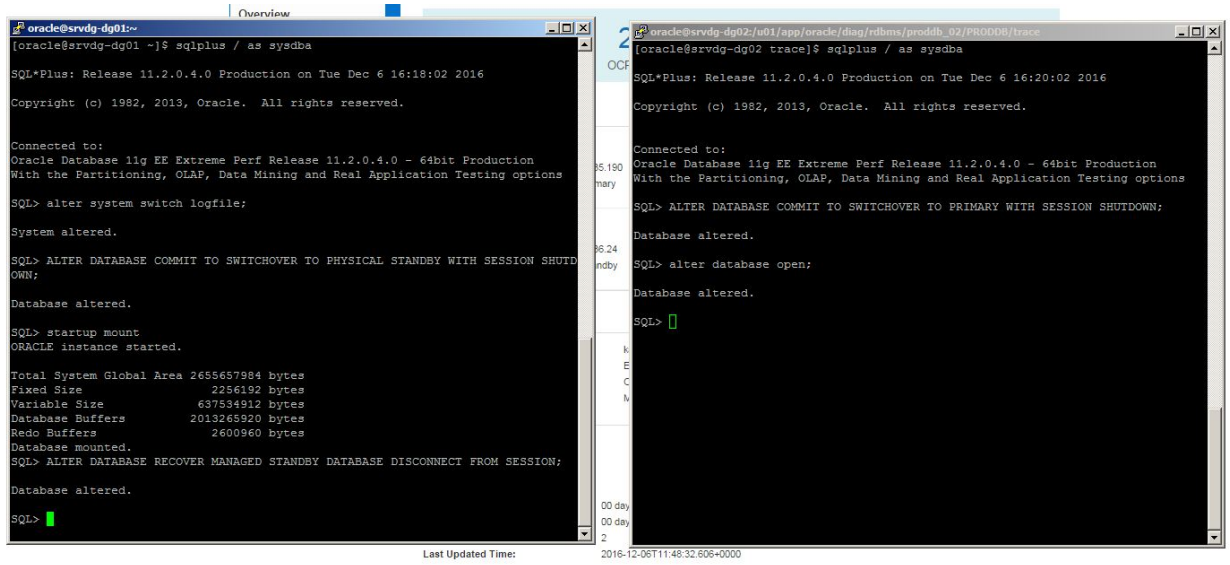
Connect to the both databases and check the DB_UNIQUE_NAME parameter. PRODDB_01 is set to the primary database, PRODDB_02 for the standby database. Also check LOG_ARCHIVE_DEST_2 parameter on the primary database. As you see, archived log files are shipped to the standby database using PRODDB_02 service.



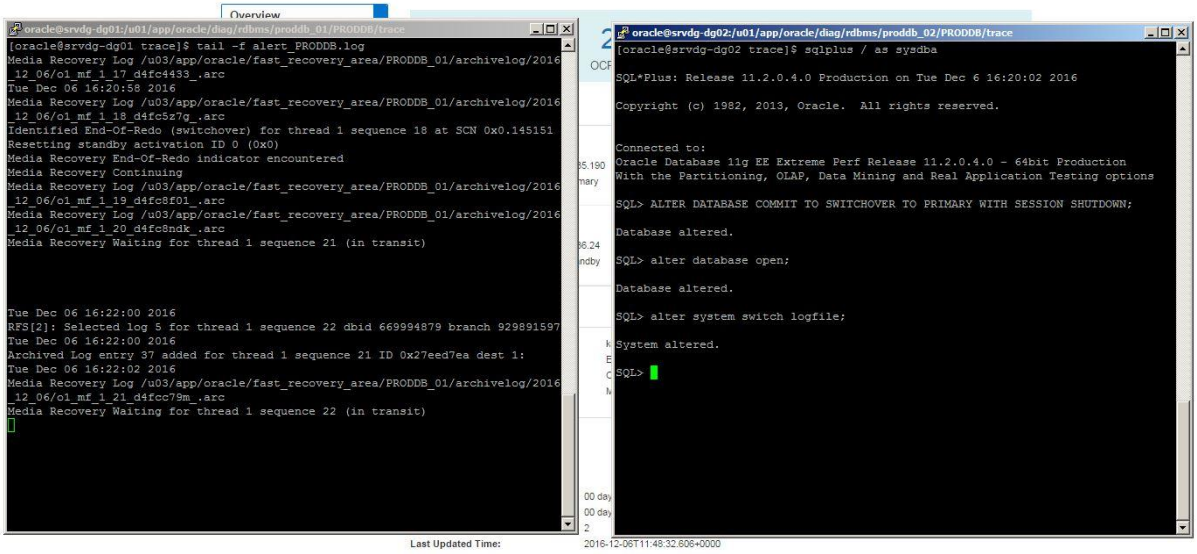
Now let's test the functionality of the standby database. First of all, open an alert.log file of the standby database, switch the log file on the primary database and check if the log file is applied



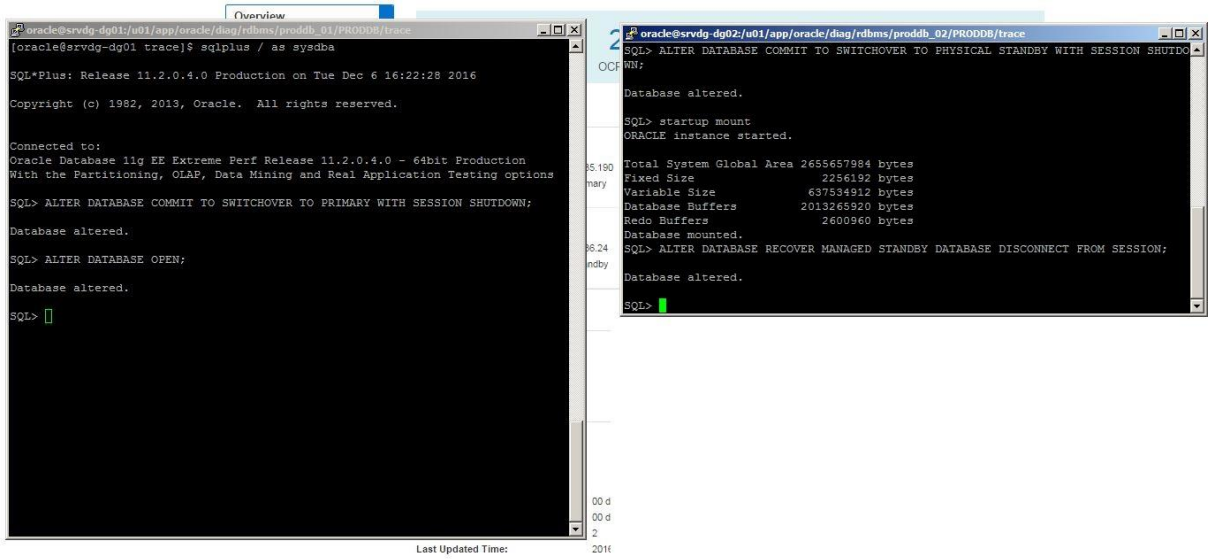
Now perform a switchover



Switch the log file and see if it is applied to the new standby database



It worked. Now switch back

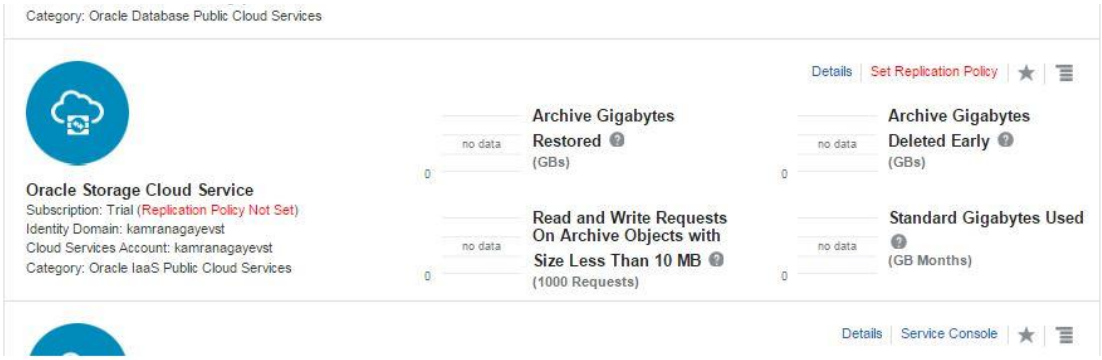


As you see, it's very easy to create a database with its own standby database in the cloud!

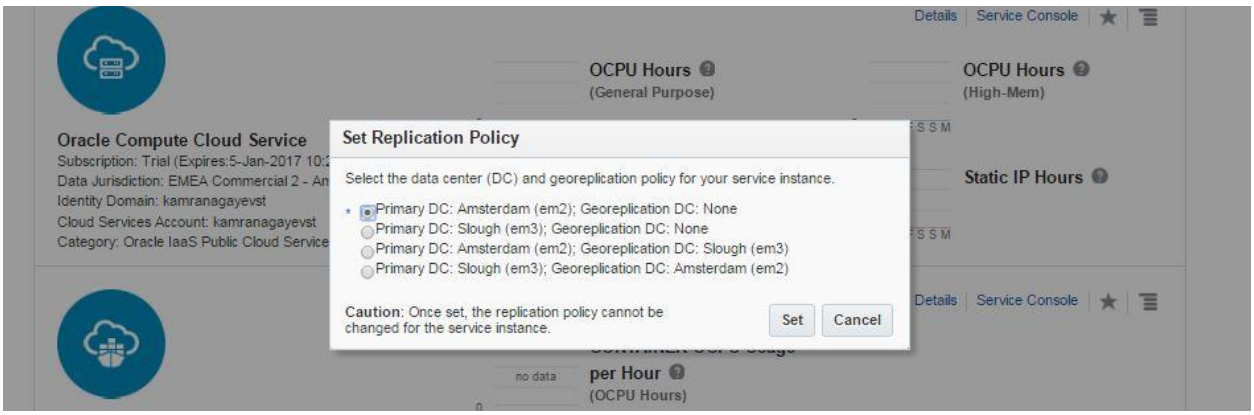
Configure and practice backup and recovery for Oracle Database in Cloud (DBaaS)

In this post I will show you how to configure backup for Oracle Database in Cloud. First of all, make sure you use Oracle Storage Cloud Service and you set the replication policy. Open the following link, scroll down to Oracle Storage Cloud Service section and click “Set Replication Policy” link:

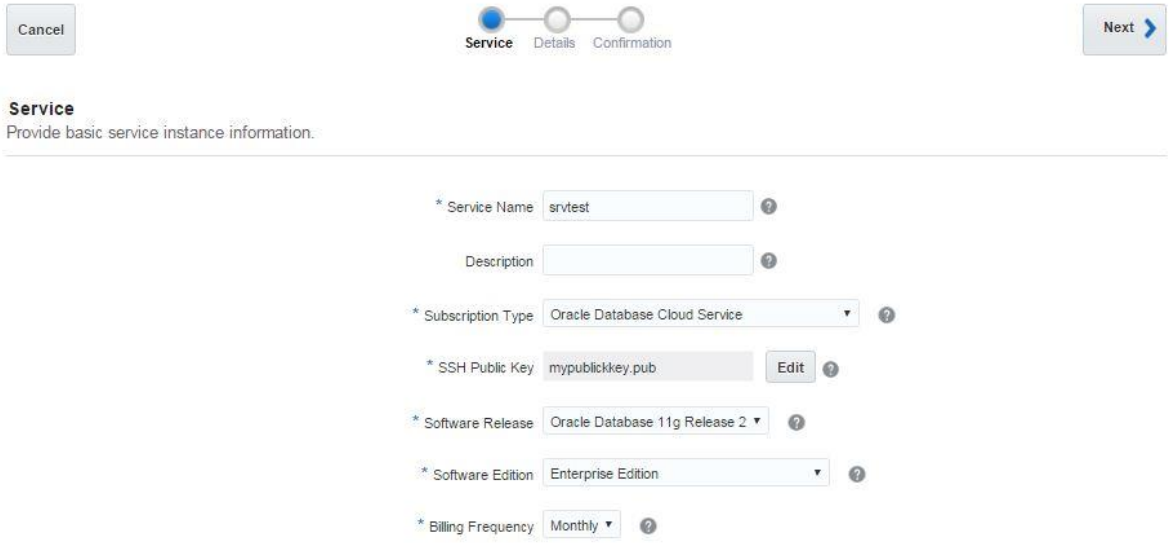
https://myservices.em2.oraclecloud.com/mycloud/faces/dashboard.jspx?showOld=true



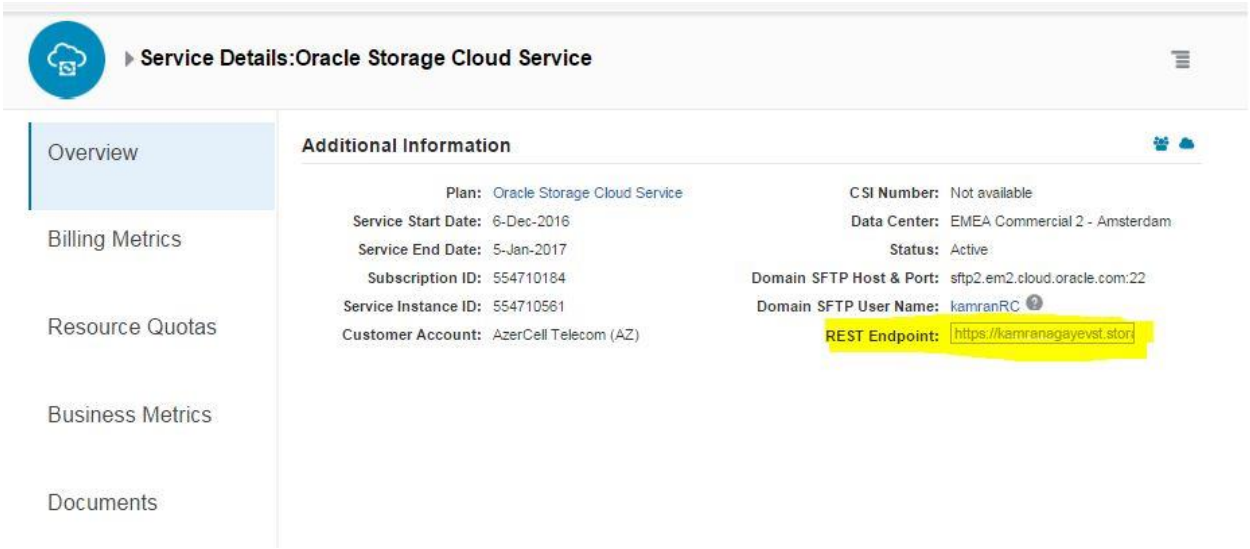
Select the data center and click Set



Next, open Oracle Database Cloud Service and create a new service. The GUI has changed and we have only 3 steps to create a database in the cloud. Provide the service name, software version and edition, upload SSH public key and click Next

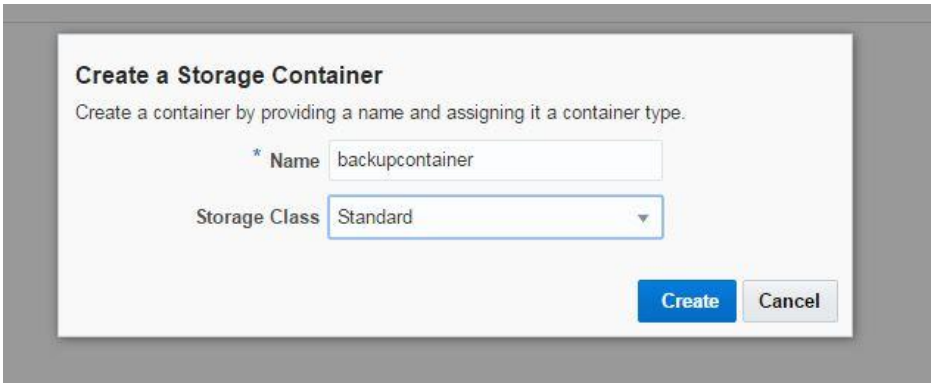


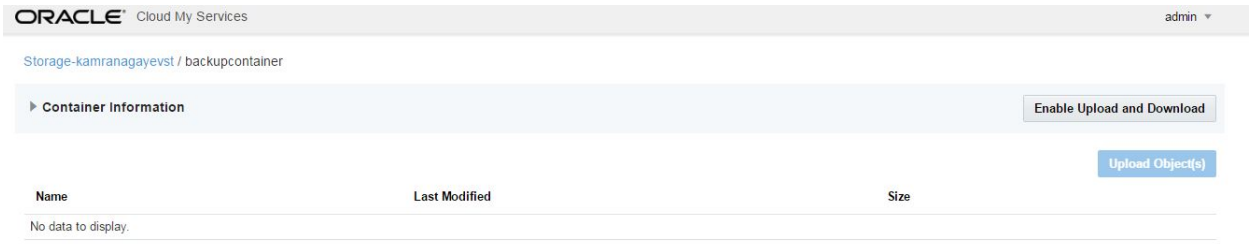
In order to enable the automatic backup of the database in the cloud, you have to create a cloud storage container. Before creating a cloud storage container, switch to Oracle Storage Cloud Service details and get the REST Endpoint:



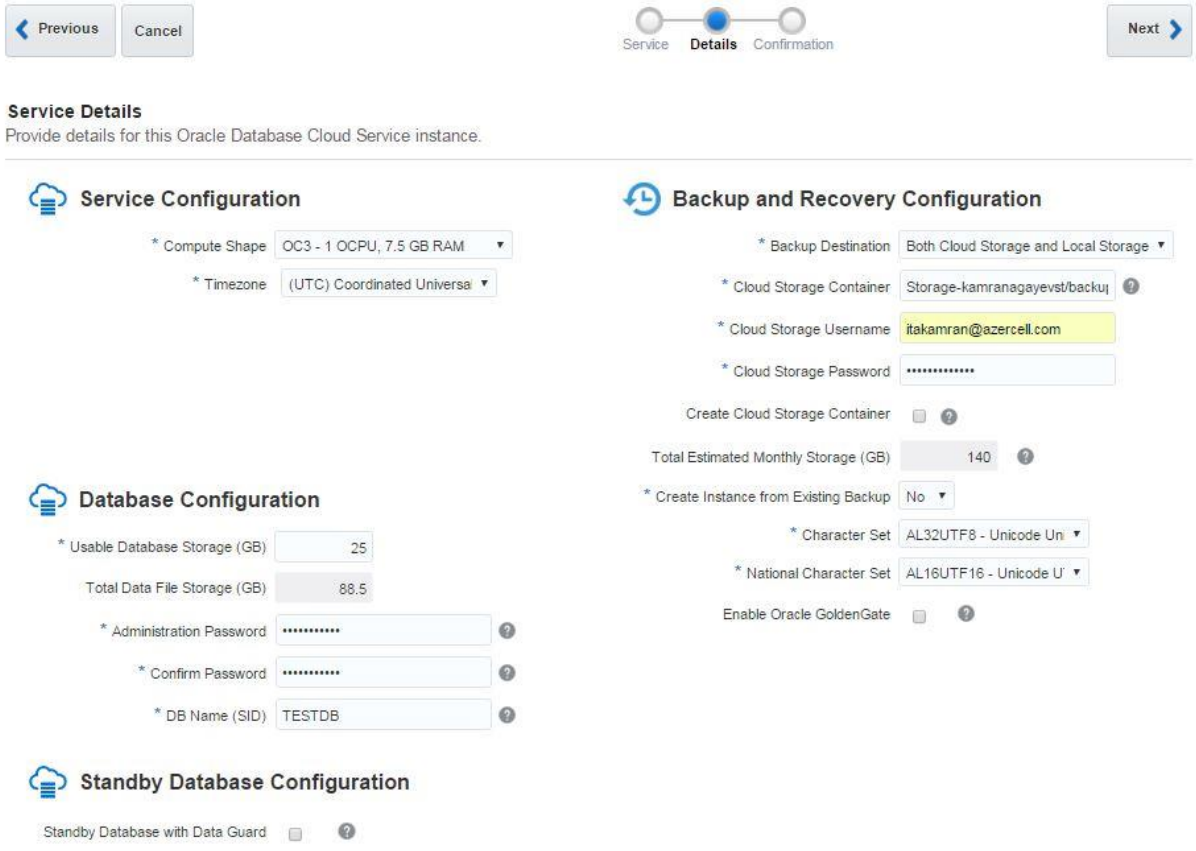
Next, open https://storageconsole.em2.oraclecloud.com/ link, provide the Service REST Endpoint and login to Oracle Storage Cloud Service:

Create a new storage container:

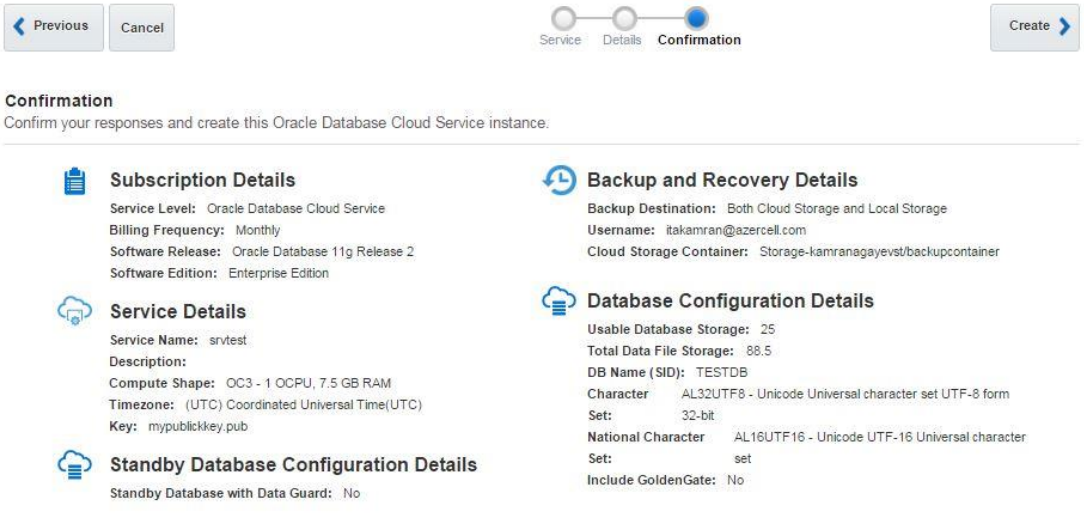




In the second screen of database service creation page, select “Both Cloud Storage and Local Storage” option as a Backup Destination, provide cloud storage container name, username and password and click Next.

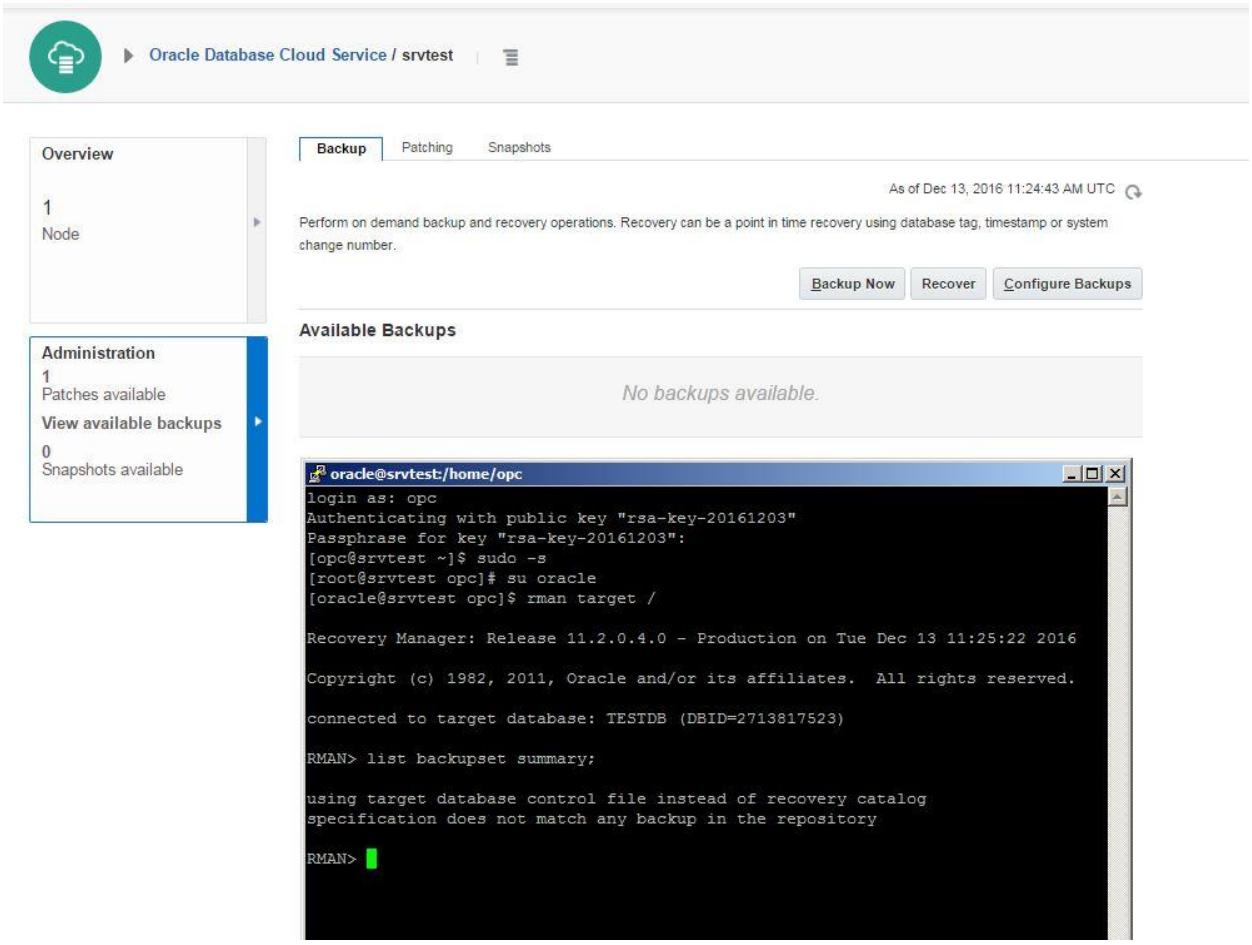


Review the configuration and click Create button.

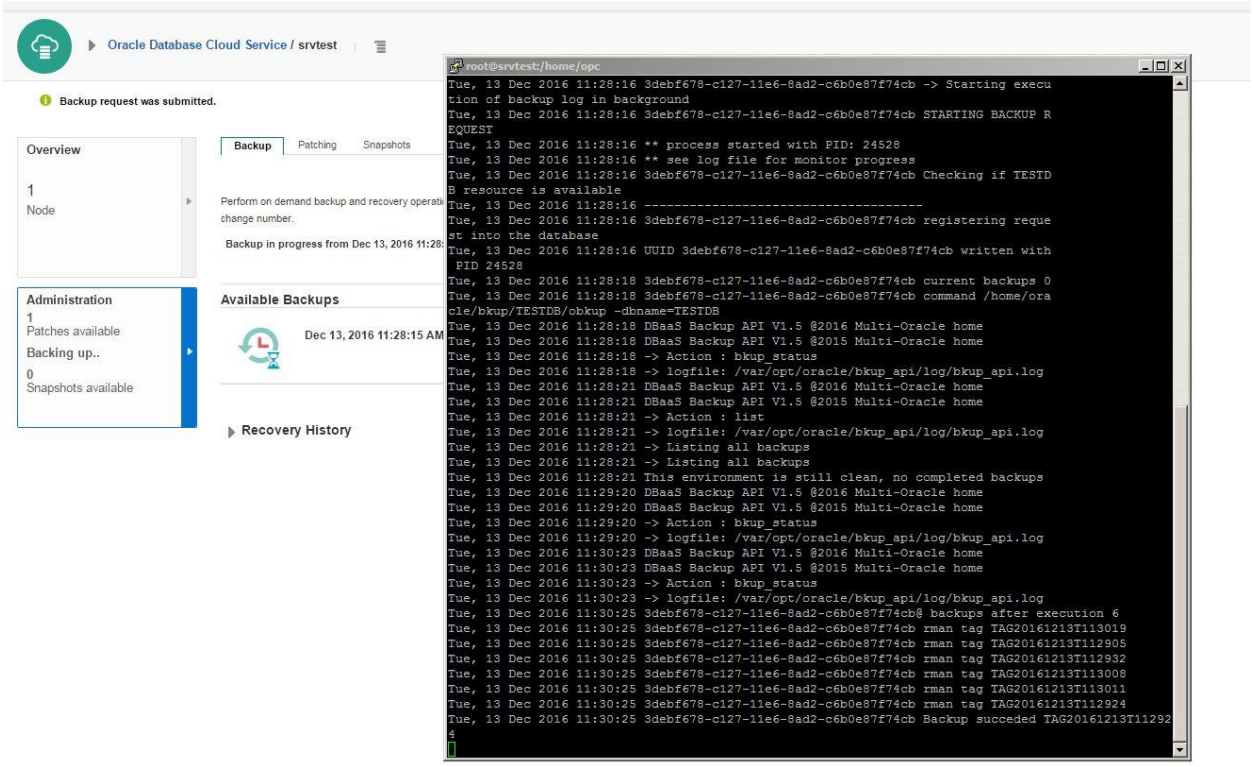


After creating the service successfully, open it and click on Administration section. From the Backup tab

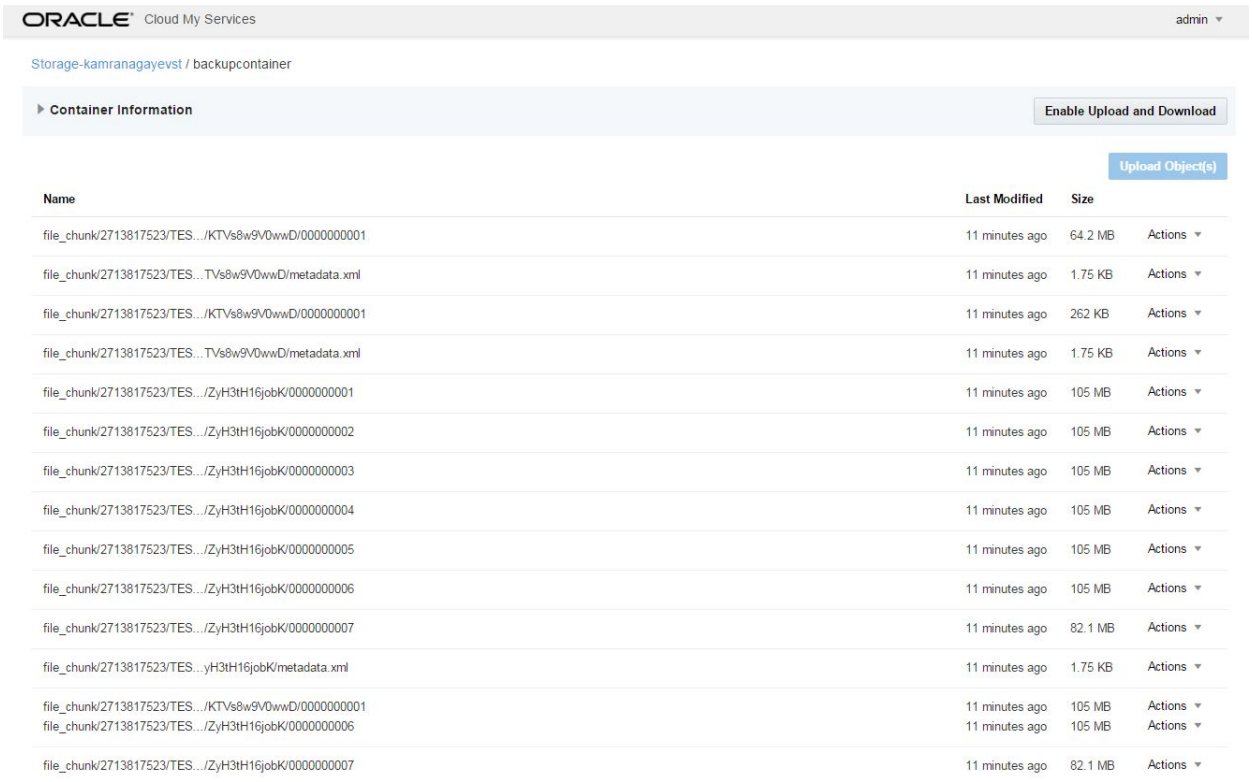
click on Backup Now button to create a backup of the database. You can use RMAN and schedule your own backups as well.



Click Backup Now and check the log file for more information:



If you switch to the storage container, you will see bunch of files created



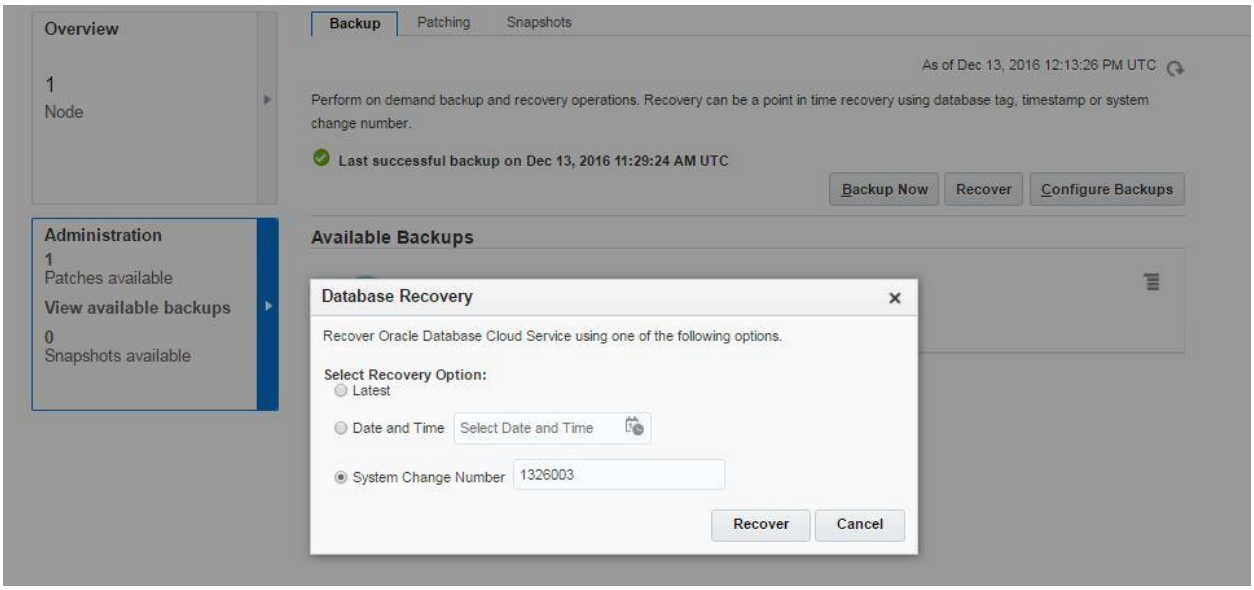
Open RMAN and run LIST BACKUPSET SUMMARY command to get list of backupsets:

```
[oracle@srvtest ~]$ rman target /

Recovery Manager: Release 11.2.0.4.0 - Production on Tue Dec 13 12:14:13 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

connected to target database: TESTDB (DBID=2713817523)

RMAN> list backupset summary;

using target database control file instead of recovery catalog

List of Backups
===============
Key     TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
------- -- -- - ----------- --------------- ------- ------- ---------- ---
1       B  A  A *           13-DEC-16       1       2       NO         DBAAS_INCR_BACKUP
2       B  A  A *           13-DEC-16       1       2       NO         DBAAS_INCR_BACKUP
3       B  F  A *           13-DEC-16       1       2       NO         TAG20161213T112905
4       B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T112924
5       B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T112932
6       B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T112932
7       B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T112932
8       B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T112932
9       B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T112932
10      B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T113008
11      B  A  A SBT_TAPE    13-DEC-16       1       1       YES        TAG20161213T113011
12      B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T113019
13      B  A  A SBT_TAPE    13-DEC-16       1       1       YES        TAG20161213T120009
14      B  F  A SBT_TAPE    13-DEC-16       1       1       NO         TAG20161213T120016

RMAN>
```

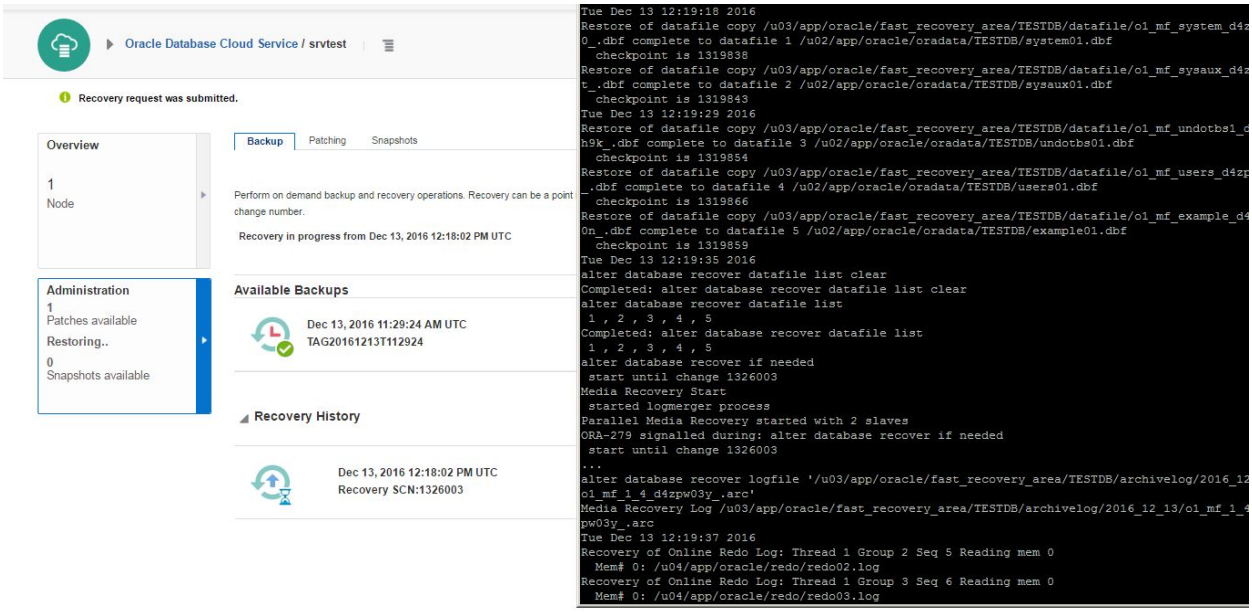
Now let's try to recover the database to the specific point in time using DBaaS wizard. For this, create a new table with some data, get the current SCN number and drop the table.

```
[oracle@srvtest ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Tue Dec 13 12:14:57 2016

Copyright (c) 1982, 2013, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create table mytable as select * from dba_objects;

Table created.

SQL> select count(1) from mytable;

  COUNT(1)
----------
     88910

SQL> select current_scn from v$database;

CURRENT_SCN
-----------
    1326003

SQL> drop table mytable purge;

Table dropped.

SQL>
```

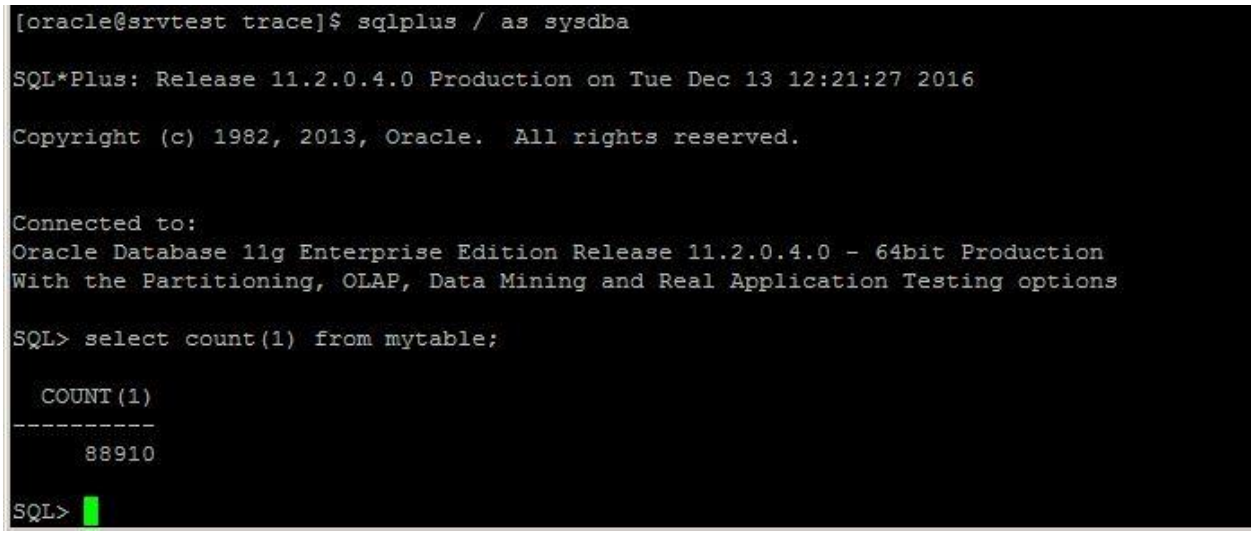
Next, switch to DBaaS backup page, click Recover, provide the SCN number and click Recover



The recover process will run in the background automatically. Check alert.log file of the database for more information:



After the recover process is completed successfully login to the database and query the table



You can also take backup and recover the database from command line interface using bkup_api utility. Now let's delete all backups, take a new backup and try the recovery.

Delete all available RMAN backups:

```
RMAN> delete backup;
```

Use bkup_api utility with bkup_start parameter to take a backup from CLI:

```
[root@srvtest spool]# /var/opt/oracle/bkup_api/bkup_api bkup_start

DBaaS Backup API V1.5 @2016 Multi-Oracle home

DBaaS Backup API V1.5 @2015 Multi-Oracle home

-> Action : bkup_start

-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log

UUID d6bf0bde-c130-11e6-8534-c6b0e87f74cb for this backup

** process started with PID: 16524

** see log file for monitor progress

_____________

[root@srvtest spool]#
```

Check the log file for more information:

```
[root@srvtest spool]# tail -f /var/opt/oracle/bkup_api/log/bkup_api.log

Tue, 13 Dec 2016 12:36:58 ** process started with PID: 16524

Tue, 13 Dec 2016 12:36:58 ** see log file for monitor progress

Tue, 13 Dec 2016 12:36:58 ————————————-

Tue, 13 Dec 2016 12:36:58 d6bf0bde-c130-11e6-8534-c6b0e87f74cb Checking if
TESTDB resource is available

Tue, 13 Dec 2016 12:36:58 d6bf0bde-c130-11e6-8534-c6b0e87f74cb has a lock
TESTDB

Tue, 13 Dec 2016 12:36:58 UUID d6bf0bde-c130-11e6-8534-c6b0e87f74cb written
with PID 16524

Tue, 13 Dec 2016 12:36:58 d6bf0bde-c130-11e6-8534-c6b0e87f74cb The process is
no longer running removing lock

Tue, 13 Dec 2016 12:36:58 d6bf0bde-c130-11e6-8534-c6b0e87f74cb registering
request into the database

Tue, 13 Dec 2016 12:37:00 d6bf0bde-c130-11e6-8534-c6b0e87f74cb current
backups 0

Tue, 13 Dec 2016 12:37:00 d6bf0bde-c130-11e6-8534-c6b0e87f74cb command
/home/oracle/bkup/TESTDB/obkup -dbname=TESTDB

Tue, 13 Dec 2016 12:38:51 d6bf0bde-c130-11e6-8534-c6b0e87f74cb@ backups after
execution 4

Tue, 13 Dec 2016 12:38:51 d6bf0bde-c130-11e6-8534-c6b0e87f74cb rman tag
TAG20161213T123750

Tue, 13 Dec 2016 12:38:51 d6bf0bde-c130-11e6-8534-c6b0e87f74cb rman tag
TAG20161213T123729

Tue, 13 Dec 2016 12:38:51 d6bf0bde-c130-11e6-8534-c6b0e87f74cb rman tag
TAG20161213T123758

Tue, 13 Dec 2016 12:38:51 d6bf0bde-c130-11e6-8534-c6b0e87f74cb rman tag
TAG20161213T123834

Tue, 13 Dec 2016 12:38:51 d6bf0bde-c130-11e6-8534-c6b0e87f74cb Backup
succeded TAG20161213T123834
```

Now having valid backups, let’s create a new table, drop it and recover it using dbaascli utility.

```
[oracle@srvtest opc]$ sqlplus / as sysdba
SQL> create table mytable2 as select * from dba_objects;
Table created.

SQL> select count(1) from mytable2;
  COUNT(1)
————-
     88911

SQL> select current_scn from v$database;
CURRENT_SCN
————-
1333654

SQL> drop table mytable2 purge;
Table dropped.
SQL> exit
```

Now use dbaascli utility and provide the SCN number to perform SCN based incomplete recovery:

```
[root@srvtest opc]# dbaascli orec –args -scn 1333654
DBAAS CLI version 1.0.0
Executing command orec –args -scn 1333654
–args : -scn 1333654
OREC version: 16.0.0.0
Starting OREC
Logfile is /var/opt/oracle/log/TESTDB/orec/orec_2016-12-13_13:41:18.log
Config file is /var/opt/oracle/orec/orec.cfg
DB name: TESTDB
OREC:: RUNNING IN NON DATAGUARD ENVIRONMENT
OREC:: Verifying scn validity…
PITR using SCN: 1333654
```

```
OREC:: Catalog mode:  Disabled
OREC:: Checking prerequirements before recovery process.
OREC:: DB Status : OPEN
OREC:: Changing instance to MOUNT stage.
OREC:: Shutting down the database… Completed.
OREC:: (RMAN) Startup MOUNT… Completed.
OREC:: Checking for PDBs directories.
OREC:: Checking for REDO logs.
OREC:: Restablishing DB instance to the original stage.
OREC:: Shutting down the database… Completed.
OREC:: Starting up database… Completed.
OREC:: Testing RMAN connection.
OREC:: Verifying backups dates ..
   :: OK
OREC:: Performing PITR using SCN number 1333654 …
INFO : DB instance is up and running after recovery procedure.
OREC:: Completed.
[root@srvtest opc]#
```

Now connect to the database and check if the table is recovered:

```
[oracle@srvtest opc]$ sqlplus / as sysdba
SQL> select count(1) from mytable2;
  COUNT(1)
———-
     88911
SQL>
```

The database backups are also stored in the flash recovery area image-16in the database host:

```
[oracle@srvtest 2016_12_13]$ pwd
/u03/app/oracle/fast_recovery_area/TESTDB/backupset/2016_12_13
[oracle@srvtest 2016_12_13]$ ls -ltr
total 117912
-rw-rw---- 1 oracle oinstall 89445376 Dec 13 12:37 o1_mf_annnn_DBAAS_INCR_BACKUP_d4ztvvk2_.bkp
-rw-rw---- 1 oracle oinstall  1971712 Dec 13 12:37 o1_mf_annnn_DBAAS_INCR_BACKUP_d4ztvypo_.bkp
-rw-rw---- 1 oracle oinstall 29310976 Dec 13 12:37 o1_mf_nnnd1_DBAAS_INCR_BACKUP_d4ztw1y7_.bkp
-rw-rw---- 1 oracle oinstall     4608 Dec 13 12:37 o1_mf_annnn_DBAAS_INCR_BACKUP_d4ztw876_.bkp
[oracle@srvtest 2016_12_13]$
```

If you want to change the automatic backup schedule, edit /etc/crontab file with a root user. Below you can see the current schedule of the database backup:

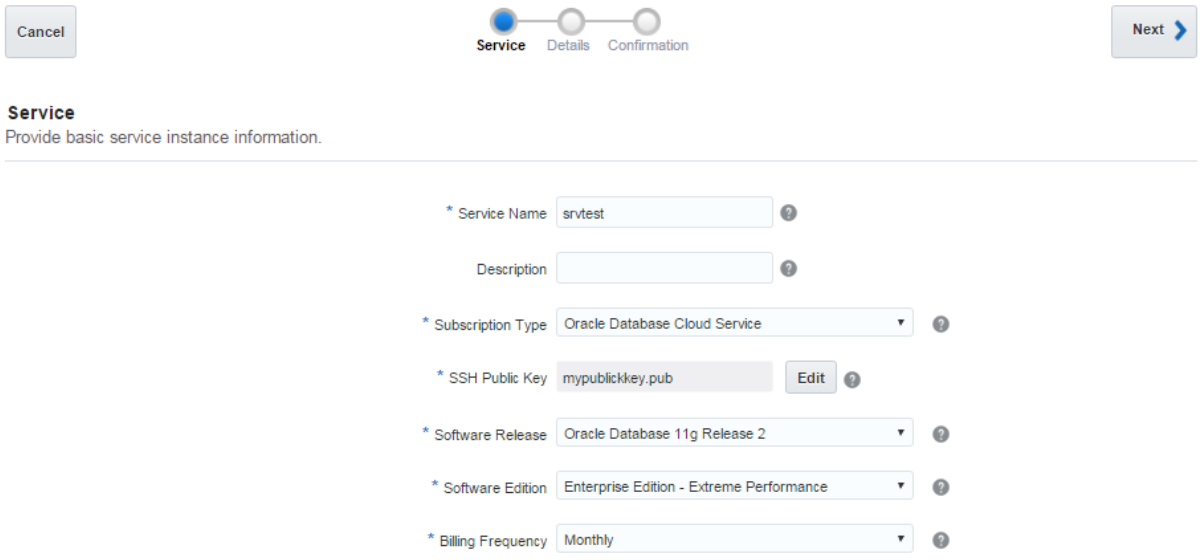
```
[root@srvtest opc]# more /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=""
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,f
ri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed

13 0 * * * root /var/opt/oracle/bkup_api/bkup_api bkup_start --dbname=TESTDB
0,30 * * * * root /home/oracle/bkup/TESTDB/obkup -dbname=TESTDB -archivelog
15 03 * * 6 oracle /var/opt/oracle/cleandb/cleandblogs.pl
[root@srvtest opc]#
```

You can use a DBaaS backup wizard, DBaaS command line interface commands and RMAN to perform backup and recovery for Oracle Database in Cloud

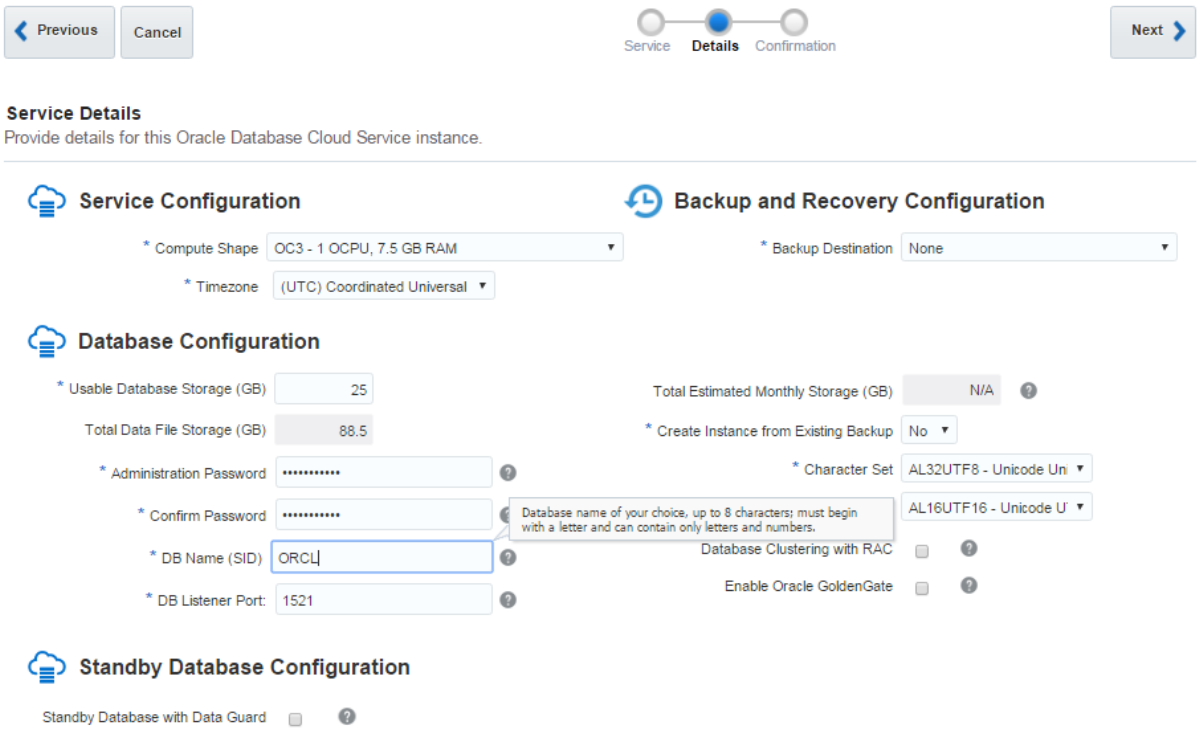
Create a Standby database in Oracle Cloud for On-Premises production database

If you have a production database and you plan to build a standby database on the different geographic location, Oracle Cloud is the best option. In this blog post you will see a step by step guide on how to create a Standby Database in Oracle Cloud for your on-premises database.

First of all, login to your Oracle Cloud account, switch to Oracle Database Cloud Service and create a new Service. Provide a service name, SSH Public Key (check above mentioned articles to see how to create a SSH public key), choose “Enterprise Edition – Extreme Performance” for Software Edition option and click Next.



We will create a standby database based on on-premises production database, so in the next screen provide any database name. We will delete it once it is created and will create a standby database using DUPLICATE DATABASE command.



Review the configuration and click Create to create a Database Cloud Service instance.

It take only 20 minutes to create a new machine, install an Oracle Software and create a new database in the cloud.

Next, create a new virtual machine in your own laptop, install Oracle 11.2.4 on Linux (OEL is preferred) and add two network cards – “Host-only Adapter” and “Bridged Adapter”. “Host-Only Adapter” is used to connect to the virtual machine from the host machine and “Bridged Adapter” is used to connect from the Virtual Machine to the outside world (internet, cloud instance and etc.). Enable both network devices, make sure you have internet connection, edit tnsnames.ora file as follows and use tnsping to ping the cloud host.

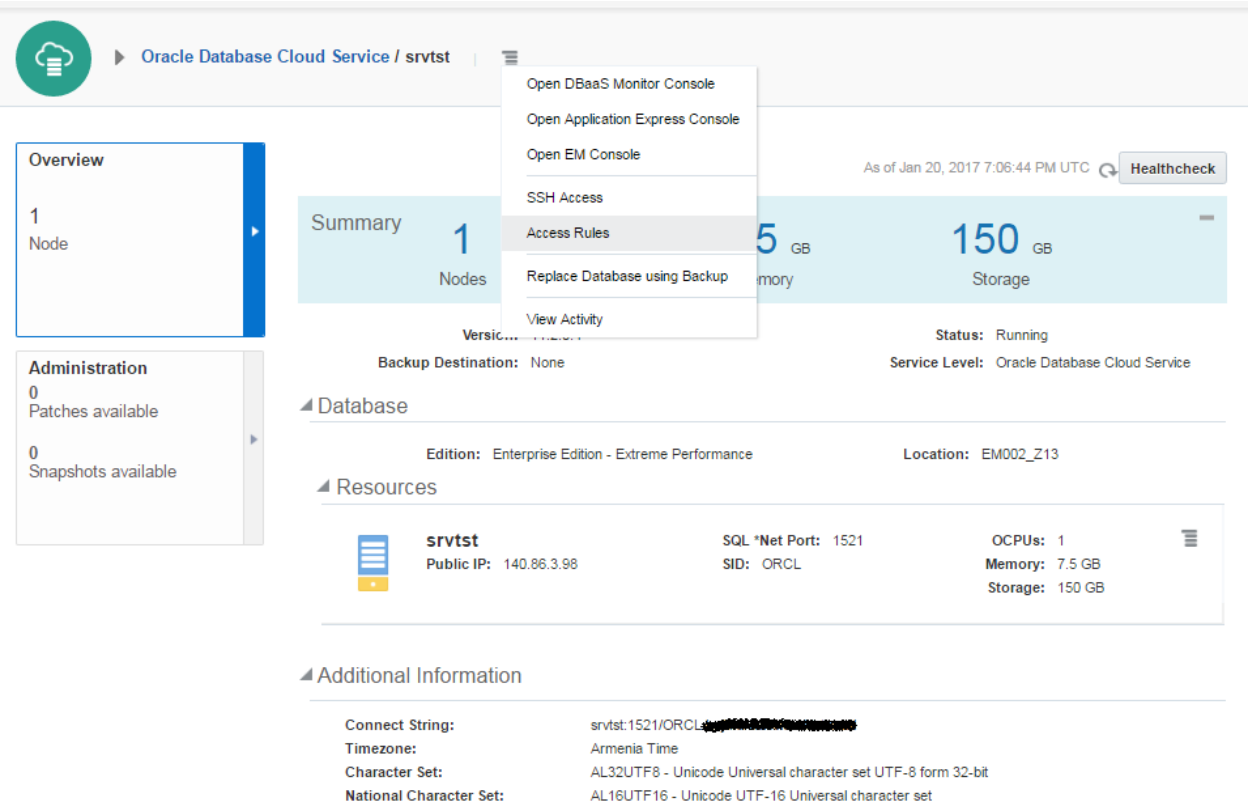
```
STBDB =
  (DESCRIPTION =
   (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 140.86.3.98)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = STBDB)
      (UR = A)
    )
  )
```

Next, use private key to connect to the cloud machine using putty and drop the ORCL database in the cloud machine

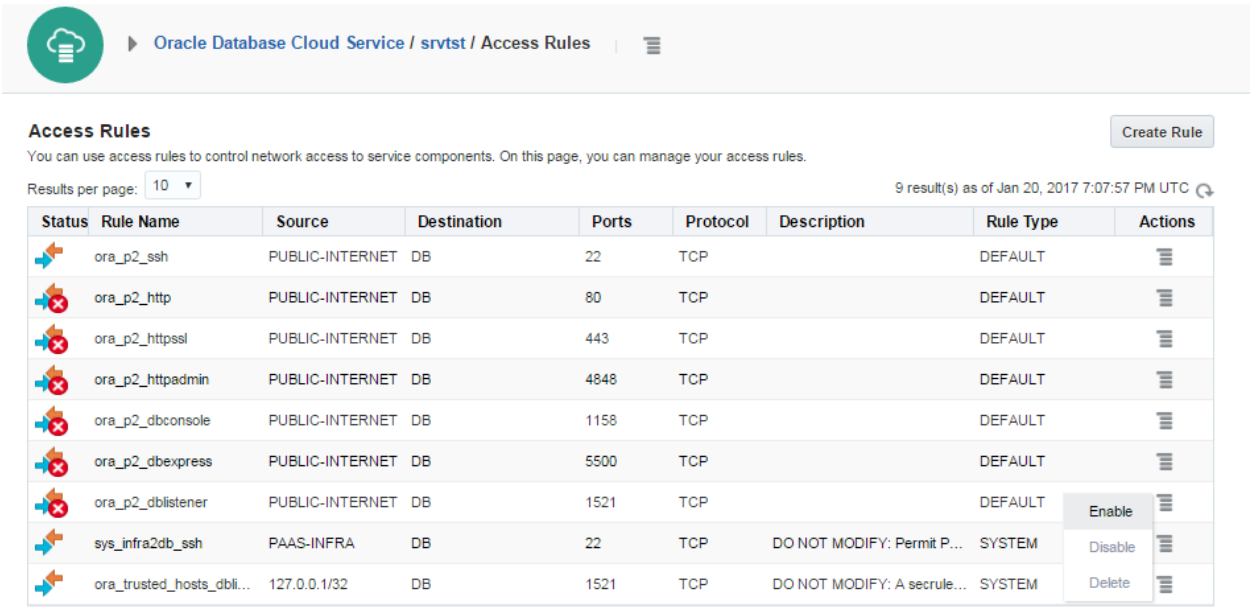
Drop the database in the cloud machine:

```
[oracle@srvtst ~]$ sqlplus / as sysdba
SQL> startup force mount exclusive restrict;
ORACLE instance started.
Total System Global Area 2655657984 bytes
Fixed Size                  2256192 bytes
Variable Size             637534912 bytes
Database Buffers         1996488704 bytes
Redo Buffers               19378176 bytes
Database mounted.

SQL> drop database;
Database dropped.
SQL>
```

Before trying to connect to the new dummy instance on the cloud machine, you have to enable dblistener access rule. Open the database service and Access Rule from the menu.



Click on Actions menu for the ora_p2_dblistener rule and enable it



Now you will be able to using tnsping to test the connection:

```
[oracle@ocm11g admin]$ tnsping STBDB

Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = 140.86.3.98)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
STBDB) (UR = A)))

OK (250 msec)

[oracle@ocm11g admin]$
```

In order to connect to the cloud machine from outside, you need to configure SSH. Open Virtual Machine box, switch to .ssh folder and generate ssh key using ssh-keygen utility as follows:

```
[oracle@ocm11g ~]$ cd .ssh

[oracle@ocm11g .ssh]$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/oracle/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/oracle/.ssh/id_rsa.
```

```
Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
1f:e8:8d:08:78:80:12:e5:c6:cb:cb:7a:97:2e:1b:02 oracle@ocm11g
The key's randomart image is:
+-[ RSA 2048]---+
|…              |
| =               |
|o =              |
|.o +     .       |
|E + o   S .      |
|.. o . o + .     |
|. +  .. o o      |
| oo.o            |
|…=.            |
+-------+
[oracle@ocm11g .ssh]$
```

Now copy the source of id_rsa.pub file and append it to the /home/oracle/.ssh/authorized_keys file at the cloud machine.

```
[oracle@ocm11g .ssh]$ more id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAn2fjBDvcycbxQxVrzFQS2URSERkdJXTdpHGw68GiQWUnCR8T8
jSwntDWH4az37Lyj7WgN0NGW7HFWC0m9EMJ/RfCPj6SXnCjdXOO2qwuxMit9B9suqm7plfQl+HpGT
rdx6KIW2UXW1M/7l2CDNjJD7zDFZ4MNwBIOtlT5lpHm61iquVeBUwFg/3fjpnk6/IjX5K0mM8gLHW
pc6WEDLcLKHgKWcVUGvY/KF1W2ehbGIo6tSDkDV2wwEj8H5G5DCxLs2Mczq1dzgt99SLVpw3s7/aG
RWrzPVRVPjmn1Y7AHnDFNFvP32V3fzKCaAHHQLjDeA6ZQyjMjBUFAxWuiymunw==
oracle@ocm11g
```

Now test the connection from virtual box to the cloud machine:

```
[oracle@ocm11g .ssh]$ ssh 140.86.3.98
The authenticity of host '140.86.3.98 (140.86.3.98)' can't be established.
RSA key fingerprint is 73:93:3c:62:41:d4:12:aa:09:07:c7:94:aa:ea:00:16.
```

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '140.86.3.98' (RSA) to the list of known hosts.
[oracle@srvtst ~]$ exit
logout
Connection to 140.86.3.98 closed.
```

Before duplicating the database, create necessary folders on the cloud machine

```
[oracle@ocm11g .ssh]$ ssh 140.86.3.98
[oracle@srvtst ~]$ mkdir -p admin/STBDB/adump
[oracle@srvtst ~]$ mkdir -p oradata/STBDB
[oracle@srvtst ~]$ mkdir flash_recovery_area
[oracle@srvtst ~]$ mkdir arch
```

Create a parameter file to start standby instance:

```
vi /home/oracle/pfile.ora
*.audit_file_dest='/home/oracle/admin/STBDB/adump'
*.control_files='/home/oracle/oradata/STBDB/control01.ctl'
*.db_file_name_convert='/u03/oracle/oradata/PROD/','/home/oracle/oradata/STBD
B/'
*.db_name='PROD'
*.db_unique_name='STBDB'
*.db_recovery_file_dest='/home/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=5g
*.fal_client='STBDB'
*.fal_server='PROD'
*.log_archive_dest_1='location=/home/oracle/arch
VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
```

```
DB_UNIQUE_NAME=STBDB'

*.log_file_name_convert='/u03/oracle/oradata/PROD/','/home/oracle/oradata/STB
DB/'

*.compatible='11.2.0.4.0'
```

Connect to SQL*Plus, create spfile and open the instance in the NOMOUNT mode:

```
[oracle@srvtst ~]$ sqlplus / as sysdba

Connected to an idle instance.


SQL> startup nomount pfile='/home/oracle/pfile.ora';

ORACLE instance started.

Total System Global Area  229683200 bytes

Fixed Size                  2251936 bytes

Variable Size             171967328 bytes

Database Buffers           50331648 bytes

Redo Buffers                5132288 bytes

SQL> create spfile from pfile='/home/oracle/pfile.ora';

File created.


SQL> shut immediate

ORA-01507: database not mounted

ORACLE instance shut down.


SQL> startup nomount;

ORACLE instance started.

Total System Global Area  229683200 bytes

Fixed Size                  2251936 bytes

Variable Size             171967328 bytes

Database Buffers           50331648 bytes

Redo Buffers                5132288 bytes

SQL>
```

Create a password file on the standby machine

```
[oracle@srvtst ~]$ orapwd
file=/u01/app/oracle/product/11.2.0/dbhome_1/dbs/orapwSTBDB password=oracle
entries=5
```

Connect to both target and auxiliary instances and duplicate the database:

```
[oracle@ocm11g dbs]$ rman target sys/oracle@PROD auxiliary sys/oracle@STBDB

connected to target database: PROD (DBID=345613202)

connected to auxiliary database: PROD (not mounted)


RMAN> duplicate target database for standby from active database;

Starting Duplicate Db at 20-JAN-17

using target database control file instead of recovery catalog

allocated channel: ORA_AUX_DISK_1

channel ORA_AUX_DISK_1: SID=171 device type=DISK

contents of Memory Script:

{

   backup as copy reuse

   targetfile  '/u03/oracle/product/11.2.4/db_1/dbs/orapwPROD' auxiliary
format

 '/u01/app/oracle/product/11.2.0/dbhome_1/dbs/orapwSTBDB'   ;

}


executing Memory Script


Starting backup at 20-JAN-17

allocated channel: ORA_DISK_1

channel ORA_DISK_1: SID=36 device type=DISK

Finished backup at 20-JAN-17

contents of Memory Script:

{
```

```
backup as copy current controlfile for standby auxiliary format
'/home/oracle/oradata/STBDB/control01.ctl';

}


executing Memory Script

Starting backup at 20-JAN-17

using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile copy

copying standby control file

output file name=/u03/oracle/product/11.2.4/db_1/dbs/snapcf_PROD.f
tag=TAG20170120T145657 RECID=3 STAMP=933778620

channel ORA_DISK_1: datafile copy complete, elapsed time: 00:02:05

Finished backup at 20-JAN-17


contents of Memory Script:

{

   sql clone 'alter database mount standby database';

}

executing Memory Script

sql statement: alter database mount standby database

contents of Memory Script:

{

   set newname for tempfile  1 to  "/home/oracle/oradata/STBDB/temp01.dbf";

   switch clone tempfile all;

   set newname for datafile  1 to "/home/oracle/oradata/STBDB/system01.dbf";

   set newname for datafile  2 to  "/home/oracle/oradata/STBDB/sysaux01.dbf";

   set newname for datafile  3 to
"/home/oracle/oradata/STBDB/undotbs01.dbf";

   set newname for datafile  4 to "/home/oracle/oradata/STBDB/users01.dbf";

   backup as copy reuse datafile  1 auxiliary format

 "/home/oracle/oradata/STBDB/system01.dbf"

datafile 2 auxiliary format  "/home/oracle/oradata/STBDB/sysaux01.dbf"

datafile 3 auxiliary format "/home/oracle/oradata/STBDB/undotbs01.dbf"
```

```
datafile 4 auxiliary format "/home/oracle/oradata/STBDB/users01.dbf"   ;
   sql 'alter system archive log current';
}

executing Memory Script
executing command: SET NEWNAME
renamed tempfile 1 to /home/oracle/oradata/STBDB/temp01.dbf in control file
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME

Starting backup at 20-JAN-17
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001 name=/u03/oracle/oradata/PROD/system01.dbf
output file name=/home/oracle/oradata/STBDB/system01.dbf
tag=TAG20170120T145917
channel ORA_DISK_1: datafile copy complete, elapsed time: 02:14:37
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002 name=/u03/oracle/oradata/PROD/sysaux01.dbf
output file name=/home/oracle/oradata/STBDB/sysaux01.dbf
tag=TAG20170120T145917
channel ORA_DISK_1: datafile copy complete, elapsed time: 01:24:17
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003 name=/u03/oracle/oradata/PROD/undotbs01.dbf
output file name=/home/oracle/oradata/STBDB/undotbs01.dbf
tag=TAG20170120T145917
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:05:15
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004 name=/u03/oracle/oradata/PROD/users01.dbf
output file name=/home/oracle/oradata/STBDB/users01.dbf
tag=TAG20170120T145917
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:56
```

```
Finished backup at 20-JAN-17

sql statement: alter system archive log current

contents of Memory Script:

{

   switch clone datafile all;

}

executing Memory Script


datafile 1 switched to datafile copy

input datafile copy RECID=3 STAMP=933824671 file
name=/home/oracle/oradata/STBDB/system01.dbf

datafile 2 switched to datafile copy

input datafile copy RECID=4 STAMP=933824671 file
name=/home/oracle/oradata/STBDB/sysaux01.dbf

datafile 3 switched to datafile copy

input datafile copy RECID=5 STAMP=933824671 file
name=/home/oracle/oradata/STBDB/undotbs01.dbf

datafile 4 switched to datafile copy

input datafile copy RECID=6 STAMP=933824671 file
name=/home/oracle/oradata/STBDB/users01.dbf

Finished Duplicate Db at 20-JAN-17

RMAN>
```

Connect to cloud database and query V$DATABASE view:

```
SQL> select name, db_unique_name, database_role, switchover_status from
v$database;

NAME      DB_UNIQUE_NAME                 DATABASE_ROLE    SWITCHOVER_STATUS

___ __________ ______- _______

PROD      STBDB                          PHYSICAL STANDBY TO PRIMARY

SQL>
```

Make sure you set LOG_ARCHIVE_DEST_2 parameter on the on-premises database and specify the instance running on the cloud machine:

```
SQL> ALTER SYSTEM SET log_archive_dest_2='SERVICE=STBDB ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=STBDB';

System altered.

SQL>
```

No switch to the cloud machine and start the apply process:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;

Database altered.

SQL>
```

Ok, the standby database is ready. Perform some logfile switches, create a new table and switch log file again. Move the standby machine and check alert.log file to see if log files are moved and applied to the standby database.

```
SQL> alter system switch logfile;

System altered.
```

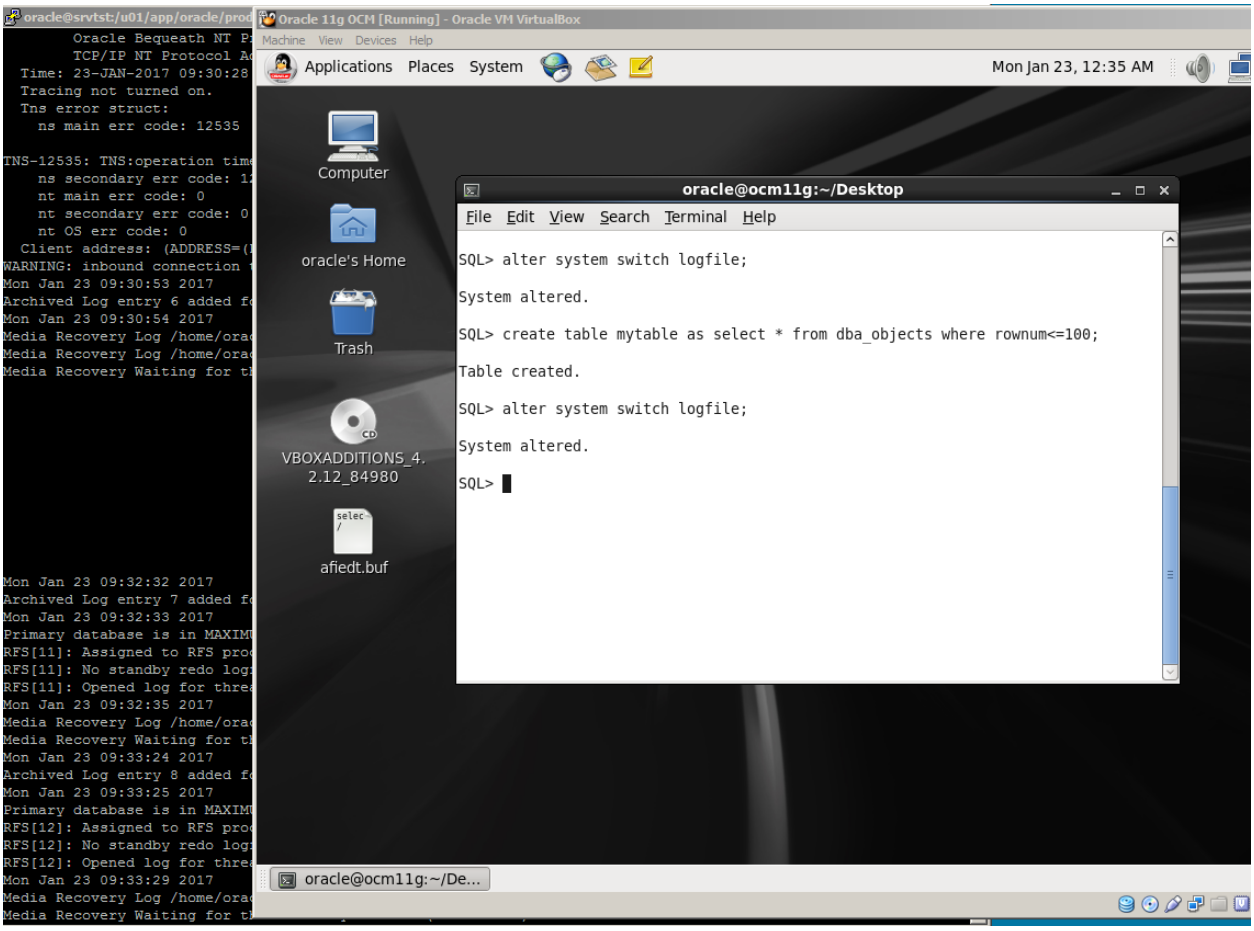
```
SQL> create table mytable as select * from dba_objects where rownum<=100;

Table created.
```

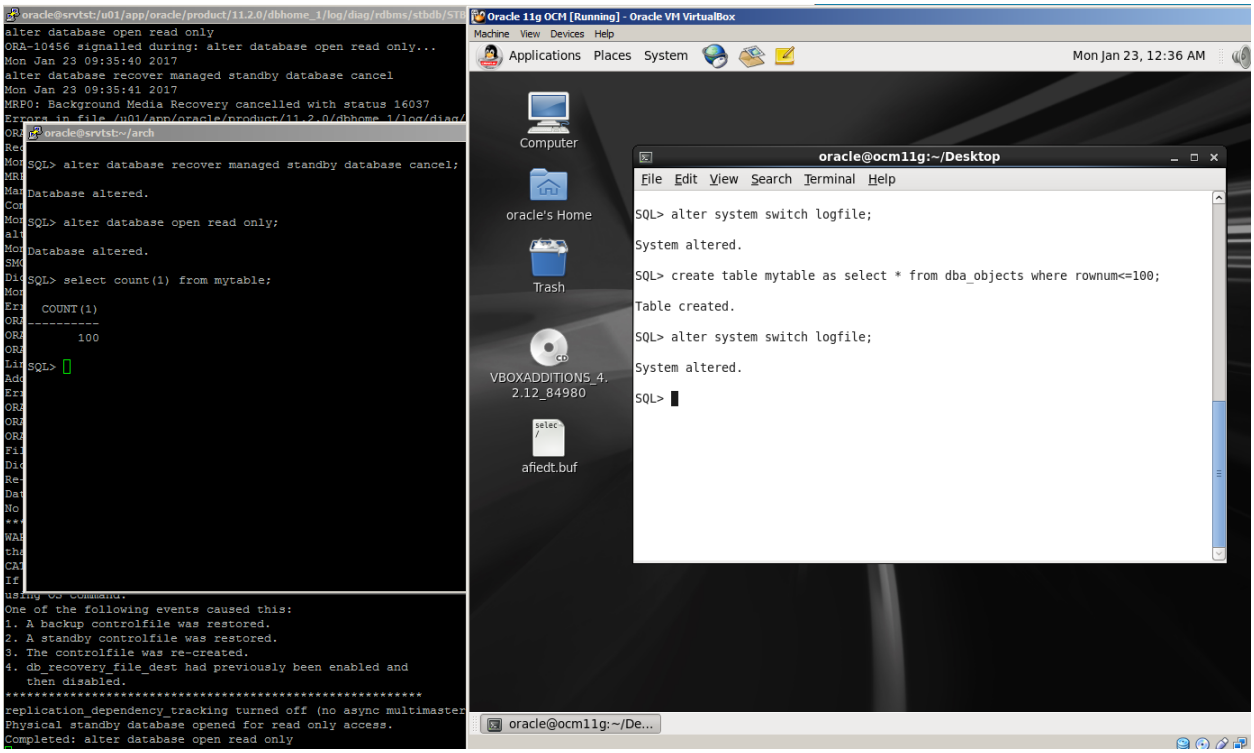
```
SQL> alter system switch logfile;

System altered.

SQL>
```

Next, open the standby database in the read only mode and see if you can query the table created on on-premises database:

```
SQL> alter database recover managed standby database cancel;
```

```
Database altered.

SQL> alter database open read only;
Database altered.

SQL> select count(1) from mytable;
  COUNT(1)
————-
       100
SQL>
```





As you see, the table has been moved within archived log file to the cloud machine and applied to the standby instance.

Performing disaster recovery with RMAN in Oracle Cloud using On-Premises backup stored in Oracle Cloud Backup Storage

In the previous blog posts you have seen how to create a disaster recovery for on-premises Oracle Database by creating a standby database in Oracle Cloud. Sometimes, you might not need to create a standby database, but just store the backup of your database in Oracle Cloud Storage and then use it to create a database in the cloud in the feature. In this blog post I will show you how to take backup of on-premises database to Oracle Cloud Storage and use it to perform a disaster recovery by restoring/recovering from backup to the instance in the cloud and perform recovery of on-premises database using backups stored in the cloud storage using RMAN.

First of all, we need to download and install a backup model to on-premises db. Open the following link and download Oracle Database Cloud Backup Module :

http://www.oracle.com/technetwork/database/availability/oracle-cloud-backup-2162729.html

Create folder to store wallets and lib file, extract the zip file and install it:

```
[oracle@ocm11g ~]$ mkdir wallet lib

[oracle@ocm11g tmp]$ java -jar opc_install.jar -serviceName Storage -
identityDomain yourIdentityDomain -opcID YourOpcId -opcPass YourOpcPassword -
walletDir /home/oracle/wallet -libDir /home/oracle/lib

Oracle Database Cloud Backup Module Install Tool, build 2016-10-07

Oracle Database Cloud Backup Module credentials are valid.

Oracle Database Cloud Backup Module wallet created in directory
/home/oracle/wallet.

Oracle Database Cloud Backup Module initialization file
/u03/oracle/product/11.2.4/db_1/dbs/opcPROD.ora created.

Downloading Oracle Database Cloud Backup Module Software Library from file
opc_linux64.zip.

Downloaded 26528348 bytes in 12 seconds. Transfer rate was 2210695
bytes/second.

Download complete.

[oracle@ocm11g tmp]$
```

The name of on-premises database is PROD. Now connect to RMAN and change the following configurations. Configure the channel to use SBT library which enable to store backups to the cloud (libopc.so) and provide OPC_FILE destination that contains Oracle Backup Cloud Service container URL:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u03/oracle/product/11.2.4/db_1/dbs/opcPROD.ora)';
```

```
new RMAN configuration parameters:

CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u03/oracle/product/11.2.4/db_1/dbs/opcPROD.ora)';

new RMAN configuration parameters are successfully stored
```

Enable autobackup of controlfile:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

```
new RMAN configuration parameters:

CONFIGURE CONTROLFILE AUTOBACKUP ON;

new RMAN configuration parameters are successfully stored
```

Set the high compression for backups to consume less space in the cloud storage:

```
RMAN> CONFIGURE COMPRESSION ALGORITHM 'HIGH';
```

```
new RMAN configuration parameters:

CONFIGURE COMPRESSION ALGORITHM 'HIGH' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE;

new RMAN configuration parameters are successfully stored
```

Change the default channel to tape (media -> Oracle Cloud Backup Storage)

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';
```

```
new RMAN configuration parameters:
```

```
CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';

new RMAN configuration parameters are successfully stored


RMAN>
```

Now connect to RMAN and run SHOW ALL command to see the backup configurations:

```
[oracle@ocm11g ~]$ rman target /


RMAN> show all;

using target database control file instead of recovery catalog

RMAN configuration parameters for database with db_unique_name PROD are:

CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default

CONFIGURE BACKUP OPTIMIZATION OFF; # default

CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE SBT_TAPE TO '%F'; #
default

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; #
default

CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 1 BACKUP TYPE TO BACKUPSET; #
default

CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default

CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE SBT_TAPE TO 1; # default

CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default

CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u03/oracle/product/11.2.4/db_1/dbs/opcPROD.ora)';

CONFIGURE MAXSETSIZE TO UNLIMITED; # default

CONFIGURE ENCRYPTION FOR DATABASE OFF; # default

CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default

CONFIGURE COMPRESSION ALGORITHM 'HIGH' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE;
```

```
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default

CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u03/oracle/product/11.2.4/db_1/dbs/snapcf_PROD.f'; # default
```

Before taking the backup, create a table at on-premises database. We will query it after disaster recovery in the cloud db.

```
SQL> create table mytable as select * from dba_objects where rownum<=100;

Table created.


SQL> select count(1) from mytable;

  COUNT(1)

----------

       100

SQL>
```

Now enable encryption (set the password for backups) and take backup of the database:

```
RMAN> set encryption on identified by "mypass" only;

executing command: SET encryption


RMAN> backup database plus archivelog;

Starting backup at 10-FEB-17

current log archived

allocated channel: ORA_SBT_TAPE_1

channel ORA_SBT_TAPE_1: SID=33 device type=SBT_TAPE

channel ORA_SBT_TAPE_1: Oracle Database Backup Service Library VER=3.16.9.21

channel ORA_SBT_TAPE_1: starting archived log backup set

channel ORA_SBT_TAPE_1: specifying archived log(s) in backup set

input archived log thread=1 sequence=48 RECID=71 STAMP=935603816

channel ORA_SBT_TAPE_1: starting piece 1 at 10-FEB-17

channel ORA_SBT_TAPE_1: finished piece 1 at 10-FEB-17
```

```
piece handle=17rs8bjd_1_1 tag=TAG20170210T175700 comment=API Version 2.0,MMS
Version 3.16.9.21

channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:25

Finished backup at 10-FEB-17


Starting backup at 10-FEB-17

using channel ORA_SBT_TAPE_1

channel ORA_SBT_TAPE_1: starting full datafile backup set

channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set

input datafile file number=00001 name=/u03/oracle/oradata/PROD/system01.dbf

input datafile file number=00002 name=/u03/oracle/oradata/PROD/sysaux01.dbf

input datafile file number=00003 name=/u03/oracle/oradata/PROD/undotbs01.dbf

input datafile file number=00004 name=/u03/oracle/oradata/PROD/users01.dbf

channel ORA_SBT_TAPE_1: starting piece 1 at 10-FEB-17

channel ORA_SBT_TAPE_1: finished piece 1 at 10-FEB-17

piece handle=18rs8bk6_1_1 tag=TAG20170210T175726 comment=API Version 2.0,MMS
Version 3.16.9.21

channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 02:57:07

Finished backup at 10-FEB-17


Starting backup at 10-FEB-17

current log archived

using channel ORA_SBT_TAPE_1

channel ORA_SBT_TAPE_1: starting archived log backup set

channel ORA_SBT_TAPE_1: specifying archived log(s) in backup set

input archived log thread=1 sequence=49 RECID=72 STAMP=935605482

input archived log thread=1 sequence=50 RECID=73 STAMP=935614475

channel ORA_SBT_TAPE_1: starting piece 1 at 10-FEB-17

channel ORA_SBT_TAPE_1: finished piece 1 at 10-FEB-17

piece handle=1ars8m0c_1_1 tag=TAG20170210T205435 comment=API Version 2.0,MMS
Version 3.16.9.21

channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:09:25

Finished backup at 10-FEB-17
```

```
Starting Control File and SPFILE Autobackup at 10-FEB-17

piece handle=c-345613202-20170210-02 comment=API Version 2.0,MMS Version
3.16.9.21

Finished Control File and SPFILE Autobackup at 10-FEB-17


RMAN>
```

The backup command completed successfully and all backups are stored in Oracle Cloud Backup Storage.

Now let's perform a disaster recovery in the cloud machine. Create a new cloud database instance, configure SSH connection from on-premises to the cloud host. Copy opc_install.zip file you have downloaded from OTN to the cloud host and install it as you did it at on-premises host. Drop the database if there's any, connect to RMAN and start it in NOMOUNT mode. Provide the RMAN password, allocate a channel as you did at on-premises database and restore the spfile:

```
RMAN> STARTUP NOMOUNT;

RMAN> set decryption identified by "mypass";


executing command: SET decryption

using target database control file instead of recovery catalog


RMAN> run

2> {

3> allocate channel t1 type 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u01/app/oracle/product/11.2.0/dbhome_1/dbs/opcPROD.ora)';

4> set dbid=345613202;

5> restore spfile to pfile '/tmp/pfile.ora' from autobackup;

6> }


allocated channel: t1

channel t1: SID=171 device type=SBT_TAPE
```

```
channel t1: Oracle Database Backup Service Library VER=3.16.9.21

executing command: SET DBID

Starting restore at 11-FEB-17

channel t1: looking for AUTOBACKUP on day: 20170211
channel t1: looking for AUTOBACKUP on day: 20170210
channel t1: AUTOBACKUP found: c-345613202-20170210-02
channel t1: restoring spfile from AUTOBACKUP c-345613202-20170210-02
channel t1: SPFILE restore from AUTOBACKUP complete
Finished restore at 11-FEB-17
released channel: t1

RMAN>
```

Server parmeter file is restored. If you need to specify different location for some parameters, create a readable parameter file from it, make your changes, create a server parameter file from it and start the database in NOMOUNT mode using the restored (and modified) spfile.

```
SQL> startup nomount force;
ORACLE instance started.

Total System Global Area 1235959808 bytes
Fixed Size                  2252784 bytes
Variable Size             385875984 bytes
Database Buffers          838860800 bytes
Redo Buffers                8970240 bytes
SQL> exit
```

Now restore controlfile from autobackup:

```
RMAN> set decryption identified by "mypass";

executing command: SET decryption

RMAN> run

2> {

3> allocate channel t1 type 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u01/app/oracle/product/11.2.0/dbhome_1/dbs/opcPROD.ora)';

4> set dbid=345613202;

5> restore controlfile from autobackup;

6> }


allocated channel: t1

channel t1: SID=134 device type=SBT_TAPE

channel t1: Oracle Database Backup Service Library VER=3.16.9.21


executing command: SET DBID


Starting restore at 11-FEB-17


channel t1: looking for AUTOBACKUP on day: 20170211

channel t1: looking for AUTOBACKUP on day: 20170210

channel t1: AUTOBACKUP found: c-345613202-20170210-02

channel t1: restoring control file from AUTOBACKUP c-345613202-20170210-02

channel t1: control file restore from AUTOBACKUP complete

output file name=/u04/app/oracle/oradata/control01.ctl

output file name=/u04/app/oracle/oradata/control02.ctl

Finished restore at 11-FEB-17

released channel: t1


RMAN>
```

Controlfile are restored. Start the database in MOUNT mode and restore the datafiles. Specify a new folder using SET NEWNAME FOR DATABASE TO command as follows:

```
RMAN> run

2> {

3> allocate channel t1 type 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u01/app/oracle/product/11.2.0/dbhome_1/dbs/opcPROD.ora)';

4> set newname for database to '/u04/app/oracle/oradata/%U.dbf';

5> restore database;

6> switch datafile all;

7> }


allocated channel: t1

channel t1: SID=133 device type=SBT_TAPE

channel t1: Oracle Database Backup Service Library VER=3.16.9.21


executing command: SET NEWNAME


Starting restore at 11-FEB-17

Starting implicit crosscheck backup at 11-FEB-17

Crosschecked 1 objects

Finished implicit crosscheck backup at 11-FEB-17


Starting implicit crosscheck copy at 11-FEB-17

Crosschecked 2 objects

Finished implicit crosscheck copy at 11-FEB-17


searching for all files in the recovery area

cataloging files...

no files cataloged
```

```
channel t1: starting datafile backup set restore

channel t1: specifying datafile(s) to restore from backup set

channel t1: restoring datafile 00001 to /u04/app/oracle/oradata/data_D-
PROD_TS-SYSTEM_FNO-1.dbf

channel t1: restoring datafile 00002 to /u04/app/oracle/oradata/data_D-
PROD_TS-SYSAUX_FNO-2.dbf

channel t1: restoring datafile 00003 to /u04/app/oracle/oradata/data_D-
PROD_TS-UNDOTBS1_FNO-3.dbf

channel t1: restoring datafile 00004 to /u04/app/oracle/oradata/data_D-
PROD_TS-USERS_FNO-4.dbf

channel t1: reading from backup piece 18rs8bk6_1_1

channel t1: piece handle=18rs8bk6_1_1 tag=TAG20170210T175726

channel t1: restored backup piece 1

channel t1: restore complete, elapsed time: 00:00:45

Finished restore at 11-FEB-17


datafile 1 switched to datafile copy

input datafile copy RECID=14 STAMP=935693831 file
name=/u04/app/oracle/oradata/data_D-PROD_TS-SYSTEM_FNO-1.dbf

datafile 2 switched to datafile copy

input datafile copy RECID=15 STAMP=935693831 file
name=/u04/app/oracle/oradata/data_D-PROD_TS-SYSAUX_FNO-2.dbf

datafile 3 switched to datafile copy

input datafile copy RECID=16 STAMP=935693831 file
name=/u04/app/oracle/oradata/data_D-PROD_TS-UNDOTBS1_FNO-3.dbf

datafile 4 switched to datafile copy

input datafile copy RECID=17 STAMP=935693831 file
name=/u04/app/oracle/oradata/data_D-PROD_TS-USERS_FNO-4.dbf

released channel: t1


RMAN>
```

Now run ALTER DATABASE RENAME FILE command to rename redo log files:

```
SQL> alter database rename file '/u03/oracle/oradata/PROD/redo03.log' to
'/u04/app/oracle/oradata/redo03.log';

Database altered.


SQL> alter database rename file '/u03/oracle/oradata/PROD/redo02.log' to
'/u04/app/oracle/oradata/redo02.log';

Database altered.


SQL> alter database rename file '/u03/oracle/oradata/PROD/redo01.log' to
'/u04/app/oracle/oradata/redo01.log';

Database altered.


SQL>
```

Now run RECOVER DATABASE command to recover the database and open the database:

```
RMAN> set decryption identified by "mypass";

executing command: SET decryption

RMAN> run

2> {

3> allocate channel t1 type 'SBT_TAPE' PARMS
'SBT_LIBRARY=/home/oracle/lib/libopc.so
ENV=(OPC_PFILE=/u01/app/oracle/product/11.2.0/dbhome_1/dbs/opcPROD.ora)';

4> recover database;

5> }


allocated channel: t1

channel t1: SID=125 device type=SBT_TAPE

channel t1: Oracle Database Backup Service Library VER=3.16.9.21


Starting recover at 11-FEB-17


starting media recovery
```

```
channel t1: starting archived log restore to default destination

channel t1: restoring archived log

archived log thread=1 sequence=49

channel t1: restoring archived log

archived log thread=1 sequence=50

channel t1: reading from backup piece 1ars8m0c_1_1

channel t1: piece handle=1ars8m0c_1_1 tag=TAG20170210T205435

channel t1: restored backup piece 1

channel t1: restore complete, elapsed time: 00:00:07

archived log file
name=/u03/app/oracle/fast_recovery_area/PROD/archivelog/2017_02_11/o1_mf_1_49
_d9yqs878_.arc thread=1 sequence=49

channel default: deleting archived log(s)

archived log file
name=/u03/app/oracle/fast_recovery_area/PROD/archivelog/2017_02_11/o1_mf_1_49
_d9yqs878_.arc RECID=75 STAMP=935693995

archived log file
name=/u03/app/oracle/fast_recovery_area/PROD/archivelog/2017_02_11/o1_mf_1_50
_d9yqs8cn_.arc thread=1 sequence=50

channel default: deleting archived log(s)

archived log file
name=/u03/app/oracle/fast_recovery_area/PROD/archivelog/2017_02_11/o1_mf_1_50
_d9yqs8cn_.arc RECID=74 STAMP=935693994

unable to find archived log

archived log thread=1 sequence=51

released channel: t1

RMAN-00571: ===========================================================

RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============

RMAN-00571: ===========================================================

RMAN-03002: failure of recover command at 02/11/2017 19:00:00

RMAN-06054: media recovery requesting unknown archived log for thread 1 with
sequence 51 and starting SCN of 1153764


RMAN> alter database open resetlogs;

database opened
```

```
RMAN>
```

Connect to SQL*Plus and query the table you have created before taking a backup at on-premises database:

```
SQL> select count(1) from mytable;


  COUNT(1)
----------
       100


SQL>
```

Great! We have successfully performed a disaster recovery of on-premises database to the cloud using RMAN backups stored in Oracle Cloud Backup Storage!

Now let's use backups stored in the cloud to perform a recovery to on-premises database. Let's create a new table, take backup of the datafile, corrupt a block of the datafile and recover it from backups stored in the cloud.

```
SQL> create table test_table tablespace users as select * from dba_objects
where rownum<=10;

Table created.


RMAN> set encryption on identified by "mypass" only;

executing command: SET encryption


RMAN> backup datafile 4;


Starting backup at 11-FEB-17

using channel ORA_SBT_TAPE_1

channel ORA_SBT_TAPE_1: starting full datafile backup set

channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set

input datafile file number=00004 name=/u03/oracle/oradata/PROD/users01.dbf

channel ORA_SBT_TAPE_1: starting piece 1 at 11-FEB-17
```

```
channel ORA_SBT_TAPE_1: finished piece 1 at 11-FEB-17

piece handle=1drsaim0_1_1 tag=TAG20170211T141008 comment=API Version 2.0,MMS
Version 3.16.9.21

channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:45

Finished backup at 11-FEB-17


Starting Control File and SPFILE Autobackup at 11-FEB-17

piece handle=c-345613202-20170211-00 comment=API Version 2.0,MMS Version
3.16.9.21

Finished Control File and SPFILE Autobackup at 11-FEB-17


RMAN> exit



SQL> SELECT header_block FROM dba_segments WHERE segment_name='TEST_TABLE';

HEADER_BLOCK

------------

       170


SQL>


[oracle@ocm11g ~]$ dd of=/u03/oracle/oradata/PROD/users01.dbf bs=8192
conv=notrunc seek=170 <<EOF

> Corruption

> Corruption

> EOF

0+1 records in

0+1 records out

23 bytes (23 B) copied, 0.000147784 s, 156 kB/s

[oracle@ocm11g ~]$ sqlplus / as sysdba


SQL> alter system flush buffer_cache;

System altered.
```

```
SQL> select count(1) from test_table;
select count(1) from test_table
                    *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 4, block # 170)
ORA-01110: data file 4: '/u03/oracle/oradata/PROD/users01.dbf'

SQL> select * from v$database_block_corruption;

     FILE#     BLOCK#     BLOCKS CORRUPTION_CHANGE# CORRUPTIO
---------- ---------- ---------- ------------------ ---------
       4      170        1            0 CORRUPT
```

Ok, we have a corrupted block. Now connect to RMAN and recover it:

```
RMAN> recover datafile 4 block 170;

Starting recover at 11-FEB-17
using channel ORA_SBT_TAPE_1
using channel ORA_DISK_1

channel ORA_SBT_TAPE_1: restoring block(s)
channel ORA_SBT_TAPE_1: specifying block(s) to restore from backup set
restoring blocks of datafile 00004
channel ORA_SBT_TAPE_1: reading from backup piece 1drsaim0_1_1
channel ORA_SBT_TAPE_1: piece handle=1drsaim0_1_1 tag=TAG20170211T141008
channel ORA_SBT_TAPE_1: restored block(s) from backup piece 1
channel ORA_SBT_TAPE_1: block restore complete, elapsed time: 00:00:15
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 11-FEB-17

RMAN> exit

[oracle@ocm11g ~]$ sqlplus / as sysdba
SQL> select count(1) from test_table;
  COUNT(1)
----------
      10
SQL>
```

As you see, we used backups stored in Oracle Cloud Backup Storage to recover a corrupted block of on-premises database.