ORACLE
UNIVERSITY

Hardware and Software
Engineered to Work Together

Exadata Database Machine: 12*c* Administration Workshop

Activity Guide
D92887GC20
Edition 2.0 | August 2016 | D97407

Learn more from Oracle University at **oracle.com/education/**

ORACLE

Author

Peter Fusek

Technical Contributors and Reviewers

Akshay Shah, Alex Tsukerman, Amit Ganesh, Andrew Babb, Aslam Edah-Tally, Barb Lundhild, Bharat Baddepudi, Bill Hodak, Boris Erlikhman, Branislav Valny, Bruce Kyro, Bruno Dagostini, Bundit Jianpinitnan, Caroline Johnston, Christian Craft, Dan Norris, Darryl Presley, Dave Winter, David Hernandez Mendoza, David Hitchcock, Deba Chatterjee, Douglas Utzig, Ed Gilowski, Eric Siglin, Francois Marcoux, Georg Schmidt, Gerlinde Frenzen, Harald van Breederode, Ira Singer, James He, James Womack, Jean-Francois Verrier, Jerry Lee, Jia Shi, Jignesh Patel, Jim Hall, Jim Spiller, Jim Viscusi, Joe Fu, Joel Goodman, Juan Loaiza, Kam Shergill, Kevin Jernigan, Kodi Umamageswaran, Krishnanjani Chitta, Lachlan Williams, Larry Justice, Lawrence To, Louis Nagode, Mahesh Subramaniam, Maria Billings, Maria Colgan, Mark Fuller, Mark Scardina, Mark Van de Wiel, Marshall Presser, Martin Jensen, Michael Cebulla, Michael Nowak, Naoki Kato, Nikolaus Dilger, Nilesh Choudhury, Ravindra Dani, Raymond Dutcher, Richard Exley, Robert Carlin, Robert Pastijn, Roger Hansen, Sabyasachi Banerjee, Sean Kim, Selcuk Aya, Scott Gossett, Sriram Palapudi, Steven Lemme, Sue Lee, Sugam Pandey, Sumeet Lahorani, Sundararaman Sridharan, Suresh Gogune, Tim Shetler, Trevor Bowen, Umesh Panchaksharaiah, Uwe Hesse, Varun Malhotra, Vern Wagman, Vijay Sridharan, Vikram Kapoor, Vimala Jacob

This book was published using: Oracle Tutor

Table of Contents

Course Practice Environment: Security Credentials

Chapter I

Course Practice Environment: Security Credentials

For OS usernames and passwords, see the following:

- If you are attending a classroom-based or live virtual class, ask your instructor or LVC producer for OS credential information.
- If you are using a self-study format, refer to the communication that you received from Oracle University for this course.

For other credentials used in this course, see the following table:

Product-Specific Credentials		
Product/Application	**Username**	**Password**
Gateway VM and VNC	`oracle`	`oracle`
Exadata Database Server VMs: `qr01dbadm01` and `qr01dbadm02`	`root`, `oracle`, `grid`	`welcome1`
Exadata Storage Server VMs: `qr01celadm01`, `qr01celadm02` and `qr01celadm03`	`root`	`welcome1`
Exadata Storage Server VMs: `qr01celadm01`, `qr01celadm02` and `qr01celadm03`	`celladmin`, `cellmonitor`	`welcome`
Enterprise Manager VM: `em12c`	`oracle`	`oracle`
Oracle Database: `dbm`	`sys`, `system`	`welcome1`
Oracle Database: `dbm`	`sh`	`sh`
Oracle ASM	`asmsnmp`	`welcome1`
Enterprise Manager Cloud Control	`sysman`	`Oracle123`
InfiniBand Switch: `qr01sw-iba01`	`nm2user`	`changeme`
Database Server ILOM	`oemuser`	`oemuser`

Practices for Lesson 1: Introduction

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

In this practice, you will be introduced to the laboratory environment used to support all the practices during this course.

Practice 1-1: Lab Environment Introduction

Overview

In this practice, you will learn how to use the laboratory environment that supports all the practices in this course.

The laboratory environment for this course is based on a Quarter Rack Database Machine. It consists of several virtual machines configured to provide two database servers and three Exadata Storage Servers.

To access the virtual machines, you will first establish a graphical session which is connected to a gateway VM. Your instructor will provide specific details for each student's gateway. From there, you will create terminal sessions as required and connect to the virtualized Database Machine servers by using SSH as described in the tasks for this practice.

During this practice, take care to remember the steps required to establish a terminal session and how to configure the environment to access ASM or Oracle Database. You will be required to repeat these steps many times during the practice exercises in this course.

Please refer to the supplied Course Practice Environment: Security Credentials document for the passwords that you will require to use the laboratory environment.

Tasks

1. Establish a terminal session connected to `qr01dbadm01` by using the `grid` operating system account.

   Note that you may see additional messages relating to server identities. Answer `yes` if you are prompted to acknowledge server authenticity.

```
$ ssh grid@qr01dbadm01
grid@qr01dbadm01's password: ????????
[grid@qr01dbadm01 ~]$
```

2. Configure your terminal session to access ASM using the `+ASM1` instance on `qr01dbadm01`.

```
[grid@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@qr01dbadm01 ~]$
```

3. Execute the following command and verify that all of the listed services are online in your laboratory environment. Your output should look like the example below. Alert your instructor if you do not have the same services online in your environment.

```
[grid@qr01dbadm01 ~]$ crsctl status resource -w "TARGET = ONLINE" -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA_QR01.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.DBFS_DG.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
```

```
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.RECO_QR01.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.asm
               ONLINE  ONLINE       qr01dbadm01              Started,STABLE
               ONLINE  ONLINE       qr01dbadm02              Started,STABLE
ora.net1.network
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.ons
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.cvu
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.dbm.db
      1        ONLINE  ONLINE       qr01dbadm01              Open,STABLE
      2        ONLINE  ONLINE       qr01dbadm02              Open,STABLE
ora.oc4j
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.qr01dbadm01.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.qr01dbadm02.vip
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
--------------------------------------------------------------------------------
[grid@qr01dbadm01 ~]$
```

4. Using SQL*Plus, connect to the clustered ASM environment as an ASM administrator. Verify that you are connected to the ASM instance `+ASM1` and exit your SQL*Plus session.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production ...

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
+ASM1

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition ...
[grid@qr01dbadm01 ~]$
```

5. Establish a new shell as the `oracle` OS user.

```
[grid@qr01dbadm01 ~]$ su - oracle
Password: ????????
[oracle@qr01dbadm01 ~]$
```

6. Configure your terminal session to access the `dbm` database using the `dbm1` instance.

```
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

7. Using SQL*Plus, connect to your database as the database administrator. Verify that you are connected to the `DBM` database and exit your SQL*Plus session.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production ...

SQL> select name from v$database;

NAME
---------
DBM

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition ...
[oracle@qr01dbadm01 ~]$
```

8. Use the `srvctl` utility to verify the status of the `DBM` database and the placement of the database instances.

```
[oracle@qr01dbadm01 ~]$ srvctl status database -d dbm
Instance dbm1 is running on node qr01dbadm01
Instance dbm2 is running on node qr01dbadm02
[oracle@qr01dbadm01 ~]$
```

Note that the `DBM` database is an administrator-managed Oracle RAC database created on two database servers; `qr01dbadm01` and `qr01dbadm02`. You will use the database instance running on `qr01dbadm01` to perform most the practices.

9. Many practices refer to SQL scripts, which are provided as an alternative to typing lengthy commands. These scripts are located in the `labs` directory under your student home directory. List the files in the `labs` directory. Notice also the `CSV`, `oeda`, `swingbench` and `TTS` subdirectories. These subdirectories contain additional files for specific practices. You will be directed to these files in the associated practice instructions.

```
[oracle@qr01dbadm01 ~]$ cd labs
[oracle@qr01dbadm01 labs]$ ls
CSV              lab04-03-15.sql  lab06-04-22.sql  lab15-01-32.sql
lab03-02-03.sql  lab04-04-03.sql  lab08-02-04.sql  lab15-01-34.sql
lab03-03-10.sql  lab04-04-04.sql  lab08-02-06.sql  lab15-01-35.sql
lab03-03-11.sql  lab06-02-03.sql  lab08-02-07.sql  lab20-01-04.sql
lab04-01-03.sql  lab06-02-04.sql  lab08-02-15.sql  lab20-01-16.sql
lab04-01-04.sql  lab06-02-08.sql  lab09-01-06.sql  lab20-01-17.sql
lab04-01-08.sql  lab06-02-11.sql  lab09-01-07.sql  lab20-01-22.sql
lab04-02-03.sql  lab06-02-20.sql  lab09-02-07.sql  lab20-02-13.sql
lab04-02-04.sql  lab06-02-30.sql  lab09-02-08.sql  oeda
lab04-02-06.sql  lab06-02-40.sql  lab09-02-09.sql  swingbench
lab04-02-07.sql  lab06-03-03.sql  lab11-01-05.sql  TTS
lab04-03-05.sql  lab06-03-04.sql  lab11-01-15.sql
lab04-03-08.sql  lab06-03-05.sql  lab12-01-22.sql
[oracle@qr01dbadm01 labs]$
```

10. Establish a separate terminal session connected to your first Exadata cell using the `celladmin` user. Confirm that you are connected to the cell and then exit the session.

Note that you may see additional messages relating to server identities. Answer `yes` if you are prompted to acknowledge server authenticity.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$ cellcli -e list cell
         qr01celadm01    online
[celladmin@qr01celadm01 ~]$ exit
logout
Connection to qr01celadm01 closed.
$
```

11. Establish another terminal session connected to your second Exadata cell using the `cellmonitor` user. Confirm you are connected to the cell and then exit the session.

    Note that you may see additional messages related to server identities. Answer `yes` if you are prompted to acknowledge server authenticity.

```
$ ssh cellmonitor@qr01celadm02
cellmonitor@qr01celadm02's password: ????????
[cellmonitor@qr01celadm02 ~]$ cellcli -e list cell
         qr01celadm02    online
[cellmonitor@qr01celadm02 ~]$ exit
logout
Connection to qr01celadm02 closed.
$
```

12. Exit your terminal sessions. We recommend that you start fresh terminal sessions at the beginning of each practice, and that you exit all of your terminal sessions at the conclusion of every practice. This eliminates the possibility that environment settings used in one practice could cause problems in following practices.

Practices for Lesson 2: Exadata Database Machine - Overview

Chapter 2

Practices for Lesson 2

There are no practices for this lesson.

Practices for Lesson 3: Exadata Database Machine Architecture

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

In these practices, you will be familiarized with the Exadata cell architecture. You will:

- Examine the Exadata processes
- Exercise Exadata high availability
- Examine the hierarchy of cell objects
- Examine Exadata Smart Flash Cache

Practice 3-1: Process Familiarization

Overview

In this practice, you will examine the Exadata cell software processes.

Tasks

1. Establish a terminal connection to `qr01celadm01` as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

2. Restart Server (`RS`) is used to start up and shut down the Cell Server (`CELLSRV`) and Management Server (`MS`). It also monitors these services to check if they need to be restarted. Locate the `RS` processes by using the following command.

```
[celladmin@qr01celadm01 ~]$ ps -ef | grep cellrs
root      1928     1  0 02:53 ?        00:00:00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrssrm -ms 1 -
cellsrv 1
root      1935  1928  0 02:53 ?        00:00:00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrsbmt -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
root      1936  1928  0 02:53 ?        00:00:00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrsmmt -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
root      1937  1928  0 02:53 ?        00:00:02
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrsomt -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
root      1938  1935  0 02:53 ?        00:00:00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrsbkm -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
root      1945  1938  0 02:53 ?        00:00:00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrssmt -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
1000      4344  4294  0 03:16 pts/0    00:00:00 grep cellrs
[celladmin@qr01celadm01 ~]$
```

3. MS provides Exadata cell management and configuration. It works in cooperation with the Exadata cell command-line interface (CellCLI). In addition, MS is responsible for sending alerts and collects some statistics in addition to those collected by CELLSRV. Locate the MS process by using the following command:

```
[celladmin@qr01celadm01 ~]$ ps -ef | grep msServer
root      2007  1939  1 02:53 ?        00:00:28 /usr/java/jdk1.7.0_72/bin/java
-client -Xms256m -Xmx512m -XX:CompileThreshold=8000 -XX:PermSize=128m -
XX:MaxPermSize=256m -Dweblogic.Name=msServer -
Djava.security.policy=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/de
ploy/wls/wlserver_10.3/server/lib/weblogic.policy -XX:-UseLargePages -
XX:ParallelGCThreads=8 -Dweblogic.ListenPort=8888 -
Djava.security.egd=file:/dev/./urandom -Xverify:none -da -
Dplatform.home=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/wl
s/wlserver_10.3 -
Dwls.home=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/wls/wls
erver_10.3/server -
Dweblogic.home=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/wl
s/wlserver_10.3/server -Dweblogic.management.discover=true -Dwlw.iterativeDev=
-Dwlw.testConsole= -Dwlw.logErrorsToConsole= -
Dweblogic.ext.dirs=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deplo
y/wls/patch_wls1036/profiles/default/sysext_manifest_classpath weblogic.Server
1000      4463  4294  0 03:20 pts/0    00:00:00 grep msServer
[celladmin@qr01celadm01 ~]$
```

4. Locate the `MS` parent process (`startWebLogic.sh`) and its parent process. Start by using the parent process number associated with `MS` in the output for step 3. When you trace back through the line of processes you will eventually find one of the `RS` processes, which demonstrates that `RS` spawns (and, when required, re-spawns) `MS`.

```
[celladmin@qr01celadm01 ~]$ ps -ef | grep 1939
root      1939  1936  0 02:53 ?        00:00:00 /bin/sh
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/msdomain/bin/star
tWebLogic.sh
root      2007  1939  0 02:53 ?        00:00:32 /usr/java/jdk1.7.0_72/bin/java
-client -Xms256m -Xmx512m -XX:CompileThreshold=8000 -XX:PermSize=128m -
XX:MaxPermSize=256m -Dweblogic.Name=msServer -
Djava.security.policy=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/de
ploy/wls/wlserver_10.3/server/lib/weblogic.policy -XX:-UseLargePages -
XX:ParallelGCThreads=8 -Dweblogic.ListenPort=8888 -
Djava.security.egd=file:/dev/./urandom -Xverify:none -da -
Dplatform.home=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/wl
s/wlserver_10.3 -
Dwls.home=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/wls/wls
erver_10.3/server -
Dweblogic.home=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/wl
s/wlserver_10.3/server -Dweblogic.management.discover=true -Dwlw.iterativeDev=
-Dwlw.testConsole= -Dwlw.logErrorsToConsole= -
Dweblogic.ext.dirs=/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deplo
y/wls/patch_wls1036/profiles/default/sysext_manifest_classpath weblogic.Server
1000      7761  4294  0 03:48 pts/0    00:00:00 grep 1939
[celladmin@qr01celadm01 ~]$ ps -ef|grep 1936
root      1936  1928  0 02:53 ?        00:00:00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrsmmt -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
root      1939  1936  0 02:53 ?        00:00:00 /bin/sh
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/msdomain/bin/star
tWebLogic.sh
1000      7781  4294  0 03:48 pts/0    00:00:00 grep 1936
[celladmin@qr01celadm01 ~]$
```

5. `CELLSRV` is the primary Exadata software component and provides the majority of Exadata storage services. `CELLSRV` is a multithreaded server. Primarily, `CELLSRV` communicates with Oracle Database to serve simple block requests, such as database buffer cache reads, and Smart Scan requests, such as table scans with projections and filters. `CELLSRV` also implements I/O Resource Management (IORM) and collects numerous statistics relating to its operations. Locate the `CELLSRV` process by using the following command:

```
[celladmin@qr01celadm01 ~]$ ps -ef | grep "/cellsrv "
root      1941  1937 20 02:53 ?        00:14:30
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellsrv 40 3000 9
5042
1000      8353  4294  0 04:03 pts/0    00:00:00 grep /cellsrv
[celladmin@qr01celadm01 ~]$
```

6. Locate the CELLSRV parent process. Use the parent process number associated with CELLSRV in the output for step 5. Note that RS spawns (and, when required, re-spawns) CELLSRV.

```
[celladmin@qr01celadm01 ~]$ ps -ef | grep 1937
root      1937  1928  0 02:53 ?        00:00:05
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellrsomt -rs_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellinit.o
ra -ms_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsms.s
tate -cellsrv_conf
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/deploy/config/cellrsos.s
tate -debug 0
root      1941  1937 20 02:53 ?        00:14:46
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellsrv 40 3000 9
5042
1000      8385  4294  0 04:04 pts/0    00:00:00 grep 1937
[celladmin@qr01celadm01 ~]$
```

7. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

8. Execute the following CellCLI command to examine the attributes of the cell. Note that the output also confirms that `CELLSRV`, `MS`, and `RS` are currently up and running. Exit CellCLI after examining the cell attributes.

```
CellCLI> list cell detail
         name:                   qr01celadm01
         cellVersion:            OSS_12.1.2.1.0_LINUX.X64_141206.1
         cpuCount:               2
         diagHistoryDays:        7
         fanCount:               0/0
         fanStatus:              normal
         flashCacheMode:         WriteThrough
         id:                     ef92136a-837c-4e1d-88d2-e01f5ab89b7b
         interconnectCount:      0
         interconnect1:          ib0
         interconnect2:          ib1
         iormBoost:              0.0
         ipaddress1:             192.168.1.105/24
         ipaddress2:             192.168.1.106/24
         kernelVersion:          2.6.39-400.243.1.el6uek.x86_64
         makeModel:              Fake hardware
         memoryGB:               4
         metricHistoryDays:      7
         offloadGroupEvents:
         offloadEfficiency:      650.9
         powerCount:             0/0
         powerStatus:            normal
         releaseVersion:         12.1.2.1.0
         releaseTrackingBug:     17885582
         status:                 online
         temperatureReading:     0.0
         temperatureStatus:      normal
         upTime:                 0 days, 1:23
         cellsrvStatus:          running
         msStatus:               running
         rsStatus:               running

CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

9. Exit all of your terminal sessions.

Practice 3-2: Exadata high availability

Overview

In this practice, you will observe some of the high availability features of Exadata Storage Server.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Execute the SQL script `/home/oracle/labs/lab03-02-03.sql`. The script contains a series of IO intensive queries, which are used in this practice to demonstrate how Oracle Database is insulated from the different Exadata failure scenarios that are demonstrated in the practice. Check on the workload periodically throughout the practice. If the workload completes before you finish all the tasks, then simply re-execute the script to maintain an active workload throughout the practice.

```
SQL> @/home/oracle/labs/lab03-02-03
SQL> set timing on
SQL> select count(*) from sales;

...
```

4. Establish a separate terminal connection to `qr01celadm01` as the `root` user. Leave your SQL*Plus terminal session and workload running in the background.

```
$ ssh root@qr01celadm01
root@qr01celadm01's password: ????????
[root@qr01celadm01 ~]$
```

5. Locate the process identification number for `CELLSRV` by using the following `ps` command.

```
[root@qr01celadm01 ~]# ps -ef | grep "/cellsrv "
root      1941  1937 20 02:53 ?        00:24:43
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellsrv 40 3000 9
5042
root      9658  9615  0 04:52 pts/0    00:00:00 grep /cellsrv
[root@qr01celadm01 ~]#
```

6. Terminate the `CELLSRV` process by using the `kill` command and the process identification number you observed in step 5.

```
[root@qr01celadm01 ~]# kill -9 1941
[root@qr01celadm01 ~]#
```

7. Re-execute the `ps` command from step 5. You should observe that `CELLSRV` is automatically restarted with a new process identification number. How was `CELLSRV` restarted?

```
[root@qr01celadm01 ~]# ps -ef | grep "/cellsrv "
root      9747  9745 76 04:53 ?        00:00:04
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellsrv 40 3000 9
5042
root     10092  9615  0 04:53 pts/0    00:00:00 grep /cellsrv
[root@qr01celadm01 ~]#
```

8. Check on the progress of your workload from step 3. You should observe that the workload continues without error.

```
...
SQL> select count(*) from sales where amount_sold > 2;

  COUNT(*)
----------
  28856060

Elapsed: 00:01:04.88
SQL> select count(*) from sales where amount_sold > 3;

  COUNT(*)
----------
  28555621

Elapsed: 00:00:13.87
SQL> select count(*) from sales where amount_sold > 4;
```

9. On `qr01celadm01`, launch the Exadata cell command-line interface (CellCLI).

```
[root@qr01celadm01 ~]# cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

10. You have already seen how Exadata automatically recovers from an unexpected process failure. Now observe the effect of restarting all the Exadata services.

```
CellCLI> alter cell restart services all

Stopping the RS, CELLSRV, and MS services...
The SHUTDOWN of services was successful.
Starting the RS, CELLSRV, and MS services...
Getting the state of RS services...  running
Starting CELLSRV services...
The STARTUP of CELLSRV services was successful.
Starting MS services...
The STARTUP of MS services was successful.

CellCLI>
```

11. Check on the progress of your workload from step 3. You should again observe that the workload continues without error.

```
...
SQL> select count(*) from sales where amount_sold > 11;

  COUNT(*)
----------
  26150787

Elapsed: 00:00:13.66
SQL> select count(*) from sales where amount_sold > 12;

  COUNT(*)
----------
  25850432

Elapsed: 00:00:11.19
SQL> select count(*) from sales where amount_sold > 13;
```

12. Exit your CellCLI session and re-execute the `ps` command from step 5. You should observe that `CELLSRV` has a different process identification number because of the restart operation executed in step 10.

```
CellCLI> exit
quitting

[root@qr01celadm01 ~]# ps -ef | grep "/cellsrv "
root     10613 10596 34 04:55 ?        00:00:09
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/cellsrv/bin/cellsrv 40 3000 9
5042
root     11162  9615  0 04:55 pts/0    00:00:00 grep /cellsrv
[root@qr01celadm01 ~]#
```

13. If the workload started in step 3 is still executing, stop it by typing `<Ctrl>-C` in your SQL*Plus session window. Wait until the workload stops.

```
...
SQL> select count(*) from sales where amount_sold > 16;

  COUNT(*)
----------
  24647749

Elapsed: 00:00:09.55
SQL> select count(*) from sales where amount_sold > 17;
^Cselect count(*) from sales where amount_sold > 17
                                   *
ERROR at line 1:
ORA-01013: user requested cancel of current operation


Elapsed: 00:00:01.30

SQL>
```

14. Exit all of your terminal sessions.

Practice 3-3: Storage Object Familiarization

Overview

In this practice, you are introduced to the hierarchy of Exadata storage objects.

Tasks

1. Establish a terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

2. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

3. In Exadata, a LUN (Logical Unit) is a logical abstraction of a storage device. LUNs are based on hard disks and flash devices. LUNs are automatically created when Exadata is initially configured. Each Exadata cell contains 12 hard disk-based LUNs along with 4 flash-based LUNs. List the LUNs on your primary Exadata cell.

```
CellCLI> list lun
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK00    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK01
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK01    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK02
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK02    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK03
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK03    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK04
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK04    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK05
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK05    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK06
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK06    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK07
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK07    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK08
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK08    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK10
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK10    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK11
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK11    normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH00
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH00   normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH01
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH01   normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH02
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH02   normal
         /opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH03
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/FLASH03   normal


CellCLI>
```

Note that the output from the virtualized Exadata cell shows LUNs with names and identifiers that are paths to virtualized disks and virtualized flash devices. On a real Exadata cell, the LUN names and identifiers are based on the PCI slot number and device number of the hard disk or flash device. For example, here is the expected output for the `LIST LUN` command on a real high capacity Exadata cell.

```
CellCLI> list lun
         0_0     0_0     normal
         0_1     0_1     normal
         0_2     0_2     normal
         0_3     0_3     normal
         0_4     0_4     normal
         0_5     0_5     normal
         0_6     0_6     normal
         0_7     0_7     normal
```

```
          0_8       0_8       normal
          0_9       0_9       normal
          0_10      0_10      normal
          0_11      0_11      normal
          1_1       1_1       normal
          2_1       2_1       normal
          4_1       4_1       normal
          5_1       5_1       normal

CellCLI>
```

4. Examine the detailed attribute listing for the LUN whose name ends with `DISK09`. Note the attribute setting `isSystemLun=FALSE`. This indicates that the LUN is not located on a system disk. Notice also that the LUN is associated with one physical disk and one cell disk.

```
CellCLI> list lun where name like '.*DISK09' detail
         name:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         cellDisk:                CD_09_qr01celadm01
         deviceName:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         diskType:                HardDisk
         id:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         isSystemLun:             FALSE
         lunSize:                 11
         physicalDrives:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         raidLevel:               "RAID 0"
         status:                  normal

CellCLI>
```

5. Exadata maintains the physical attributes of each hard disk in a `physicaldisk` object. A `physicaldisk` object is automatically created for each hard disk. Examine the attributes for the hard disk associated with LUN you examined in the previous step.

```
CellCLI> list physicaldisk where luns like '.*DISK09' detail
         name:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         deviceName:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         diskType:                HardDisk
         luns:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         physicalInsertTime:      2015-02-17T03:31:42+00:00
         physicalSerial:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         physicalSize:            11
         status:                  normal


CellCLI>
```

6. A cell disk is a higher level storage abstraction. Each cell disk is based on a LUN and contains additional attributes and metadata. Examine the attributes for the cell disk-based on the LUN you examined in step 4.

```
CellCLI> list celldisk CD_09_qr01celadm01 detail
         name:                   CD_09_qr01celadm01
         comment:
         creationTime:           2015-02-20T00:57:10+00:00
         deviceName:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         devicePartition:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         diskType:               HardDisk
         errorCount:             0
         freeSpace:              0
         id:                     f369c761-a9a1-4d1b-aaa0-51fc32edbc42
         interleaving:           none
         lun:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         physicalDisk:
/opt/oracle/cell12.1.2.1.0_LINUX.X64_141206.1/disks/raw/DISK09
         raidLevel:              "RAID 0"
         size:                   2G
         status:                 normal

CellCLI>
```

7. A grid disk defines an area of storage on a cell disk. Grid disks are consumed by ASM and are used as the storage for ASM disk groups. Each cell disk can contain a number of grid disks. Examine the grid disks associated with the cell disk you examined in the previous step. Note the names and sizes of the grid disks.

```
CellCLI> list griddisk where celldisk=CD_09_qr01celadm01 detail
         name:                   DATA_QR01_CD_09_qr01celadm01
         asmDiskGroupName:       DATA_QR01
         asmDiskName:            DATA_QR01_CD_09_QR01CELADM01
         asmFailGroupName:       QR01CELADM01
         availableTo:
         cachingPolicy:          default
         cellDisk:               CD_09_qr01celadm01
         comment:
         creationTime:           2015-03-11T03:23:37+00:00
         diskType:               HardDisk
         errorCount:             0
         id:                     9a11b79a-1fa7-4527-85bf-28ad5cb98cac
         offset:                 400M
         size:                   720M
         status:                 active

         name:                   DBFS_DG_CD_09_qr01celadm01
         asmDiskGroupName:       DBFS_DG
         asmDiskName:            DBFS_DG_CD_09_QR01CELADM01
         asmFailGroupName:       QR01CELADM01
```

```
         availableTo:
         cachingPolicy:          default
         cellDisk:               CD_09_qr01celadm01
         comment:
         creationTime:           2015-03-11T03:23:35+00:00
         diskType:               HardDisk
         errorCount:             0
         id:                     e304a244-75de-4ea6-b92d-b480e2226615
         offset:                 48M
         size:                   352M
         status:                 active

         name:                   RECO_QR01_CD_09_qr01celadm01
         asmDiskGroupName:       RECO_QR01
         asmDiskName:            RECO_QR01_CD_09_QR01CELADM01
         asmFailGroupName:       QR01CELADM01
         availableTo:
         cachingPolicy:          default
         cellDisk:               CD_09_qr01celadm01
         comment:
         creationTime:           2015-03-11T03:23:40+00:00
         diskType:               HardDisk
         errorCount:             0
         id:                     9923a077-dc5e-4f53-adc5-4cc3eaaa9916
         offset:                 1.09375G
         size:                   928M
         status:                 active

CellCLI>
```

8. Establish a terminal connection to `qr01dbadm01` as the `grid` user and configure the session to access ASM using the `+ASM1` instance.

```
$ ssh grid@qr01dbadm01
grid@qr01dbadm01's password: ????????
[grid@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@qr01dbadm01 ~]$
```

9. Using SQL*Plus, connect to ASM as `sysasm`.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

10. Locate the grid disks from step 7 inside ASM (use the SQL script `/home/oracle/labs/lab03-03-10.sql` if you prefer). Check that the sizes reported by ASM match the grid disk attributes reported in step 7. Note the capitalization of the value in the `like` string.

```
SQL> select name, path, state, total_mb from v$asm_disk
2>   where name like '%_CD_09_QR01CELADM01';

NAME
------------------------------
PATH
--------------------------------------------------------------
STATE      TOTAL_MB
-------- ----------
DBFS_DG_CD_09_QR01CELADM01
o/192.168.1.105;192.168.1.106/DBFS_DG_CD_09_qr01celadm01
NORMAL          352

RECO_QR01_CD_09_QR01CELADM01
o/192.168.1.105;192.168.1.106/RECO_QR01_CD_09_qr01celadm01
NORMAL          928

DATA_QR01_CD_09_QR01CELADM01
o/192.168.1.105;192.168.1.106/DATA_QR01_CD_09_qr01celadm01
NORMAL          720


SQL>
```

11. Determine which ASM disk group the grid disks from step 7 are assigned to (use the SQL script `/home/oracle/labs/lab03-03-11.sql` if you prefer). Note the capitalization of the value in the `like` string.

```
SQL> select d.name disk, dg.name diskgroup
2>   from v$asm_disk d, v$asm_diskgroup dg
3>   where dg.group_number = d.group_number
4>   and d.name like '%_CD_09_QR01CELADM01';

DISK                           DISKGROUP
------------------------------ ------------------------------
DBFS_DG_CD_09_QR01CELADM01     DBFS_DG
RECO_QR01_CD_09_QR01CELADM01   RECO_QR01
DATA_QR01_CD_09_QR01CELADM01   DATA_QR01

SQL>
```

12. Exit all your SQL*Plus and CellCLI sessions.

Practice 3-4: Exadata Smart Flash Cache Familiarization

Overview

In this practice, you are introduced to Exadata Smart Flash Cache.

Tasks

1. Establish a terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

2. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~] $ cellcli
CellCLI: Release 12.1.2.1.0 – Production...

CellCLI>
```

3. List the cell disks associated with the flash modules in your Exadata cell. By default, there should be 4 cell disks having names that start with `FD`.

```
CellCLI> list celldisk where disktype=flashdisk
         FD_00_qr01celadm01      normal
         FD_01_qr01celadm01      normal
         FD_02_qr01celadm01      normal
         FD_03_qr01celadm01      normal

CellCLI>
```

4. By default, Exadata Smart Flash Cache is configured across all the flash-based cell disks. Use the `LIST FLASHCACHE DETAIL` command to confirm that Exadata Smart Flash Cache is configured on your flash-based cell disks. Note that the size of the Exadata Smart Flash Cache on your laboratory cells is much smaller than what you would observe on a real cell; however, all of the other attributes would be similar on a real cell.

```
CellCLI> list flashcache detail
         name:                   qr01celadm01_FLASHCACHE
         cellDisk:
FD_03_qr01celadm01,FD_02_qr01celadm01,FD_01_qr01celadm01,FD_00_qr01celadm01
         creationTime:           2016-06-02T02:54:32+00:00
         degradedCelldisks:
         effectiveCacheSize:     1.0625G
         id:                     3a4cbd94-2ebb-4fc2-8c95-1221cc26d8a5
         size:                   1.0625G
         status:                 normal

CellCLI>
```

5. In addition to Exadata Smart Flash Cache, Exadata Smart Flash Log provides a mechanism for improving the latency of database redo log write operations. Exadata Smart Flash Log uses a small portion of high-performance flash memory as temporary storage to facilitate low latency redo log writes. By default, Exadata Smart Flash Log uses a total of 512 MB on each Exadata Storage Server; however, on your laboratory cells Exadata Smart Flash Log is configured to consume 256 MB on each cell. Use the `LIST FLASHLOG DETAIL` command to examine the Exadata Smart Flash Log area on this cell.

```
CellCLI> list flashlog detail
         name:                   qr01celadm01_FLASHLOG
         cellDisk:
FD_01_qr01celadm01,FD_02_qr01celadm01,FD_00_qr01celadm01,FD_03_qr01celadm01
         creationTime:           2016-06-02T02:54:30+00:00
         degradedCelldisks:
         effectiveSize:          256M
         efficiency:             100.0
         id:                     11b847a1-3e09-4185-91bf-887976154371
         size:                   256M
         status:                 normal

CellCLI>
```

6. Use the `LIST FLASHCACHECONTENT DETAIL` command to show information about the data inside Exadata Smart Flash Cache. You can see that each entry contains a series of attributes relating to a database object in the cache. For each object, you can see how much data is being cached along with the number of cache hits and misses. This information can help you to assess cache efficiency for specific database objects.

```
CellCLI> list flashcachecontent detail
...
         cachedKeepSize:         0
         cachedSize:             262144
         cachedWriteSize:        0
         columnarCacheSize:      0
         columnarKeepSize:       0
         dbID:                   2080757153
         dbUniqueName:           DBM
         hitCount:               11345
         missCount:              9
         objectNumber:           4294967294
         tableSpaceNumber:       0

         cachedKeepSize:         0
         cachedSize:             65536
         cachedWriteSize:        0
         columnarCacheSize:      0
         columnarKeepSize:       0
         dbID:                   2080757153
         dbUniqueName:           DBM
```

```
          hitCount:                0
          missCount:               0
          objectNumber:            4294967295
          tableSpaceNumber:        4

          cachedKeepSize:          0
          cachedSize:              81920
          cachedWriteSize:         0
          columnarCacheSize:       0
          columnarKeepSize:        0
          dbID:                    2080757153
          dbUniqueName:            DBM
          hitCount:                0
          missCount:               0
          objectNumber:            4294967295
          tableSpaceNumber:        2

CellCLI>
```

7. Exit your CellCLI session.

Practices for Lesson 4: Key Capabilities of Exadata Database Machine

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In these practices, you are introduced to four major capabilities of Exadata, namely:

- Smart Scan
- Exadata Hybrid Columnar Compression
- Exadata Smart Flash Cache
- Storage Index

Practice 4-1: Smart Scan

Overview

In this practice, you are introduced to the Smart Scan capability of Exadata. You will execute a query with and without Smart Scan enabled and you will examine statistics to measure the effect of Smart Scan.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user and configure the session to enable command timing.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 ...

SQL> set timing on
SQL>
```

3. Execute the following query (or execute the SQL script `/home/oracle/labs/lab04-01-03.sql`) and verify that the statistics are at or near zero values. If any statistics are significantly greater than zero then reconnect (using `connect sh`) and try again.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                           .0546875
physical write total bytes                                                 0
cell physical IO interconnect bytes                                 .0546875
cell physical IO bytes saved during optimized file creation                0
cell physical IO bytes saved during optimized RMAN file restore            0
cell physical IO bytes eligible for predicate offload                      0
cell physical IO bytes saved by storage index                              0
cell physical IO bytes sent directly to DB node to balance CPU             0
cell physical IO interconnect bytes returned by smart scan                 0
cell physical IO bytes saved by columnar cache                             0
cell physical write bytes saved by smart file initialization               0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                 0
cell physical write IO bytes eligible for offload                          0
cell physical write IO host network bytes written during offloa            0

14 rows selected.

Elapsed: 00:00:00.04
SQL>
```

4. Execute the following query (or execute the SQL script `/home/oracle/labs/lab04-01-04.sql`). Note the optimizer hint that disables Smart Scan for the query.

```
SQL> select /*+ OPT_PARAM('cell_offload_processing' 'false') */
  2  count(*) from sales
  3  where amount_sold = 1;

  COUNT(*)
----------
    301261

Elapsed: 00:00:58.45
SQL>
```

5. Repeat the statistics query from step 3 (or execute the SQL script `/home/oracle/labs/lab04-01-03.sql`). Note that all of the data processed by the query in step 4 (`physical read total bytes`) is returned to the database server over the storage network (`cell physical IO interconnect bytes`).

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        1800.70313
physical write total bytes                                                0
cell physical IO interconnect bytes                              1800.70313
cell physical IO bytes saved during optimized file creation              0
cell physical IO bytes saved during optimized RMAN file restore          0
cell physical IO bytes eligible for predicate offload                    0
cell physical IO bytes saved by storage index                            0
cell physical IO bytes sent directly to DB node to balance CPU           0
cell physical IO interconnect bytes returned by smart scan               0
cell physical IO bytes saved by columnar cache                           0
cell physical write bytes saved by smart file initialization             0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                0
cell physical write IO bytes eligible for offload                        0
cell physical write IO host network bytes written during offloa          0

14 rows selected.

Elapsed: 00:00:00.03
SQL>
```

6. Reconnect to your database in order to reset the session level statistics.

```
SQL> connect sh/??
Connected.
SQL>
```

7. Repeat the statistics query from step 3 (or execute the SQL script `/home/oracle/labs/lab04-01-03.sql`) and verify that the statistics are again at or near zero values. If any statistics are significantly greater than zero then reconnect again and retry.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                                 0
physical write total bytes                                                0
cell physical IO interconnect bytes                                       0
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.04
SQL>
```

8. Execute the following query (or execute the SQL script `/home/oracle/labs/lab04-01-08.sql`). This is the same query as in step 4; however, this time there is no optimizer hint to disable Smart Scan.

```
SQL> select count(*) from sales where amount_sold = 1;

  COUNT(*)
----------
    301261

Elapsed: 00:00:10.33
SQL>
```

9. Repeat the statistics query from step 3 (or execute the SQL script `/home/oracle/labs/lab04-01-03.sql`). Note that the query still performs approximately 1800 MB of I/O (`physical read total bytes`). However, this time only about 4.2 MB is actually returned to the database server (`cell physical IO interconnect bytes`). This is Smart Scan in action.

   Also note that in this case, Smart Scan is acting on all of the I/O associated with the query. This is the case because `cell physical IO bytes eligible for predicate offload` equals `physical read total bytes`, and `cell physical IO interconnect bytes returned by smart scan` equals `cell physical IO interconnect bytes`.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        1800.70313
physical write total bytes                                                0
cell physical IO interconnect bytes                              4.19950867
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            1800.70313
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       4.19950867
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       1800.70313
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.02
SQL>
```

10. Exit your SQL*Plus session.

Practice 4-2: Exadata Hybrid Columnar Compression

Overview

In this practice, you are introduced to Exadata Hybrid Columnar Compression. You will create compressed copies of an existing database table and examine the level of compression you achieve.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user and configure the session to enable command timing.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 ...

SQL> set timing on
SQL>
```

3. Determine the size of the uncompressed `MYCUSTOMERS` table (use the SQL script `/home/oracle/labs/lab04-02-03.sql` if you prefer).

```
SQL> col segment_name format a30
SQL> select segment_name, sum(bytes)/1024/1024 MB
  2  from user_segments
  3  where segment_name like 'MYCUST%'
  4  group by segment_name;

SEGMENT_NAME                           MB
------------------------------ ----------
MYCUSTOMERS                           654

Elapsed: 00:00:00.29
SQL>
```

4. Verify that the `CUSTOMERS` table is uncompressed (use the SQL script `/home/oracle/labs/lab04-02-04.sql` if you prefer).

```
SQL> col table_name format a30
SQL> select table_name, compression, compress_for
  2  from user_tables
  3  where table_name like 'MYCUST%';

TABLE_NAME                     COMPRESS COMPRESS_FOR
------------------------------ -------- ----------------------
MYCUSTOMERS                    DISABLED

Elapsed: 00:00:00.01
SQL>
```

5. Exadata Hybrid Columnar Compression achieves its highest levels of compression with data that is direct-path inserted. Execute the following `ALTER SESSION` commands to ensure the use of direct-path inserts later in the practice.

```
SQL> alter session force parallel query;

Session altered.

Elapsed: 00:00:00.01
SQL> alter session force parallel ddl;

Session altered.

Elapsed: 00:00:00.00
SQL> alter session force parallel dml;

Session altered.

Elapsed: 00:00:00.00
SQL>
```

6. Create a compressed copy of the MYCUSTOMERS table by using the QUERY HIGH warehouse compression mode (use the SQL script /home/oracle/labs/lab04-02-06.sql if you prefer).

```
SQL> create table mycust_query
  2  column store compress for query high
  3  parallel 4 nologging as select * from mycustomers;

Table created.

Elapsed: 00:00:38.68
SQL>
```

7. Create a compressed copy of the MYCUSTOMERS table using the ARCHIVE HIGH archive compression mode (use the SQL script /home/oracle/labs/lab04-02-07.sql if you prefer). Note that it may take a few minutes for the table to be created.

```
SQL> create table mycust_archive
  2  column store compress for archive high
  2  parallel 4 nologging as select * from mycustomers;

Table created.

Elapsed: 00:01:21.48
SQL>
```

8. Verify the compression mode settings for the tables you just created (use the SQL script /home/oracle/labs/lab04-02-04.sql again if you prefer).

```
SQL> select table_name, compression, compress_for
  2  from user_tables
  3  where table_name like 'MYCUST%';

TABLE_NAME                     COMPRESS COMPRESS_FOR
------------------------------ -------- ----------------------
MYCUSTOMERS                    DISABLED
MYCUST_ARCHIVE                 ENABLED  ARCHIVE HIGH
MYCUST_QUERY                   ENABLED  QUERY HIGH

Elapsed: 00:00:00.05
SQL>
```

9. Compare the size of the original uncompressed table with the two compressed copies you created (use the SQL script /home/oracle/labs/lab04-02-03.sql if you prefer).

```
SQL> select segment_name,sum(bytes)/1024/1024 MB
  2  from user_segments
  3  where segment_name like 'MYCUST%'
  4  group by segment_name;

SEGMENT_NAME                           MB
------------------------------ ----------
MYCUSTOMERS                           654
MYCUST_ARCHIVE                        113
MYCUST_QUERY                          159

Elapsed: 00:00:00.47
SQL>
```

10. Drop the compressed tables that you created in this practice.

```
SQL> drop table mycust_query;

Table dropped.

Elapsed: 00:00:00.46
SQL> drop table mycust_archive;

Table dropped.

Elapsed: 00:00:00.08
SQL>
```

11. Exit your SQL*Plus session.

Practice 4-3: Exadata Smart Flash Cache

Overview

In this practice, you will examine the use of Exadata Smart Flash Cache. You will execute a series of record lookups and use database statistics to verify the use of Exadata Smart Flash Cache. You will also compare the execution statistics with and without the use of Exadata Smart Flash Cache.

Tasks

1. Establish a terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

2. Execute the following two commands to drop and then re-create Exadata Smart Flash Cache on all of your Exadata cells. You must perform this action so that Exadata Smart Flash Cache is empty at the beginning of this practice; thus ensuring consistent results later in the practice. To do this, you will use the distributed command line utility (`dcli`) that is provided with Exadata. Using `dcli` you can execute cell-level administrative commands simultaneously on multiple Exadata cells. A more detailed discussion of `dcli` features and options is provided later in the course. Be careful not to add any extra spaces in the server list following the `dcli -c` command line option.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e drop flashcache
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE successfully dropped
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE successfully dropped
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE successfully dropped
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e create flashcache all
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE successfully created
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE successfully created
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE successfully created
[celladmin@qr01celadm01 ~]$
```

3. Establish a separate terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

4. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

5. Execute the following query (or execute the SQL script `/home/oracle/labs/lab04-03-05.sql`) and verify that the statistics are at or near zero values. If any statistics are significantly greater than zero then reconnect and retry.

```
SQL> select a.name, b.value from v$sysstat a, v$mystat b
  2  where a.statistic# = b.statistic# and
  3  (a.name like '%flash cache read hits'
  4  or a.name like 'cell phy%'
  5  or a.name like 'physical read tot%'
  6  or a.name like 'physical read req%');

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
physical read total IO requests                                           0
physical read total multi block requests                                  0
physical read requests optimized                                          0
physical read total bytes optimized                                       0
physical read total bytes                                                 0
cell physical IO interconnect bytes                                       0
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0
cell flash cache read hits                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

17 rows selected.

SQL>
```

6. Flush the buffer cache to ensure that the queries in step 8 must retrieve the required data from the Exadata cells.

```
SQL> alter system flush buffer_cache;

System altered.

SQL>
```

7. Configure the session to display server output.

```
SQL> set serveroutput on
SQL>
```

8. The following PL/SQL block performs 500 record lookups spread across a reasonably large table. The workload is representative of the scattered record access normally associated with an OLTP application. Execute the PL/SQL block against your database (or execute the SQL script `/home/oracle/labs/lab04-03-08.sql`). Note that the workload may take a few minutes to complete.

```
SQL> declare
       a number;
       s number := 0;
     begin
       for n in 1 .. 500 loop
         select cust_credit_limit into a from customers
           where cust_id=n*2000;
         s := s+a;
       end loop;
       dbms_output.put_line('Transaction total = '||s);
     end;
     /
Transaction total = 3620500

PL/SQL procedure successfully completed.

SQL>
```

9. Repeat the statistics query from step 5 (or execute the SQL script `/home/oracle/labs/lab04-03-05.sql`). Note the high number of IO requests (`physical read total IO requests`) relative to the low number of optimized requests (`physical read requests optimized` and `cell flash cache read hits`). This indicates that the queries were mostly satisfied by using physical disk reads and is indicative of a recently emptied cache.

```
SQL> select a.name, b.value from v$sysstat a, v$mystat b
  2  where a.statistic# = b.statistic# and
  3  (a.name like '%flash cache read hits'
  4  or a.name like 'cell phy%'
  5  or a.name like 'physical read tot%'
  6  or a.name like 'physical read req%');

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
physical read total IO requests                                         943
physical read total multi block requests                                 65
physical read requests optimized                                        162
physical read total bytes optimized                                 1343488
physical read total bytes                                          22732800
cell physical IO interconnect bytes                                22732800
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0
cell flash cache read hits                                              162
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

17 rows selected.

SQL>
```

10. Reconnect to your database in order to reset the session level statistics.

```
SQL> connect sh/??
Connected.
SQL>
```

11. Repeat the statistics query from step 5 (or execute the SQL script `/home/oracle/labs/lab04-03-05.sql`) and verify that the statistics are again at or near zero values. If any statistics are significantly greater than zero then reconnect again and retry.

```
SQL> select a.name, b.value from v$sysstat a, v$mystat b
  2  where a.statistic# = b.statistic# and
  3  (a.name like '%flash cache read hits'
  4  or a.name like 'cell phy%'
  5  or a.name like 'physical read tot%'
  6  or a.name like 'physical read req%');

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
physical read total IO requests                                           0
physical read total multi block requests                                  0
physical read requests optimized                                          0
physical read total bytes optimized                                       0
physical read total bytes                                                 0
cell physical IO interconnect bytes                                       0
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0
cell flash cache read hits                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

17 rows selected.

SQL>
```

12. Again, flush the buffer cache to ensure that the queries in the next step must retrieve the required data from the Exadata cells. Also, configure your new session to display server output.

```
SQL> alter system flush buffer_cache;

System altered.

SQL> set serveroutput on
SQL>
```

13. Re-execute the PL/SQL block introduced in step 8 (or execute the SQL script `/home/oracle/labs/lab04-03-08.sql`). Confirm that the result is the same.

```
SQL> declare
       a number;
       s number := 0;
     begin
       for n in 1 .. 500 loop
         select cust_credit_limit into a from customers
           where cust_id=n*2000;
         s := s+a;
       end loop;
       dbms_output.put_line('Transaction total = '||s);
     end;
     /
Transaction total = 3620500

PL/SQL procedure successfully completed.

SQL>
```

14. Repeat the statistics query from step 5 (or execute the SQL script `/home/oracle/labs/lab04-03-05.sql`). Compare the values for `cell flash cache read hits` and `physical read total IO requests`. They should be much closer together, indicating that most of the I/Os were satisfied by Exadata Smart Flash Cache.

```
SQL> select a.name, b.value from v$sysstat a, v$mystat b
  2  where a.statistic# = b.statistic# and
  3  (a.name like '%flash cache read hits'
  4  or a.name like 'cell phy%'
  5  or a.name like 'physical read tot%'
  6  or a.name like 'physical read req%');

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
physical read total IO requests                                        1011
physical read total multi block requests                                 20
physical read requests optimized                                        983
physical read total bytes optimized                                15548416
physical read total bytes                                          19914752
cell physical IO interconnect bytes                                19914752
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0

NAME                                                                  VALUE
---------------------------------------------------------------- ----------
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0
cell flash cache read hits                                              983
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

17 rows selected.

SQL>
```

In an earlier practice, you saw how to obtain general information about Exadata Smart Flash Cache on an Exadata cell using the `LIST FLASHCACHECONTENT` CellCLI comannd. Over the remainder of this practice, you will learn how to isolate specific information in Exadata Smart Flash Cache.

15. Use the following query (or execute the SQL script `/home/oracle/labs/lab04-03-15.sql`) to determine the number of optimized physical reads (reads optimized by Exadata Smart Flash Cache or Exadata storage index) for the `SH.CUSTOMERS` table. Note the tablespace number (`TS#`) and object number (`DATAOBJ#`) associated with the table.

```
SQL> select owner, object_name, tablespace_name, ts#, dataobj#,
  2  statistic_name, value
  3  from v$segment_statistics
  4  where owner='SH' and object_name='CUSTOMERS'
  5  and statistic_name='optimized physical reads';

OWNER
--------------------------------------------------------------------------------
--
OBJECT_NAME
--------------------------------------------------------------------------------
--
TABLESPACE_NAME                       TS#   DATAOBJ#
------------------------------ ----------------------
STATISTIC_NAME                                                       VALUE
---------------------------------------------------------------- ----------
SH
CUSTOMERS
SH                                      7      20473
optimized physical reads                                               477


SQL>
```

16. Back in the terminal session connected to `qr01celadm01`, launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

17. Use the tablespace number (`TS#`) and object number (`DATAOBJ#`) you gathered in step 15 to query the Exadata Smart Flash Cache. The output relates specifically to the `SH.CUSTOMERS` table.

```
CellCLI> list flashcachecontent where objectnumber=20473 and
tablespacenumber=7 and dbuniquename=DBM detail
         cachedKeepSize:          0
         cachedSize:              13189120
         cachedWriteSize:         0
         columnarCacheSize:       0
         columnarKeepSize:        0
         dbID:                    2080757153
         dbUniqueName:            DBM
         hitCount:                160
         missCount:               159
         objectNumber:            20473
         tableSpaceNumber:        7

CellCLI>
```

Note that in step 15 the value for `optimized physical reads` is 477 while the `hitCount` observed in this step is 160. Why is this so? In the remaining time allocated for this practice, query the Exadata Smart Flash Cache `hitCount` values for the other cells (`qr01celadm02` and `qr01celadm03`) and compare the `hitCount` total across all the cells with the `optimized physical reads` value observed in step 15. Can you explain your observations?

18. Exit your terminal sessions.

Practice 4-4: Storage Index

Overview

In this practice, you are introduced to the storage index capability of Exadata. You will execute a query multiple times and examine statistics to measure the effect of storage index.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Execute the following query (or execute the SQL script `/home/oracle/labs/lab04-04-03.sql`) and verify that the statistics are at or near zero values. If any statistics are significantly greater than zero then reconnect and retry.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                            .09375
physical write total bytes                                                0
cell physical IO interconnect bytes                                  .09375
cell physical IO bytes saved during optimized file creation              0
cell physical IO bytes saved during optimized RMAN file restore          0
cell physical IO bytes eligible for predicate offload                    0
cell physical IO bytes saved by storage index                            0
cell physical IO bytes sent directly to DB node to balance CPU           0
cell physical IO interconnect bytes returned by smart scan               0
cell physical IO bytes saved by columnar cache                           0
cell physical write bytes saved by smart file initialization             0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

SQL>
```

4. Execute the following query (or execute the SQL script `/home/oracle/labs/lab04-04-04.sql`).

```
SQL> select cust_gender,count(*) from mycustomers
  2  where cust_income_level = 'C: 50000 - 69999'
  3  group by cust_gender;

C   COUNT(*)
- ----------
M     135817
F     136226

SQL>
```

5. Repeat the statistics query from step 3 (or execute the SQL script `/home/oracle/labs/lab04-04-03.sql`).

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        650.296875
physical write total bytes                                                0
cell physical IO interconnect bytes                                3.811203
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload              650.1875
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan         3.701828
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                         650.1875
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

SQL>
```

The statistics show that the query in step 4 was conducted using Smart Scan. Note, however, that `cell physical IO bytes saved by storage index` is zero. This is because storage indexes are memory structures which do not persist when the Exadata cells are restarted. They are dynamically built when tables are referenced for the first time after the cells restart. Now that the `mycustomers` table has been scanned as a result of the query in step 4, all subsequent queries on the `mycustomers` table can benefit from whatever storage indexes the Exadata cells automatically create.

6. Reconnect to your database in order to reset the session level statistics.

```
SQL> connect sh/??
Connected.
SQL>
```

7. Repeat the statistics query from step 3 (or execute the SQL script `/home/oracle/labs/lab04-04-03.sql`) and verify that the statistics are again at or near zero values. If any statistics are significantly greater than zero then reconnect again and retry.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                                 0
physical write total bytes                                                0
cell physical IO interconnect bytes                                       0
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa          0

14 rows selected.

SQL>
```

8. Re-execute the query from step 4 (or execute the SQL script `/home/oracle/labs/lab04-04-04.sql`).

```
SQL> select cust_gender,count(*) from mycustomers
  2  where cust_income_level = 'C: 50000 - 69999'
  3  group by cust_gender;

C   COUNT(*)
- ----------
M     135817
F     136226

SQL>
```

9. Repeat the statistics query from step 3 (or execute the SQL script `/home/oracle/labs/lab04-04-03.sql`).

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                 MB
---------------------------------------------------------------- ----------
physical read total bytes                                          650.1875
physical write total bytes                                                0
cell physical IO interconnect bytes                              3.59671783
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload              650.1875
cell physical IO bytes saved by storage index                    595.398438
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       3.59671783
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                 MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       54.7890625
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa          0

14 rows selected.

SQL>
```

This time you will see that the query in step 8 benefits from the storage index. Instead of conducting more than 650 MB of I/O inside the cells, storage indexes were used to bypass more than 595 MB of I/O. In other words, approximately 55 MB of I/O was conducted instead of 650 MB. Queries that benefit from storage indexes can execute more quickly using fewer resources which allows other workloads to benefit from the unused I/O resources.

10. Exit your SQL*Plus session.

Practices for Lesson 5: Exadata Database Machine Initial Configuration

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In this practice, you will be introduced to the Oracle Exadata Deployment Assistant configuration tool.

Practice 5-1: Using the Exadata Configuration Tool

Overview

In this practice, you will be introduced to the Oracle Exadata Deployment Assistant configuration tool (Exadata configuration tool). You will use the configuration tool to generate a set of configuration files for example Database Machine implementation scenarios.

Tasks

1. Establish a terminal session connected to `qr01dbadm01` using the `oracle` OS user. Ensure that you specify the `-X` option for `ssh`.

```
$ ssh -X oracle@qr01dbadm01
oracle@qr01dbadm01 password: ????????
[oracle@qr01dbadm01 ~]$
```

2. The latest Oracle Exadata Deployment Assistant is provided as a patch on My Oracle Support. In your laboratory environment, a version of the Oracle Exadata Deployment Assistant is located under `/home/oracle/labs/oeda/linux-x64` on `qr01dbadm01`. Change directory to the directory containing the Oracle Exadata Deployment Assistant.

```
[oracle@qr01dbadm01 ~]$ cd labs/oeda/linux-x64
[oracle@qr01dbadm01 linux-x64]$
```

3. Start the Exadata configuration tool.

```
[oracle@qr01dbadm01 linux-x64]$ ./config.sh
```

4. Complete a configuration session for an X6-2 Full Rack Database Machine by using the information in the following table. Leave the default values for fields that are not specified in the following table. Examine the options and additional information presented on each page. When you are satisfied with each page, click “Next >” to proceed unless otherwise directed.

Step	Window/Page Description	Configuration Settings and Other Instructions
a.	Welcome	Leave the default selection (Oracle Exadata Database Machine).
b.	Customer Details	Customer Name: `Example Industries` Application: `Example Full Rack` Network Domain Name: `example.com` DNS: `10.7.7.5` NTP: `10.7.7.5`
c.	Hardware Selection	This is your deployment: `X6-2 Full Rack HC 8TB`
d.	Rack Details	Leave the default values.
e.	Define Customer Networks	Subnet 1 (Admin) Gateway: `10.7.7.1` Subnet 2 (Client) Gateway: `172.16.1.1`
f.	Administration Network	Starting IP Address for Pool: `10.7.7.101`
g.	Client Ethernet Network	Starting IP Address for Pool: `172.16.1.101`

Step	Window/Page Description	Configuration Settings and Other Instructions
h.	InfiniBand Network	Leave the default values.
i.	Backup Network	Leave unconfigured.
j.	Identify Compute Node OS	Leave the default configuration.
k.	Capacity-on-Demand	Click Confirmed.
l.	Review and Edit	Review and proceed.
m.	Define Clusters	Cluster Name: `dm01` Click “All >>” and then proceed.
n.	Cluster 1	Check the `Role Separated` option in the Users and Groups section. Grid Infrastructure Home: `12.1.0.2 DBBP160419` Database Home Location: `12.1.0.2 DBBP160419`
o.	Review and Edit	Review and proceed.
p.	Alerting	Leave unconfigured.
q.	Platinum Configuration	Uncheck the `Capture data for Platinum configuration` option.
r.	Oracle Config. Manager	Leave unconfigured.
s.	Auto Service Request	Leave unconfigured.
t.	Grid Control Agent	Leave unconfigured.
u.	Comments	Review and proceed.
v.	Generate	Click “Next >” as instructed.
w.	Select Destination Folder	Leave the default location and click Save.
x.	Finish	Note the names and locations of the generated files. Click “Installation template” and examine the configuration details in the resulting web browser page. When you are finished, close the web browser and click Finish to end the configuration session.

5. Back in your terminal session, examine the contents of the ExadataConfigurations directory, which contains your newly generated configuration files.

```
[oracle@qr01dbadm01 linux-x64]$ ls -hs ExadataConfigurations
total 536K
 12K databasemachine.xml
4.0K Example_Industries-dm01-checkip.sh
232K Example_Industries-dm01-InstallationTemplate.html
4.0K Example_Industries-dm01-platinum.csv
8.0K Example_Industries-dm01-preconf_rack_0.csv
244K Example_Industries-dm01.xml
 32K Example_Industries-dm01.zip
[oracle@qr01dbadm01 linux-x64]$
```

6. Examine the `Example_Industries-dm01.xml` file (using the `more` command or `vi` if you prefer). This is one of the main configuration files that drive the Database Machine deployment process. Examine the file and confirm that the details within it match your inputs to the configuration tool. Take a few minutes to examine the other configuration files as well.

   Congratulations, you have now used the Exadata configuration tool to generate a set of configuration files for a Database Machine deployment scenario. In the next part of this practice you will use the configuration tool to import your configuration information and make some changes.

7. Restart the Exadata configuration tool.

```
[oracle@qr01dbadm01 linux-x64]$ ./config.sh
```

8. In your previous configuration session, you created configuration files for an X6-2 Full Rack Database Machine with many default settings. In this configuration session, you will import your previous configuration and make a series of changes by using the information in the following table. If you wish, feel free to experiment with the different configuration options. When you are satisfied with each page, click "Next >" to proceed unless otherwise directed.

Step	Window/Page Description	Configuration Settings and Other Instructions
a.	Welcome	Click Import... Select `/home/oracle/labs/oeda/linux-x64/ExadataConfigurations/Example_Industries-dm01.xml` and click Open. Click OK to acknowledge the resulting dialog box.
b.	Customer Details	Review and proceed.
c.	Hardware Selection	Review and proceed.
d.	Rack Details	Review and proceed.

<table>
<tr><td>e.</td><td>Define Customer Networks</td><td>Click Advanced... and examine the advanced options.
Select the <code>Enable Network VLAN</code> option and click OK.
Subnet 1 (Admin) VLAN ID: Specify a unique numeric value between <code>1</code> and <code>4079</code>
Subnet 2 (Client) VLAN ID: Specify a unique numeric value between <code>1</code> and <code>4095</code></td></tr>
<tr><td>f.</td><td>Administration Network</td><td>Click Modify... and examine the resulting dialog.
Alter the configuration to use a non-default naming convention by altering any of the name format strings and click Save.</td></tr>
<tr><td>g.</td><td>Client Ethernet Network</td><td>Click Modify... and examine the resulting dialog.
Alter the configuration to use a non-default naming convention by altering any of the name format strings and click Save.</td></tr>
<tr><td>h.</td><td>InfiniBand Network</td><td>Click Modify... and examine the resulting dialog.
Alter the configuration to use a non-default naming convention by altering any of the name format strings and click Save.</td></tr>
<tr><td>i.</td><td>Backup Network</td><td>Leave unconfigured.</td></tr>
<tr><td>j.</td><td>Identify Compute Node OS</td><td>Click the <code>Enable Capacity-on-Demand</code> option and use the slider to set the number of enabled CPU cores on each database server.</td></tr>
<tr><td>k.</td><td>Review and Edit</td><td>Review and proceed.</td></tr>
<tr><td>l.</td><td>Define Clusters</td><td>Review and proceed.</td></tr>
<tr><td>m.</td><td>Cluster 1</td><td>Customize the storage configuration in the Disk Group Details section as follows:
<table>
<tr><th>Disk Group</th><th>Name</th><th>Redundancy</th><th>Size</th></tr>
<tr><td>DATA</td><td><code>DATAC1</code></td><td><code>HIGH</code></td><td><code>90%</code></td></tr>
<tr><td>RECO</td><td><code>RECOC1</code></td><td><code>HIGH</code></td><td><code>10%</code></td></tr>
</table></td></tr>
<tr><td>n.</td><td>Review and Edit</td><td>Review and proceed.</td></tr>
<tr><td>o.</td><td>Alerting</td><td>Leave unconfigured.</td></tr>
<tr><td>p.</td><td>Platinum Configuration</td><td>Leave unconfigured.</td></tr>
<tr><td>q.</td><td>Oracle Config. Manager</td><td>Leave unconfigured.</td></tr>
<tr><td>r.</td><td>Auto Service Request</td><td>Leave unconfigured.</td></tr>
<tr><td>s.</td><td>Grid Control Agent</td><td>Leave unconfigured.</td></tr>
<tr><td>t.</td><td>Comments</td><td>Review and proceed.</td></tr>
<tr><td>u.</td><td>Generate</td><td>Click “Next >” as instructed.</td></tr>
<tr><td>v.</td><td>Select Destination Folder</td><td>Enter <code>ExadataConfigurations2</code> as the destination folder (using the File Name field) and click Save.</td></tr>
</table>

w.	Finish	Note the names and locations of the generated files. Click “Installation template” and examine the configuration details in the resulting web browser page. When you are finished, close the web browser and click Finish to end the configuration session.

9. Back in your terminal session, compare your configuration files and verify the changes that you made in the second configuration session. For example:

```
[oracle@qr01dbadm01 linux-x64]$ xml_grep diskGroupSize
ExadataConfigurations*/Example_Industries-dm01.xml
<?xml version="1.0" ?>
<xml_grep version="0.7" date="Fri Jun  3 06:02:52 2016">
<file filename="ExadataConfigurations2/Example_Industries-
dm01.xml">
  <diskGroupSize>2037G</diskGroupSize>
  <diskGroupSize>367416G</diskGroupSize>
  <diskGroupSize>40824G</diskGroupSize>
</file>
<file filename="ExadataConfigurations/Example_Industries-
dm01.xml">
  <diskGroupSize>2037G</diskGroupSize>
  <diskGroupSize>349888G</diskGroupSize>
  <diskGroupSize>87444G</diskGroupSize>
</file>
</xml_grep>
[oracle@qr01dbadm01 linux-x64]$
```

10. Restart the Exadata configuration tool and create a configuration for an elastic configuration with virtualized database servers by using the information in the following table. Feel free again to experiment with the different configuration options if you wish. When you are satisfied with each page, click “Next >” to proceed unless otherwise directed.

Step	Window/Page Description	Configuration Settings and Other Instructions
a.	Welcome	Leave the default selection (Oracle Exadata Database Machine).
b.	Customer Details	Customer Name: `Example Industries` Application: `Example with Virtualization` Network Domain Name: `example.com` Name Prefix: `vm01` DNS: `10.7.7.5` NTP: `10.7.7.5`
c.	Hardware Selection	This is your deployment: `X6-2 Elastic Rack HC 8TB`
d.	Rack Details	Compute Node count: `4` Storage Cell count: `6`

Step	Window/Page Description	Configuration Settings and Other Instructions
e.	Define Customer Networks	Subnet 1 (Admin) Gateway: `10.7.7.1` Subnet 2 (Client) Gateway: `172.16.1.1`
f.	Administration Network	Starting IP Address for Pool: `10.7.7.11`
g.	Client Ethernet Network	Starting IP Address for Pool: `172.16.1.11`
h.	InfiniBand Network	Leave the default values.
i.	Backup Network	Leave unconfigured.
j.	Identify Compute Node OS	Click "All OVM"
k.	Capacity-on-Demand	Click Confirmed.
l.	Review and Edit	Review and proceed.
m.	Define Clusters	Number of Clusters to create: `2` Cluster 1 tab: Cluster Name: `c1` Click "All >>" Cluster 2 tab: Cluster Name: `c2` Click "All >>"
n.	Cluster 1	Examine the options for `Virtual Guest Size`. Also click `Oracle VM Defaults...` and examine the available options. Otherwise, create your own custom configuration or leave the default values.
o.	Cluster 2	Create your own custom configuration or leave the default values.
p.	Review and Edit	Review and proceed.
q.	Alerting	Leave unconfigured.
r.	Platinum Configuration	Uncheck the `Capture data for Platinum configuration` option.
s.	Oracle Config. Manager	Leave unconfigured.
t.	Auto Service Request	Leave unconfigured.
u.	Grid Control Agent	Leave unconfigured.
v.	Comments	Review and proceed.

Step	Window/Page Description	Configuration Settings and Other Instructions
w.	Generate	Click “Next >” as instructed.
x.	Select Destination Folder	Enter `ExadataConfigurations3` as the destination folder (using the File Name field) and click Save.
y.	Finish	Note the names and locations of the generated files. Click “Installation template” and examine the configuration details in the resulting web browser page. Close the web browser when you are finished examining the installation template.

11. Use any remaining time to experiment with the Exadata configuration tool to create different configurations. Exit your terminal session when you are finished.

Practices for Lesson 6: Exadata Storage Server Configuration

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In these practices, you will perform a variety of Exadata configuration tasks, including cell configuration and storage reconfiguration. You will also consume Exadata storage using ASM, configure Exadata storage security, exercise the privileges associated with the different cell user accounts and use the distributed command line utility (`dcli`).

Practice 6-1: Cell configuration

Overview

In this practice you examine, set and validate some Exadata cell parameters.

Tasks

1. Establish a terminal connection to `qr01celadm01` as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

2. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

3. Execute the following CellCLI command to examine the attributes of the cell.

```
CellCLI> list cell detail
         name:                   qr01celadm01
         cellVersion:            OSS_12.1.2.1.0_LINUX.X64_141206.1
         cpuCount:               2
         diagHistoryDays:        7
         fanCount:               0/0
         fanStatus:              normal
         flashCacheMode:         WriteThrough
         id:                     ef92136a-837c-4e1d-88d2-e01f5ab89b7b
         interconnectCount:      0
         interconnect1:          ib0
         interconnect2:          ib1
         iormBoost:              0.0
         ipaddress1:             192.168.1.105/24
         ipaddress2:             192.168.1.106/24
         kernelVersion:          2.6.39-400.243.1.el6uek.x86_64
         makeModel:              Fake hardware
         memoryGB:               4
         metricHistoryDays:      7
         offloadGroupEvents:
         offloadEfficiency:      791.1
         powerCount:             0/0
         powerStatus:            normal
         releaseVersion:         12.1.2.1.0
         releaseTrackingBug:     17885582
         status:                 online
         temperatureReading:     0.0
         temperatureStatus:      normal
         upTime:                 4 days, 21:22
         cellsrvStatus:          running
         msStatus:               running
         rsStatus:               running

CellCLI>
```

4. Configure the cell to send email alerts to a fictitious Exadata administrator.

```
CellCLI> alter cell smtpServer='my_mail.example.com',          -
>                   smtpFromAddr='john.doe@example.com',        -
>                   smtpFrom='John Doe',                        -
>                   smtpToAddr='jane.smith@example.com',        -
>                   notificationPolicy='critical,warning,clear', -
>                   notificationMethod='mail'
Cell qr01celadm01 successfully altered

CellCLI>
```

5. Reexamine the cell configuration to verify the changes you made in step 4.

```
CellCLI> list cell detail
         name:                   qr01celadm01
         cellVersion:            OSS_12.1.2.1.0_LINUX.X64_141206.1
         cpuCount:               2
         diagHistoryDays:        7
         fanCount:               0/0
         fanStatus:              normal
         flashCacheMode:         WriteThrough
         id:                     ef92136a-837c-4e1d-88d2-e01f5ab89b7b
         interconnectCount:      0
         interconnect1:          ib0
         interconnect2:          ib1
         iormBoost:              0.0
         ipaddress1:             192.168.1.105/24
         ipaddress2:             192.168.1.106/24
         kernelVersion:          2.6.39-400.243.1.el6uek.x86_64
         makeModel:              Fake hardware
         memoryGB:               4
         metricHistoryDays:      7
         notificationMethod:     mail
         notificationPolicy:     critical,warning,clear
         offloadGroupEvents:
         offloadEfficiency:      791.1
         powerCount:             0/0
         powerStatus:            normal
         releaseVersion:         12.1.2.1.0
         releaseTrackingBug:     17885582
         smtpFrom:               "John Doe"
         smtpFromAddr:           john.doe@example.com
         smtpServer:             my_mail.example.com
         smtpToAddr:             jane.smith@example.com
         status:                 online
         temperatureReading:     0.0
         temperatureStatus:      normal
         upTime:                 4 days, 21:23
         cellsrvStatus:          running
         msStatus:               running
         rsStatus:               running

CellCLI>
```

6. Execute the following CellCLI command to validate the email attributes configured for the cell.

   Note: Executing this command attempts to send a test email to each of the configured email addresses. The validation process only confirms the ability to successfully send a test email using the specified configuration. The validation process does not confirm the existence of the target email account, nor does it confirm successful receipt of the test email message. In this case an error message is observed because the target email server (`my_mail.example.com`) does not really exist.

```
CellCLI> alter cell validate mail


CELL-02578: An error was detected in the SMTP configuration:
CELL-05503: An error was detected during notification. The text
of the associated internal error is: Unknown SMTP host:
my_mail.example.com.
The notification recipient is jane.smith@example.com.


Please verify your SMTP configuration.


CellCLI>
```

7. Execute the following CellCLI command to perform a complete internal check of the cell configuration settings.

```
CellCLI> alter cell validate configuration
Cell qr01celadm01 successfully altered


CellCLI>
```

   Note that the `ALTER CELL VALIDATE CONFIGURATION` command does not perform I/O tests against the cell's hard disks and flash modules. You must use the `CALIBRATE` command to perform such tests. The `CALIBRATE` command can only be executed in a CellCLI session initiated by the `root` user.

8. Exit your CellCLI session.

Practice 6-2: Storage reconfiguration

Overview

In this practice, you alter the Database Machine storage configuration. The approach used in this practice allows the storage reconfiguration to occur while the system is running and databases remain available. The procedure is based on one of the methods described in the section entitled *Resizing Storage Grid Disks* in the *Oracle Exadata Database Machine Maintenance Guide*.

In this practice you will reconfigure the `RECO_QR01` ASM disk group. The aim is to resize the disk group and underlying Exadata grid disks so that some of the available free space can be used to create another disk group in the following practices.

Often storage is reconfigured to alter the balance of space allocated to the default DATA and RECO disk groups. In such cases, both disk groups are reconfigured in parallel, with the space freed from one disk group immediately consumed by the other disk group. See the *Oracle Exadata Database Machine Maintenance Guide* for more information.

Note: To complete this practice successfully, you must follow the instructions carefully and replicate the commands exactly. Failure to do so could result in unrecoverable damage to your lab environment. Please take care.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `grid` user, and configure the terminal session to access ASM using the `+ASM1` instance on `qr01dbadm01`.

```
$ ssh grid@qr01dbadm01
grid@qr01dbadm01's password: ????????
[grid@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@qr01dbadm01 ~]$
```

2. Using SQL*Plus, connect to ASM as an ASM administrator.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Examine the ASM disk groups in your environment using the following query (or execute the SQL script `/home/oracle/labs/lab06-02-03.sql`). Note that the `RECO_QR01` disk group is mostly free space.

```
SQL> select name, total_mb, free_mb, total_mb-free_mb used_mb
  2  from v$asm_diskgroup;

NAME                              TOTAL_MB    FREE_MB    USED_MB
------------------------------ ---------- ---------- ----------
DATA_QR01                            25920      11220      14700
DBFS_DG                              12672       2432      10240
RECO_QR01                            33408      32580        828

SQL>
```

4. The following query shows a summary of the space utilization for the disks in each disk group (use the SQL script `/home/oracle/labs/lab06-02-04.sql` if you prefer). For each disk group the query shows the number of associated disks, the size of each disk and the minimum and maximum amount of free space on the disks. Examine the output to ensure that the disk group being reconfigured is reasonably well balanced. In particular ensure that none of the disks are at or near capacity because this may cause problems with later rebalancing operations. If required rebalance the disk group using the `ALTER DISKGROUP ... REBALANCE` command prior to proceeding.

```
SQL> col name format a15
SQL> select dg.name, count(*), d.total_mb, d.os_mb,
  2  min(d.free_mb) MIN_FREE_MB, max(d.free_mb) MAX_FREE_MB
  3  from v$asm_disk d, v$asm_diskgroup dg
  4  where dg.group_number=d.group_number and d.mount_status='CACHED'
  5  group by dg.name, d.total_mb, d.os_mb;

NAME              COUNT(*)   TOTAL_MB      OS_MB MIN_FREE_MB MAX_FREE_MB
--------------- ---------- ---------- ---------- ----------- -----------
DBFS_DG                 36        352        352           4          84
DATA_QR01               36        720        720         292         328
RECO_QR01               36        928        928         888         916

SQL>
```

5. Resize the ASM disks in the `RECO_QR01` disk group so that each ASM disk shrinks from 928 MB to 608 MB.

```
SQL> alter diskgroup reco_qr01 resize all size 608m
  2  rebalance power 1024;

Diskgroup altered.

SQL>
```

6. Repeat the earlier queries from steps 3 and 4 and verify the effect of shrinking the `RECO_QR01` disk group.

```
SQL> select name, total_mb, free_mb, total_mb-free_mb used_mb
  2  from v$asm_diskgroup;

NAME              TOTAL_MB    FREE_MB    USED_MB
--------------- ---------- ---------- ----------
DATA_QR01            25920      11220      14700
DBFS_DG              12672       2432      10240
RECO_QR01            21888      21060        828

SQL> select dg.name, count(*), d.total_mb, d.os_mb,
  2  min(d.free_mb) MIN_FREE_MB, max(d.free_mb) MAX_FREE_MB
  3  from v$asm_disk d, v$asm_diskgroup dg
  4  where dg.group_number=d.group_number and d.mount_status='CACHED'
  5  group by dg.name, d.total_mb, d.os_mb;

NAME              COUNT(*)   TOTAL_MB      OS_MB MIN_FREE_MB MAX_FREE_MB
--------------- ---------- ---------- ---------- ----------- -----------
DBFS_DG                 36        352        352           4          84
DATA_QR01               36        720        720         292         328
RECO_QR01               36        608        928         568         596

SQL>
```

At this point you have shrunk the `RECO_QR01` disk group so that each ASM disk now consumes 608 MB. However, each underlying Exadata grid disk still consumes 928 MB, as shown by the value in the `OS_MB` column.

In the next part of this practice, you will also resize each grid disk associated with the `RECO_QR01` disk group. To use the method outlined in this practice requires at least enough free space so that the disk group can tolerate the temporary loss of the storage from two Exadata Storage Servers. Therefore, since the `RECO_QR01` disk group now totals 21888 MB spanning three Exadata Storage Servers, the method outlined in this practice requires 14592 MB (two-thirds of 21888 MB) of free space, which is easily the case in your practice environment.

You can vary the approach outlined in this practice to cater for situations where you have enough free space to tolerate the temporary loss of only one Exadata Storage Server from the disk group. This option is discussed later in the practice. However, if you do not have enough free space to tolerate the temporary loss of even one Exadata Storage Server from your disk group, you will not be able to use an online rebalancing method and you will need to consider alternatives such as using backup and recovery.

7. Check that no ASM rebalance operations are currently active across the cluster. If required, periodically repeat the query until no rows are returned. **Do not proceed to the next step until the query returns no rows.**

```
SQL> select * from gv$asm_operation;

no rows selected

SQL>
```

8. Drop the disks associated with the cell `qr01celadm01` (use the SQL script `/home/oracle/labs/lab06-02-08.sql` if you prefer). Note that a disk group rebalance is specified in order to maintain data redundancy.

```
SQL> alter diskgroup reco_qr01
  2  drop disks in failgroup qr01celadm01
  3  rebalance power 1024;

Diskgroup altered.

SQL>
```

9. Monitor the rebalance operation using the following query.

```
SQL> select * from gv$asm_operation;

  INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         2            3 REBAL RESYNC    DONE       1024

         0

         2            3 REBAL RESILVER  DONE       1024

         0

  INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ----------- ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------

         2            3 REBAL REBALANCE WAIT       1024

         0

         2            3 REBAL COMPACT   WAIT       1024
```

```
  INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         0

         1            3 REBAL RESYNC    DONE       1024       1024          0
         0          0           0
         0

         1            3 REBAL RESILVER  DONE       1024       1024          0

  INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         0          0           0
         0

         1            3 REBAL REBALANCE RUN        1024       1024          4
        68        403           0
         0


  INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         1            3 REBAL COMPACT   WAIT       1024       1024          0
         0          0           0
         0


8 rows selected.

SQL>
```

10. Periodically repeat the query to monitor the rebalance operation, which will take a few minutes to complete in your laboratory environment. When the query returns no results, the rebalance operation is completed. **Do not proceed to the next step until the rebalance operation completes.**

```
SQL> select * from gv$asm_operation;

no rows selected

SQL>
```

11. Use the following query (or execute the SQL script `/home/oracle/labs/lab06-02-11.sql`) to confirm that the disks are dropped (`HEADER_STATUS=FORMER` and `MOUNT_STATUS=CLOSED`).

```
SQL> select path, free_mb, header_status, mount_status
  2  from v$asm_disk
  3  where path like '%RECO_QR01%celadm01';

PATH
-------------------------------------------------------------------------------
   FREE_MB HEADER_STATU MOUNT_S
---------- ------------ -------
o/192.168.1.105;192.168.1.106/RECO_QR01_CD_10_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_06_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_00_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_05_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_08_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_03_qr01celadm01
         0 FORMER       CLOSED
```

```
PATH
--------------------------------------------------------------------------------
   FREE_MB HEADER_STATU MOUNT_S
---------- ------------ -------
o/192.168.1.105;192.168.1.106/RECO_QR01_CD_09_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_04_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_01_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_07_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_02_qr01celadm01
         0 FORMER       CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_11_qr01celadm01
         0 FORMER       CLOSED


12 rows selected.

SQL>
```

12. Establish a separate terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user. Maintain your ASM administrator SQL session as you will require this throughout the rest of the practice.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

13. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

14. Examine the grid disks on `qr01celadm01`. Notice the `ASMModeStatus` for the dropped disks.

```
CellCLI> list griddisk attributes name, size, ASMModeStatus
         DATA_QR01_CD_00_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_01_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_02_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_03_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_04_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_05_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_06_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_07_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_08_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_09_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_10_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_11_qr01celadm01    720M    ONLINE
         DBFS_DG_CD_00_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_01_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_02_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_03_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_04_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_05_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_06_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_07_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_08_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_09_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_10_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_11_qr01celadm01      352M    ONLINE
         RECO_QR01_CD_00_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_01_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_02_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_03_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_04_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_05_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_06_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_07_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_08_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_09_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_10_qr01celadm01    928M    UNUSED
         RECO_QR01_CD_11_qr01celadm01    928M    UNUSED

CellCLI>
```

15. Drop the grid disks on `qr01celadm01` previously associated with the `RECO_QR01` disk group.

```
CellCLI> drop griddisk all prefix=reco_qr01
GridDisk RECO_QR01_CD_00_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_01_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_02_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_03_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_04_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_05_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_06_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_07_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_08_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_09_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_10_qr01celadm01 successfully dropped
GridDisk RECO_QR01_CD_11_qr01celadm01 successfully dropped

CellCLI>
```

16. Create a new set of grid disks. Use the same grid disk prefix as before, but specify a size of 608 MB for each grid disk.

```
CellCLI> create griddisk all harddisk prefix=RECO_QR01, size=608M
GridDisk RECO_QR01_CD_00_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_01_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_02_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_03_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_04_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_05_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_06_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_07_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_08_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_09_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_10_qr01celadm01 successfully created
GridDisk RECO_QR01_CD_11_qr01celadm01 successfully created

CellCLI>
```

17. Re-examine the grid disk on `qr01celadm01`. Notice the reconfigured grid disks with the `RECO_QR01` prefix.

```
CellCLI> list griddisk attributes name, size, ASMModeStatus
         DATA_QR01_CD_00_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_01_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_02_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_03_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_04_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_05_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_06_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_07_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_08_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_09_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_10_qr01celadm01    720M    ONLINE
         DATA_QR01_CD_11_qr01celadm01    720M    ONLINE
         DBFS_DG_CD_00_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_01_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_02_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_03_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_04_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_05_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_06_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_07_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_08_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_09_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_10_qr01celadm01      352M    ONLINE
         DBFS_DG_CD_11_qr01celadm01      352M    ONLINE
         RECO_QR01_CD_00_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_01_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_02_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_03_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_04_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_05_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_06_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_07_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_08_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_09_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_10_qr01celadm01    608M    UNUSED
         RECO_QR01_CD_11_qr01celadm01    608M    UNUSED

CellCLI>
```

18. Exit your CellCLI session but keep your terminal session open. You will require a terminal session connected to `qr01celadm01` as the `celladmin` user later in the practice.

```
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

19. Back in your ASM administrator SQL session, re-execute the following query (or execute the SQL script `/home/oracle/labs/lab06-02-11.sql`). Notice that the reconfigured grid disks are listed with `HEADER_STATUS=CANDIDATE`.

```
SQL> select path, free_mb, header_status, mount_status
  2  from v$asm_disk
  3  where path like '%RECO_QR01%celadm01';

PATH
--------------------------------------------------------------------------------
   FREE_MB HEADER_STATU MOUNT_S
---------- ------------ -------
o/192.168.1.105;192.168.1.106/RECO_QR01_CD_10_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_06_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_00_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_05_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_08_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_03_qr01celadm01
         0 CANDIDATE    CLOSED
```

```
PATH
--------------------------------------------------------------------------------
   FREE_MB HEADER_STATU MOUNT_S
---------- ------------ -------
o/192.168.1.105;192.168.1.106/RECO_QR01_CD_09_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_04_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_01_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_07_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_02_qr01celadm01
         0 CANDIDATE    CLOSED

o/192.168.1.105;192.168.1.106/RECO_QR01_CD_11_qr01celadm01
         0 CANDIDATE    CLOSED


12 rows selected.

SQL>
```

20. Add the reconfigured grid disks back into the `RECO_QR01` disk group, and at the same time drop the disks associated with the cell `qr01celadm02` (use the SQL script `/home/oracle/labs/lab06-02-20.sql` if you prefer).

```
SQL> alter diskgroup reco_qr01 add disk
  2  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_00_qr01celadm01',
  3  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_01_qr01celadm01',
  4  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_02_qr01celadm01',
  5  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_03_qr01celadm01',
  6  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_04_qr01celadm01',
  7  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_05_qr01celadm01',
  8  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_06_qr01celadm01',
  9  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_07_qr01celadm01',
 10  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_08_qr01celadm01',
 11  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_09_qr01celadm01',
 12  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_10_qr01celadm01',
 13  'o/192.168.1.105;192.168.1.106/RECO_QR01_CD_11_qr01celadm01'
 14  drop disks in failgroup qr01celadm02
 15  rebalance power 1024;

Diskgroup altered.

SQL>
```

The preceding operation requires at least enough free space so that the disk group can tolerate the temporary loss of the storage from two Exadata Storage Servers. In situations with only enough free space to tolerate the temporary loss of one Exadata Storage Server from the disk group, you can perform the preceding operation in 2 steps. Firstly, add the reconfigured disks back into the disk group, and then drop the next set of disks. This two-step approach requires less free space to work; however, it results in 2 rebalance operations, which requires more time to perform.

21. Monitor the rebalance operation by using the following query.

```
SQL> select * from gv$asm_operation;

...

   INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         2            3 REBAL COMPACT   WAIT       1024

         0


8 rows selected.

SQL>
```

22. Periodically repeat the query to monitor the rebalance operation. When the query returns no results the rebalance operation is completed. **Do not proceed to the next step until the rebalance operation completes.**

```
SQL> select * from gv$asm_operation;

no rows selected

SQL>
```

23. Re-execute the query from step 4 (use the SQL script `/home/oracle/labs/lab06-02-04.sql` if you prefer). Now you can see the `RECO_QR01` disk group in a partially reconfigured state. At this point the storage associated with the `RECO_QR01` disk group has been reconfigured on `qr01celadm01` (12 disks) and the disks on `qr01celadm02` have been dropped.

```
SQL> select dg.name, count(*), d.total_mb, d.os_mb,
  2  min(d.free_mb) MIN_FREE_MB, max(d.free_mb) MAX_FREE_MB
  3  from v$asm_disk d, v$asm_diskgroup dg
  4  where dg.group_number=d.group_number and d.mount_status='CACHED'
  5  group by dg.name, d.total_mb, d.os_mb;

NAME              COUNT(*)   TOTAL_MB      OS_MB MIN_FREE_MB MAX_FREE_MB
--------------- ---------- ---------- ---------- ----------- -----------
DBFS_DG                 36        352        352           4          84
DATA_QR01               36        720        720         292         328
RECO_QR01               12        608        608         572         592
RECO_QR01               12        608        928         576         592

SQL>
```

24. Establish a separate terminal connection to the `qr01celadm02` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm02
celladmin@qr01celadm02's password: ????????
[celladmin@qr01celadm02 ~]$
```

25. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm02 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production ...

CellCLI>
```

26. Examine the grid disks on `qr01celadm02`. Notice again the `ASMModeStatus` for the dropped disks.

    Note: If the ASMModeStatus for the `RECO_QR01` prefixed grid disks is `ONLINE` then you have either dropped the wrong disks in step 20 or you are connected to the wrong cell (you should now be connected to `qr01celadm02`). In either case, **do not proceed to the next step until you have resolved the problem.**

```
CellCLI> list griddisk attributes name, size, ASMModeStatus
         DATA_QR01_CD_00_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_01_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_02_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_03_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_04_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_05_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_06_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_07_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_08_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_09_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_10_qr01celadm02    720M    ONLINE
         DATA_QR01_CD_11_qr01celadm02    720M    ONLINE
         DBFS_DG_CD_00_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_01_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_02_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_03_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_04_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_05_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_06_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_07_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_08_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_09_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_10_qr01celadm02      352M    ONLINE
         DBFS_DG_CD_11_qr01celadm02      352M    ONLINE
         RECO_QR01_CD_00_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_01_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_02_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_03_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_04_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_05_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_06_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_07_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_08_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_09_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_10_qr01celadm02    928M    UNUSED
         RECO_QR01_CD_11_qr01celadm02    928M    UNUSED

CellCLI>
```

27. Drop the grid disks on `qr01celadm02` previously associated with the `RECO_QR01` disk group.

```
CellCLI> drop griddisk all prefix=reco_qr01
GridDisk RECO_QR01_CD_00_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_01_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_02_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_03_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_04_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_05_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_06_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_07_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_08_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_09_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_10_qr01celadm02 successfully dropped
GridDisk RECO_QR01_CD_11_qr01celadm02 successfully dropped

CellCLI>
```

28. Create a new set of grid disks. Use the same grid disk prefix as before, but specify a size of 608 MB for each grid disk.

```
CellCLI> create griddisk all harddisk prefix=RECO_QR01, size=608M
GridDisk RECO_QR01_CD_00_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_01_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_02_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_03_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_04_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_05_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_06_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_07_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_08_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_09_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_10_qr01celadm02 successfully created
GridDisk RECO_QR01_CD_11_qr01celadm02 successfully created

CellCLI>
```

29. Exit your CellCLI session.

```
CellCLI> exit
quitting

[celladmin@qr01celadm02 ~]$
```

30. Back in your ASM administrator SQL session, add the reconfigured grid disks on `qr01celadm02` back into the `RECO_QR01` disk group, and at the same time drop the disks associated with the cell `qr01celadm03` (use the SQL script `/home/oracle/labs/lab06-02-30.sql` if you prefer).

```
SQL> alter diskgroup reco_qr01 add disk
  2  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_00_qr01celadm02',
  3  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_01_qr01celadm02',
  4  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_02_qr01celadm02',
  5  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_03_qr01celadm02',
  6  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_04_qr01celadm02',
  7  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_05_qr01celadm02',
  8  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_06_qr01celadm02',
  9  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_07_qr01celadm02',
 10  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_08_qr01celadm02',
 11  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_09_qr01celadm02',
 12  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_10_qr01celadm02',
 13  'o/192.168.1.107;192.168.1.108/RECO_QR01_CD_11_qr01celadm02'
 14  drop disks in failgroup qr01celadm03
 15  rebalance power 1024;

Diskgroup altered.

SQL>
```

As was the case in step 20, the preceding operation requires at least enough free space so that the disk group can tolerate the temporary loss of the storage from two Exadata Storage Servers. In situations with only enough free space to tolerate the temporary loss of one Exadata Storage Server from the disk group, you can perform the preceding operation in 2 steps by separately adding the reconfigured disks and then dropping the next set of disks.

31. Monitor the rebalance operation as before.

```
SQL> select * from gv$asm_operation;

...

   INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         2            3 REBAL COMPACT   WAIT       1024

         0


8 rows selected.

SQL>
```

32. Periodically repeat the query to monitor the rebalance operation. When the query returns no results, the rebalance operation is completed. **Do not proceed to the next step until the rebalance operation completes.**

```
SQL> select * from gv$asm_operation;

no rows selected

SQL>
```

33. Re-execute the query from step 4 (use the SQL script `/home/oracle/labs/lab06-02-04.sql` if you prefer). Now the storage associated with the `RECO_QR01` disk group has been reconfigured on two cells (24 disks on `qr01celadm01` and `qr01celadm02`) and the disks on the remaining cell (`qr01celadm03`) have been dropped.

```
SQL> select dg.name, count(*), d.total_mb, d.os_mb,
  2  min(d.free_mb) MIN_FREE_MB, max(d.free_mb) MAX_FREE_MB
  3  from v$asm_disk d, v$asm_diskgroup dg
  4  where dg.group_number=d.group_number and d.mount_status='CACHED'
  5  group by dg.name, d.total_mb, d.os_mb;

NAME              COUNT(*)   TOTAL_MB      OS_MB MIN_FREE_MB MAX_FREE_MB
--------------- ---------- ---------- ---------- ----------- -----------
DBFS_DG                 36        352        352           4          84
DATA_QR01               36        720        720         292         328
RECO_QR01               24        608        608         572         592

SQL>
```

34. Establish a separate terminal connection to the `qr01celadm03` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm03
celladmin@qr01celadm03's password: ????????
[celladmin@qr01celadm03 ~]$
```

35. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm03 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production ...

CellCLI>
```

36. Examine the grid disks on `qr01celadm03`. Notice again the `ASMModeStatus` for the dropped disks.

    Note: If the ASMModeStatus for the `RECO_QR01` prefixed grid disks is `ONLINE` then you have either dropped the wrong disks in step 30 or you are connected to the wrong cell (you should now be connected to `qr01celadm03`). In either case, **do not proceed to the next step until you have resolved the problem.**

```
CellCLI> list griddisk attributes name, size, ASMModeStatus
         DATA_QR01_CD_00_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_01_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_02_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_03_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_04_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_05_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_06_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_07_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_08_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_09_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_10_qr01celadm03    720M    ONLINE
         DATA_QR01_CD_11_qr01celadm03    720M    ONLINE
         DBFS_DG_CD_00_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_01_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_02_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_03_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_04_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_05_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_06_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_07_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_08_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_09_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_10_qr01celadm03      352M    ONLINE
         DBFS_DG_CD_11_qr01celadm03      352M    ONLINE
         RECO_QR01_CD_00_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_01_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_02_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_03_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_04_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_05_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_06_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_07_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_08_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_09_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_10_qr01celadm03    928M    UNUSED
         RECO_QR01_CD_11_qr01celadm03    928M    UNUSED

CellCLI>
```

37. Drop the grid disks on `qr01celadm03` previously associated with the `RECO_QR01` disk group.

```
CellCLI> drop griddisk all prefix=reco_qr01
GridDisk RECO_QR01_CD_00_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_01_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_02_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_03_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_04_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_05_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_06_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_07_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_08_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_09_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_10_qr01celadm03 successfully dropped
GridDisk RECO_QR01_CD_11_qr01celadm03 successfully dropped

CellCLI>
```

38. Create a new set of grid disks. Use the same grid disk prefix as before, but specify a size of 608 MB for each grid disk.

```
CellCLI> create griddisk all harddisk prefix=RECO_QR01, size=608M
GridDisk RECO_QR01_CD_00_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_01_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_02_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_03_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_04_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_05_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_06_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_07_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_08_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_09_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_10_qr01celadm03 successfully created
GridDisk RECO_QR01_CD_11_qr01celadm03 successfully created

CellCLI>
```

39. Exit your CellCLI session.

```
CellCLI> exit
quitting

[celladmin@qr01celadm03 ~]$
```

40. Back in your ASM administrator SQL session, add the reconfigured grid disks on `qr01celadm03` back into the `RECO_QR01` disk group (use the SQL script `/home/oracle/labs/lab06-02-40.sql` if you prefer).

```
SQL> alter diskgroup reco_qr01 add disk
  2  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_00_qr01celadm03',
  3  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_01_qr01celadm03',
  4  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_02_qr01celadm03',
  5  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_03_qr01celadm03',
  6  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_04_qr01celadm03',
  7  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_05_qr01celadm03',
  8  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_06_qr01celadm03',
  9  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_07_qr01celadm03',
 10  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_08_qr01celadm03',
 11  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_09_qr01celadm03',
 12  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_10_qr01celadm03',
 13  'o/192.168.1.109;192.168.1.110/RECO_QR01_CD_11_qr01celadm03'
 14  rebalance power 1024;

Diskgroup altered.

SQL>
```

41. Monitor the rebalance operation as before.

```
SQL> select * from gv$asm_operation;

...

   INST_ID GROUP_NUMBER OPERA PASS      STAT      POWER     ACTUAL      SOFAR
---------- ------------ ----- --------- ---- ---------- ---------- ----------
  EST_WORK   EST_RATE EST_MINUTES ERROR_CODE
---------- ---------- ----------- --------------------------------------------
    CON_ID
----------
         2            3 REBAL COMPACT   WAIT       1024

         0


8 rows selected.

SQL>
```

42. Periodically repeat the query to monitor the rebalance operation. When the query returns no results the rebalance operation is completed. **Do not proceed to the next step until the rebalance operation completes.**

```
SQL> select * from gv$asm_operation;

no rows selected


SQL>
```

43. Re-execute the query from step 4 (use the SQL script `/home/oracle/labs/lab06-02-04.sql` if you prefer). Now the storage associated with the `RECO_QR01` disk group has been reconfigured on all three cells.

```
SQL> select dg.name, count(*), d.total_mb, d.os_mb,
  2  min(d.free_mb) MIN_FREE_MB, max(d.free_mb) MAX_FREE_MB
  3  from v$asm_disk d, v$asm_diskgroup dg
  4  where dg.group_number=d.group_number and d.mount_status='CACHED'
  5  group by dg.name, d.total_mb, d.os_mb;

NAME              COUNT(*)   TOTAL_MB      OS_MB MIN_FREE_MB MAX_FREE_MB
--------------- ---------- ---------- ---------- ----------- -----------
DBFS_DG                 36        352        352           4          84
DATA_QR01               36        720        720         292         328
RECO_QR01               36        608        608         560         596

SQL>
```

In the final part of this practice, the free space created by reconfiguring the `RECO_QR01` disk group will be used to create another set of grid disks.

44. Launch the Exadata cell command-line interface (CellCLI) on `qr01celadm01`.

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production ...

CellCLI>
```

45. Use the following CellCLI command to show the free space on each cell disk.

```
CellCLI> list celldisk attributes name, freeSpace where freeSpace != 0
         CD_00_qr01celadm01      320M
         CD_01_qr01celadm01      320M
         CD_02_qr01celadm01      320M
         CD_03_qr01celadm01      320M
         CD_04_qr01celadm01      320M
         CD_05_qr01celadm01      320M
         CD_06_qr01celadm01      320M
         CD_07_qr01celadm01      320M
         CD_08_qr01celadm01      320M
         CD_09_qr01celadm01      320M
         CD_10_qr01celadm01      320M
         CD_11_qr01celadm01      320M

CellCLI>
```

46. Create a set of grid disks which consume all of the available free space. Specify `prefix=DATA2_QR01`.

```
CellCLI> create griddisk all harddisk prefix=DATA2_QR01
GridDisk DATA2_QR01_CD_00_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_01_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_02_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_03_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_04_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_05_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_06_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_07_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_08_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_09_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_10_qr01celadm01 successfully created
GridDisk DATA2_QR01_CD_11_qr01celadm01 successfully created

CellCLI>
```

47. Examine the newly created grid disks. Note that they are now ready to be consumed in an ASM disk group.

```
CellCLI> list griddisk attributes name, size, ASMModeStatus where name like
'DATA2.*'
         DATA2_QR01_CD_00_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_01_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_02_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_03_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_04_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_05_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_06_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_07_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_08_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_09_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_10_qr01celadm01   320M    UNUSED
         DATA2_QR01_CD_11_qr01celadm01   320M    UNUSED

CellCLI>
```

48. Exit your CellCLI session but keep your terminal session open.

```
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

49. In step 46, a set of grid disks was created on `qr01celadm01`. Use the following command to create similar grid disks on `qr01celadm02` and `qr01celadm03`.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm02,qr01celadm03 cellcli -e
create griddisk all harddisk prefix=DATA2_QR01
qr01celadm02: GridDisk DATA2_QR01_CD_00_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_01_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_02_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_03_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_04_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_05_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_06_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_07_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_08_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_09_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_10_qr01celadm02 successfully created
qr01celadm02: GridDisk DATA2_QR01_CD_11_qr01celadm02 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_00_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_01_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_02_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_03_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_04_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_05_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_06_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_07_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_08_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_09_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_10_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_11_qr01celadm03 successfully created
[celladmin@qr01celadm01 ~]$
```

50. Use the following command to verify the existence of the newly created grid disks on all three cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e list griddisk attributes name, size, ASMModeStatus where name like
\'DATA2.*\'"
qr01celadm01: DATA2_QR01_CD_00_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_01_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_02_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_03_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_04_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_05_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_06_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_07_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_08_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_09_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_10_qr01celadm01      320M    UNUSED
qr01celadm01: DATA2_QR01_CD_11_qr01celadm01      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_00_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_01_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_02_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_03_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_04_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_05_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_06_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_07_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_08_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_09_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_10_qr01celadm02      320M    UNUSED
qr01celadm02: DATA2_QR01_CD_11_qr01celadm02      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_00_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_01_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_02_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_03_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_04_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_05_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_06_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_07_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_08_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_09_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_10_qr01celadm03      320M    UNUSED
qr01celadm03: DATA2_QR01_CD_11_qr01celadm03      320M    UNUSED
[celladmin@qr01celadm01 ~]$
```

51. Exit your CellCLI and SQL*Plus sessions.

Practice 6-3: Consuming grid disks using ASM

Overview

In this practice, you consume some newly created Exadata grid disks using ASM.

Assumptions

Before beginning this practice you must complete Practice 6-2. Your ability to complete this practice depends on the existence of the grid disks that are created in practice 6-2.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `grid` user, and configure the terminal session to access ASM using the `+ASM1` instance on `qr01dbadm01`.

```
$ ssh grid@qr01dbadm01
grid@qr01dbadm01's password: ????????
[grid@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@qr01dbadm01 ~]$
```

2. By using SQL*Plus, connect to ASM as an ASM administrator.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

The ASM environment you are using is clustered across the entire Database Machine. All of the grid disks on all of the Exadata cells are visible inside ASM. In the previous practice you configured some new grid disks. Now you will consume those grid disks using ASM.

3. Execute the following query (or execute the SQL script `/home/oracle/labs/lab06-03-03.sql`). Note the capitalization inside the `like` string. The output shows a series of disks having `HEADER_STATUS=CANDIDATE`. This indicates that the disks do not belong to an ASM disk group and are ready to be consumed by ASM. Examine the `PATH` output. Can you determine the Exadata cell that contains each grid disk?

```
SQL> select name, header_status, path from v$asm_disk
  2  where path like 'o/%/DATA2%'
  3  and header_status='CANDIDATE';

NAME                           HEADER_STATU
------------------------------ ------------
PATH
--------------------------------------------------------------------------------
                               CANDIDATE
o/192.168.1.105;192.168.1.106/DATA2_QR01_CD_05_qr01celadm01

                               CANDIDATE
o/192.168.1.109;192.168.1.110/DATA2_QR01_CD_06_qr01celadm03

                               CANDIDATE
o/192.168.1.107;192.168.1.108/DATA2_QR01_CD_05_qr01celadm02

...

                               CANDIDATE
o/192.168.1.105;192.168.1.106/DATA2_QR01_CD_11_qr01celadm01

                               CANDIDATE
o/192.168.1.105;192.168.1.106/DATA2_QR01_CD_03_qr01celadm01

                               CANDIDATE
o/192.168.1.107;192.168.1.108/DATA2_QR01_CD_10_qr01celadm02


36 rows selected.

SQL>
```

Exadata grid disks are consumed inside ASM in two ways. You can add disks to an existing ASM disk group, which you did in the previous practice when you added disks back into the `RECO_QR01` disk group. Or you can create a new ASM disk group based on Exadata grid disks.

4. Create a new ASM disk group consuming all the grid disks that you created in the previous practice (use the SQL script `/home/oracle/labs/lab06-03-04.sql` if you prefer).

```
SQL> create diskgroup data2_qr01 normal redundancy
  2  disk 'o/*/DATA2_QR01*'
  3  attribute 'compatible.rdbms' = '12.1.0.2.0',
  4  'compatible.asm' = '12.1.0.2.0',
  5  'cell.smart_scan_capable' = 'TRUE',
  6  'au_size' = '4M';

Diskgroup created.

SQL>
```

The newly created disk group can be used to house Oracle data files in the same way as an ASM disk group based on any other storage. To complement the recommended `AU_SIZE` setting of 4 MB, you should set the initial extent size to 8 MB for large segments. The recommended approaches are discussed in the lesson entitled *Optimizing Database Performance with Exadata*.

5. Examine your newly created disk group using the following query (or execute the SQL script `/home/oracle/labs/lab06-03-05.sql`). Note how the grid disks from each different Exadata cell are automatically grouped into separate failure groups.

```
SQL> select d.path, dg.name GNAME, d.failgroup, d.state
  2  from v$asm_disk d, v$asm_diskgroup dg
  3  where d.group_number = dg.group_number
  4  and dg.name = 'DATA2_QR01';

PATH
--------------------------------------------------------------------------------
GNAME                          FAILGROUP                      STATE
------------------------------ ------------------------------ --------
o/192.168.1.109;192.168.1.110/DATA2_QR01_CD_08_qr01celadm03
DATA2_QR01                     QR01CELADM03                   NORMAL

o/192.168.1.109;192.168.1.110/DATA2_QR01_CD_05_qr01celadm03
DATA2_QR01                     QR01CELADM03                   NORMAL

o/192.168.1.107;192.168.1.108/DATA2_QR01_CD_06_qr01celadm02
DATA2_QR01                     QR01CELADM02                   NORMAL

...

o/192.168.1.105;192.168.1.106/DATA2_QR01_CD_05_qr01celadm01
DATA2_QR01                     QR01CELADM01                   NORMAL

o/192.168.1.105;192.168.1.106/DATA2_QR01_CD_07_qr01celadm01
DATA2_QR01                     QR01CELADM01                   NORMAL

o/192.168.1.105;192.168.1.106/DATA2_QR01_CD_09_qr01celadm01
DATA2_QR01                     QR01CELADM01                   NORMAL


36 rows selected.

SQL>
```

6. Drop the disk group you created in step 4.

```
SQL> drop diskgroup data2_qr01;

Diskgroup dropped.

SQL>
```

7. Exit your SQL*Plus session.

Practice 6-4: Configuring Exadata storage security

Overview

In this practice, you configure Exadata storage security.

Assumptions

Before beginning this practice you must complete Practice 6-2. Your ability to complete this practice depends on the existence of the grid disks that are created in practice 6-2.

Tasks

Exadata storage security has two modes; ASM-scoped security and database-scoped security. ASM-scoped security must be implemented before database-scoped security can be configured. In the first part of this practice, you will configure ASM-scoped security across your lab environment.

1. Establish a terminal connection to `qr01dbadm01` as the `grid` user, and configure the terminal session to access ASM using the `+ASM1` instance on `qr01dbadm01`.

```
$ ssh grid@qr01dbadm01
grid@qr01dbadm01's password: ????????
[grid@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
[grid@qr01dbadm01 ~]$
```

2. Using SQL*Plus, connect to ASM as an ASM administrator.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Note the `DB_UNIQUE_NAME` setting for the ASM environment and then exit SQL*Plus.

```
SQL> show parameter unique

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
db_unique_name                       string      +ASM
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[grid@qr01dbadm01 ~]$
```

4. Use the `su` command to assume the privileges of the `root` user.

```
[grid@qr01dbadm01 ~]$ su
Password: ????????
[root@qr01dbadm01 grid]#
```

To implement ASM-scoped security you must shut down the ASM cluster and any databases that are using it. However, in many cases it is easier to simply shut down the cluster across the Exadata Database Machine.

5. Shut down the cluster across your Exadata Database Machine environment.

```
[root@qr01dbadm01 grid]# crsctl stop cluster -all
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm01'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN2.lsnr' on
'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN3.lsnr' on
'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.dbm.db' on 'qr01dbadm01'
...
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'qr01dbadm02'
CRS-2673: Attempting to stop 'ora.DBFS_DG.dg' on 'qr01dbadm02'
CRS-2673: Attempting to stop 'ora.dbm.db' on 'qr01dbadm02'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on
'qr01dbadm02'
...
CRS-2792: Shutdown of Cluster Ready Services-managed resources
on 'qr01dbadm02' has completed
CRS-2677: Stop of 'ora.RECO_QR01.dg' on 'qr01dbadm01' succeeded
CRS-2673: Attempting to stop 'ora.DATA_QR01.dg' on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.DBFS_DG.dg' on 'qr01dbadm01'
...
CRS-2792: Shutdown of Cluster Ready Services-managed resources
on 'qr01dbadm01' has completed
CRS-2677: Stop of 'ora.crsd' on 'qr01dbadm01' succeeded
CRS-2673: Attempting to stop 'ora.ctssd' on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.evmd' on 'qr01dbadm01'
...
CRS-2673: Attempting to stop 'ora.cssd' on 'qr01dbadm01'
CRS-2677: Stop of 'ora.cssd' on 'qr01dbadm01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'qr01dbadm01'
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm01' succeeded
[root@qr01dbadm01 grid]#
```

6. Exit the `root` user session.

```
[root@qr01dbadm01 grid]# exit
exit
[grid@qr01dbadm01 ~]$
```

7. Leave the current terminal session active and establish a separate terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

8. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 – Production...

CellCLI>
```

9. Use the `CREATE KEY` command to generate a random hexadecimal key string. Then exit CellCLI.

```
CellCLI> create key
        aecacf517c96683eb33eaff589a59818
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

10. Use the `ASSIGN KEY` command to assign the previously generated security key to the Oracle ASM cluster on all of the cells accessed by the ASM cluster. Use the `DB_UNIQUE_NAME` observed earlier. Note that this is case-sensitive.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e assign key for +ASM=\'aecacf517c96683eb33eaff589a59818\'"
qr01celadm01: Key for +ASM successfully created
qr01celadm02: Key for +ASM successfully created
qr01celadm03: Key for +ASM successfully created
[celladmin@qr01celadm01 ~]$
```

11. Use the ALTER GRIDDISK command to configure security on the grid disks that you want the ASM cluster to access. Set the Oracle ASM DB_UNIQUE_NAME in the availableTo attribute of each grid disk.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e alter griddisk all availableTo=\'+ASM\'"
qr01celadm01: GridDisk DATA2_QR01_CD_00_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_01_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_02_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_03_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_04_qr01celadm01 successfully altered
...
qr01celadm03: GridDisk RECO_QR01_CD_07_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_08_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_09_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_10_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_11_qr01celadm03 successfully altered
[celladmin@qr01celadm01 ~]$
```

12. Return to your grid user session on qr01dbadm01 and change directory to /etc/oracle/cell/network-config.

```
[grid@qr01dbadm01 ~]$ cd /etc/oracle/cell/network-config
[grid@qr01dbadm01 network-config]$
```

13. Create a cellkey.ora file containing the key value from step 9 and the DB_UNIQUE_NAME for the ASM cluster.

```
[grid@qr01dbadm01 network-config]$ cat << END > cellkey.ora
> key=aecacf517c96683eb33eaff589a59818
> asm=+ASM
> END
[grid@qr01dbadm01 network-config]$
```

14. Confirm the contents of the cellkey.ora file.

```
[grid@qr01dbadm01 network-config]$ cat cellkey.ora
key=aecacf517c96683eb33eaff589a59818
asm=+ASM
[grid@qr01dbadm01 network-config]$
```

15. Set the file permissions and verify the settings.

```
[grid@qr01dbadm01 network-config]$ chown grid:asmadmin
cellkey.ora
[grid@qr01dbadm01 network-config]$ chmod 640 cellkey.ora
[grid@qr01dbadm01 network-config]$ ls -l cellkey.ora
-rw-r----- 1 grid asmadmin 46 Jun  7 03:48 cellkey.ora
[grid@qr01dbadm01 network-config]$
```

16. Copy the `cellkey.ora` file to every server in the ASM cluster.

```
[grid@qr01dbadm01 network-config]$ scp cellkey.ora
qr01dbadm02:/etc/oracle/cell/network-config
cellkey.ora                                    100%   46     0.0KB/s   00:00
[grid@qr01dbadm01 network-config]$
```

17. Use the `su` command to assume the privileges of the `root` user.

```
[grid@qr01dbadm01 network-config]$ su
Password: ????????
[root@qr01dbadm01 network-config]#
```

18. Restart the cluster.

```
[root@qr01dbadm01 network-config]# crsctl start cluster -all
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm02'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm02'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm01'
CRS-2676: Start of 'ora.cssdmonitor' on 'qr01dbadm02' succeeded
...
CRS-2676: Start of 'ora.asm' on 'qr01dbadm01' succeeded
CRS-2672: Attempting to start 'ora.storage' on 'qr01dbadm01'
CRS-2676: Start of 'ora.storage' on 'qr01dbadm01' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'qr01dbadm01'
CRS-2676: Start of 'ora.crsd' on 'qr01dbadm01' succeeded
[root@qr01dbadm01 network-config]#
```

19. Verify that all the Oracle cluster resources restart using the following command. It may take a few minutes for all of the cluster resources to restart so periodically re-execute the command until all the resources start. **Do not proceed to the next step before all of the cluster resources restart.**

```
[root@qr01dbadm01 network-config]# crsctl stat res -w "TARGET = ONLINE" -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA_QR01.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.DBFS_DG.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.RECO_QR01.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.asm
               ONLINE  ONLINE       qr01dbadm01              Started,STABLE
               ONLINE  ONLINE       qr01dbadm02              Started,STABLE
ora.net1.network
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.ons
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.cvu
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.dbm.db
      1        ONLINE  ONLINE       qr01dbadm01              Open,STABLE
      2        ONLINE  ONLINE       qr01dbadm02              Open,STABLE
ora.oc4j
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.qr01dbadm01.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
```

```
ora.qr01dbadm02.vip
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
--------------------------------------------------------------------------------
[root@qr01dbadm01 network-config]#
```

20. Exit the `root` user session and change back to the `grid` user home directory.

```
[root@qr01dbadm01 network-config]# exit
exit
[grid@qr01dbadm01 network-config]$ cd
[grid@qr01dbadm01 ~]$
```

ASM-scoped security is now configured. The fact that ASM and Oracle Database restarted shows that the ASM environment can access the grid disks configured on the Exadata storage. To further prove this is the case, you will now create a disk group on some of the grid disks.

21. Using SQL*Plus, connect to ASM as an ASM administrator.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

22. Create a new ASM disk group consuming the grid disks created in a previous practice (use the SQL script `/home/oracle/labs/lab06-04-22.sql` if you prefer).

```
SQL> create diskgroup data2_qr01_asm_sec normal redundancy
  2  disk 'o/*/DATA2_QR01*'
  3  attribute 'compatible.rdbms' = '12.1.0.2.0',
  4  'compatible.asm' = '12.1.0.2.0',
  5  'cell.smart_scan_capable' = 'TRUE',
  6  'au_size' = '4M';

Diskgroup created.

SQL>
```

23. Drop the newly created disk group and exit SQL*Plus.

```
SQL> drop diskgroup data2_qr01_asm_sec;

Diskgroup dropped.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[grid@qr01dbadm01 ~]$
```

As expected, the ASM cluster is able to access the Exadata storage using ASM-scoped security. Now imagine that another ASM cluster is configured and some of the grid disks are assigned to it. In the next part of this practice you will reconfigure the Exadata storage and see the effect.

24. Return to your `celladmin` user session on `qr01celadm01` and launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

25. Use the `CREATE KEY` command to generate another random hexadecimal key string. Then exit CellCLI.

```
CellCLI> create key
         4b03b5b2b54c871de54784b8064dabdd
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

26. Use the `ASSIGN KEY` command to assign the security key generated in step 25 to another Oracle ASM cluster called `+ASM2`.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e assign key for +ASM2=\'4b03b5b2b54c871de54784b8064dabdd\'"
qr01celadm01: Key for +ASM2 successfully created
qr01celadm02: Key for +ASM2 successfully created
qr01celadm03: Key for +ASM2 successfully created
[celladmin@qr01celadm01 ~]$
```

27. Confirm the key assignment. Note that each cell now has two key assignments for different ASM clusters.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list key
qr01celadm01: +ASM       aecacf517c96683eb33eaff589a59818
qr01celadm01: +ASM2      4b03b5b2b54c871de54784b8064dabdd
qr01celadm02: +ASM       aecacf517c96683eb33eaff589a59818
qr01celadm02: +ASM2      4b03b5b2b54c871de54784b8064dabdd
qr01celadm03: +ASM       aecacf517c96683eb33eaff589a59818
qr01celadm03: +ASM2      4b03b5b2b54c871de54784b8064dabdd
[celladmin@qr01celadm01 ~]$
```

28. Drop the grid disks having `prefix=DATA2_QR01`.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e drop griddisk all prefix=DATA2_QR01
qr01celadm01: GridDisk DATA2_QR01_CD_00_qr01celadm01 successfully dropped
qr01celadm01: GridDisk DATA2_QR01_CD_01_qr01celadm01 successfully dropped
qr01celadm01: GridDisk DATA2_QR01_CD_02_qr01celadm01 successfully dropped
qr01celadm01: GridDisk DATA2_QR01_CD_03_qr01celadm01 successfully dropped
qr01celadm01: GridDisk DATA2_QR01_CD_04_qr01celadm01 successfully dropped
...
qr01celadm03: GridDisk DATA2_QR01_CD_07_qr01celadm03 successfully dropped
qr01celadm03: GridDisk DATA2_QR01_CD_08_qr01celadm03 successfully dropped
qr01celadm03: GridDisk DATA2_QR01_CD_09_qr01celadm03 successfully dropped
qr01celadm03: GridDisk DATA2_QR01_CD_10_qr01celadm03 successfully dropped
qr01celadm03: GridDisk DATA2_QR01_CD_11_qr01celadm03 successfully dropped
[celladmin@qr01celadm01 ~]$
```

29. Create a new set of grid disks with the `availableTo` attribute set to `+ASM2`.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e create griddisk all harddisk prefix=DATA2_QR01,
availableTo=\'+ASM2\'"
qr01celadm01: GridDisk DATA2_QR01_CD_00_qr01celadm01 successfully created
qr01celadm01: GridDisk DATA2_QR01_CD_01_qr01celadm01 successfully created
qr01celadm01: GridDisk DATA2_QR01_CD_02_qr01celadm01 successfully created
qr01celadm01: GridDisk DATA2_QR01_CD_03_qr01celadm01 successfully created
qr01celadm01: GridDisk DATA2_QR01_CD_04_qr01celadm01 successfully created
...
qr01celadm03: GridDisk DATA2_QR01_CD_07_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_08_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_09_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_10_qr01celadm03 successfully created
qr01celadm03: GridDisk DATA2_QR01_CD_11_qr01celadm03 successfully created
[celladmin@qr01celadm01 ~]$
```

Note that the `ALTER GRIDDISK` command could have been used instead of dropping and recreating the grid disks. However, the `ALTER GRIDDISK` command can only be run against all the grid disks (regardless of prefix) or it can be used to modify an individual grid disk. Hence, using the `ALTER GRIDDISK` command in this case would require 36 separate commands (or equivalent scripting).

30. Confirm the `availableTo` attribute setting for all the grid disks.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list griddisk attributes name,availableTo
qr01celadm01: DATA2_QR01_CD_00_qr01celadm01    +ASM2
qr01celadm01: DATA2_QR01_CD_01_qr01celadm01    +ASM2
qr01celadm01: DATA2_QR01_CD_02_qr01celadm01    +ASM2
qr01celadm01: DATA2_QR01_CD_03_qr01celadm01    +ASM2
qr01celadm01: DATA2_QR01_CD_04_qr01celadm01    +ASM2
...
qr01celadm03: RECO_QR01_CD_07_qr01celadm03     +ASM
qr01celadm03: RECO_QR01_CD_08_qr01celadm03     +ASM
qr01celadm03: RECO_QR01_CD_09_qr01celadm03     +ASM
qr01celadm03: RECO_QR01_CD_10_qr01celadm03     +ASM
qr01celadm03: RECO_QR01_CD_11_qr01celadm03     +ASM
[celladmin@qr01celadm01 ~]$
```

ASM-scoped security is now reconfigured. Now see what happens when `+ASM` attempts to use the grid disks assigned to `+ASM2`.

31. Return to your `qr01dbadm01` terminal session. Using SQL*Plus, connect to ASM as an ASM administrator.

```
[grid@qr01dbadm01 ~]$ sqlplus / as sysasm

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

32. Create an ASM disk group referencing the grid disks assigned to `+ASM2` (use the SQL script `/home/oracle/labs/lab06-04-22.sql` if you prefer). Note that the command fails with an error message indicating that the disks do not exist.

```
SQL> create diskgroup data2_qr01_asm_sec normal redundancy
  2  disk 'o/*/DATA2_QR01*'
  3  attribute 'compatible.rdbms' = '12.1.0.2.0',
  4  'compatible.asm' = '12.1.0.2.0',
  5  'cell.smart_scan_capable' = 'TRUE',
  6  'au_size' = '4M';
create diskgroup data2_qr01_asm_sec normal redundancy
*
ERROR at line 1:
ORA-15018: diskgroup cannot be created
ORA-15031: disk specification 'o/*/DATA2_QR01*' matches no disks


SQL>
```

33. Execute the following query to confirm that the disks are not visible to ASM. Exadata storage security limits the visibility of grid disks to the environments that are allowed to access them.

```
SQL> select * from v$asm_disk where path like '%DATA2_QR01%';

no rows selected

SQL>
```

34. Exit your SQL*Plus session.

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[grid@qr01dbadm01 ~]$
```

So far you have examined ASM-scoped security. Database-scoped security can only be implemented after ASM-scoped security is already in place. The configuration process for database-scoped security very similar to the process you have already used for ASM-scoped security. In the next part of this practice you will configure database-scoped security for one Oracle Database. In an Exadata environment with multiple databases you would be required to repeat the process for each database.

35. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

36. Using SQL*Plus, connect as a database administrator.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

37. Note the DB_UNIQUE_NAME setting for the database and then exit SQL*Plus.

```
SQL> show parameter unique

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
db_unique_name                       string      dbm
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[oracle@qr01dbadm01 ~]$
```

38. Use the su command to assume the privileges of the root user.

```
[oracle@qr01dbadm01 ~]$ su
Password: ????????
[root@qr01dbadm01 oracle]#
```

To implement database-scoped security you must also shut down the ASM cluster and any databases that are using it. However, in many cases it is easier to simply shut down the cluster across the Exadata Database Machine.

39. Shut down the cluster across your Exadata Database Machine environment.

```
[root@qr01dbadm01 oracle]# /u01/app/12.1.0.2/grid/bin/crsctl stop cluster -all
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
'qr01dbadm01'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
'qr01dbadm02'
...
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm01' succeeded
CRS-2677: Stop of 'ora.cssd' on 'qr01dbadm02' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'qr01dbadm02'
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm02' succeeded
[root@qr01dbadm01 oracle]#
```

40. Exit the root user session.

```
[root@qr01dbadm01 oracle]# exit
exit
[oracle@qr01dbadm01 ~]$
```

41. Return to your celladmin user session on qr01celadm01 and launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 – Production...

CellCLI>
```

42. Use the `CREATE KEY` command to generate a random hexadecimal key string. Then exit CellCLI.

```
CellCLI> create key
         2877208d48fa273d86ee6492cd6fe331
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

43. Use the `ASSIGN KEY` command to assign the security key generated in step 42 to your Oracle Database on all the cells that you want the database to access. Use the `DB_UNIQUE_NAME` observed in step 37. Note that this is case-sensitive.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e assign key for dbm=\'2877208d48fa273d86ee6492cd6fe331\'"
qr01celadm01: Key for dbm successfully created
qr01celadm02: Key for dbm successfully created
qr01celadm03: Key for dbm successfully created
[celladmin@qr01celadm01 ~]$
```

44. Use the `ALTER GRIDDISK` command to configure security on the grid disks. For database-scoped security you must set the `availableTo` attribute of each grid disk to include both the ASM environment and the database which are allowed to access the grid disk. Use the following command to make all the grid disks accessible to the `dbm` database. Note that in other environments you would typically assign groups of grid disks to different databases.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e alter griddisk all availableTo=\'+ASM,dbm\'"
qr01celadm01: GridDisk DATA2_QR01_CD_00_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_01_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_02_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_03_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_04_qr01celadm01 successfully altered
...
qr01celadm03: GridDisk RECO_QR01_CD_07_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_08_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_09_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_10_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_11_qr01celadm03 successfully altered
[celladmin@qr01celadm01 ~]$
```

45. Return to your `oracle` user session on `qr01dbadm01` and create a directory at `$ORACLE_HOME/admin/<DB_UNIQUE_NAME>/pfile` where `<DB_UNIQUE_NAME>` represents the `DB_UNIQUE_NAME` setting for the database. Change into the newly created directory.

```
[oracle@qr01dbadm01 ~]$ mkdir -p $ORACLE_HOME/admin/dbm/pfile
[oracle@qr01dbadm01 ~]$ cd $ORACLE_HOME/admin/dbm/pfile
[oracle@qr01dbadm01 pfile]$
```

46. Create a `cellkey.ora` file containing the key value from step 42 and the `DB_UNIQUE_NAME` for the ASM cluster associated with the database.

```
[oracle@qr01dbadm01 pfile]$ cat << END > cellkey.ora
> key=2877208d48fa273d86ee6492cd6fe331
> asm=+ASM
> END
[oracle@qr01dbadm01 pfile]$
```

47. Confirm the contents of the `cellkey.ora` file.

```
[oracle@qr01dbadm01 pfile]$ cat cellkey.ora
key=2877208d48fa273d86ee6492cd6fe331
asm=+ASM
[oracle@qr01dbadm01 pfile]$
```

48. Set the file permissions and verify the settings. Note that on a Database Machine you would need to configure the `cellkey.ora` file on every database server associated with the database.

```
[oracle@qr01dbadm01 pfile]$ chmod 600 cellkey.ora
[oracle@qr01dbadm01 pfile]$ ls -l cellkey.ora
-rw------- 1 oracle oinstall 46 Jun  7 04:34 cellkey.ora
[oracle@qr01dbadm01 pfile]$
```

49. Copy the `cellkey.ora` file to every server in the database cluster.

```
[oracle@qr01dbadm01 pfile]$ ssh qr01dbadm02 "mkdir -p
/u01/app/oracle/product/12.1.0.2/dbhome_1/admin/dbm/pfile"
[oracle@qr01dbadm01 pfile]$ scp cellkey.ora
qr01dbadm02:/u01/app/oracle/product/12.1.0.2/dbhome_1/admin/dbm/pfile
cellkey.ora                                   100%   46     0.0KB/s   00:00
[oracle@qr01dbadm01 pfile]$
```

50. Use the `su` command to assume the privileges of the `root` user.

```
[oracle@qr01dbadm01 pfile]$ su
Password: ????????
[root@qr01dbadm01 pfile]#
```

51. Restart the cluster.

```
[root@qr01dbadm01 pfile]# /u01/app/12.1.0.2/grid/bin/crsctl start cluster -all
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm02'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm02'
...
CRS-2672: Attempting to start 'ora.storage' on 'qr01dbadm02'
CRS-2676: Start of 'ora.storage' on 'qr01dbadm02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'qr01dbadm02'
CRS-2676: Start of 'ora.crsd' on 'qr01dbadm02' succeeded
[root@qr01dbadm01 pfile]#
```

52. Verify that all the Oracle cluster resources restart using the following command. It may take a few minutes for all of the cluster resources to restart so periodically re-execute the command until all the resources start. **Do not proceed to the next step before all of the cluster resources restart.**

```
[root@qr01dbadm01 pfile]# /u01/app/12.1.0.2/grid/bin/crsctl stat res -w
"TARGET = ONLINE" -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA_QR01.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.DBFS_DG.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.RECO_QR01.dg
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.asm
               ONLINE  ONLINE       qr01dbadm01              Started,STABLE
               ONLINE  ONLINE       qr01dbadm02              Started,STABLE
ora.net1.network
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
ora.ons
               ONLINE  ONLINE       qr01dbadm01              STABLE
               ONLINE  ONLINE       qr01dbadm02              STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.cvu
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.dbm.db
      1        ONLINE  ONLINE       qr01dbadm01              Open,STABLE
      2        ONLINE  ONLINE       qr01dbadm02              Open,STABLE
ora.oc4j
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.qr01dbadm01.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
```

```
ora.qr01dbadm02.vip
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       qr01dbadm02              STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       qr01dbadm01              STABLE
--------------------------------------------------------------------------------
[root@qr01dbadm01 pfile]#
```

53. Exit the `root` user session and change back to the `oracle` user home directory.

```
[root@qr01dbadm01 pfile]# exit
exit
[oracle@qr01dbadm01 pfile]$ cd
[oracle@qr01dbadm01 ~]$
```

Database-scoped security is now configured. The fact that ASM and Oracle Database restarted shows that the both environments can access the grid disks configured on the Exadata storage.

54. Confirm that the database can access its storage by connecting to the database and executing a query.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL> select count(*) from customers;

  COUNT(*)
----------
   3252128

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[oracle@qr01dbadm01 ~]$
```

In the final part of the practice you will remove the Exadata storage security that you have configured during this practice.

55. Use the `su` command to assume the privileges of the `root` user.

```
[oracle@qr01dbadm01 ~]$ su
Password: ????????
[root@qr01dbadm01 oracle]#
```

56. Shut down the cluster.

```
[root@qr01dbadm01 oracle]# /u01/app/12.1.0.2/grid/bin/crsctl stop cluster -all
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN2.lsnr' on 'qr01dbadm01'
...
CRS-2677: Stop of 'ora.cssd' on 'qr01dbadm01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'qr01dbadm01'
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm02' succeeded
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm01' succeeded
[root@qr01dbadm01 oracle]#
```

57. Back in your `celladmin` session on `qr01celadm01`, use the following command to clear the `availableTo` grid disk attribute.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e alter griddisk all availableTo=\'\'"
qr01celadm01: GridDisk DATA2_QR01_CD_00_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_01_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_02_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_03_qr01celadm01 successfully altered
qr01celadm01: GridDisk DATA2_QR01_CD_04_qr01celadm01 successfully altered
...
qr01celadm03: GridDisk RECO_QR01_CD_07_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_08_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_09_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_10_qr01celadm03 successfully altered
qr01celadm03: GridDisk RECO_QR01_CD_11_qr01celadm03 successfully altered
[celladmin@qr01celadm01 ~]$
```

58. Clear the key assignment for the `dbm` database.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e assign key for dbm=\'\'"
qr01celadm01: Key for dbm successfully dropped
qr01celadm02: Key for dbm successfully dropped
qr01celadm03: Key for dbm successfully dropped
[celladmin@qr01celadm01 ~]$
```

59. Clear the key assignments for `+ASM` and `+ASM2`.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e assign key for +ASM=\'\'"
qr01celadm01: Key for +ASM successfully dropped
qr01celadm02: Key for +ASM successfully dropped
qr01celadm03: Key for +ASM successfully dropped
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
"cellcli -e assign key for +ASM2=\'\'"
qr01celadm01: Key for +ASM2 successfully dropped
qr01celadm02: Key for +ASM2 successfully dropped
qr01celadm03: Key for +ASM2 successfully dropped
[celladmin@qr01celadm01 ~]$
```

60. Return to your `root` session and remove the `cellkey.ora` files that you created earlier in the practice.

```
[root@qr01dbadm01 oracle]# rm
/u01/app/oracle/product/12.1.0.2/dbhome_1/admin/dbm/pfile/cellkey.ora
rm: remove regular file
`/u01/app/oracle/product/12.1.0.2/dbhome_1/admin/dbm/pfile/cellkey.ora'? y
[root@qr01dbadm01 oracle]# ssh qr01dbadm02 "rm
/u01/app/oracle/product/12.1.0.2/dbhome_1/admin/dbm/pfile/cellkey.ora"
root@qr01dbadm02's password: ????????
[root@qr01dbadm01 oracle]# rm /etc/oracle/cell/network-config/cellkey.ora
rm: remove regular file `/etc/oracle/cell/network-config/cellkey.ora'? y
[root@qr01dbadm01 oracle]# ssh qr01dbadm02 "rm /etc/oracle/cell/network-
config/cellkey.ora"
root@qr01dbadm02's password: ????????
[root@qr01dbadm01 oracle]#
```

61. Restart the cluster.

```
[root@qr01dbadm01 oracle]# /u01/app/12.1.0.2/grid/bin/crsctl start cluster -
all
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm02'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm02'
...
CRS-2672: Attempting to start 'ora.storage' on 'qr01dbadm02'
CRS-2676: Start of 'ora.storage' on 'qr01dbadm02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'qr01dbadm02'
CRS-2676: Start of 'ora.crsd' on 'qr01dbadm02' succeeded
[root@qr01dbadm01 oracle]#
```

62. Exit all of your terminal sessions.

Practice 6-5: Cell user accounts

Overview

In this practice you exercise the privileges available to the `celladmin` and `cellmonitor` Exadata administration accounts.

Tasks

1. Establish a terminal connection to `qr01celadm01` as the `cellmonitor` user.

```
$ ssh cellmonitor@qr01celadm01
cellmonitor@qr01celadm01's password: ????????
[cellmonitor@qr01celadm01 ~]$
```

2. Launch the Exadata cell command-line interface (CellCLI).

```
[cellmonitor@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 – Production...

CellCLI>
```

The `cellmonitor` user can only view Exadata cell objects using the CellCLI `LIST` command.

3. Confirm that `cellmonitor` can view the Exadata cell attributes.

```
CellCLI> list cell detail
         name:                   qr01celadm01
         cellVersion:            OSS_12.1.2.1.0_LINUX.X64_141206.1
         cpuCount:               2
         diagHistoryDays:        7
         fanCount:               0/0
         fanStatus:              normal
         flashCacheMode:         WriteThrough
         id:                     ef92136a-837c-4e1d-88d2-e01f5ab89b7b
         interconnectCount:      0
         interconnect1:          ib0
         interconnect2:          ib1
         iormBoost:              0.0
         ipaddress1:             192.168.1.105/24
         ipaddress2:             192.168.1.106/24
         kernelVersion:          2.6.39-400.243.1.el6uek.x86_64
         makeModel:              Fake hardware
         memoryGB:               4
         metricHistoryDays:      7
         notificationMethod:     mail
         notificationPolicy:     critical,warning,clear
         offloadGroupEvents:
         offloadEfficiency:      791.1
         powerCount:             0/0
         powerStatus:            normal
         releaseVersion:         12.1.2.1.0
         releaseTrackingBug:     17885582
         smtpFrom:               "John Doe"
         smtpFromAddr:           john.doe@example.com
         smtpServer:             my_mail.example.com
         smtpToAddr:             jane.smith@example.com
         status:                 online
         temperatureReading:     0.0
         temperatureStatus:      normal
         upTime:                 5 days, 2:53
         cellsrvStatus:          running
         msStatus:               running
         rsStatus:               running

CellCLI>
```

4. Confirm that `cellmonitor` cannot modify the Exadata cell attributes.

```
CellCLI> alter cell smtpToAddr='admin@example.com'
CELL-01520: This command is not permitted in monitor mode.

CellCLI>
```

5. Confirm that `cellmonitor` cannot create or modify the Exadata cell objects.

```
CellCLI> create celldisk all harddisk
CELL-01520: This command is not permitted in monitor mode.

CellCLI> alter griddisk all comment="Here is a comment"
CELL-01520: This command is not permitted in monitor mode.

CellCLI>
```

6. Establish a terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

7. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

In previous practices you have already seen how the `celladmin` user can create, modify and drop Exadata cell objects. In fact, the `celladmin` user can execute any CellCLI command except for the `CALIBRATE` command. The `CALIBRATE` command can only be executed by the `root` user.

8. Confirm that `celladmin` cannot run the `CALIBRATE` command.

```
CellCLI> calibrate
CELL-01522: CALIBRATE must be run as the root user id.

CellCLI>
```

9. Exit your CellCLI sessions.

Practice 6-6: Using the distributed command line utility (dcli)

Overview

The distributed command line utility (`dcli`) is a utility program which is provided with Database Machine. Its purpose is to provide a means to simultaneously execute monitoring and administration commands across multiple servers.

In earlier practices you used `dcli` to execute CellCLI commands across multiple Exadata cells. In this practice you will extend your use of `dcli` by performing the initial configuration required to enable `dcli` to issue commands to all of your cells from your database server (`qr01dbadm01`). You will also exercise some additional `dcli` functions.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user.

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$
```

2. Create a file named `mycells` which contains the names of your Exadata cells on separate lines.

```
[oracle@qr01dbadm01 ~]$ cat << END > mycells
> qr01celadm01
> qr01celadm02
> qr01celadm03
> END
[oracle@qr01dbadm01 ~]$
```

3. Confirm the existence of an SSH private/public key pair for the `oracle` OS user. In situations where you want to configure `dcli` for a user without an SSH key pair, you would use the `ssh-keygen` command to create one.

```
[oracle@qr01dbadm01 ~]$ ls ~/.ssh/id_rsa*
/home/oracle/.ssh/id_rsa  /home/oracle/.ssh/id_rsa.pub
[oracle@qr01dbadm01 ~]$
```

4. Execute the following command to configure SSH user-equivalence between your database server OS account (`oracle`) and the `celladmin` user on the cells specified in the `mycells` file.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells -k
celladmin@qr01celadm01's password: ????????
celladmin@qr01celadm03's password: ????????
celladmin@qr01celadm02's password: ????????
qr01celadm01: ssh key added
qr01celadm02: ssh key added
qr01celadm03: ssh key added
[oracle@qr01dbadm01 ~]$
```

You have now completed the one-time configuration required to enable `dcli` between the `oracle` user on `qr01dbadm01` and the `celladmin` user on each Exadata cell. After `dcli` is executed once with the `-k` option, subsequent `dcli` commands between the same servers and user accounts do not require the `-k` option and do not require a password.

5. The primary use for `dcli` is to simultaneously execute CellCLI commands across multiple cells. Use the following command to check on the status of your Exadata cells.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells cellcli -e list cell
qr01celadm01: qr01celadm01       online
qr01celadm02: qr01celadm02       online
qr01celadm03: qr01celadm03       online
[oracle@qr01dbadm01 ~]$
```

6. `dcli` can also be used to execute any non-interactive operating system commands on multiple cells and/or database servers. You can use quotes to surround compound commands and commands that contain pipes. Execute the following example command or construct an alternative command of your own.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells "cat /proc/meminfo | grep Mem"
qr01celadm01: MemTotal:        4104976 kB
qr01celadm01: MemFree:           71316 kB
qr01celadm02: MemTotal:        4104976 kB
qr01celadm02: MemFree:           62508 kB
qr01celadm03: MemTotal:        4104976 kB
qr01celadm03: MemFree:           61516 kB
[oracle@qr01dbadm01 ~]$
```

7. `dcli` is not just limited to monitoring. It is often used to ensure that consistent settings are applied across multiple systems. Use the following commands to view and adjust IORM settings on your Exadata cells.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells cellcli -e alter iormplan
objective=auto
qr01celadm01: IORMPLAN successfully altered
qr01celadm02: IORMPLAN successfully altered
qr01celadm03: IORMPLAN successfully altered
[oracle@qr01dbadm01 ~]$ dcli -g mycells cellcli -e list iormplan attributes
objective
qr01celadm01: auto
qr01celadm02: auto
qr01celadm03: auto
[oracle@qr01dbadm01 ~]$ dcli -g mycells cellcli -e alter iormplan
objective=basic
qr01celadm01: IORMPLAN successfully altered
qr01celadm02: IORMPLAN successfully altered
qr01celadm03: IORMPLAN successfully altered
[oracle@qr01dbadm01 ~]$ dcli -g mycells cellcli -e list iormplan attributes
objective
qr01celadm01: basic
qr01celadm02: basic
qr01celadm03: basic
[oracle@qr01dbadm01 ~]$
```

8. Sometimes `dcli` commands return a lot of output. The following command uses wildcards and a `WHERE` condition to return the current metric observations for small write I/O requests for every disk-based cell disk across all the cells. Execute the command and examine the output for your cells.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells "cellcli -e list metriccurrent where
name like \'CD_IO_RQ_W_S.?\' and metricobjectname like \'CD.*\'"
qr01celadm01: CD_IO_RQ_W_SM      CD_00_qr01celadm01      38,691 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_01_qr01celadm01      2,606 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_02_qr01celadm01      2,645 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_03_qr01celadm01      1,952 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_04_qr01celadm01      4,744 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_05_qr01celadm01      3,092 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_06_qr01celadm01      4,299 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_07_qr01celadm01      1,520 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_08_qr01celadm01      1,721 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_09_qr01celadm01      6,313 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_10_qr01celadm01      5,256 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_11_qr01celadm01      3,640 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_00_qr01celadm02      105,698 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_01_qr01celadm02      18,832 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_02_qr01celadm02      3,014 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_03_qr01celadm02      3,637 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_04_qr01celadm02      2,750 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_05_qr01celadm02      33,290 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_06_qr01celadm02      4,591 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_07_qr01celadm02      2,823 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_08_qr01celadm02      17,328 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_09_qr01celadm02      2,320 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_10_qr01celadm02      5,257 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_11_qr01celadm02      6,443 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_00_qr01celadm03      135,592 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_01_qr01celadm03      1,963 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_02_qr01celadm03      2,455 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_03_qr01celadm03      3,394 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_04_qr01celadm03      3,682 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_05_qr01celadm03      32,892 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_06_qr01celadm03      4,005 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_07_qr01celadm03      3,060 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_08_qr01celadm03      2,614 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_09_qr01celadm03      1,351 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_10_qr01celadm03      5,965 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_11_qr01celadm03      3,984 IO requests
[oracle@qr01dbadm01 ~]$
```

9. This command is essentially the same as the command you just executed in step 8. It uses the `-r` option along with a regular expression string to specify which output `dcli` should delete from the output. The result is that only the output that does not match the `-r` regular expression is returned to the user. Execute the command and examine the output for your cells.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells -r '.*CD_0.*' "cellcli -e list
metriccurrent where name like \'CD_IO_RQ_W_S.?\' and metricobjectname like
\'CD.*\'"
.*CD_0.*: ['qr01celadm01', 'qr01celadm02', 'qr01celadm03']
qr01celadm01: CD_IO_RQ_W_SM      CD_10_qr01celadm01      5,256 IO requests
qr01celadm01: CD_IO_RQ_W_SM      CD_11_qr01celadm01      3,640 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_10_qr01celadm02      5,257 IO requests
qr01celadm02: CD_IO_RQ_W_SM      CD_11_qr01celadm02      6,443 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_10_qr01celadm03      5,965 IO requests
qr01celadm03: CD_IO_RQ_W_SM      CD_11_qr01celadm03      3,984 IO requests
[oracle@qr01dbadm01 ~]$
```

`dcli` can also be used to copy files to numerous remote systems. You will exercise this capability in the next series of steps.

10. Create a small text file that contains a short message identifying you. Name the file according to your assigned student account. Confirm the existence and contents of the file.

```
[oracle@qr01dbadm01 ~]$ cat << END > message.txt
> Hello World!
> END
[oracle@qr01dbadm01 ~]$ cat message.txt
Hello World!
[oracle@qr01dbadm01 ~]$
```

11. Use `dcli` with the `-f` option to copy your file to the default home directory of the `celladmin` user on your Exadata cells.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells -f message.txt
[oracle@qr01dbadm01 ~]$
```

12. Use the following command to confirm the success of the operation in the previous step.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells cat message.txt
qr01celadm01: Hello World!
qr01celadm02: Hello World!
qr01celadm03: Hello World!
[oracle@qr01dbadm01 ~]$
```

In addition to copying a file to multiple remote locations, `dcli` can copy a file and execute it simultaneously on the specified remote systems. You will exercise this capability in the next series of steps.

13. Create a simple shell script such as the one shown below.

```
[oracle@qr01dbadm01 ~]$ cat << END > script.sh
> HST=\`hostname -s\`
> DTE=\`date\`
> echo -n \`cat message.txt\`
> echo " on \${HST} at \${DTE}."
> END
[oracle@qr01dbadm01 ~]$
```

14. Use the `chmod` command to make your newly created script file executable.

```
[oracle@qr01dbadm01 ~]$ chmod +x script.sh
[oracle@qr01dbadm01 ~]$
```

15. Use `dcli` with the `-x` option to copy and run the script you just created.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells -x script.sh
qr01celadm01: Hello World! on qr01celadm01 at Tue Jun  7 06:18:57 UTC 2016.
qr01celadm02: Hello World! on qr01celadm02 at Tue Jun  7 06:18:48 UTC 2016.
qr01celadm03: Hello World! on qr01celadm03 at Tue Jun  7 06:19:02 UTC 2016.
[oracle@qr01dbadm01 ~]$
```

Note that script files with the `.scl` extension are run by the CellCLI utility on the remote server.

16. Use `dcli` in conjunction with `rm` to delete the files you copied to your cells during this practice. Please be careful not to mistakenly delete any other files.

```
[oracle@qr01dbadm01 ~]$ dcli -g mycells rm -v message.txt script.sh
qr01celadm01: removed `message.txt'
qr01celadm01: removed `script.sh'
qr01celadm02: removed `message.txt'
qr01celadm02: removed `script.sh'
qr01celadm03: removed `message.txt'
qr01celadm03: removed `script.sh'
[oracle@qr01dbadm01 ~]$
```

17. Exit your terminal session.

Practices for Lesson 7: I/O Resource Management

Chapter 7

Practices for Lesson 7

There are no practices for this lesson.

Practices for Lesson 8: Recommendations for Optimizing Database Performance

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

In these practices, you will explore the following performance optimization techniques and technologies:

- Configuring write back flash cache
- Using Exadata Hybrid Columnar Compression
- Testing index elimination

Practice 8-1: Configuring Write Back Flash Cache

Overview

In this practice you will reconfigure Exadata Smart Flash Cache so that write operations can be serviced by flash only, instead of using disks. This mode of operation is known as write back flash cache.

Tasks

Reconfiguring Exadata Storage Servers to enable write back flash cache can be achieved in a rolling manner (one cell at a time) or all-at-once. In this practice you will reconfigure all the cells at once, which requires that Oracle Database and Grid Infrastructure are shut down on all database servers. To configure write back flash cache in a rolling manner some additional steps and checks are recommended. Refer to My Oracle Support bulletin 1500257.1 for details.

1. Establish a terminal connection to `qr01dbadm01` as the `root` user.

```
$ ssh root@qr01dbadm01
root@qr01dbadm01's password: ????????
[root@qr01dbadm01 ~]#
```

2. Shut down the cluster. **Do not proceed to the next step until this step completes.**

```
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/crsctl stop cluster -all
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on 'qr01dbadm02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on 'qr01dbadm01'
...
CRS-2677: Stop of 'ora.cssd' on 'qr01dbadm01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'qr01dbadm01'
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm02' succeeded
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm01' succeeded
[root@qr01dbadm01 ~]#
```

3. Leave the `root` terminal session active and establish a separate terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

4. Examine the cell `flashCacheMode` attribute setting on all of the cells. By default, each cell is configured in `WriteThrough` mode, which means that write operations must be persisted to disk regardless of whether or not the data resides inside Exadata Smart Flash Cache.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list cell attributes flashCacheMode
qr01celadm01: WriteThrough
qr01celadm02: WriteThrough
qr01celadm03: WriteThrough
[celladmin@qr01celadm01 ~]$
```

5. Drop the existing Exadata Smart Flash Cache.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e drop flashcache
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE successfully dropped
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE successfully dropped
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE successfully dropped
[celladmin@qr01celadm01 ~]$
```

6. Stop `cellsrv` on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter cell shutdown services cellsrv
qr01celadm01:
qr01celadm01: Stopping CELLSRV services...
qr01celadm01: The SHUTDOWN of CELLSRV services was successful.
qr01celadm02:
qr01celadm02: Stopping CELLSRV services...
qr01celadm02: The SHUTDOWN of CELLSRV services was successful.
qr01celadm03:
qr01celadm03: Stopping CELLSRV services...
qr01celadm03: The SHUTDOWN of CELLSRV services was successful.
[celladmin@qr01celadm01 ~]$
```

7. Enable write back flash cache on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter cell flashCacheMode = WriteBack
qr01celadm01: Cell qr01celadm01 successfully altered
qr01celadm02: Cell qr01celadm02 successfully altered
qr01celadm03: Cell qr01celadm03 successfully altered
[celladmin@qr01celadm01 ~]$
```

8. Restart `cellsrv` on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter cell startup services cellsrv
qr01celadm01:
qr01celadm01: Starting CELLSRV services...
qr01celadm01: The STARTUP of CELLSRV services was successful.
qr01celadm02:
qr01celadm02: Starting CELLSRV services...
qr01celadm02: The STARTUP of CELLSRV services was successful.
qr01celadm03:
qr01celadm03: Starting CELLSRV services...
qr01celadm03: The STARTUP of CELLSRV services was successful.
[celladmin@qr01celadm01 ~]$
```

9. Re-create Exadata Smart Flash Cache on all the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e create flashcache all
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE successfully created
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE successfully created
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE successfully created
[celladmin@qr01celadm01 ~]$
```

10. Verify the cell `flashCacheMode` attribute setting on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list cell attributes flashCacheMode
qr01celadm01: WriteBack
qr01celadm02: WriteBack
qr01celadm03: WriteBack
[celladmin@qr01celadm01 ~]$
```

Now that write back flash cache is enabled, each cache may contain updated data which is different from the data copy held on disk. This is called dirty data, and you can monitor the amount of dirty data inside Exadata Smart Flash Cache using cell metrics.

11. Use the following command to view the amount of dirty data currently inside Exadata Smart Flash Cache on each cell. Because the caches have only just been created and no databases are currently using the cells, the amount should be zero.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list metriccurrent FC_BY_DIRTY
qr01celadm01: FC_BY_DIRTY   FLASHCACHE      0.000 MB
qr01celadm02: FC_BY_DIRTY   FLASHCACHE      0.000 MB
qr01celadm03: FC_BY_DIRTY   FLASHCACHE      0.000 MB
[celladmin@qr01celadm01 ~]$
```

12. Return to you root terminal session and restart the cluster.

```
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/crsctl start cluster -all
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm02'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm02'
...
CRS-2672: Attempting to start 'ora.storage' on 'qr01dbadm02'
CRS-2676: Start of 'ora.storage' on 'qr01dbadm02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'qr01dbadm02'
CRS-2676: Start of 'ora.crsd' on 'qr01dbadm02' succeeded
[root@qr01dbadm01 ~]#
```

13. Wait for a few minutes until your database restarts. Periodically execute the following command to monitor the status of the database. **Proceed to the next step only after your database is started.**

```
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/srvctl status database -d dbm
Instance dbm1 is not running on node qr01dbadm01
Instance dbm2 is not running on node qr01dbadm02
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/srvctl status database -d dbm
Instance dbm1 is running on node qr01dbadm01
Instance dbm2 is running on node qr01dbadm02
[root@qr01dbadm01 ~]#
```

14. Return to your `celladmin` terminal session and view the amount of dirty data in each cell cache.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list metriccurrent FC_BY_DIRTY
qr01celadm01: FC_BY_DIRTY        FLASHCACHE      1.281 MB
qr01celadm02: FC_BY_DIRTY        FLASHCACHE      1.234 MB
qr01celadm03: FC_BY_DIRTY        FLASHCACHE      1.297 MB
[celladmin@qr01celadm01 ~]$
```

At this point, you have configured write back flash cache and you have confirmed that it is being used. Over time you would see more dirty data reported in the cache as you transact against your databases.

In the final part of this practice, you will reconfigure the cells once more and revert back to write through flash cache.

15. Return to you root terminal session and shut down the cluster. **Do not proceed to the next step until this step completes.**

```
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/crsctl stop cluster -all
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.crsd' on 'qr01dbadm02'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
'qr01dbadm01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN3.lsnr' on 'qr01dbadm01'
...
CRS-2677: Stop of 'ora.cssd' on 'qr01dbadm01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'qr01dbadm01'
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm02' succeeded
CRS-2677: Stop of 'ora.diskmon' on 'qr01dbadm01' succeeded
[root@qr01dbadm01 ~]#
```

Switching back to write through flash cache involves essentially the same steps as switching to write back flash cache. However, because write back flash cache is currently enabled you must first flush the flash cache before dropping it. Flushing Exadata Smart Flash Cache ensures that all modified data is written to disk.

16. Flush the write back flash cache on all of the cells. Note that the flush operation may take a few minutes to complete. If you receive a message indicating that the flush operation timed out for some of the flash disks, repeat this step until your output matches the example shown below.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter flashcache all flush
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE altered successfully
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE altered successfully
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE altered successfully
[celladmin@qr01celadm01 ~]$
```

17. Drop the Exadata Smart Flash Cache on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e drop flashcache
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE successfully dropped
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE successfully dropped
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE successfully dropped
[celladmin@qr01celadm01 ~]$
```

18. Stop `cellsrv` on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter cell shutdown services cellsrv
qr01celadm01:
qr01celadm01: Stopping CELLSRV services...
qr01celadm01: The SHUTDOWN of CELLSRV services was successful.
qr01celadm02:
qr01celadm02: Stopping CELLSRV services...
qr01celadm02: The SHUTDOWN of CELLSRV services was successful.
qr01celadm03:
qr01celadm03: Stopping CELLSRV services...
qr01celadm03: The SHUTDOWN of CELLSRV services was successful.
[celladmin@qr01celadm01 ~]$
```

19. Set the `flashCacheMode` attribute back to `WriteThrough`.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter cell flashCacheMode = WriteThrough
qr01celadm01: Cell qr01celadm01 successfully altered
qr01celadm02: Cell qr01celadm02 successfully altered
qr01celadm03: Cell qr01celadm03 successfully altered
 [celladmin@qr01celadm01 ~]$
```

20. Restart `cellsrv` on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e alter cell startup services cellsrv
qr01celadm01:
qr01celadm01: Starting CELLSRV services...
qr01celadm01: The STARTUP of CELLSRV services was successful.
qr01celadm02:
qr01celadm02: Starting CELLSRV services...
qr01celadm02: The STARTUP of CELLSRV services was successful.
qr01celadm03:
qr01celadm03: Starting CELLSRV services...
qr01celadm03: The STARTUP of CELLSRV services was successful.
[celladmin@qr01celadm01 ~]$
```

21. Re-create Exadata Smart Flash Cache on all the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e create flashcache all
qr01celadm01: Flash cache qr01celadm01_FLASHCACHE successfully created
qr01celadm02: Flash cache qr01celadm02_FLASHCACHE successfully created
qr01celadm03: Flash cache qr01celadm03_FLASHCACHE successfully created
[celladmin@qr01celadm01 ~]$
```

22. Verify the cell `flashCacheMode` attribute setting on all of the cells.

```
[celladmin@qr01celadm01 ~]$ dcli -c qr01celadm01,qr01celadm02,qr01celadm03
cellcli -e list cell attributes flashCacheMode
qr01celadm01: WriteThrough
qr01celadm02: WriteThrough
qr01celadm03: WriteThrough
[celladmin@qr01celadm01 ~]$
```

23. Return to you root terminal session and restart the cluster.

```
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/crsctl start cluster -all
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'qr01dbadm02'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm01'
CRS-2672: Attempting to start 'ora.evmd' on 'qr01dbadm02'
...
CRS-2672: Attempting to start 'ora.storage' on 'qr01dbadm02'
CRS-2676: Start of 'ora.storage' on 'qr01dbadm02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'qr01dbadm02'
CRS-2676: Start of 'ora.crsd' on 'qr01dbadm02' succeeded
[root@qr01dbadm01 ~]#
```

24. Wait for a few minutes until your database restarts. Periodically execute the following command to monitor the status of the database. **Proceed to the next practice only after your database is started.**

```
[root@qr01dbadm01 ~]# /u01/app/12.1.0.2/grid/bin/srvctl status database -d dbm
Instance dbm1 is running on node qr01dbadm01
Instance dbm2 is running on node qr01dbadm02
[root@qr01dbadm01 ~]#
```

25. Exit all of your terminal sessions.

Practice 8-2: Using Exadata Hybrid Columnar Compression

Overview

In this practice, you will examine the performance of Exadata Hybrid Columnar Compression. You will compare predicted and actual compression ratios using an example dataset. You will also examine how bulk data loading and query operations are affected using the different compression modes.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Configure the session to display server output and timing statistics.

```
SQL> set serveroutput on
SQL> set timing on
SQL>
```

4. Use `dbms_compression.get_compression_ratio` to predict the expected compression ratio for the `MYCUSTOMERS` table using all the different Exadata Hybrid Columnar Compression modes (use the SQL script `/home/oracle/labs/lab08-02-04.sql` if you prefer). Note that this step can take more than 10 minutes to complete in your laboratory environment.

```
SQL> declare
  2    b_cmp number;
  3    b_ucmp number;
  4    r_cmp number;
  5    r_ucmp number;
  6    cmp_ratio number(6,2);
  7    cmp_type varchar2(1024);
  8  begin
  9    dbms_compression.get_compression_ratio('USERS','SH',
 10      'MYCUSTOMERS',NULL,DBMS_COMPRESSION.COMP_QUERY_LOW,
 11      b_cmp,b_ucmp, r_cmp, r_ucmp, cmp_ratio, cmp_type);
 12    dbms_output.put_line('Table: MYCUSTOMERS');
 13    dbms_output.put_line('Compression Ratio: '||cmp_ratio);
 14    dbms_output.put_line('Compression Type: '|| cmp_type);
 15  end;
 16  /
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 1000001 rows
Table: MYCUSTOMERS
Compression Ratio: 2.5
Compression Type: "Compress Query Low"

PL/SQL procedure successfully completed.

Elapsed: 00:02:55.88
SQL> declare
  2    b_cmp number;
  3    b_ucmp number;
  4    r_cmp number;
  5    r_ucmp number;
  6    cmp_ratio number(6,2);
  7    cmp_type varchar2(1024);
  8  begin
  9    dbms_compression.get_compression_ratio('USERS','SH',
 10      'MYCUSTOMERS',NULL,DBMS_COMPRESSION.COMP_QUERY_HIGH,
 11      b_cmp,b_ucmp, r_cmp, r_ucmp, cmp_ratio, cmp_type);
 12    dbms_output.put_line('Table: MYCUSTOMERS');
 13    dbms_output.put_line('Compression Ratio: '||cmp_ratio);
 14    dbms_output.put_line('Compression Type: '|| cmp_type);
 15  end;
 16  /
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 1000001 rows
```

```
Table: MYCUSTOMERS
Compression Ratio: 4.2
Compression Type: "Compress Query High"

PL/SQL procedure successfully completed.

Elapsed: 00:01:40.31
SQL> declare
  2    b_cmp number;
  3    b_ucmp number;
  4    r_cmp number;
  5    r_ucmp number;
  6    cmp_ratio number(6,2);
  7    cmp_type varchar2(1024);
  8  begin
  9    dbms_compression.get_compression_ratio('USERS', 'SH',
 10      'MYCUSTOMERS',NULL,DBMS_COMPRESSION.COMP_ARCHIVE_LOW,
 11      b_cmp,b_ucmp, r_cmp, r_ucmp, cmp_ratio, cmp_type);
 12    dbms_output.put_line('Table: MYCUSTOMERS');
 13    dbms_output.put_line('Compression Ratio: '||cmp_ratio);
 14    dbms_output.put_line('Compression Type: '|| cmp_type);
 15  end;
 16  /
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 1000001 rows
Table: MYCUSTOMERS
Compression Ratio: 4.6
Compression Type: "Compress Archive Low"

PL/SQL procedure successfully completed.

Elapsed: 00:01:50.79
SQL> declare
  2    b_cmp number;
  3    b_ucmp number;
  4    r_cmp number;
  5    r_ucmp number;
  6    cmp_ratio number(6,2);
  7    cmp_type varchar2(1024);
  8  begin
  9    dbms_compression.get_compression_ratio('USERS', 'SH',
 10      'MYCUSTOMERS',NULL,DBMS_COMPRESSION.COMP_ARCHIVE_HIGH,
 11      b_cmp,b_ucmp, r_cmp, r_ucmp, cmp_ratio, cmp_type);
 12    dbms_output.put_line('Table: MYCUSTOMERS');
 13    dbms_output.put_line('Compression Ratio: '||cmp_ratio);
 14    dbms_output.put_line('Compression Type: '|| cmp_type);
 15  end;
 16  /
```

```
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 1000001 rows
Table: MYCUSTOMERS
Compression Ratio: 6.4
Compression Type: "Compress Archive High"

PL/SQL procedure successfully completed.

Elapsed: 00:02:29.60
SQL>
```

5. Exadata Hybrid Columnar Compression achieves its highest levels of compression with data that is direct-path inserted. Execute the following `ALTER SESSION` commands to ensure the use of direct-path inserts later in the practice.

```
SQL> alter session force parallel query;

Session altered.

Elapsed: 00:00:00.00
SQL> alter session force parallel ddl;

Session altered.

Elapsed: 00:00:00.00
SQL> alter session force parallel dml;

Session altered.

Elapsed: 00:00:00.01
```

6. Use the following commands (or execute the SQL script `/home/oracle/labs/lab08-02-06.sql`) to create compressed copies of the `MYCUSTOMERS` table. Notice the relative difference in the time taken to create each table by using the different compression modes.

```
SQL> create table mycust_query_low
  2  column store compress for query low
  3  nologging parallel 4
  4  as select * from mycustomers;

Table created.

Elapsed: 00:02:53.51
SQL> create table mycust_query_high
  2  column store compress for query high
  3  nologging parallel 4
  4  as select * from mycustomers;

Table created.

Elapsed: 00:00:34.43
SQL> create table mycust_archive_low
  2  column store compress for archive low
  3  nologging parallel 4
  4  as select * from mycustomers;

Table created.

Elapsed: 00:00:44.75
SQL> create table mycust_archive_high
  2  column store compress for archive high
  3  nologging parallel 4
  4  as select * from mycustomers;

Table created.

Elapsed: 00:01:22.92
SQL>
```

7. Use the following query (or execute the SQL script `/home/oracle/labs/lab08-02-07.sql`) to compare the size of the original uncompressed table with the newly created compressed copies, and to compare the actual compression ratios achieved with the ones predicted in step 4.

```
SQL> col segment_name format a30
SQL> select segment_name,MB,
  2  round((max(MB) over (partition by X))/MB,1) COMP_RATIO
  3  from (select segment_name,sum(bytes)/1024/1024 MB,1 X
  4  from user_segments
  5  where segment_name like 'MYCUST%'
  6  group by segment_name)
  7  order by 2 desc;

SEGMENT_NAME                           MB COMP_RATIO
------------------------------ ---------- ----------
MYCUSTOMERS                           654          1
MYCUST_QUERY_LOW                      260        2.5
MYCUST_QUERY_HIGH                     160        4.1
MYCUST_ARCHIVE_LOW                    150        4.4
MYCUST_ARCHIVE_HIGH                   109          6

Elapsed: 00:00:01.20
SQL>
```

In the next part of the practice, you will compare direct path insert performance for compressed and uncompressed tables. On each occasion you will perform the same transaction twice. The first time will help to prime the system to ensure more consistent results. You should take particular note of the timings for the second insert command. This will help you to determine the impact of Exadata Hybrid Columnar Compression on bulk data loading operations.

8. As a baseline, execute the following transactions to load data into the uncompressed `MYCUSTOMERS` table. Note the time taken to perform the second insert.

```
SQL> insert /*+APPEND */ into mycustomers
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:22.06
SQL> commit;

Commit complete.

Elapsed: 00:00:00.10
SQL> insert /*+APPEND */ into mycustomers
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:06.85
SQL> commit;

Commit complete.

Elapsed: 00:00:00.06
SQL>
```

9. Execute the same insert transactions against the `COMPRESS FOR QUERY LOW` copy of the table. Note the time taken to perform the second insert. You may observe that the time for this insert is better than the uncompressed insert in step 8. In this case, the cost of performing the compression is offset by the lower number of I/O operations that are required. This characteristic is one of the reasons why query compression is well suited to data warehouse environments where large data loads exist.

```
SQL> insert /*+APPEND */ into mycust_query_low
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:03.99
SQL> commit;

Commit complete.

Elapsed: 00:00:00.04
SQL> insert /*+APPEND */ into mycust_query_low
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:03.84
SQL> commit;

Commit complete.

Elapsed: 00:00:00.03
SQL>
```

10. Execute the same insert transactions against the `COMPRESS FOR QUERY HIGH` copy of the table. Note the time taken to perform the second insert and compare it with the previous results.

```
SQL> insert /*+APPEND */ into mycust_query_high
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:05.00
SQL> commit;

Commit complete.

Elapsed: 00:00:00.04
SQL> insert /*+APPEND */ into mycust_query_high
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:04.46
SQL> commit;

Commit complete.

Elapsed: 00:00:00.08
SQL>
```

11. Execute the same insert transactions against the COMPRESS FOR ARCHIVE LOW copy of the table. Note the time taken to perform the second insert. You should observe that the load times are steadily increasing as more aggressive compression modes are used.

```
SQL> insert /*+APPEND */ into mycust_archive_low
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:05.27
SQL> commit;

Commit complete.

Elapsed: 00:00:00.04
SQL> insert /*+APPEND */ into mycust_archive_low
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:05.28
SQL> commit;

Commit complete.

Elapsed: 00:00:00.02
SQL>
```

12. Execute the insert transaction against the `COMPRESS FOR ARCHIVE HIGH` copy of the table. Note the time taken to perform the second insert. This time you should observe a more substantial cost for the data compression. This is because `COMPRESS FOR ARCHIVE HIGH` uses a more costly compression algorithm to achieve higher levels of compression. This extra cost is generally acceptable in archiving situations because the data does not change (or changes very little) after it is loaded.

```
SQL> insert /*+APPEND */ into mycust_archive_high
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:06.67
SQL> commit;

Commit complete.

Elapsed: 00:00:00.03
SQL> insert /*+APPEND */ into mycust_archive_high
  2  select * from seed_data;

500000 rows created.

Elapsed: 00:00:07.40
SQL> commit;

Commit complete.

Elapsed: 00:00:00.03
SQL>
```

In the final part of the practice, you will compare query performance for compressed and uncompressed tables.

13. Reconnect to your database as the `sh` user. This clears the session-level statistics, which will be used later to compare query performance.

```
SQL> connect sh/??
Connected.
SQL>
```

14. Execute the following test query against the uncompressed table.

```
SQL> select avg(cust_credit_limit) from mycustomers;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7683.31668

Elapsed: 00:00:10.93
SQL>
```

15. Examine the I/O statistics for the query you just ran (use the SQL script `/home/oracle/labs/lab08-02-15.sql` if you prefer). This will provide a baseline for later comparison.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                              MB
---------------------------------------------------------------- ----------
physical read total bytes                                        849.617188
physical write total bytes                                                0
cell physical IO interconnect bytes                              55.8248749
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            849.429688
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       55.6373749
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                              MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       849.664063
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.10
SQL>
```

16. Reconnect to your database as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

17. Execute the following test query against the `COMPRESS FOR QUERY LOW` copy of the table.

```
SQL> select avg(cust_credit_limit) from mycust_query_low;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7683.31668

Elapsed: 00:00:18.24
SQL>
```

18. Examine the I/O statistics for the query you just ran (use the SQL script `/home/oracle/labs/lab08-02-15.sql` if you prefer). Compare the statistics with the results observed in step 15. Note the substantial decrease in the I/O required to satisfy the query.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        340.734375
physical write total bytes                                                0
cell physical IO interconnect bytes                               21.457222
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            338.085938
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       18.8087845
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       719.677283
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.05
SQL>
```

19. Reconnect to your database as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

20. Execute the following test query against the `COMPRESS FOR QUERY HIGH` copy of the table.

```
SQL> select avg(cust_credit_limit) from mycust_query_high;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7683.31668

Elapsed: 00:00:16.40
SQL>
```

21. Examine the I/O statistics for the query you just ran (use the SQL script `/home/oracle/labs/lab08-02-15.sql` if you prefer). Compare the statistics with the results observed previously. Note again the decline in the I/O required to satisfy the query.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                 MB
---------------------------------------------------------------- ----------
physical read total bytes                                        210.578125
physical write total bytes                                                0
cell physical IO interconnect bytes                               20.350296
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            207.953125
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan        17.725296
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                 MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       720.139211
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.03
SQL>
```

22. Reconnect to your database as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

23. Execute the following test query against the `COMPRESS FOR ARCHIVE LOW` copy of the table.

```
SQL> select avg(cust_credit_limit) from mycust_archive_low;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7683.31668

Elapsed: 00:00:13.80
SQL>
```

24. Examine the I/O statistics for the query you just ran (use the SQL script `/home/oracle/labs/lab08-02-15.sql` if you prefer). Compare the statistics with the results observed previously. Note the continued decline in the I/O required to satisfy the query.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        196.335938
physical write total bytes                                                0
cell physical IO interconnect bytes                              17.6317673
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload              194.0625
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       15.3583298
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       722.061086
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.03
SQL>
```

25. Reconnect to your database as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

26. Execute the following test query against the `COMPRESS FOR ARCHIVE HIGH` copy of the table. Compare the query performance using the uncompressed table and all of the different compressed versions of the table. Note that in some cases, the queries against the compressed tables outperformed the query against the uncompressed table. With compression, you can often observe improved query performance for scanning queries because less I/O is required.

```
SQL> select avg(cust_credit_limit) from mycust_archive_high;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7683.31668

Elapsed: 00:00:07.44
SQL>
```

27. Examine the I/O statistics for the query you just ran (use the SQL script `/home/oracle/labs/lab08-02-15.sql` if you prefer). Compare the statistics with the results previously observed. Note again the decline in the I/O required to satisfy the query.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        144.476563
physical write total bytes                                                0
cell physical IO interconnect bytes                              18.5637817
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload               140.625
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       14.7122192
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       721.553273
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.03
SQL>
```

28. Exit your SQL*Plus session.

Practice 8-3: Testing Index elimination

Overview

In this practice, you make an index invisible so that you can test the effect of removing the index on your queries without actually dropping the index.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Configure your session to display timing statistics and execution plans. Then flush the database buffer cache to ensure consistent results in the following steps.

```
SQL> set timing on
SQL> set autotrace on explain
SQL> alter system flush buffer_cache;

System altered.

Elapsed: 00:00:01.20
SQL>
```

4. Execute the following query. Note that the execution plan uses an index range scan on the CUSTOMERS_PK index. Note also the time taken to execute the query using the index.

```
SQL> select avg(cust_credit_limit) from customers
  2  where cust_id between 200000 and 320000;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7696.89836

Elapsed: 00:00:05.99

Execution Plan
----------------------------------------------------------
Plan hash value: 3524092294

-------------------------------------------------------------------------------
| Id  | Operation                            | Name         | Rows  | Bytes |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                     |              |     1 |    10 |
|   1 |  SORT AGGREGATE                      |              |     1 |    10 |
|   2 |   TABLE ACCESS BY INDEX ROWID BATCHED| CUSTOMERS    |  120K|  1171K|
|*  3 |    INDEX RANGE SCAN                  | CUSTOMERS_PK |  120K|       |
-------------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("CUST_ID">=200000 AND "CUST_ID"<=320000)

SQL>
```

5. Reconfigure your session to disable the automatic output of execution plans.

```
SQL> set autotrace off
SQL>
```

6. Make the CUSTOMERS_PK index invisible. An invisible index still exists and is maintained by DML operations, but it is not used by the optimizer for queries.

```
SQL> alter index customers_pk invisible;

Index altered.

Elapsed: 00:00:00.16
SQL>
```

7. The index you have just made invisible is associated with a primary key constraint. Use the following query to check the status of the constraint. Note that even though the index is invisible, the associated constraint is still enabled.

```
SQL> select status from user_constraints
  2  where constraint_name='CUSTOMERS_PK';

STATUS
--------
ENABLED

Elapsed: 00:00:00.64
SQL>
```

8. Reconfigure your session to automatically show execution plans.

```
SQL> set autotrace on explain
SQL>
```

9. Re-execute the query from step 4. Notice that an Exadata Smart Scan is used rather than an index range scan.

```
SQL> select avg(cust_credit_limit) from customers
  2  where cust_id between 200000 and 320000;

AVG(CUST_CREDIT_LIMIT)
----------------------
            7696.89836

Elapsed: 00:00:12.80

Execution Plan
----------------------------------------------------------
Plan hash value: 296924608

-----------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows   | Bytes | Cost (%CPU)|
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |      1 |    10 | 23479   (1)|
|   1 |  SORT AGGREGATE              |           |      1 |    10 |            |
|*  2 |   TABLE ACCESS STORAGE FULL| CUSTOMERS |   120K|  1171K| 23479   (1)|
-----------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage("CUST_ID"<=320000 AND "CUST_ID">=200000)
       filter("CUST_ID"<=320000 AND "CUST_ID">=200000)

SQL>
```

Compare the time taken to execute the query with and without the index. Note that in some cases, Exadata Smart Scan may deliver better query performance than using an index.

However, this may not always be the case. Even in cases where an index delivers better query performance you might choose to remove it if you determine that the un-indexed query performance is acceptable and the index is otherwise unnecessary. Removing unnecessary indexes saves space and improves DML performance by eliminating the maintenance operations associated with the index. If you decide not to remove the index, you can quickly and easily make it visible.

10. Make the index visible again.

```
SQL> alter index customers_pk visible;

Index altered.

Elapsed: 00:00:00.02
SQL>
```

11. Exit your SQL*Plus session.

Practices for Lesson 9: Using Smart Scan

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

In these practices, you will exercise Exadata Smart Scan and examine various statistics and wait events to determine what is occurring.

Practice 9-1: Monitoring Exadata Smart Scan

Overview

In this practice, you will examine various Exadata Smart Scan statistics and measures which can be observed within Oracle Database.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Flush the buffer cache to ensure a consistent starting point for this practice, and reconnect to the database to reset the session statistics. Then, configure your session to display query execution plans and statement timings.

```
SQL> alter system flush buffer_cache;

System altered.

SQL> connect sh/??
Connected.
SQL> set autotrace on explain
SQL> set timing on
SQL>
```

4. Execute the following query. You can identify whether Smart Scan is possible by examining the query execution plan. Smart Scan is indicated for this query by the `TABLE ACCESS STORAGE FULL` operation. You can also see that the `WHERE` clause predicate (`occupation = 'Farming'`) can be evaluated by Exadata.

```
SQL> select count(*) from cust_info where occupation = 'Farming';

  COUNT(*)
----------
     27244

Elapsed: 00:00:05.84

Execution Plan
----------------------------------------------------------
Plan hash value: 1273666552

------------------------------------------------------------------------------
| Id  | Operation                  | Name       | Rows   | Bytes | Cost (%CPU)|
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |            |     1 |      7 |  8718   (1)|
|   1 |  SORT AGGREGATE            |            |     1 |      7 |            |
|*  2 |   TABLE ACCESS STORAGE FULL| CUST_INFO  |  229K|  1570K|  8718   (1)|
------------------------------------------------------------------------------



Predicate Information (identified by operation id):
---------------------------------------------------


   2 - storage("OCCUPATION"='Farming')
       filter("OCCUPATION"='Farming')

SQL>
```

5. Reconfigure your session to disable the automatic output of execution plans.

```
SQL> set autotrace off
SQL>
```

6. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-01-06.sql`) and examine the statistics for the current session. Since the query in step 4 is the only query that has been executed during this session we can safely assume that the statistics relate to that query. The statistics show that approximately 250 MB of IO was performed to scan the `cust_info` table (`physical read total bytes`), and that almost all of the IO was eligible for Smart Scan (`cell physical IO bytes eligible for predicate offload`). The statistics further show that Smart Scan returned approximately 400 KB of data back to the database server (`cell physical IO interconnect bytes returned by smart scan`).

```
SQL> SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
  2  WHERE s.statistic# = m.statistic# AND
  3  (s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%'
  4  OR s.name LIKE 'cell IO%');

NAME                                                                    MB
---------------------------------------------------------------- ----------
physical read total bytes                                        250.132813
physical write total bytes                                                0
cell physical IO interconnect bytes                              1.02911377
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            249.515625
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan        .41192627
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                    MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       249.515625
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.06
SQL>
```

7. Execute the following query (or execute the SQL script /home/oracle/labs/lab09-01-07.sql) and examine the cell wait events associated with the current session. Note that the amount of time associated with the cell smart table scan wait event accounts for most of the execution time observed in step 4. This is normal for a query using Smart Scan. You may also see some other wait events that relate to other activities in the session, such as executing the statistics queries in this step and the prior step.

```
SQL> SELECT DISTINCT event, total_waits,
  2  time_waited/100 wait_secs, average_wait/100 avg_wait_secs
  3  FROM V$SESSION_EVENT e, V$MYSTAT s
  4  WHERE event LIKE 'cell%' AND e.sid = s.sid;

EVENT                                                            TOTAL_WAITS
---------------------------------------------------------------- -----------
 WAIT_SECS AVG_WAIT_SECS
---------- -------------
cell single block physical read                                           91
       .28          .003

cell smart table scan                                                    255
      5.65         .0222


Elapsed: 00:00:00.28
SQL>
```

Examining the statistics and wait events associated with the query executed at step 4 indicates that the query did make efficient use of Exadata Smart Scan just as the query execution plan suggested. In the next part of this practice you will consider a scenario where the query execution plan indicates the use of Smart Scan but the statistics and wait events suggest something more.

8. Leave your current SQL session active and establish a second terminal connection to qr01dbadm01 as the oracle user. Configure this terminal to access the dbm database (dbm1 instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

9. In the second terminal, connect to your database with SQL*Plus. Log in as the `sh` user and set the SQL prompt to `UPDATE>` so that you can easily distinguish this session from your other SQL session.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL> set sqlprompt "UPDATE> "
UPDATE>
```

10. In the second terminal, execute the following command to update a substantial number of customer records. Following the update, flush the buffer cache. This simulates a long running update transaction where the updated blocks (and associated rollback segment blocks) have been aged out of the buffer cache. **Leave the transaction in this terminal window uncommitted for now. Do not proceed to the next step until the update command completes and the buffer cache is flushed.**

```
UPDATE> update cust_info set
     2  affinity_card = 0
     3  where occupation = 'Farming';

27244 rows updated.

UPDATE> alter system flush buffer_cache;

System altered.

UPDATE>
```

11. Switch back to your first SQL session, leaving the second terminal session in the background for now.
12. Back in the first SQL session, reconnect to the database to establish a fresh database session as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

13. Configure the session to display query execution plans.

```
SQL> set autotrace on explain
SQL>
```

14. Re-execute the query from step 4. Notice again that the query execution plan indicates the use of Smart Scan. Notice also that the execution time increases substantially compared with step 4.

```
SQL> select count(*) from cust_info where occupation = 'Farming';

  COUNT(*)
----------
     27244

Elapsed: 00:00:30.01

Execution Plan
----------------------------------------------------------
Plan hash value: 1273666552

------------------------------------------------------------------------------
| Id  | Operation                   | Name      | Rows   | Bytes | Cost (%CPU)|
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |           |     1 |     7 |  8718   (1)|
|   1 |  SORT AGGREGATE             |           |     1 |     7 |            |
|*  2 |   TABLE ACCESS STORAGE FULL| CUST_INFO |   229K|  1570K|  8718   (1)|
------------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage("OCCUPATION"='Farming')
       filter("OCCUPATION"='Farming')

SQL>
```

15. Reconfigure your session to disable the automatic output of execution plans.

```
SQL> set autotrace off
SQL>
```

16. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-01-06.sql`) and examine the statistics for the current session. Compare the output with the observations at step 6. This time the same amount of IO is eligible for Smart Scan (`cell physical IO bytes eligible for predicate offload`); however, substantially more data was transported to the database server (`cell physical IO interconnect bytes returned by smart scan`).

```
SQL> SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
  2  WHERE s.statistic# = m.statistic# AND
  3  (s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%'
  4  OR s.name LIKE 'cell IO%');

NAME                                                                MB
---------------------------------------------------------------- ----------
physical read total bytes                                        254.007813
physical write total bytes                                                0
cell physical IO interconnect bytes                              147.894104
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            249.515625
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       143.401917
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       249.515625
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.05
SQL>
```

17. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-01-07.sql`) and examine the cell wait events associated with the current session. Compare the output with the observations at step 7. Notice that this time a significant amount of time is associated with the `cell single block physical read` wait event. This is because the uncommitted update transaction has forced a substantial number of reads to be transferred to the traditional buffer cache read-consistency path and since the required blocks were not in the buffer cache a large number of single block physical reads were required. Notice also the amount of time associated with the different wait events and how they correlate with the overall query execution time. Clearly, the efficiency and performance of Smart Scan were severely compromised by the pending transaction.

```
SQL> SELECT DISTINCT event, total_waits,
  2  time_waited/100 wait_secs, average_wait/100 avg_wait_secs
  3  FROM V$SESSION_EVENT e, V$MYSTAT s
  4  WHERE event LIKE 'cell%' AND e.sid = s.sid;

EVENT                                                            TOTAL_WAITS
---------------------------------------------------------------- -----------
 WAIT_SECS AVG_WAIT_SECS
---------- -------------
cell single block physical read                                          576
     21.58         .0375

cell smart table scan                                                    160
      8.11         .0507


Elapsed: 00:00:00.41
SQL>
```

In the final part of this practice, you will consider another scenario where the query execution plan indicates the use of Smart Scan but the statistics and wait events suggest something different.

18. Switch back to the second SQL session which contains the update transaction.
19. Commit the current transaction and then execute another update command to update a substantial number of customer records. This time, don't flush the buffer cache after the update command. **Like you did in step 10, leave the transaction in this terminal window uncommitted for now. Do not proceed to the next step until the update command completes.**

```
UPDATE> commit;

Commit complete.

UPDATE> update cust_info set
      2  affinity_card = 1
      3  where occupation = 'Farming';

27244 rows updated.

UPDATE>
```

20. Back in the first SQL session, reconnect to the database to establish a fresh database session as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

21. Configure your newly created session to display query execution plans and summary statistics for each SQL command.

```
SQL> set autotrace on
SQL>
```

22. Re-execute the query from step 4. Notice again that the query execution plan indicates the use of Smart Scan. However, this time the summary statistics show a very small number of `physical reads` (zero in the example shown below) and a comparatively large number of `consistent gets`, which indicates that most of the IO for this query was satisfied by using the buffer cache.

```
SQL> select count(*) from cust_info where occupation = 'Farming';

  COUNT(*)
----------
     27244

Elapsed: 00:00:01.12

Execution Plan
----------------------------------------------------------
Plan hash value: 1273666552

------------------------------------------------------------------------------
| Id  | Operation                  | Name      | Rows  | Bytes | Cost (%CPU)|
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |           |     1 |     7 |  8718   (1)|
|   1 |  SORT AGGREGATE            |           |     1 |     7 |            |
|*  2 |   TABLE ACCESS STORAGE FULL| CUST_INFO |  229K|  1570K|  8718   (1)|
------------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage("OCCUPATION"='Farming')
       filter("OCCUPATION"='Farming')


Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
     104805  consistent gets
          0  physical reads
    1321072  redo size
        544  bytes sent via SQL*Net to client
        551  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed

SQL>
```

23. Reconfigure your session to disable automatic tracing.

```
SQL> set autotrace off
SQL>
```

24. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-01-06.sql`) and examine the statistics for the current session. Now the statistics relate to the query at step 22. Notice that all the statistics show very little cell activity and no `cell physical IO interconnect bytes returned by smart scan`.

```
SQL> SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
  2  WHERE s.statistic# = m.statistic# AND
  3  (s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%'
  4  OR s.name LIKE 'cell IO%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                              .125
physical write total bytes                                                0
cell physical IO interconnect bytes                                    .125
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.03
SQL>
```

25. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-01-07.sql`) and examine the cell wait events associated with the current session. This time there are no wait events indicating the use of smart scan. Evidently the database kernel realized that the query could be satisfied using blocks in the buffer cache. So even though the query plan indicted the use of Smart Scan, the database kernel used the buffer cache at runtime and avoided nearly all IO against the cells.

```
SQL> SELECT DISTINCT event, total_waits,
  2  time_waited/100 wait_secs, average_wait/100 avg_wait_secs
  3  FROM V$SESSION_EVENT e, V$MYSTAT s
  4  WHERE event LIKE 'cell%' AND e.sid = s.sid;

EVENT                                                            TOTAL_WAITS
---------------------------------------------------------------- -----------
 WAIT_SECS AVG_WAIT_SECS
---------- -------------
cell single block physical read                                           16
       .09         .0058


Elapsed: 00:00:00.39
SQL>
```

In this practice you have seen some scenarios where the optimizer indicated the use of Smart Scan. However, you have also seen that depending on the situation, the performance of Smart Scan may be impacted by other concurrent transactions, or Smart Scan may be skipped, partially or completely, if the database kernel can make use of information in the buffer cache to avoid IO operations.

26. Switch back to the second SQL session which contains the update transaction.
27. Commit the transaction.

```
UPDATE> commit;

Commit complete.

UPDATE>
```

28. Exit all your terminal sessions.

Practice 9-2: Monitoring Cell Wait Events for Parallel Query

Overview

In this practice, you consider strategies for monitoring cell wait events when parallel query is used.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the `sh` user.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Configure your session to display query execution plans and statement timings.

```
SQL> set autotrace on explain
SQL> set timing on
SQL>
```

4. Typically Exadata Smart Scan is used in conjunction with parallel query. Configure your session to force the use of parallel query.

```
SQL> alter session force parallel query parallel 2;

Session altered.

Elapsed: 00:00:00.00
SQL>
```

5. Execute the following query. The query execution plan indicates the use of parallel query and Exadata Smart Scan.

```
SQL> select count(*) from cust_info where occupation = 'Farming';

  COUNT(*)
----------
     27244

Elapsed: 00:00:06.27

Execution Plan
----------------------------------------------------------
Plan hash value: 3555626242

------------------------------------------------------------------------------
| Id  | Operation                   | Name      | Rows  | Bytes | Cost
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |           |     1 |     7 |  4838
|   1 |  SORT AGGREGATE             |           |     1 |     7 |
|   2 |   PX COORDINATOR            |           |       |       |
|   3 |    PX SEND QC (RANDOM)      | :TQ10000  |     1 |     7 |
|   4 |     SORT AGGREGATE          |           |     1 |     7 |
|   5 |      PX BLOCK ITERATOR      |           |   229K|  1570K|  4838
|*  6 |       TABLE ACCESS STORAGE FULL| CUST_INFO |   229K|  1570K|  4838
------------------------------------------------------------------------------


Predicate Information (identified by operation id):
---------------------------------------------------

   6 - storage("OCCUPATION"='Farming')
       filter("OCCUPATION"='Farming')

Note
-----
   - Degree of Parallelism is 2 because of session

SQL>
```

6. Reconfigure your session to disable the automatic output of execution plans.

```
SQL> set autotrace off
SQL>
```

7. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-02-07.sql`) and examine the statistics for the current session. The statistics confirm the use of Smart Scan for the query in step 5.

```
SQL> SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
  2  WHERE s.statistic# = m.statistic# AND
  3  (s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%'
  4  OR s.name LIKE 'cell IO%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        249.640625
physical write total bytes                                                0
cell physical IO interconnect bytes                              .543540955
cell physical IO bytes saved during optimized file creation               0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload            249.515625
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan       .418540955
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                       249.515625
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

Elapsed: 00:00:00.05
SQL>
```

8. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-02-08.sql`) and examine the cell wait events associated with the current session. Notice that there are very few (if any) waits.

```
SQL> SELECT DISTINCT event, total_waits,
  2  time_waited/100 wait_secs, average_wait/100 avg_wait_secs
  3  FROM V$SESSION_EVENT e, V$MYSTAT s
  4  WHERE event LIKE 'cell%' AND e.sid = s.sid;

EVENT                                                             TOTAL_WAITS
----------------------------------------------------------------- -----------
 WAIT_SECS AVG_WAIT_SECS
---------- -------------
cell single block physical read                                            16
       .07         .0043


Elapsed: 00:00:00.22
SQL>
```

Based on the statistics in step 7, you might reasonably expect to see wait events for `cell smart table scan` in step 8. What happened? Because parallel query was used, the query IO was performed by parallel server processes. The associated wait events are connected to the parallel server sessions, not the current session. Note that this behavior is symptomatic of parallel query and is not Exadata-specific. So when parallel query is used, the wait events must be observed differently. The rest of the practice shows two alternative strategies for observing the wait events.

9. Execute the following query (or execute the SQL script `/home/oracle/labs/lab09-02-09.sql`) to display the cell wait events across the entire system.

```
SQL> col event format a35
SQL> select event, total_waits,
  2  time_waited/100 wait_secs, average_wait/100 avg_wait_secs
  3  from v$system_event where event like 'cell%';

EVENT                               TOTAL_WAITS  WAIT_SECS AVG_WAIT_SECS
----------------------------------- ----------- ---------- -------------
cell smart table scan                     10299     312.97         .0304
cell statistics gather                      117         .1         .0009
cell smart file creation                  14047     116.76         .0083
cell single block physical read           10208       66.9         .0066
cell multiblock physical read              3199      44.27         .0138
cell list of blocks physical read            12        .06          .005

6 rows selected.

Elapsed: 00:00:00.05
SQL>
```

10. Re-execute the parallel query.

```
SQL> select count(*) from cust_info where occupation =
'Farming';


  COUNT(*)
----------
     27244

Elapsed: 00:00:00.69
SQL>
```

11. Re-execute the following query (or execute the SQL script `/home/oracle/labs/lab09-02-09.sql`) to again display the cell wait events across the entire system. Compare the output with the output from step 9. The differences are the cell wait events associated with the query at step 10.

```
SQL> select event, total_waits,
  2  time_waited/100 wait_secs, average_wait/100 avg_wait_secs
  3  from v$system_event where event like 'cell%';

EVENT                                TOTAL_WAITS  WAIT_SECS AVG_WAIT_SECS
------------------------------------ ----------- ---------- -------------
cell smart table scan                      10601     313.74         .0296
cell statistics gather                       117         .1         .0009
cell smart file creation                   14047     116.76         .0083
cell single block physical read            10208       66.9         .0066
cell multiblock physical read               3199      44.27         .0138
cell list of blocks physical read             12        .06          .005

6 rows selected.

Elapsed: 00:00:00.01
SQL>
```

Using system level wait event statistics is a simple way to monitor parallel query wait events as long as you are the only user of the system and you do not wish to monitor concurrent operations. Often this is not the case. The final part of this practice shows another method that can be used to isolate the wait events associated with a specific parallel query operation regardless of concurrency.

12. Execute the following query to determine the default trace file for the current session. Take note of the directory path since that location will also be the default location for other trace files.

```
SQL> select value from v$diag_info
  2  where name = 'Default Trace File';

VALUE
--------------------------------------------------------------
/u01/app/oracle/diag/rdbms/dbm/dbm1/trace/dbm1_ora_86399.trc

Elapsed: 00:00:00.18
SQL>
```

13. Use the `dbms_session.set_identifier` procedure to set a client identifier (`PQ1`) for the current session. The client identifier will help to locate trace information associated with the current session and any parallel query sessions that perform work in behalf of the current session.

```
SQL> exec dbms_session.set_identifier(client_id=>'PQ1')

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.17
SQL>
```

14. Use the `dbms_monitor.client_id_trace_enable` procedure to start recording trace information for the `PQ1` client identifier. Notice that `waits=>true` is specified to ensure that wait information is recorded in the trace.

```
SQL> exec dbms_monitor.client_id_trace_enable(client_id=>'PQ1', waits=>true,
binds=>false)

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.20
SQL>
```

15. Re-execute the parallel query.

```
SQL> select count(*) from cust_info where occupation =
'Farming';

  COUNT(*)
----------
     27244

Elapsed: 00:00:00.68
SQL>
```

16. Stop the trace gathering started in step 14 and exit SQL*Plus.

```
SQL> exec dbms_monitor.client_id_trace_disable(client_id=>'PQ1')

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.00
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0...
[oracle@qr01dbadm01 ~]$
```

17. Change directories to the location of the trace file observed in step 12.

```
[oracle@qr01dbadm01 ~]$ cd /u01/app/oracle/diag/rdbms/dbm/dbm1/trace
[oracle@qr01dbadm01 trace]$
```

18. Search for trace files containing `CLIENT ID:(PQ1)`. The resulting output will include the trace file for the client SQL session that was listed in the query output from step 12. It will also include trace files associated with any parallel query servers that performed work for the same client.

```
[oracle@qr01dbadm01 trace]$ grep "CLIENT ID:(PQ1)" *
dbm1_ora_86399.trc:*** CLIENT ID:(PQ1) 2016-06-09 04:28:48.093
dbm1_p000_114880.trc:*** CLIENT ID:(PQ1) 2016-06-09 04:29:01.215
dbm1_p001_114882.trc:*** CLIENT ID:(PQ1) 2016-06-09 04:29:01.215
[oracle@qr01dbadm01 trace]$
```

Note that trace files relating to your query may also exist on other nodes in your cluster if your parallel query was executed across multiple database servers.

19. Examine the trace files listed in step 18. The parallel query server trace files will display the `cell smart table scan` events associated with the parallel query.

```
[oracle@qr01dbadm01 trace]$ grep cell dbm1_p001_114882.trc
...
WAIT #139732971106048: nam='cell smart table scan' ela= 19
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180278511
WAIT #139732971106048: nam='cell smart table scan' ela= 492
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180279122
WAIT #139732971106048: nam='cell smart table scan' ela= 15
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180279227
WAIT #139732971106048: nam='cell smart table scan' ela= 563
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180279819
WAIT #139732971106048: nam='cell smart table scan' ela= 15
cellhash#=3097415979 p2=0 p3=0 obj#=20476 tim=610180279931
WAIT #139732971106048: nam='cell smart table scan' ela= 1702
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180281693
WAIT #139732971106048: nam='cell smart table scan' ela= 18
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180281843
WAIT #139732971106048: nam='cell smart table scan' ela= 2252
cellhash#=3097415979 p2=0 p3=0 obj#=20476 tim=610180284151
WAIT #139732971106048: nam='cell smart table scan' ela= 20
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180284255
WAIT #139732971106048: nam='cell smart table scan' ela= 734
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180285029
WAIT #139732971106048: nam='cell smart table scan' ela= 398
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180286970
WAIT #139732971106048: nam='cell smart table scan' ela= 5
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180287012
WAIT #139732971106048: nam='cell smart table scan' ela= 6737
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180293842
WAIT #139732971106048: nam='cell smart table scan' ela= 24
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180294037
WAIT #139732971106048: nam='cell smart table scan' ela= 2252
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180296392
WAIT #139732971106048: nam='cell smart table scan' ela= 16
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180296535
WAIT #139732971106048: nam='cell smart table scan' ela= 4280
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180300845
WAIT #139732971106048: nam='cell smart table scan' ela= 2790
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180303820
WAIT #139732971106048: nam='cell smart table scan' ela= 18
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180303932
WAIT #139732971106048: nam='cell smart table scan' ela= 9455
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180313415
WAIT #139732971106048: nam='cell smart table scan' ela= 972
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180314524
WAIT #0: nam='cell smart table scan' ela= 525
cellhash#=2877408282 p2=0 p3=0 obj#=20476 tim=610180316862
WAIT #0: nam='cell smart table scan' ela= 644
cellhash#=3097415979 p2=0 p3=0 obj#=20476 tim=610180317537
WAIT #0: nam='cell smart table scan' ela= 622
cellhash#=2271898598 p2=0 p3=0 obj#=20476 tim=610180318201
[oracle@qr01dbadm01 trace]$
```

20. Exit your terminal session.

Practices for Lesson 10: Consolidation Options and Recommendations

Chapter 10

Practices for Lesson 10

There are no practices for this lesson.

Practices for Lesson 11: Migrating Databases to Exadata Database Machine

Chapter 11

Practices for Lesson 11: Overview

Practices Overview

In this practice you will use Oracle Recovery Manager (RMAN), in conjunction with the transportable tablespace feature of Oracle Database, to migrate data from a big endian platform to Database Machine, a little endian platform.

Practice 11-1: Migrating to Databases Machine using Transportable Tablespaces

Overview

In this practice, you will migrate data from an AIX platform to Database Machine. You will use Oracle Recovery Manager (RMAN) to perform endian conversion on a data file and also to load the data file into ASM. You will then use the transportable tablespace feature of Oracle Database to make the data file part of your database on Database Machine.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. From the `oracle` user home directory, change directory into the `TTS` subdirectory under the `labs` directory.

```
[oracle@qr01dbadm01 ~]$ cd labs/TTS
[oracle@qr01dbadm01 TTS]$
```

3. List the contents of the `TTS` directory. You should see two files. `soe_TTS_AIX.dbf` is a data file sourced from an Oracle database running on AIX. It contains a tablespace called `SOE` which houses numerous database objects belonging to a schema named `SOE`. `expSOE_TTS.dmp` is a data pump export file, which contains the transportable tablespace metadata associated with the `SOE` tablespace.

```
[oracle@qr01dbadm01 TTS]$ ls -l
total 1230568
-rw-r--r-- 1 oracle oinstall     565248 Oct 15  2010 expSOE_TTS.dmp
-rw-r--r-- 1 oracle oinstall 1258299392 Oct 15  2010 soe_TTS_AIX.dbf
[oracle@qr01dbadm01 TTS]$
```

Note that the files associated with the transportable tablespace are staged on the filesystem of your database server. This arrangement is being used only to facilitate the practice in this classroom setting and is not a recommended practice for migrating data to Database Machine. Various solutions exist for staging data files in a real-world Database Machine setting, including using a database file system (DBFS). Configuring DBFS on Database Machine is examined in the lesson entitled *Bulk Data Loading with Database Machine*.

4. Connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 TTS]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

5. Execute the following query to confirm the Database Machine platform and endian format (or execute the SQL script `/home/oracle/labs/lab11-01-05.sql`). Note that Database Machine is a little endian format system.

```
SQL> select d.platform_name, endian_format
  2  from v$transportable_platform tp, v$database d
  3  where tp.platform_name = d.platform_name;

PLATFORM_NAME
--------------------------------------------------------------------------------
ENDIAN_FORMAT
--------------
Linux x86 64-bit
Little


SQL>
```

6. Query V$TRANSPORTABLE_PLATFORM to display all the platforms supported by the transportable tablespace feature along with each platform's endian format. Note that AIX is a big endian format system.

```
SQL> col platform_name format a40
SQL> select * from v$transportable_platform;

PLATFORM_ID PLATFORM_NAME                            ENDIAN_FORMAT      CON_ID
----------- ---------------------------------------- -------------- ----------
          1 Solaris[tm] OE (32-bit)                  Big                     0
          2 Solaris[tm] OE (64-bit)                  Big                     0
          7 Microsoft Windows IA (32-bit)            Little                  0
         10 Linux IA (32-bit)                        Little                  0
          6 AIX-Based Systems (64-bit)               Big                     0
          3 HP-UX (64-bit)                           Big                     0
          5 HP Tru64 UNIX                            Little                  0
          4 HP-UX IA (64-bit)                        Big                     0
         11 Linux IA (64-bit)                        Little                  0
         15 HP Open VMS                              Little                  0
          8 Microsoft Windows IA (64-bit)            Little                  0

PLATFORM_ID PLATFORM_NAME                            ENDIAN_FORMAT      CON_ID
----------- ---------------------------------------- -------------- ----------
          9 IBM zSeries Based Linux                  Big                     0
         13 Linux x86 64-bit                         Little                  0
         16 Apple Mac OS                             Big                     0
         12 Microsoft Windows x86 64-bit             Little                  0
         17 Solaris Operating System (x86)           Little                  0
         18 IBM Power Based Linux                    Big                     0
         19 HP IA Open VMS                           Little                  0
         20 Solaris Operating System (x86-64)        Little                  0
         21 Apple Mac OS (x86-64)                    Little                  0

20 rows selected.

SQL>
```

7. Exit your SQL*Plus session. Then, launch Oracle Recovery Manager (RMAN) and connect to your database as shown below:

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0...
[oracle@qr01dbadm01 TTS]$ rman target /

Recovery Manager: Release 12.1.0.2.0 - Production...

connected to target database: DBM (DBID=1355636031)

RMAN>
```

8. Use the `CONVERT DATAFILE` command as shown below to perform endian conversion on the AIX-based Oracle data file in your `TTS` directory. The command also loads the data file into ASM. Take note of the ASM file name for your converted data file.

```
RMAN> convert datafile
2> '/home/oracle/labs/TTS/soe_TTS_AIX.dbf'
3> to platform="Linux x86 64-bit"
4> from platform="AIX-Based Systems (64-bit)"
5> parallelism=1
6> format '+DATA_QR01';

Starting conversion at target at 09-JUN-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=43 instance=dbm1 device type=DISK
channel ORA_DISK_1: starting datafile conversion
input file name=/home/oracle/labs/TTS/soe_TTS_AIX.dbf
converted datafile=+DATA_QR01/DBM/DATAFILE/soe.274.914043297
channel ORA_DISK_1: datafile conversion complete, elapsed time:
00:02:15
Finished conversion at target at 09-JUN-16

RMAN>
```

9. Exit RMAN and connect to your database using SQL*Plus. Log in as the database administrator.

```
RMAN> exit

Recovery Manager complete.
[oracle@qr01dbadm01 TTS]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production ...

SQL>
```

10. When a transportable tablespace is mapped into a destination database, by default, all the database objects (such as tables and indexes) are created in the same user schema as in the source database, and those users must already exist in the destination database. Create a user named `SOE` and grant `SOE` the `CONNECT` and `RESOURCE` roles. Remember your chosen password as you will need to use it later in the practice.

```
SQL> create user soe identified by ??? account unlock;

User created.

SQL> grant connect,resource to soe;

Grant succeeded.

SQL>
```

11. Create a directory object that references the directory which houses your transportable tablespace export file. After the directory object is created, exit your SQL*Plus session.

```
SQL> create directory tts as '/home/oracle/labs/TTS';

Directory created.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[oracle@qr01dbadm01 TTS]$
```

12. Execute the data pump import utility (`impdp`) to import the transportable tablespace metadata. Use the ASM file name you obtained in step 8 in your setting for the `transport_datafiles` parameter.

```
[oracle@qr01dbadm01 TTS]$ impdp system dumpfile=expSOE_TTS.dmp directory=tts
logfile=imp_SOE.log
transport_datafiles='+DATA_QR01/DBM/DATAFILE/soe.274.914043297'

Import: Release 12.1.0.2.0 - Production on Thu Jun 9 05:01:27 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.
Password: ????????

Connected to: Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage
Management, OLAP,
Advanced Analytics and Real Application Testing options
Master table "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully
loaded/unloaded
Source time zone version is 14 and target time zone version is 18.
import done in AL32UTF8 character set and AL16UTF16 NCHAR character set
export done in US7ASCII character set and AL16UTF16 NCHAR character set
Starting "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01":  system/********
dumpfile=expSOE_TTS.dmp directory=tts logfile=imp_SOE.log
transport_datafiles=+DATA_QR01/DBM/DATAFILE/soe.274.914043297
Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
Processing object type TRANSPORTABLE_EXPORT/TABLE
Processing object type TRANSPORTABLE_EXPORT/INDEX
Processing object type TRANSPORTABLE_EXPORT/CONSTRAINT/CONSTRAINT
Processing object type TRANSPORTABLE_EXPORT/INDEX_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/CONSTRAINT/REF_CONSTRAINT
Processing object type TRANSPORTABLE_EXPORT/INDEX/FUNCTIONAL_AND_BITMAP/INDEX
Processing object type
TRANSPORTABLE_EXPORT/INDEX/STATISTICS/FUNCTIONAL_AND_BITMAP/INDEX_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/TABLE_STATISTICS
Processing object type TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
Job "SYSTEM"."SYS_IMPORT_TRANSPORTABLE_01" successfully completed at Thu Jun 9
05:02:12 2016 elapsed 0 00:00:35

[oracle@qr01dbadm01 TTS]$
```

13. Transportable tablespaces are generated in read-only mode. Connect to your database by using SQL*Plus. Log in as the database administrator and restore your newly migrated `SOE` tablespace to read/write mode.

```
[oracle@qr01dbadm01 TTS]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL> alter tablespace soe read write;

Tablespace altered.

SQL>
```

14. Connect as the newly created `SOE` user.

```
SQL> connect soe/???
Connected.
SQL>
```

15. Query the data dictionary (or execute the SQL script `/home/oracle/labs/lab11-01-15.sql`) to view a summary of the database objects belonging to `SOE`.

```
SQL> select segment_type,tablespace_name,count(*)
  2  from user_segments
  3  group by segment_type,tablespace_name;

SEGMENT_TYPE       TABLESPACE_NAME                  COUNT(*)
------------------ ------------------------------ ----------
TABLE PARTITION    SOE                                    64
INDEX              SOE                                    19
TABLE              SOE                                     3
INDEX PARTITION    SOE                                    64

SQL>
```

16. Query some of the migrated data to confirm that it is available. You have completed the migration of a tablespace to Database Machine.

```
SQL> select * from warehouses;

WAREHOUSE_ID WAREHOUSE_NAME                      LOCATION_ID
------------ ----------------------------------- -----------
           1 Southlake, Texas                           1400
           2 San Francisco                              1500
           3 New Jersey                                 1600
           4 Seattle, Washington                        1700
           5 Toronto                                    1800
           6 Sydney                                     2200
           7 Mexico City                                3200
           8 Beijing                                    2000
           9 Bombay                                     2100
          10 Paris                                      3240
          11 Warehouse Number 11                        1252

WAREHOUSE_ID WAREHOUSE_NAME                      LOCATION_ID
------------ ----------------------------------- -----------
          12 Warehouse Number 12                        9176
          13 Warehouse Number 13                        3766
          14 Warehouse Number 14                        3766
          15 Warehouse Number 15                        3766
          16 Warehouse Number 16                        3766
          17 Warehouse Number 17                        8971
          18 Warehouse Number 18                        8971
          19 Warehouse Number 19                        8971
          20 Warehouse Number 20                        8971

20 rows selected.

SQL>
```

17. Exit your SQL*Plus session.

Practices for Lesson 12: Bulk Data Loading

Chapter 12

Practices for Lesson 12: Overview

Practices Overview

In this practice, you will perform a bulk data load on Database Machine.

Practice 12-1: Bulk Data Loading with Database Machine

Overview

In this practice, you will perform a bulk data load on Database Machine. You will configure a database file system (DBFS) and use it to stage a CSV formatted file. You will then use the external table feature of Oracle Database to reference the CSV file. Finally, you will use a `CREATE TABLE AS SELECT` statement to copy the CSV file data into a table in your database.

Note that this practice familiarizes students with the process required to configure DBFS in an Exadata environment using Linux as the database server OS, and some of the tasks performed during this practice are Linux-specific. Furthermore, this practice does not contain all of the optional steps for configuring DBFS on Exadata Database Machine. For information on additional configuration options see My Oracle Support note 1054431.1.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `root` user.

```
$ ssh root@qr01dbadm01
root@qr01dbadm01's password: ????????
[root@qr01dbadm01 ~]#
```

2. Configure the OS to enable the `oracle` user to use the Linux `fuse` (Filesystem in User Space) kernel module.

```
[root@qr01dbadm01 ~]# usermod -a -G fuse oracle
[root@qr01dbadm01 ~]# echo user_allow_other > /etc/fuse.conf
[root@qr01dbadm01 ~]# chmod 644 /etc/fuse.conf
[root@qr01dbadm01 ~]#
```

Note that the configuration performed in this step is required on every database server where the DBFS will be mounted. In a production environment, you could repeat this step on each database server or use the `dcli` utility to perform the configuration at once across multiple database servers.

3. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance). **It is important that you create a fresh `oracle` user session after you perform step 2 so that the session can inherit the newly configured group membership. If you use a previously started terminal session then you will experience problems later in the practice.**

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

4. Connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

5. Create a new tablespace to house your database file system (DBFS).

```
SQL> create bigfile tablespace dbfs
  2  datafile '+DBFS_DG' size 200M;

Tablespace created.

SQL>
```

Note that in this practice you will configure DBFS inside your assigned student database. This arrangement is only being used to facilitate the practice in this classroom setting and is not a recommended practice for configuring DBFS to facilitate bulk data loading on Database Machine. For bulk data loading in a real-world Database Machine setting, Oracle recommends that you create a separate dedicated database instance for DBFS on your Database Machine. See My Oracle Support notes 1054431.1 and 1191144.1 for specific recommendations.

6. Create a new database user to support DBFS and grant the database user the required system privileges and roles as shown below. Remember your chosen password as you will need to use it later in the practice. Exit your SQL*Plus session when you are finished.

```
SQL> create user dbfs identified by ????
  2  quota unlimited on dbfs;

User created.

SQL> grant create session, create table,
  2  create procedure, dbfs_role to dbfs;

Grant succeeded.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[oracle@qr01dbadm01 ~]$
```

7. Create a new directory named `DBFS` under the `oracle` user home directory. This directory will act as the anchor for your database file system mount point. Note that this directory needs to be created on every database server where the DBFS will be mounted.

```
[oracle@qr01dbadm01 ~]$ mkdir DBFS
[oracle@qr01dbadm01 ~]$
```

8. Change directory to `$ORACLE_HOME/rdbms/admin`.

```
[oracle@qr01dbadm01 ~]$ cd $ORACLE_HOME/rdbms/admin
[oracle@qr01dbadm01 admin]$
```

9. Connect to your database with SQL*Plus. Log in as the newly created `dbfs` user.

```
[oracle@qr01dbadm01 admin]$ sqlplus dbfs/????

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

10. Run the `dbfs_create_filesystem` script file to create the database objects for your DBFS store. The first parameter (`dbfs`) specifies the tablespace where the DBFS store is created. The second parameter (`mydbfs`) specifies the name of the DBFS store.

```
SQL> @dbfs_create_filesystem dbfs mydbfs
...
No errors.
--------
CREATE STORE:
begin dbms_dbfs_sfs.createFilesystem(store_name => 'mydbfs', tbl_name =>
'mydbfs', tbl_tbs => 'dbfs', lob_tbs => 'dbfs', do_partition => false,
partition_key => 1, do_compress => false, compression => '', do_dedup =>
false,
do_encrypt => false); end;
--------
REGISTER STORE:
begin dbms_dbfs_content.registerStore(store_name=> 'mydbfs', provider_name =>
'sample1', provider_package => 'dbms_dbfs_sfs'); end;
--------
MOUNT STORE:
begin dbms_dbfs_content.mountStore(store_name=>'mydbfs',
store_mount=>'mydbfs');
end;
--------
CHMOD STORE:
declare m integer; begin m := dbms_fuse.fs_chmod('/mydbfs', 16895); end;
No errors.
SQL>
```

11. Exit SQL*Plus and change back to the `oracle` user's home directory.

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[oracle@qr01dbadm01 admin]$ cd
[oracle@qr01dbadm01 ~]$
```

12. Create a file named passwd.txt, which contains the password for your DBFS database user.

```
[oracle@qr01dbadm01 ~]$ echo ???? > passwd.txt
[oracle@qr01dbadm01 ~]$
```

13. Launch the DBFS client (dbfs_client) by using the following command. Running dbfs_client in this manner mounts your database file system.

```
[oracle@qr01dbadm01 ~]$ nohup $ORACLE_HOME/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS < passwd.txt
&
[1] 5376
nohup: appending output to `nohup.out'

[oracle@qr01dbadm01 ~]$
```

Note that you have just started DBFS using the simplest and most direct method. Optional configuration steps can be performed to enable DBFS to use a wallet-based password and to enable DBFS to be managed automatically as a cluster resource. See My Oracle Support note 1054431.1 for details.

14. Use the ps command to locate your dbfs_client process.

```
[oracle@qr01dbadm01 ~]$ ps -ef | grep dbfs_client
oracle     5376 129147  0 05:46 pts/1    00:00:00
/u01/app/oracle/product/12.1.0.2/dbhome_1/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS
oracle     5580 129147  0 05:46 pts/1    00:00:00 grep
dbfs_client
[oracle@qr01dbadm01 ~]$
```

15. Use the df command to confirm that your database file system is mounted.

```
[oracle@qr01dbadm01 ~]$ df /home/oracle/DBFS
Filesystem      1K-blocks  Used Available Use% Mounted on
dbfs-dbfs@dbm:/    198592   160    198432   1% /home/oracle/DBFS
[oracle@qr01dbadm01 ~]$
```

16. Normally you would transfer files into your DBFS staging area by using a network file transfer mechanism such as SCP or FTP, or by reading them off a mass storage device (such as magnetic tape for example) attached to your Database Machine. To facilitate this practice, a CSV formatted data file is already located in the labs/CSV directory under the oracle user home directory. Copy the CSV file into your DBFS staging area as shown below. Then change directory into your DBFS staging area and confirm the presence of the CSV data file.

```
[oracle@qr01dbadm01 ~]$ cp labs/CSV/customers.csv DBFS/mydbfs/
[oracle@qr01dbadm01 ~]$ cd DBFS/mydbfs/
[oracle@qr01dbadm01 mydbfs]$ ls -l
total 7376
-rw-r--r-- 1 oracle oinstall 7552705 Jun  9 05:48 customers.csv
[oracle@qr01dbadm01 mydbfs]$
```

17. Use the `head` command to show the first 10 lines of data inside the CSV file.

```
[oracle@qr01dbadm01 mydbfs]$ head customers.csv
55,"Bruce","Lange","hi","INDIA",1999,"Bruce.Lange@oracle.com",166
74,"Claude","Brown","d","SWITZERLAND",4737,"Claude.Brown@oracle.com",163
81,"Max","Capshaw","us","AMERICA",650,"Max.Capshaw@oracle.com",165
97,"Roy","Kazan","zhs","CHINA",2707,"Roy.Kazan@oracle.com",156
111,"Robert","Young","i","ITALY",4895,"Robert.Young@oracle.com",169
146,"Ridley","Schneider","th","THAILAND",3775,"Ridley.Schneider@oracle.com",16
8
245,"Matthias","Russell","i","ITALY",2839,"Matthias.Russell@oracle.com",158
252,"Edward","Hoskins","ja","JAPAN",2469,"Edward.Hoskins@oracle.com",153
262,"Grace","Sen","zhs","CHINA",500,"Grace.Sen@oracle.com",162
281,"Gtz","Peckinpah","th","THAILAND",1128,"Gtz.Peckinpah@oracle.com",151
[oracle@qr01dbadm01 mydbfs]$
```

Your CSV data file is now staged inside DBFS. In the next section of this practice, you will create an external table to reference the staged data.

18. Connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 mydbfs]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

19. Create a directory object that points to your DBFS staging directory.

```
SQL> create directory staging as '/home/oracle/DBFS/mydbfs';

Directory created.

SQL>
```

20. Grant read and write permissions on your newly created staging directory object to the `sh` user.

```
SQL> grant read, write on directory staging to sh;

Grant succeeded.

SQL>
```

21. Connect as the `sh` user.

```
SQL> connect sh/??
Connected.
SQL>
```

22. Create an external table that references the data in your DBFS-staged CSV data file. You can use the command shown below without any modifications (or execute the SQL script `/home/oracle/labs/lab12-01-22.sql`).

```
SQL> create table ext_customers
(
  customer_id      number(12),
  cust_first_name  varchar2(30),
  cust_last_name   varchar2(30),
  nls_language     varchar2(3),
  nls_territory    varchar2(30),
  credit_limit     number(9,2),
  cust_email       varchar2(100),
  account_mgr_id   number(6)
)
organization external
(
  type oracle_loader
  default directory staging
  access parameters
  (
    records delimited by newline
    badfile staging:'custxt%a_%p.bad'
    logfile staging:'custxt%a_%p.log'
    fields terminated by ',' optionally enclosed by '"'
    missing field values are null
    (
      customer_id, cust_first_name, cust_last_name, nls_language,
      nls_territory, credit_limit, cust_email, account_mgr_id
    )
  )
  location ('customers.csv')
)
parallel
reject limit unlimited;

Table created.

SQL>
```

23. Configure your session to display query execution plans.

```
SQL> set autotrace on explain
SQL>
```

24. Execute the following query to display the number of records in the external table. Note that the full table scan of the external table is executed in parallel.

```
SQL> select count(*) from ext_customers;

  COUNT(*)
----------
    100000


Execution Plan
----------------------------------------------------------
Plan hash value: 3054877561

------------------------------------------------------------------------------
| Id  | Operation                         | Name           | Rows  | Cost (%CPU)|
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                  |                |     1 |    95   (2)|
|   1 |  SORT AGGREGATE                   |                |     1 |            |
|   2 |   PX COORDINATOR                  |                |       |            |
|   3 |    PX SEND QC (RANDOM)            | :TQ10000       |     1 |            |
|   4 |     SORT AGGREGATE                |                |     1 |            |
|   5 |      PX BLOCK ITERATOR            |                |  102K|    95   (2)|
|   6 |       EXTERNAL TABLE ACCESS FULL| EXT_CUSTOMERS |  102K|    95   (2)|
------------------------------------------------------------------------------



Note
-----
   - Degree of Parallelism is 4 because of table property

SQL>
```

25. Reconfigure your session to disable automatic query plan display.

```
SQL> set autotrace off
SQL>
```

26. Use a `CREATE TABLE AS SELECT` command to load the external table data contained in the CSV file into a new table in your database.

```
SQL> create table loaded_customers
  2  as select * from ext_customers;

Table created.

SQL>
```

27. Query your newly loaded table to confirm the number of records that were loaded and then exit SQL*Plus.

```
SQL> select count(*) from loaded_customers;

  COUNT(*)
----------
    100000

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production...
[oracle@qr01dbadm01 mydbfs]$
```

28. List the contents of your current directory (which should be your DBFS staging area). In addition to your CSV data file, you should now see a log file that was generated when you referenced the external table.

```
[oracle@qr01dbadm01 mydbfs]$ ls -l
total 7379
-rw-r--r-- 1 oracle oinstall 7552705 Jun  9 05:48 customers.csv
-rw-r--r-- 1 oracle asmadmin    2752 Jun  9 05:53 custxt000_8277.log
[oracle@qr01dbadm01 mydbfs]$
```

29. Examine the contents of the log file.

```
[oracle@qr01dbadm01 mydbfs]$ cat custxt000_8277.log
...
 LOG file opened at 06/09/16 05:58:24

Field Definitions for table EXT_CUSTOMERS
  Record format DELIMITED BY NEWLINE
  Data in file has same endianness as the platform
  Rows with all null fields are accepted

  Fields in Data Source:

    CUSTOMER_ID                     CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    CUST_FIRST_NAME                 CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    CUST_LAST_NAME                  CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    NLS_LANGUAGE                    CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    NLS_TERRITORY                   CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    CREDIT_LIMIT                    CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    CUST_EMAIL                      CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
    ACCOUNT_MGR_ID                  CHAR (255)
      Terminated by ","
      Enclosed by """ and """
      Trim whitespace same as SQL Loader
[oracle@qr01dbadm01 mydbfs]$
```

30. Change back to the `oracle` user's home directory.

```
[oracle@qr01dbadm01 mydbfs]$ cd
[oracle@qr01dbadm01 ~]$
```

31. Use the `fusermount -u` command to unmount your database file system.

```
[oracle@qr01dbadm01 ~]$ fusermount -u /home/oracle/DBFS
[1]+  Done                    nohup $ORACLE_HOME/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS < passwd.txt
[oracle@qr01dbadm01 ~]$
```

32. Unmounting your database file system terminates the `dbfs_client` process you started earlier. Use the `ps` command to verify that your `dbfs_client` process is no longer running.

```
[oracle@qr01dbadm01 ~]$ ps -ef|grep dbfs_client
oracle    18083 129147  0 06:33 pts/1    00:00:00 grep dbfs_client
[oracle@qr01dbadm01 ~]$
```

What do you think would happen if you referenced the external table (`EXT_CUSTOMERS`) now? Try it.

So far, you have mounted your DBFS exclusively on `qr01dbadm01`. In the final part of this practice you will mirror the configuration on `qr01dbadm02` and mount DBFS on both of your database servers.

33. Establish a separate terminal connection to `qr01dbadm02` as the `root` user.

```
$ ssh root@qr01dbadm02
root@qr01dbadm02's password: ????????
[root@qr01dbadm02 ~]#
```

34. Configure the OS to enable the `oracle` user to use the Linux `fuse` (Filesystem in User Space) kernel module.

```
[root@qr01dbadm02 ~]# usermod -a -G fuse oracle
[root@qr01dbadm02 ~]# echo user_allow_other > /etc/fuse.conf
[root@qr01dbadm02 ~]# chmod 644 /etc/fuse.conf
[root@qr01dbadm02 ~]#
```

35. Establish a new shell as the `oracle` OS user and create a new directory named `DBFS` to act as the anchor for your database file system mount point.

```
[root@qr01dbadm02 ~]# su - oracle
[oracle@qr01dbadm02 ~]$ mkdir DBFS
[oracle@qr01dbadm02 ~]$
```

36. Create a file named `passwd.txt`, which contains the password for your DBFS database user.

```
[oracle@qr01dbadm02 ~]$ echo ???? > passwd.txt
[oracle@qr01dbadm02 ~]$
```

37. Configure your OS environment to access the dbm database.

```
[oracle@qr01dbadm02 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm02 ~]$
```

38. Mount your database file system on qr01dbadm02.

```
[oracle@qr01dbadm02 ~]$ nohup $ORACLE_HOME/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS < passwd.txt
&
[1] 81229
nohup: appending output to `nohup.out'

[oracle@qr01dbadm02 ~]$
```

39. Confirm the contents of the file system.

```
[oracle@qr01dbadm02 ~]$ ls -l DBFS/mydbfs
total 7379
-rw-r--r-- 1 oracle oinstall 7552705 Jun  9 05:48 customers.csv
-rw-r--r-- 1 oracle asmadmin    2752 Jun  9 05:53 custxt000_8277.log
[oracle@qr01dbadm02 ~]$
```

40. Create a new file in the file system.

```
[oracle@qr01dbadm02 ~]$ echo "Test" > DBFS/mydbfs/test.txt
[oracle@qr01dbadm02 ~]$
```

41. Return to your qr01dbadm01 session and mount the database file system.

```
[oracle@qr01dbadm01 ~]$  nohup $ORACLE_HOME/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS < passwd.txt
&
[1] 34148
nohup: appending output to `nohup.out'

[oracle@qr01dbadm01 ~]$
```

42. Confirm the contents of the file system, including the file you created on qr01dbadm02.

```
[oracle@qr01dbadm01 ~]$ ls -l DBFS/mydbfs
total 7380
-rw-r--r-- 1 oracle oinstall 7552705 Jun  9 05:48 customers.csv
-rw-r--r-- 1 oracle asmadmin    2752 Jun  9 05:53 custxt000_8277.log
-rw-r--r-- 1 oracle oinstall       5 Jun  9 07:05 test.txt
[oracle@qr01dbadm01 ~]$ cat DBFS/mydbfs/test.txt
Test
[oracle@qr01dbadm01 ~]$
```

43. Unmount your database file system on `qr01dbadm01`.

```
[oracle@qr01dbadm01 ~]$ fusermount -u /home/oracle/DBFS
[1]+  Done                    nohup $ORACLE_HOME/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS < passwd.txt
[oracle@qr01dbadm01 ~]$
```

44. Unmount your database file system on `qr01dbadm02`.

```
[oracle@qr01dbadm02 ~]$ fusermount -u /home/oracle/DBFS
[1]+  Done                    nohup $ORACLE_HOME/bin/dbfs_client
dbfs@dbm -o allow_other,direct_io /home/oracle/DBFS < passwd.txt
[oracle@qr01dbadm02 ~]$
```

45. Exit your terminal sessions.

Practices for Lesson 13: Exadata Database Machine Platform Monitoring - Introduction

Chapter 13

Practices for Lesson 13

There are no practices for this lesson.

Practices for Lesson 14: Configuring Enterprise Manager Cloud Control to Monitor Exadata Database Machine

Chapter 14

Practices for Lesson 14: Overview

Practices Overview

In these practices, you will configure Enterprise Manager Cloud Control 12*c* to monitor Exadata Database Machine. You will also perform a selection of post-discovery configuration and verification tasks.

Practice 14-1: Configuring Enterprise Manager Cloud Control 12*c* to Monitor Exadata Database Machine

Overview

In this practice, you will configure Enterprise Manager Cloud Control 12*c* to monitor Exadata Database Machine. Specifically, you will:

- Deploy Management Agents to Database Machine servers
- Discover the Database Machine system components
- Discover Oracle Grid Infrastructure and Oracle Database software components
- Configure a Database Machine Services Dashboard

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the oracle user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Confirm that the `dbm` database is running on `qr01dbadm01` and `qr01dbadm02`.

```
[oracle@qr01dbadm01 ~]$ srvctl status database -d dbm
Instance dbm1 is running on node qr01dbadm01
Instance dbm2 is running on node qr01dbadm02
[oracle@qr01dbadm01 ~]$
```

3. Establish a terminal session connected to `em12c` by using the `oracle` OS user. Ensure that you specify the `–X` option for `ssh`.

   Note that you may see additional messages relating to server identities. Answer `yes` if you are prompted to acknowledge server authenticity.

```
$ ssh –X oracle@em12c
oracle@em12c password: ????????
[oracle@em12c ~]$
```

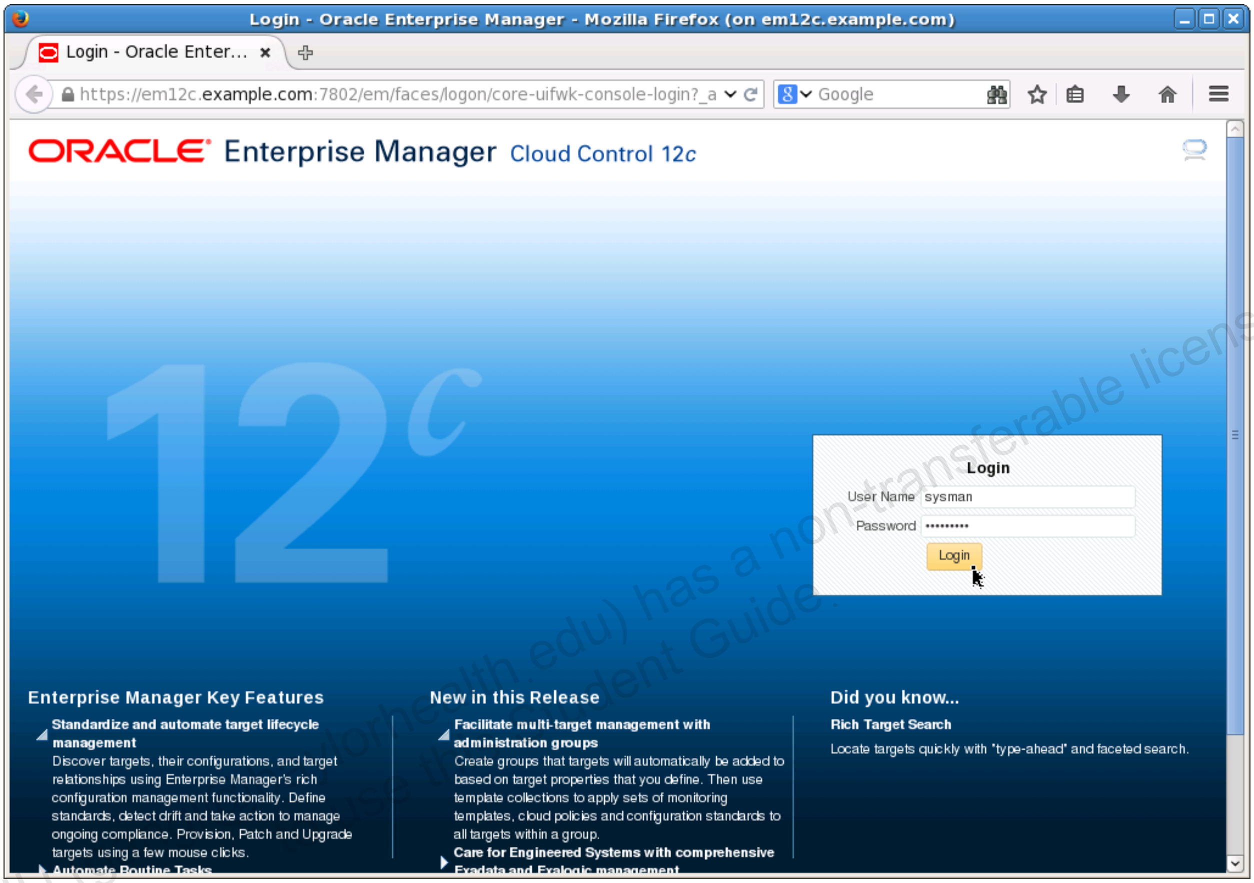
4. Verify that your Enterprise Manager environment is up and running.

```
[oracle@em12c ~]$ /u01/app/oracle/product/middleware/oms/bin/emctl status oms
Oracle Enterprise Manager Cloud Control 12c Release 4
Copyright (c) 1996, 2014 Oracle Corporation.  All rights reserved.
WebTier is Up
Oracle Management Server is Up
[oracle@em12c ~]$
```

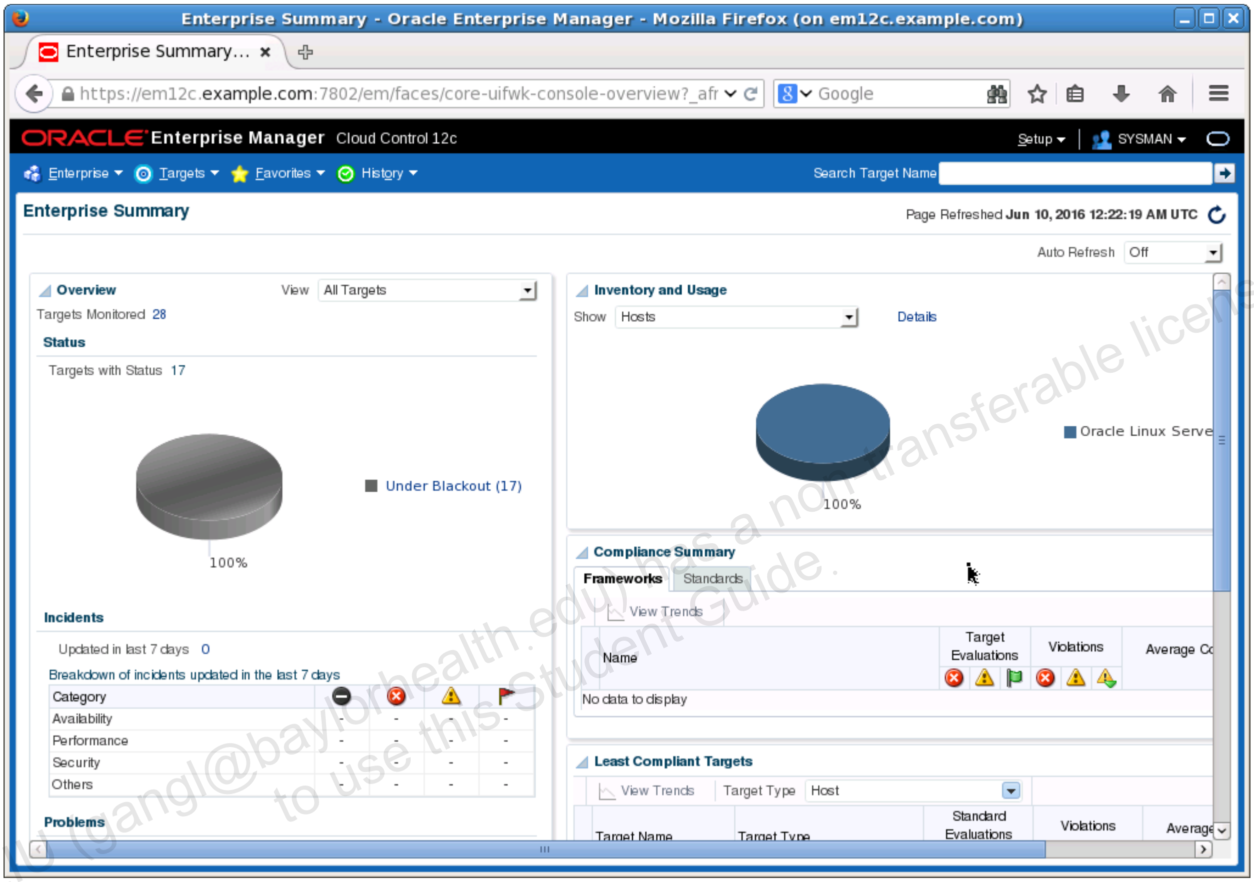
5. Start the Firefox web browser.

```
[oracle@em12c ~]$ firefox &
[1] 4953
```

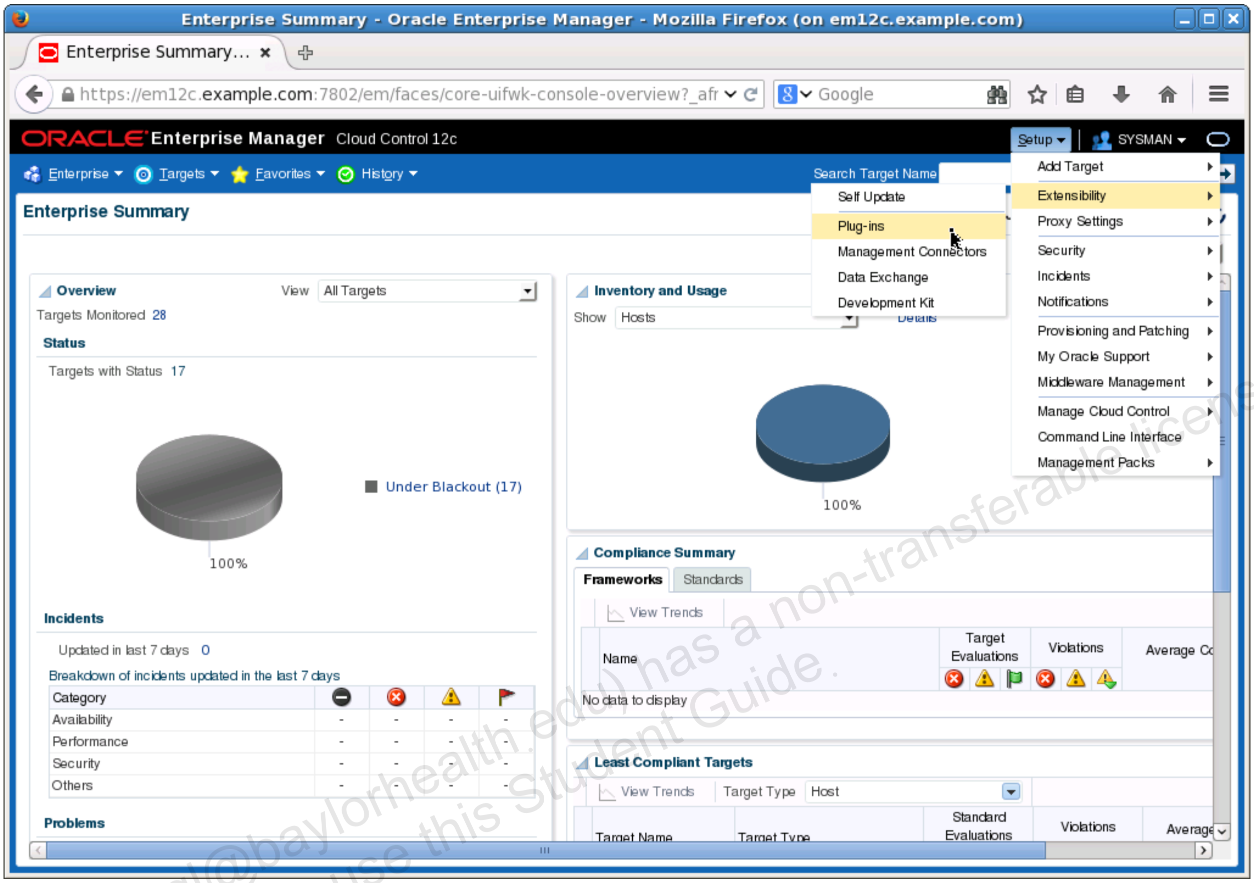
6. Log in to Enterprise Manager Cloud Control 12*c* as the `sysman` user.



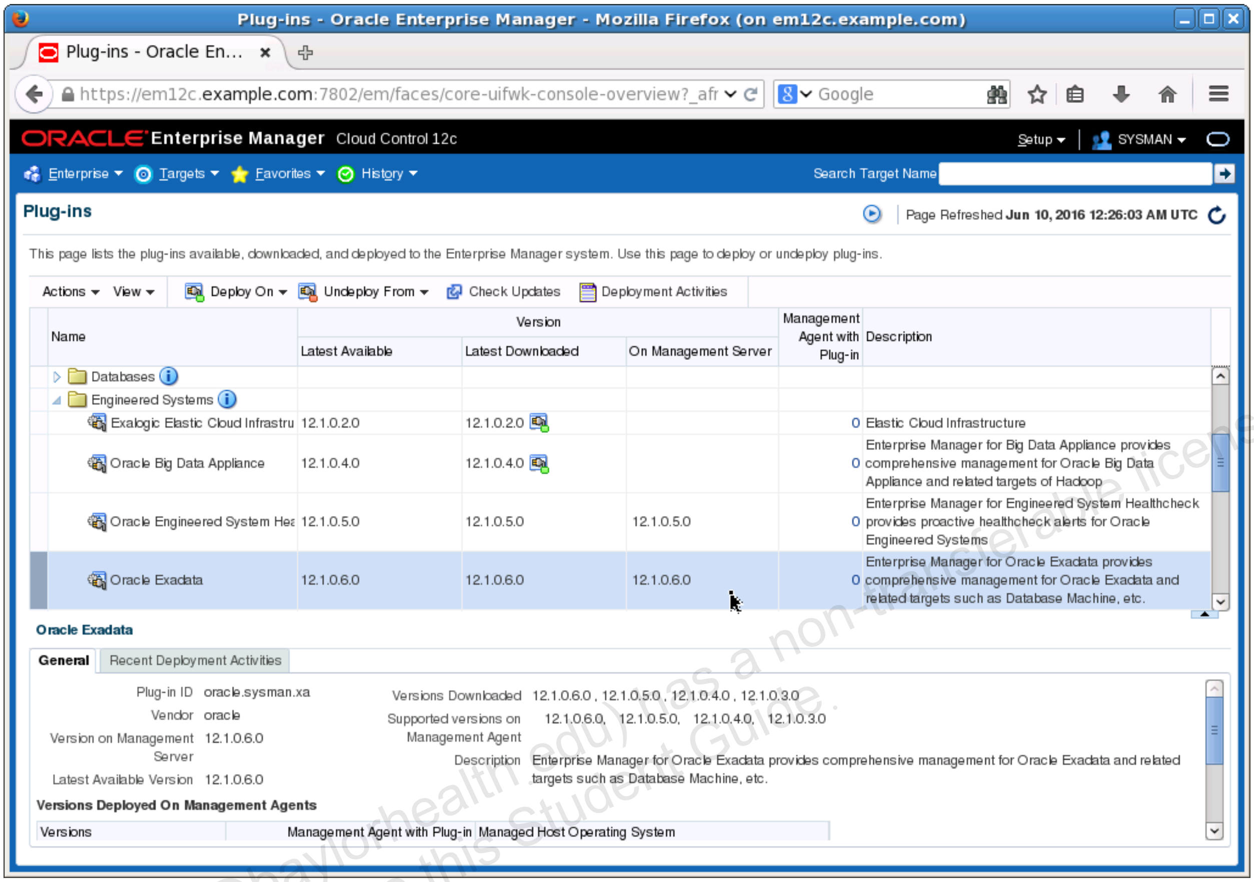
7. When the Enterprise Summary page appears, you should notice that all the currently defined targets are under blackout. These targets relate to the Enterprise Manager Cloud Control 12*c* environment that you are using and they have been intentionally blacked out to eliminate the metric collection overhead associated with them. Apart from this change, the Enterprise Manager Cloud Control 12*c* environment that you are using is essentially the same as a fresh installation.



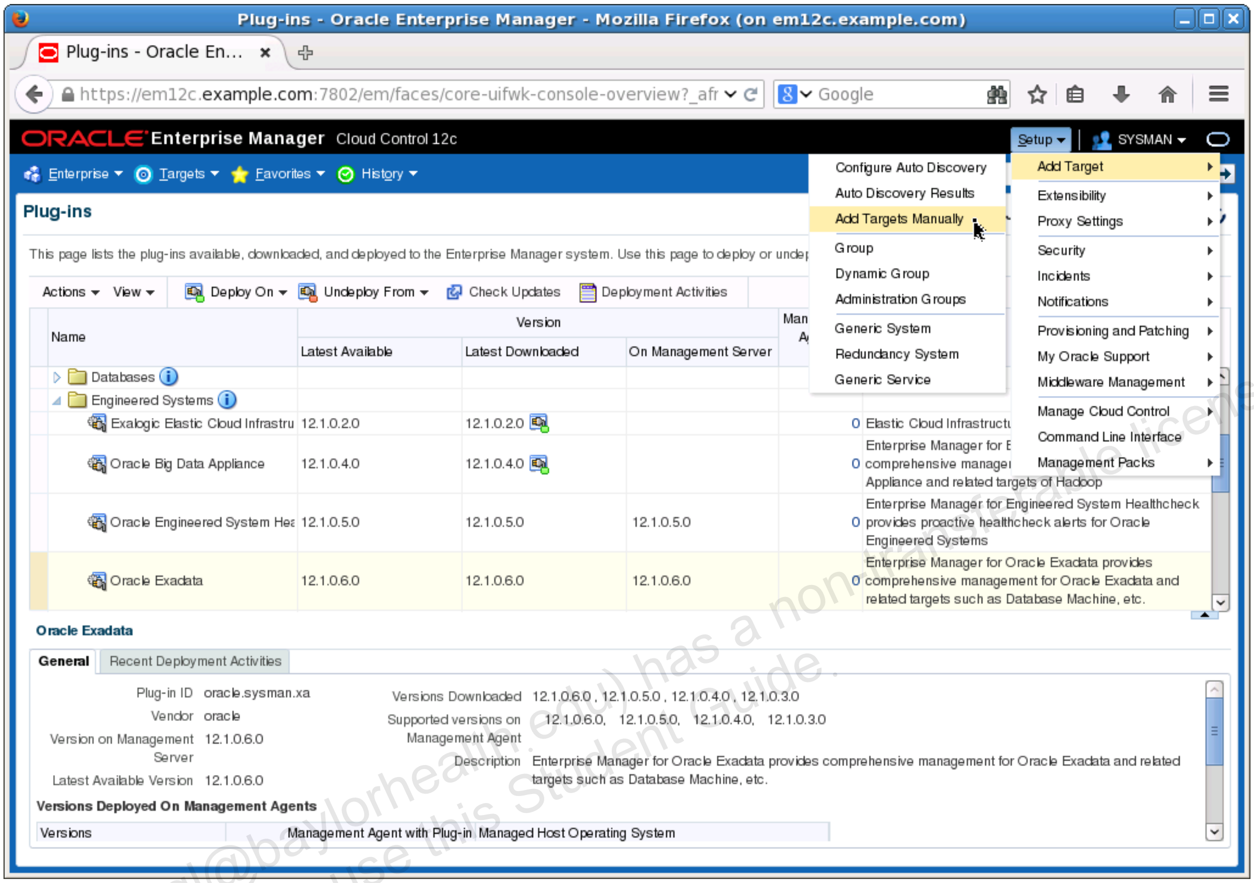
8. Before you commence configuring Cloud Control to monitor Exadata, confirm that the Exadata plug-in is available and ready to use. Select the Setup > Extensibility > Plug-ins menu command.



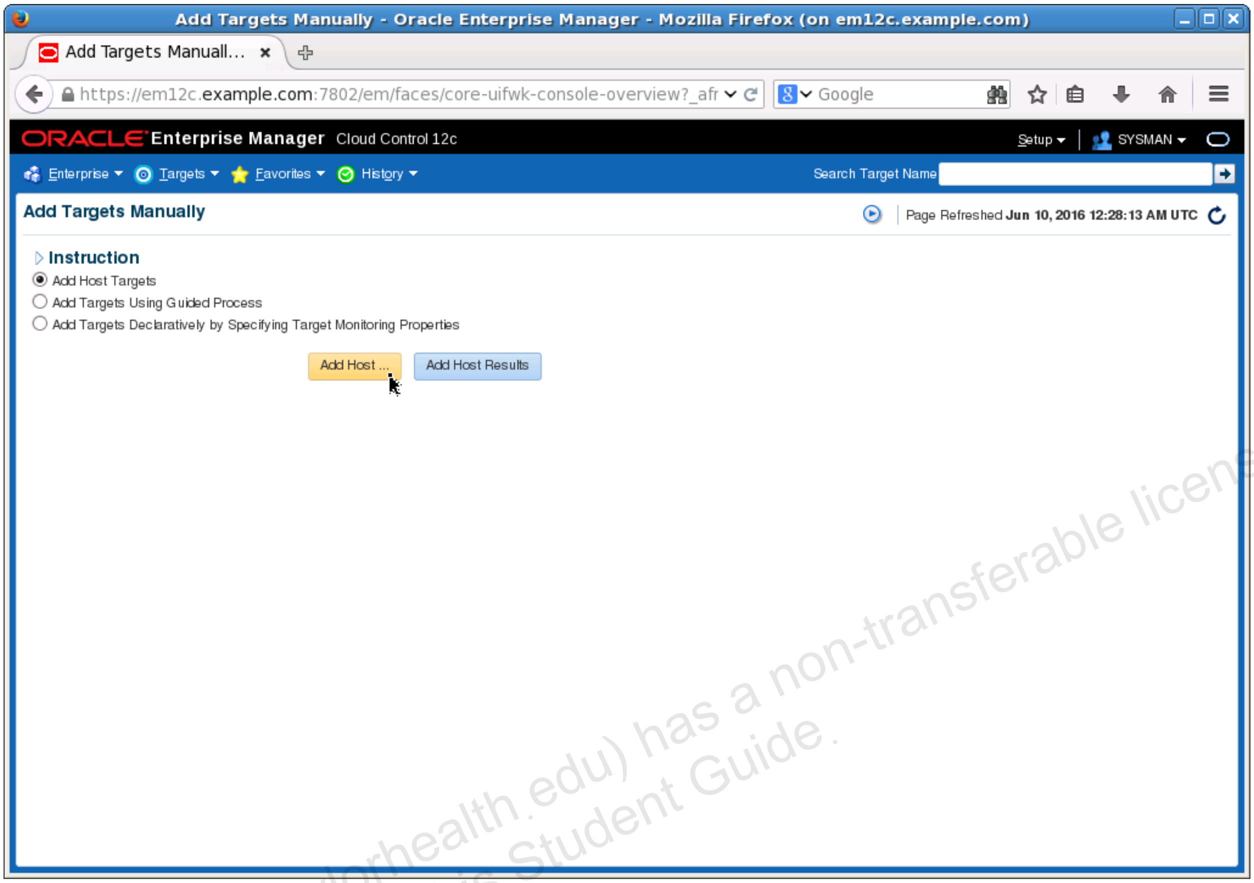
9. On the Plug-ins page, expand the Engineered Systems list item and confirm that the Oracle Exadata plug-in is available and installed on the Management Server.



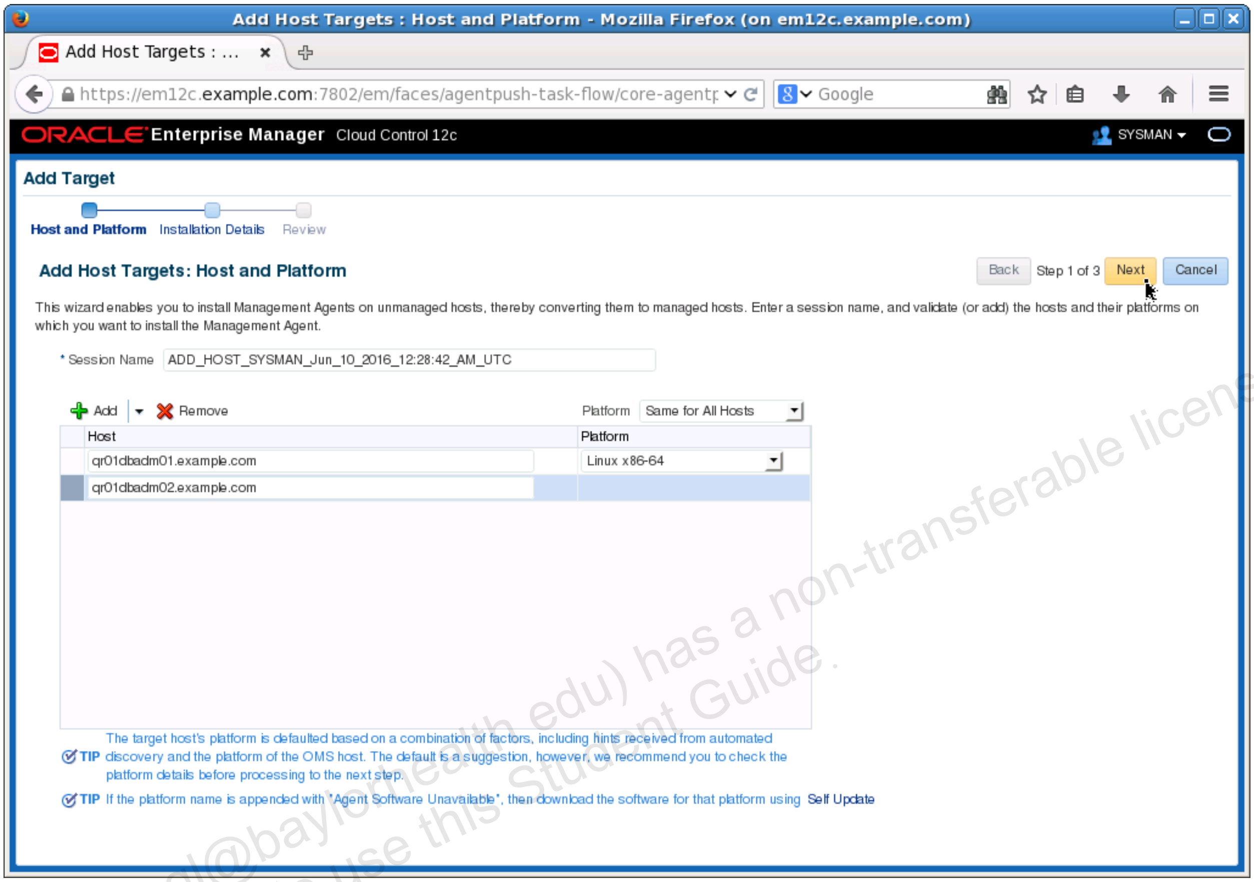
10. The first phase of configuration involves deploying the Enterprise Manager Agent to all the database server hosts on the Exadata Database Machine. To start this process, select the Setup > Add Target > Add Targets Manually menu command.



11. Ensure that the Add Host Targets option is selected and click Add Host to proceed.

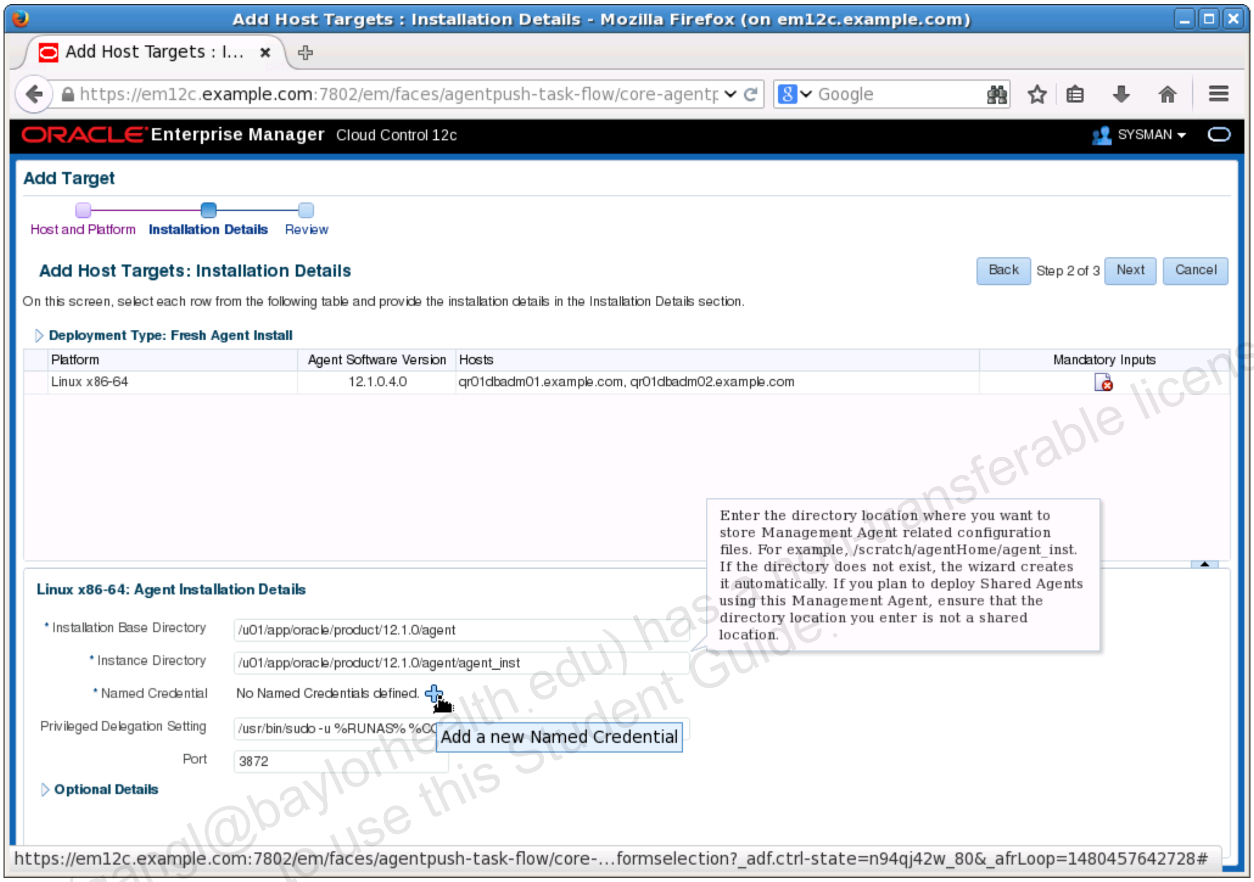


12. On the Host and Platform page, add host entries for `qr01dbadm01.example.com` and `qr01dbadm02.example.com`. Select "Same for All Hosts" in the Platform option list and ensure that `Linux x86-64` is selected for both hosts. Finally, click Next to proceed.

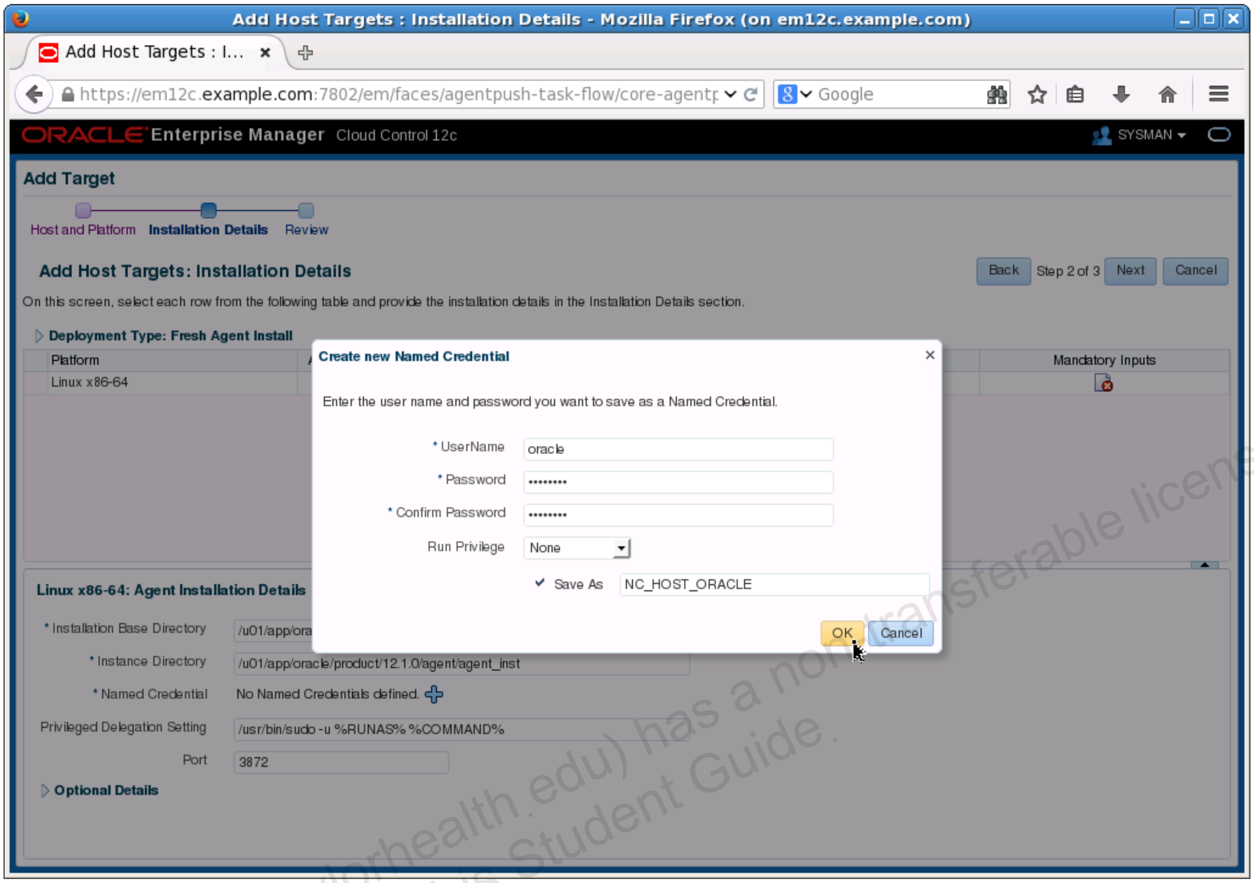


13. On the Installation Details page, enter the following installation details:
   - Installation Base Directory: `/u01/app/oracle/product/12.1.0/agent`
   - Instance Directory: `/u01/app/oracle/product/12.1.0/agent/agent_inst`

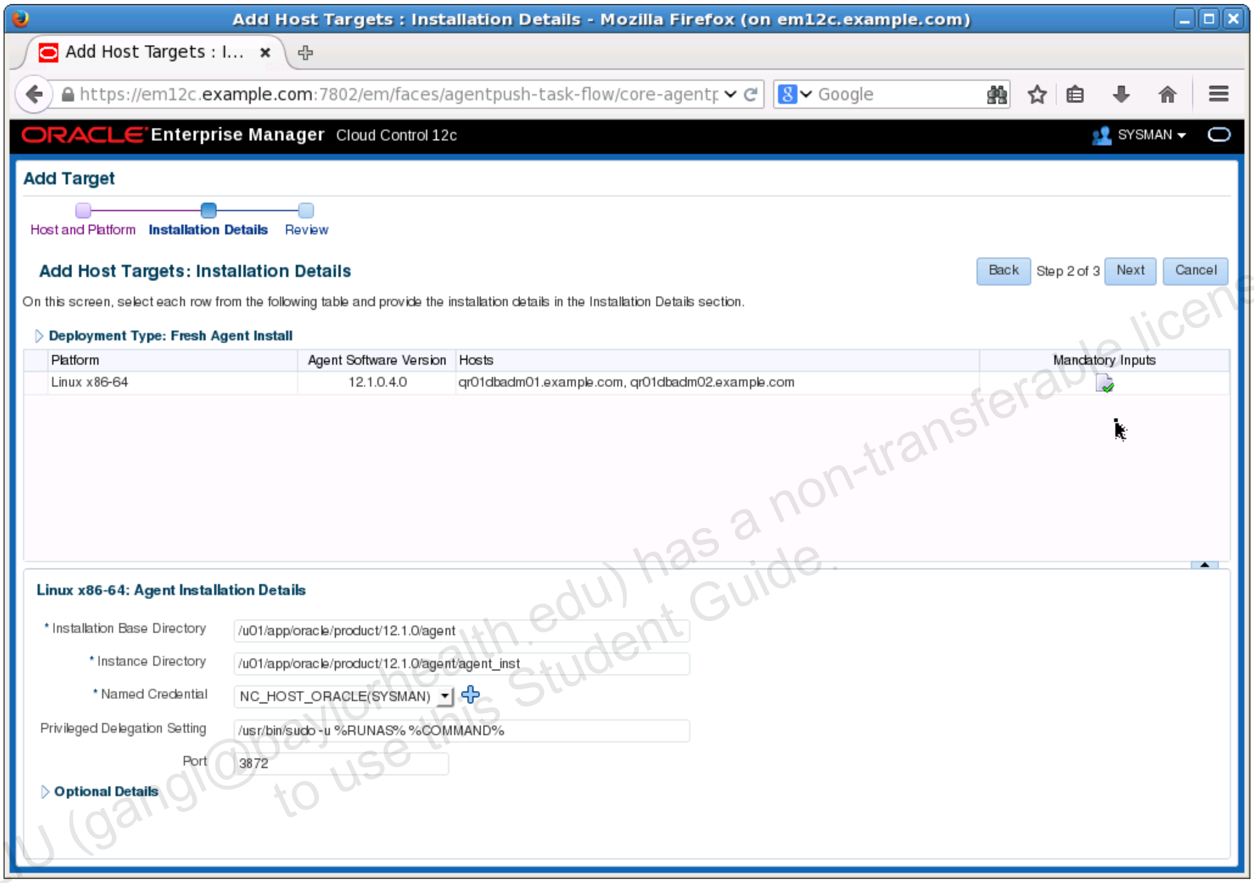
   Then, click the plus icon to add a new Named Credential.



14. Create a new named credential for the `oracle` user on the Exadata database servers.



15. Notice that the icon in the Mandatory Inputs column now includes a green check mark indicating that the required installation details have been supplied. Notice also that the Privileged Delegation Setting command uses the `sudo` utility. For this command to work, you must ensure that `sudo` is appropriately configured on the Exadata database servers. Leave your Enterprise Manager session running while you perform the required configuration in the following steps.



16. Establish a terminal session connected to `qr01dbadm01` as the system administrator (`root` OS user).

```
$ ssh root@qr01dbadm01
root@qr01dbadm01's password: ????????
[root@qr01dbadm01 ~]#
```

17. Execute the following command to add the required `sudo` configuration entries to `/etc/sudoers`. (Alternatively, use a text editor to modify the file if you prefer.)

```
[root@qr01dbadm01 ~]# cat << END >> /etc/sudoers
> oracle ALL=(ALL) NOPASSWD:ALL
> Defaults:oracle !requiretty
> END
[root@qr01dbadm01 ~]#
```

18. Verify that the required `sudo` configuration entries are in `/etc/sudoers`.

```
[root@qr01dbadm01 ~]# tail -2 /etc/sudoers
oracle ALL=(ALL) NOPASSWD:ALL
Defaults:oracle !requiretty
[root@qr01dbadm01 ~]#
```

At this point, you have configured `sudo` on one of your database servers. In the next three steps, you will replicate the same configuration on the other database server.

19. Establish a terminal session connected to `qr01dbadm02` as the system administrator (`root` OS user).

```
$ ssh root@qr01dbadm02
root@qr01dbadm01's password: ????????
[root@qr01dbadm02 ~]#
```

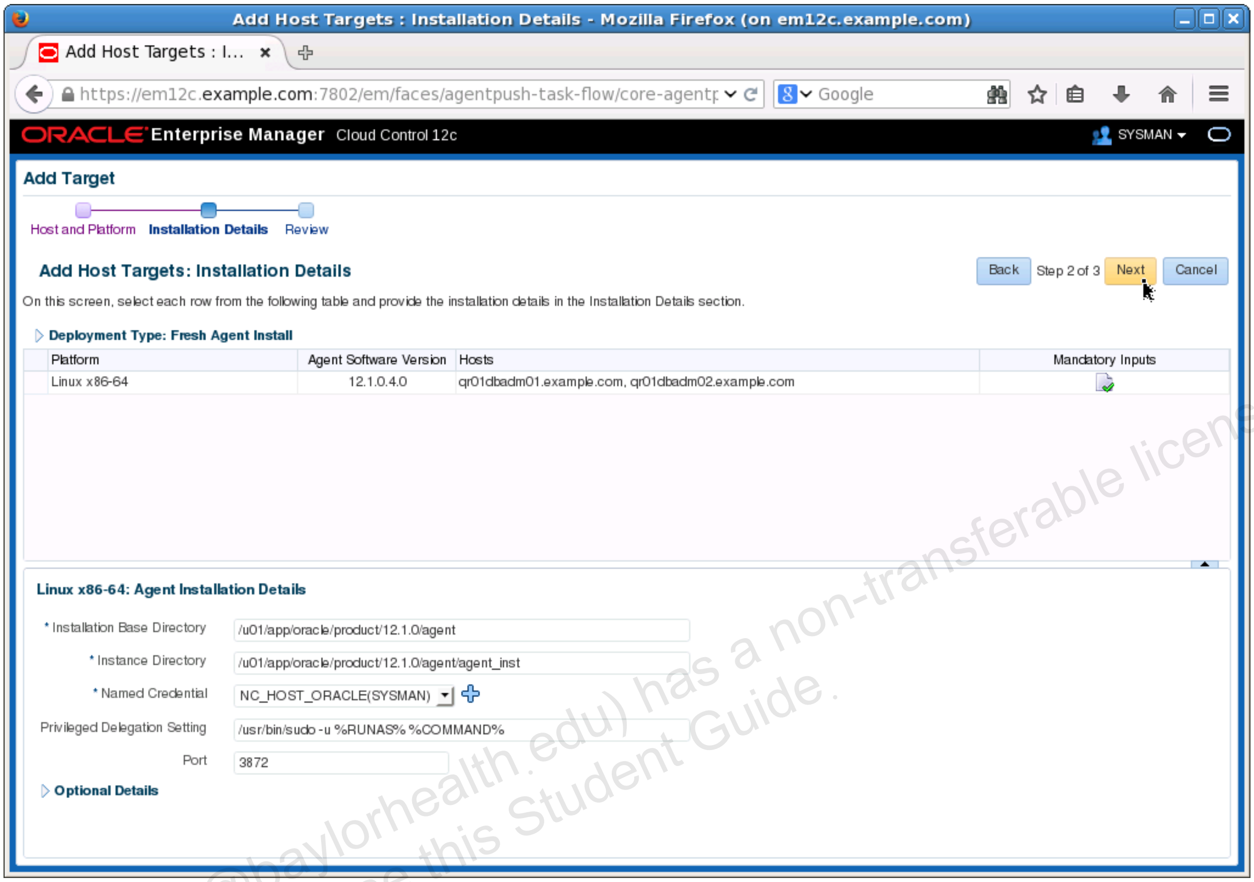
20. Execute the following command to add the required `sudo` configuration entries to `/etc/sudoers` (alternatively, use a text editor to modify the file if you prefer).

```
[root@qr01dbadm02 ~]# cat << END >> /etc/sudoers
> oracle ALL=(ALL) NOPASSWD:ALL
> Defaults:oracle !requiretty
> END
[root@qr01dbadm02 ~]#
```

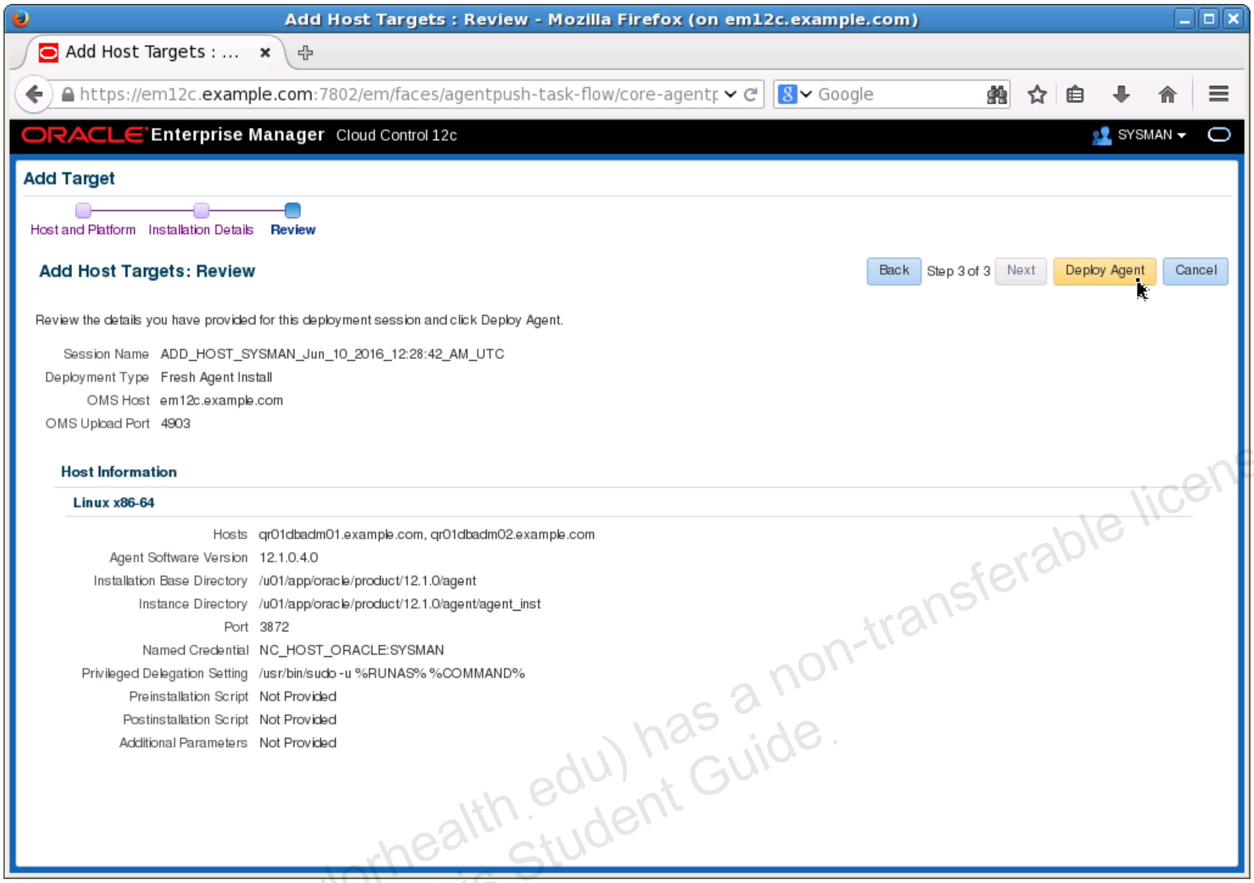
21. Verify that the required `sudo` configuration entries are in `/etc/sudoers`.

```
[root@qr01dbadm02 ~]# tail -2 /etc/sudoers
oracle ALL=(ALL) NOPASSWD:ALL
Defaults:oracle !requiretty
[root@qr01dbadm02 ~]#
```

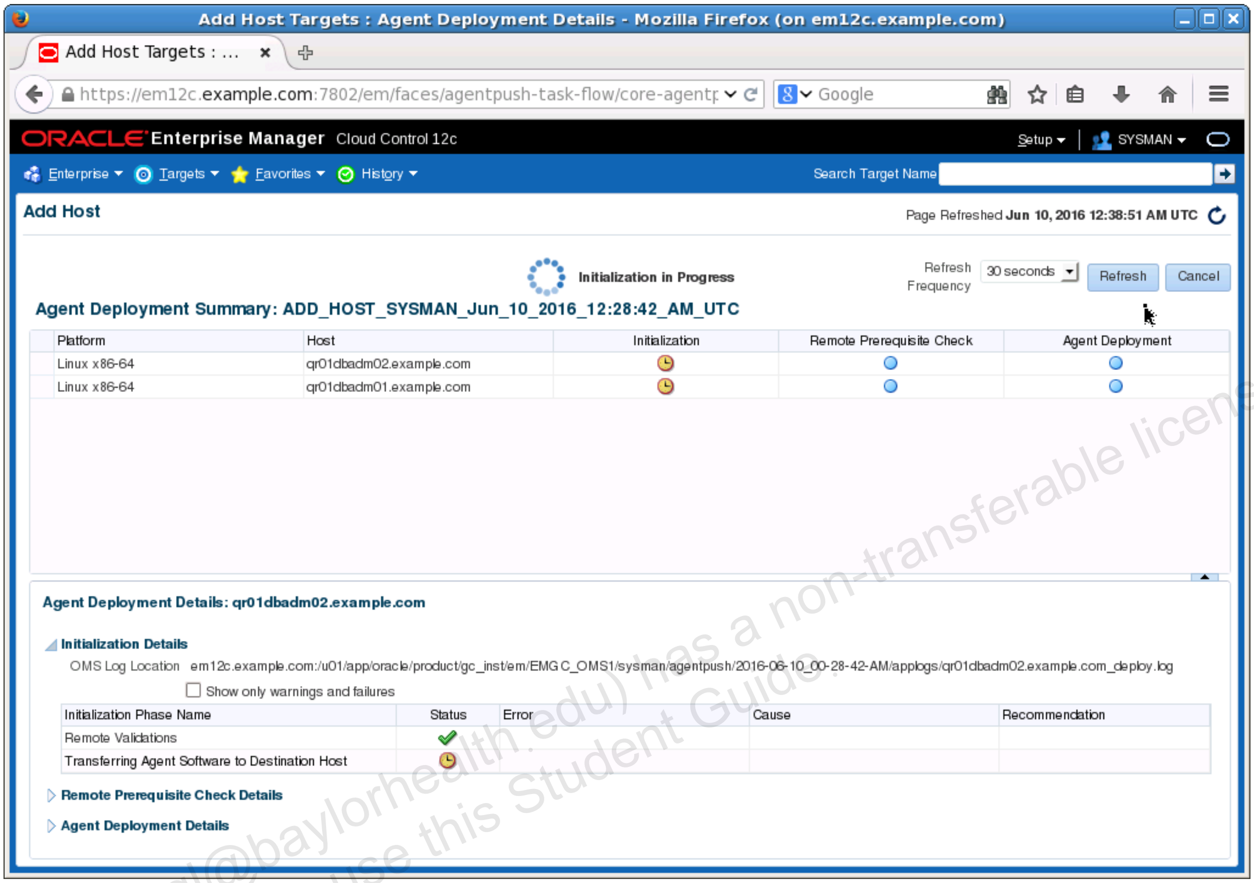
22. After the sudo configuration is complete on qr01dbadm01 and qr01dbadm02, return to your Enterprise Manager session and click Next to proceed.



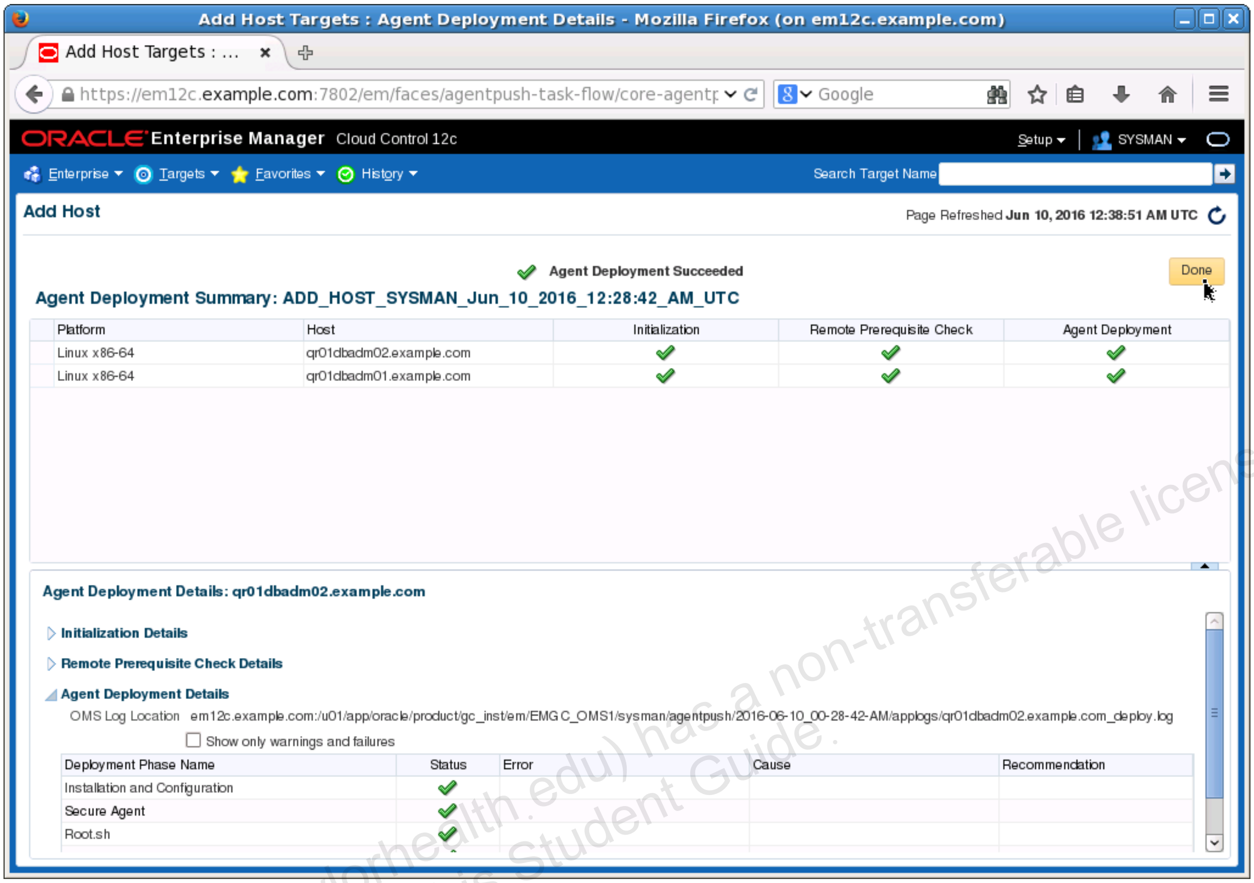
23. Review the agent deployment details that you provided and click Deploy Agent.



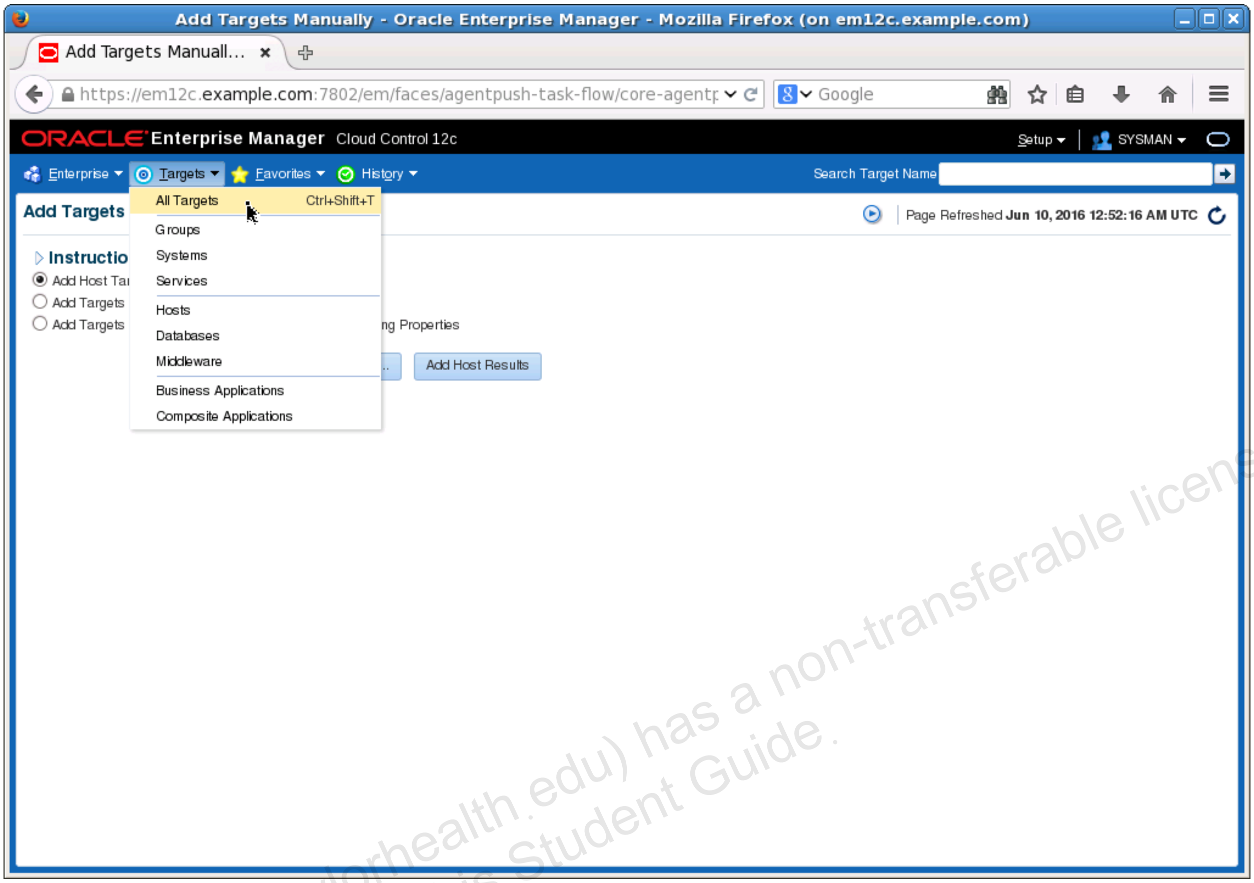
24. Agent deployment can take between 5 and 10 minutes in your laboratory environment. You can monitor agent deployment by using the Add Host Status page. Do not navigate away from this page during agent deployment.



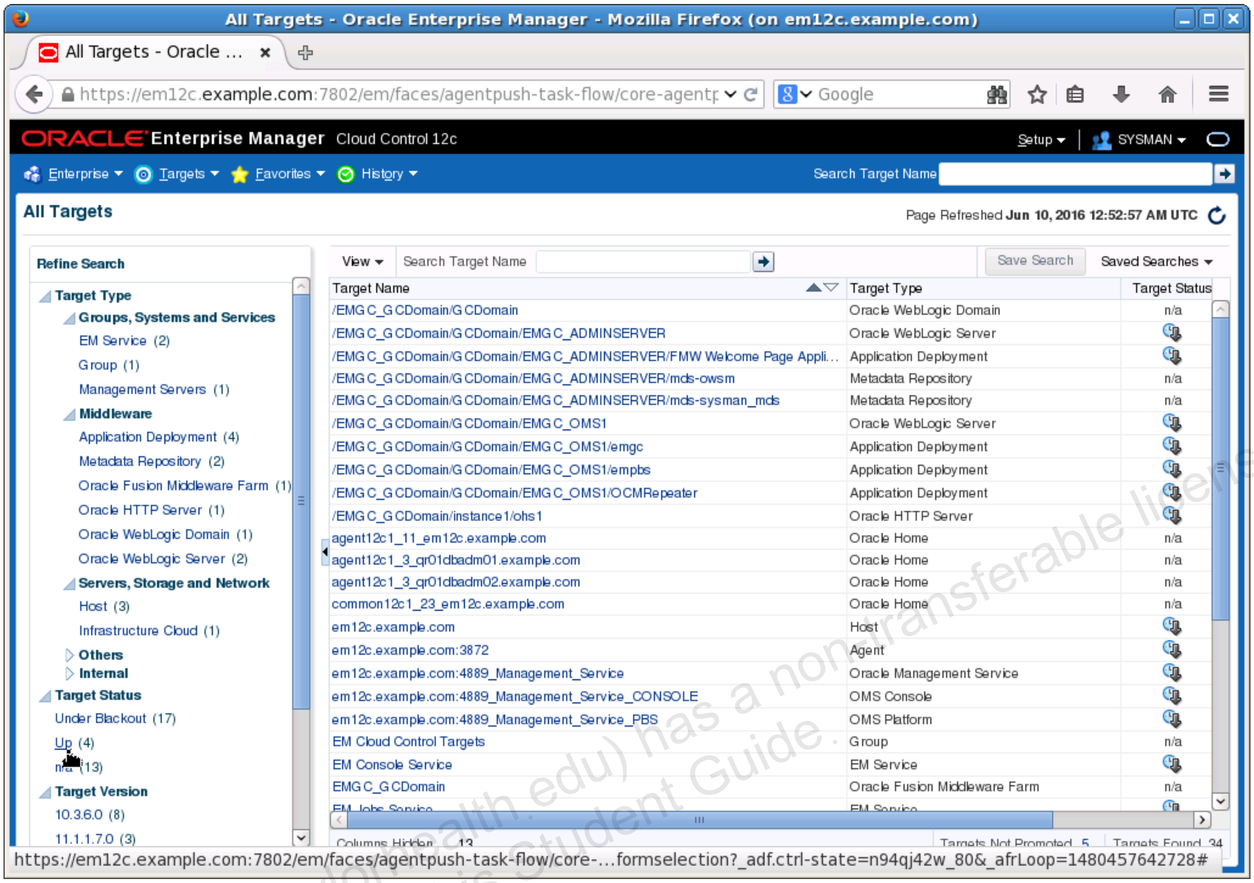
25. When you receive a notification indicating that agent deployment succeeded, click Done.



26. Select the Targets > All Targets menu command.

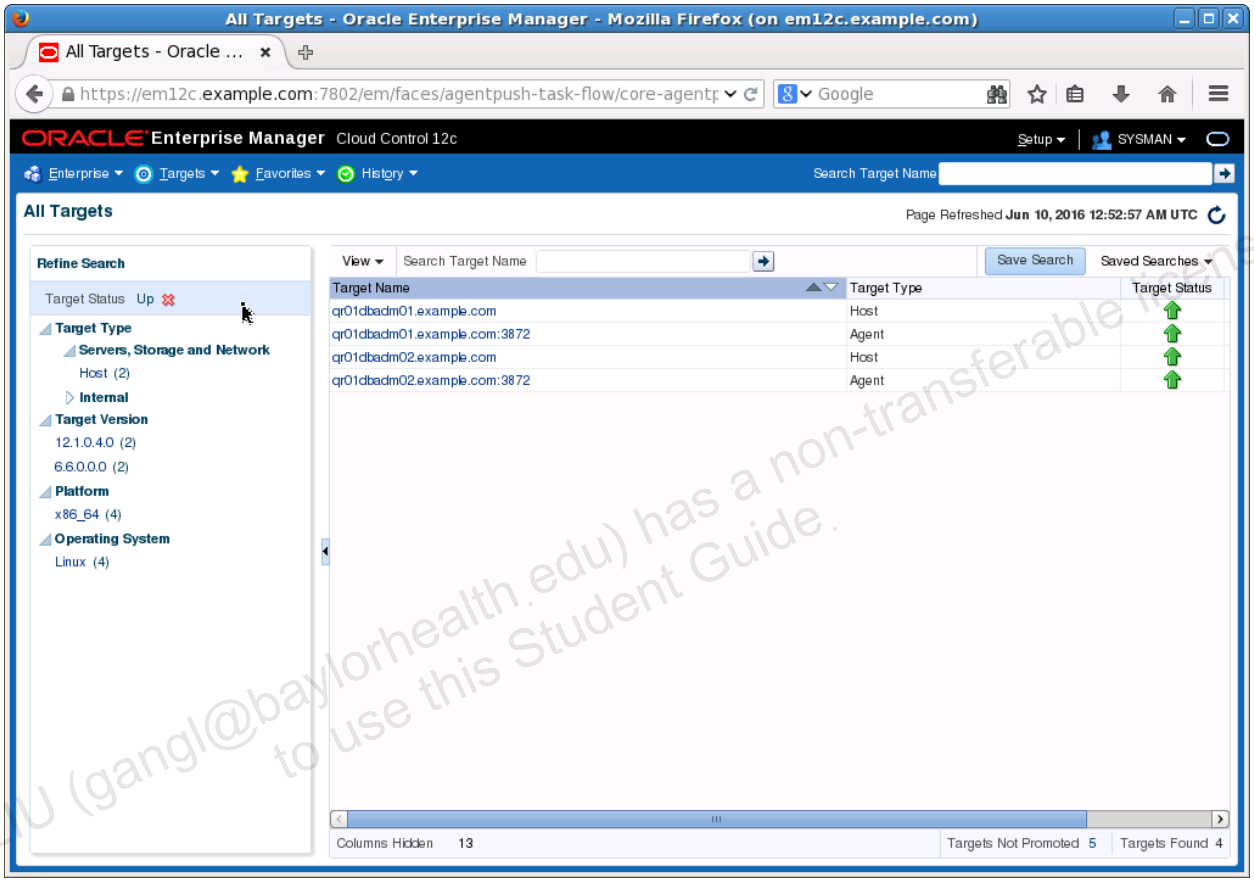


27. In the Refine Search pane, select the Target Status > Up option.



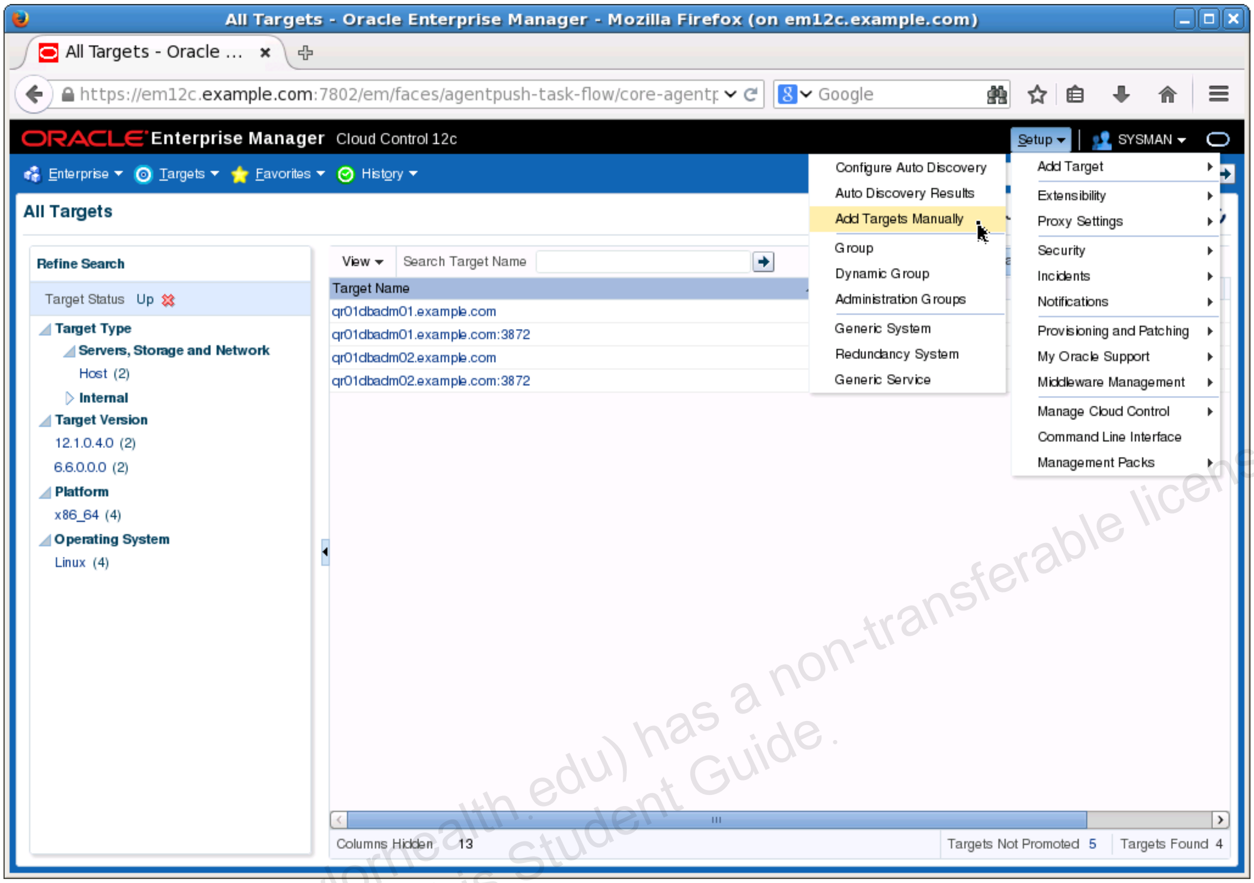
28. You should see the following targets:
- Host: `qr01dbadm01.example.com`
- Agent: `qr01dbadm01.example.com:3872`
- Host: `qr01dbadm02.example.com`
- Agent: `qr01dbadm02.example.com:3872`

If you do not see all these targets, wait for a few moments and refresh the page. Continue to wait until all four targets are displayed.

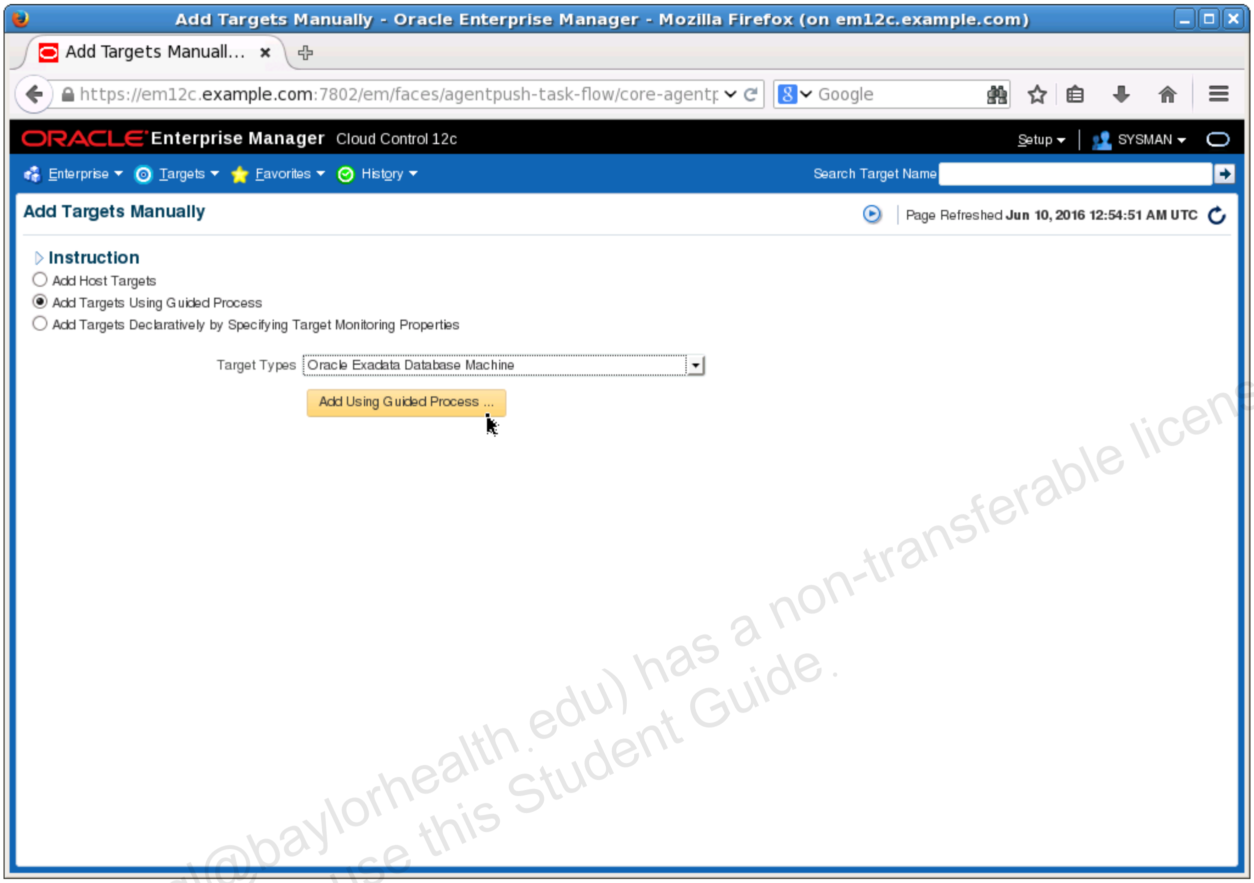


At this point, you have successfully deployed the Enterprise Manager Agent to all the database server hosts on the Exadata Database Machine. Next, you will discover the Exadata Database Machine.

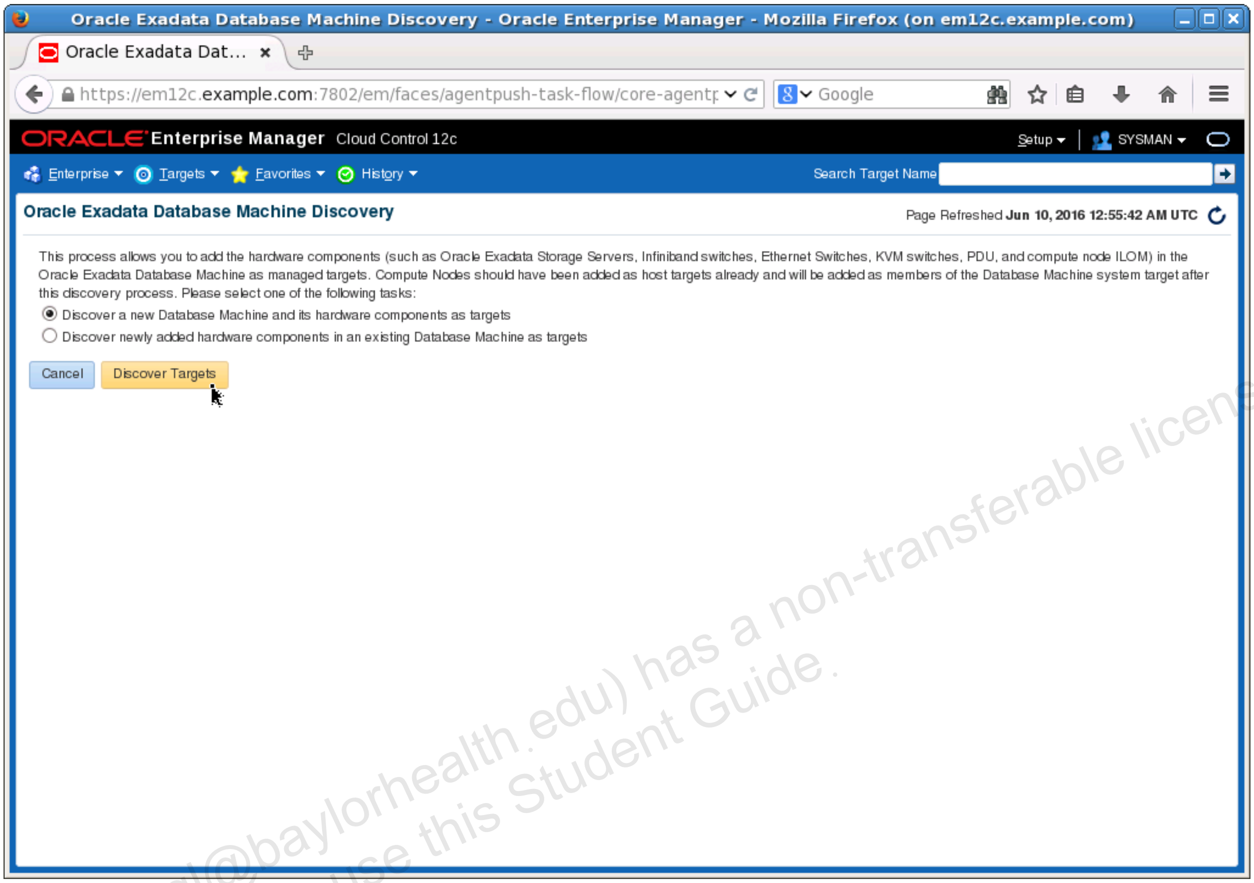
29. Select the Setup > Add Target > Add Targets Manually menu command.



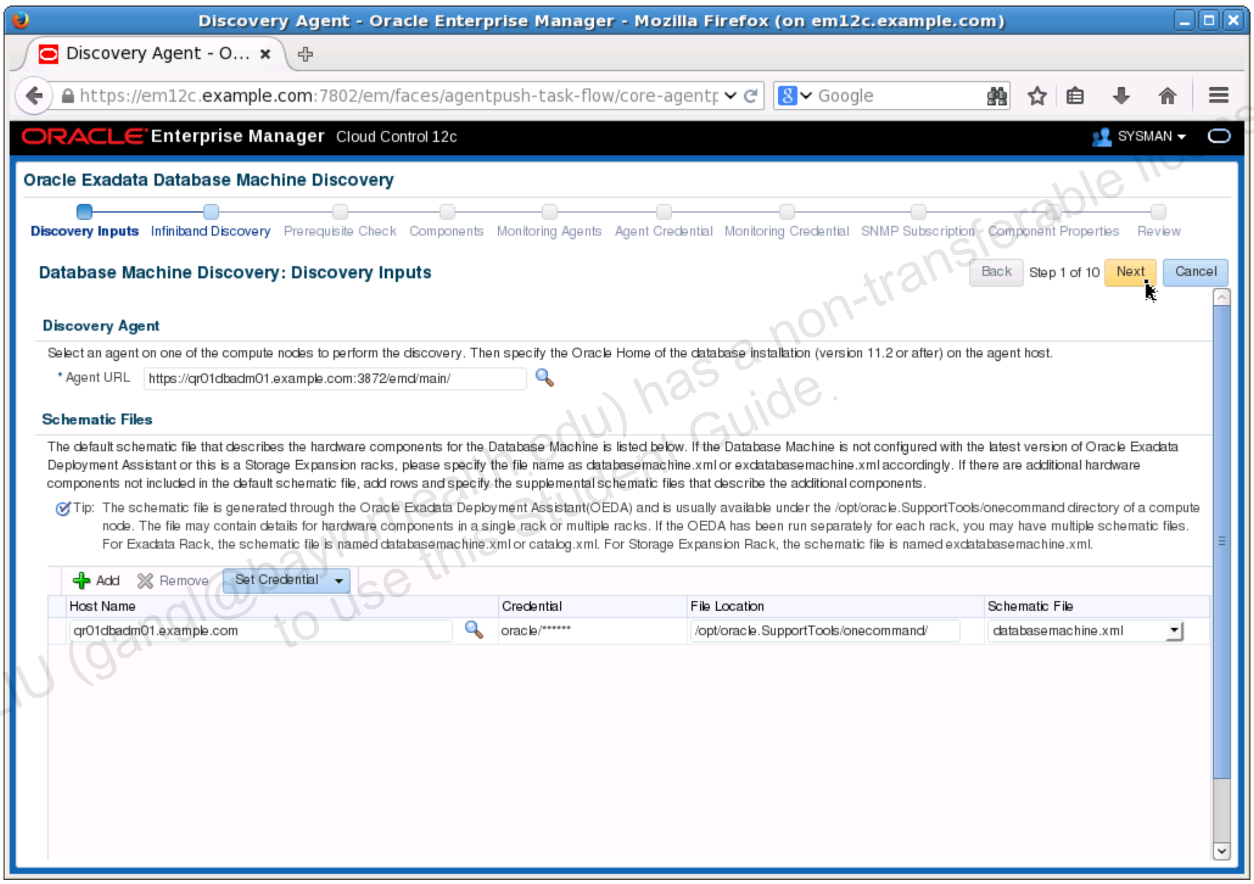
30. On the Add Targets Manually page, select the Add Targets Using Guided Process option. Then select Oracle Exadata Database Machine from the Target Types drop-down list. Finally, click Add Using Guided Process to start the discovery process.



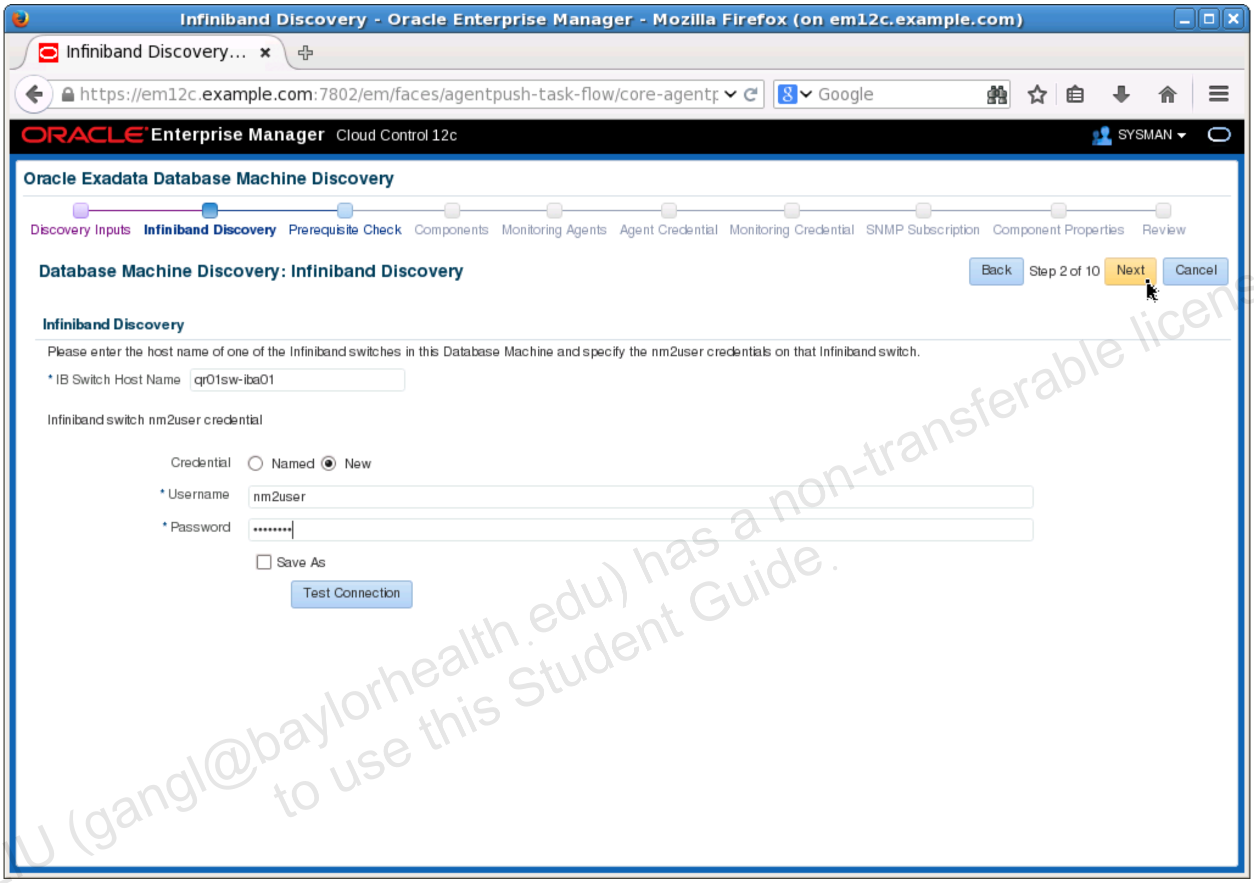
31. On the Oracle Exadata Database Machine Discovery page, select the “Discover a new Database Machine and its hardware components as targets” option and click Discover Targets to proceed.



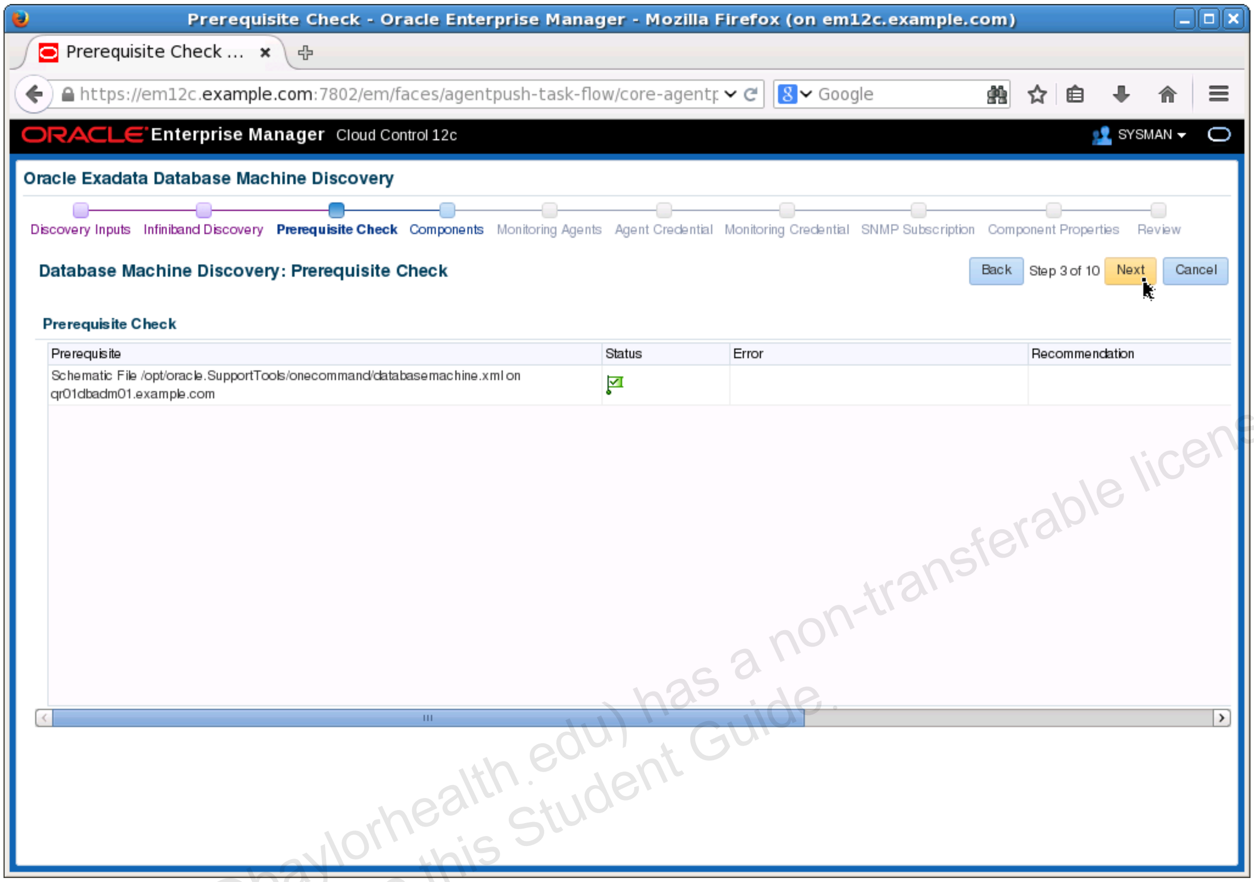
32. On the Discovery Inputs page, in the Discovery Agent section, specify the following:
    - Agent URL: `https://qr01dbadm01.example.com:3872/emd/main/`
    You can specify the Agent URL by clicking the magnifying glass icon and selecting the agent associated with `qr01dbadm01.example.com` if you prefer.
    Note that you must specify the Agent URL for a database server that contains the Database Machine schematic file (typically `databasemchine.xml`), which describes the Database Machine components and configuration settings. This file is generated during the Database Machine initial configuration process, and it typically resides on the first database server.
    Also, click Set Credential > All Hosts and select the named credential that you configured earlier in step 14.
    Finally, click Next to continue.



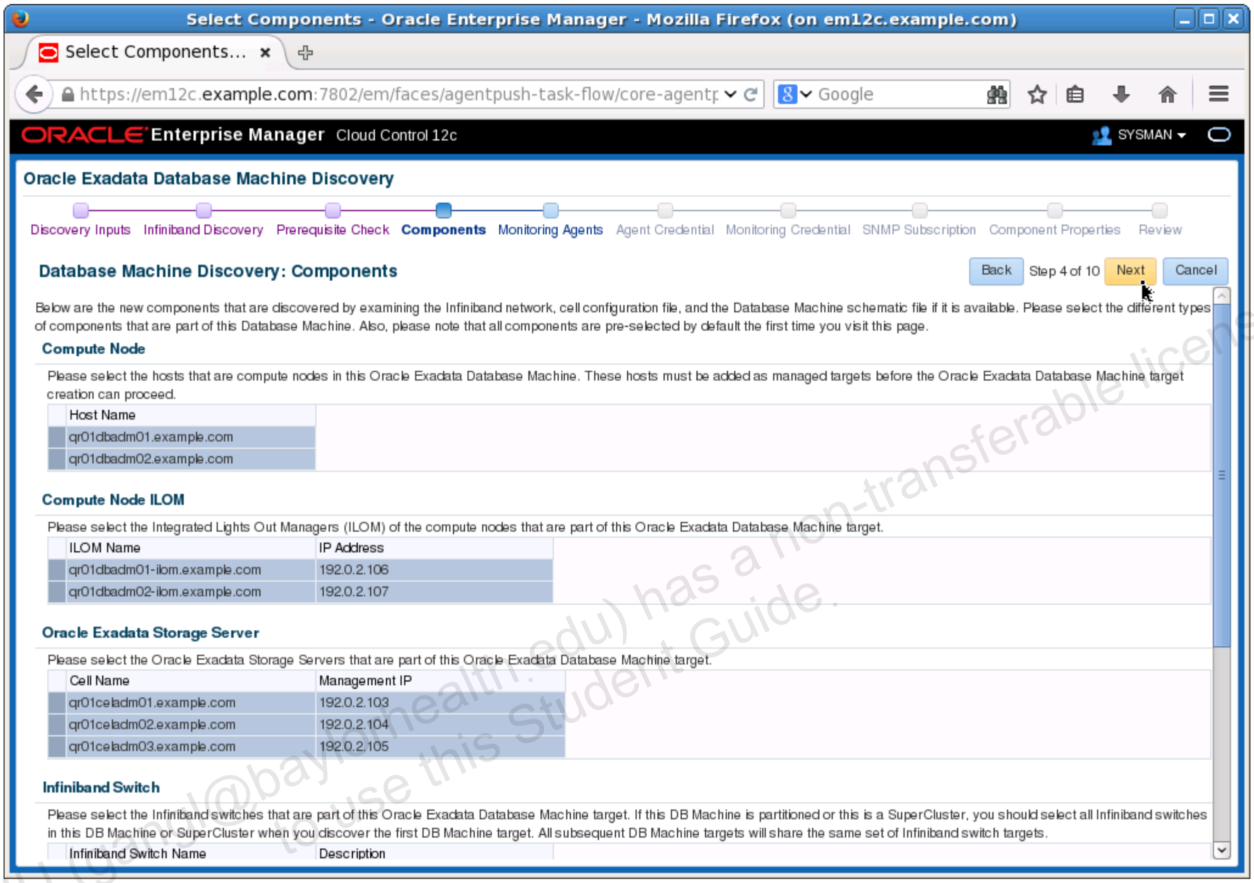
33. On the Infiniband Discovery page, the IB Switch Host Name should be automatically populated by using one of the InfiniBand switches defined in the Database Machine schematic file (`databasemchine.xml`). Select the option to create a new credential and specify the credentials for the `nm2user` user.
    If you want, you can click Test Connection to verify that you have entered the credentials correctly. Finally, click Next to proceed.



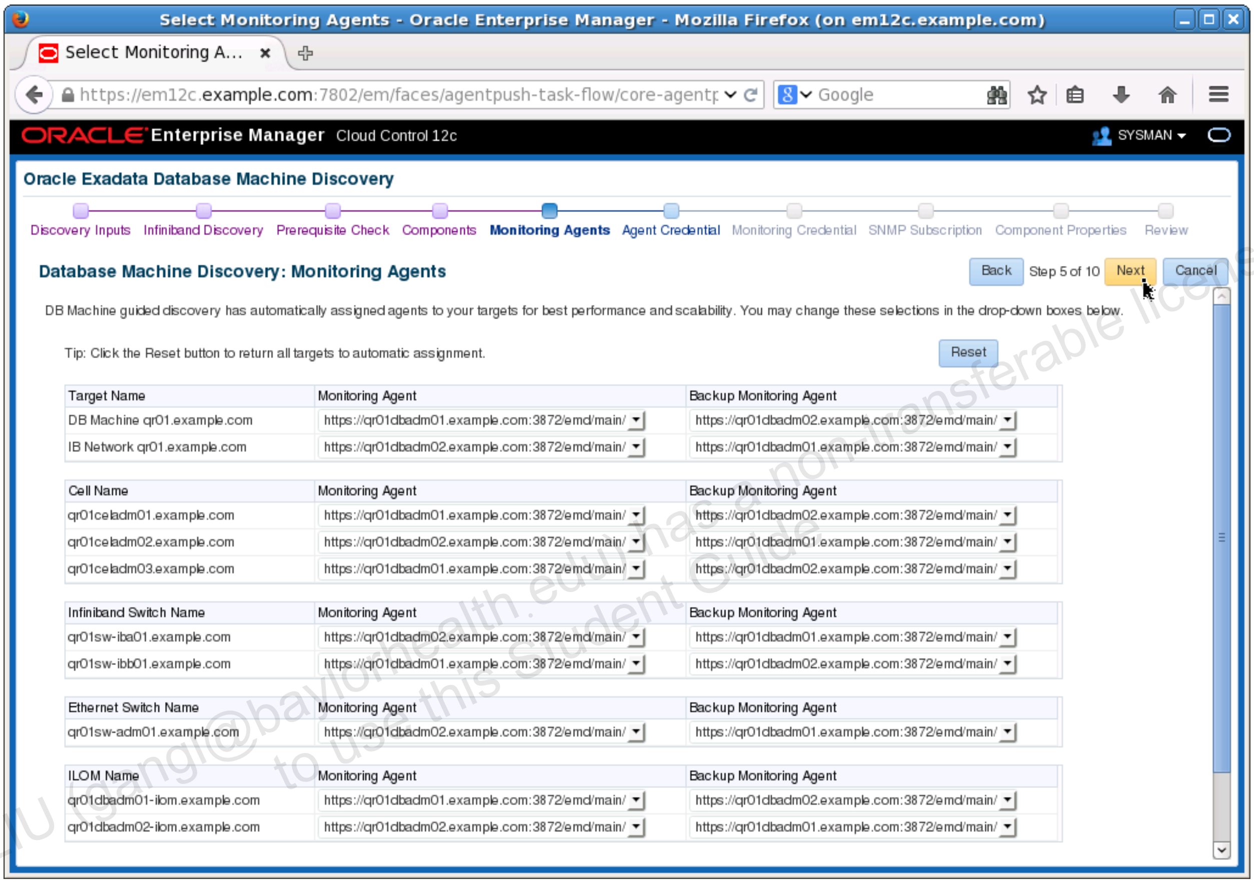
34. Confirm that the prerequisite check status flag contains a green check mark indicating success. Click Next to proceed.



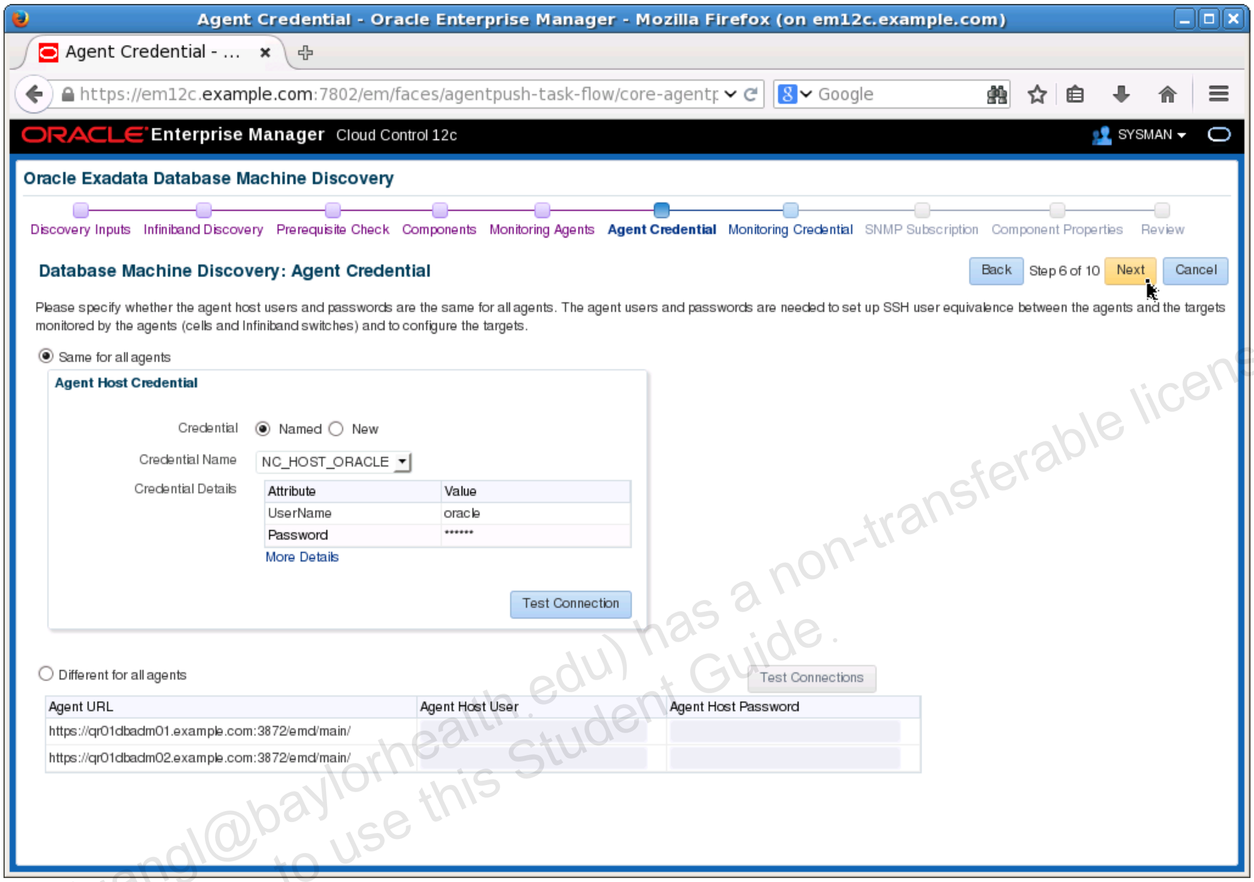
35. The Components page shows all of the components listed in the Database Machine schematic file (`databasemchine.xml`). By default, all of the components are selected. You may optionally deselect components; however, those components will not be monitored by Enterprise Manager. Ensure that all of the components are selected and click Next to proceed.



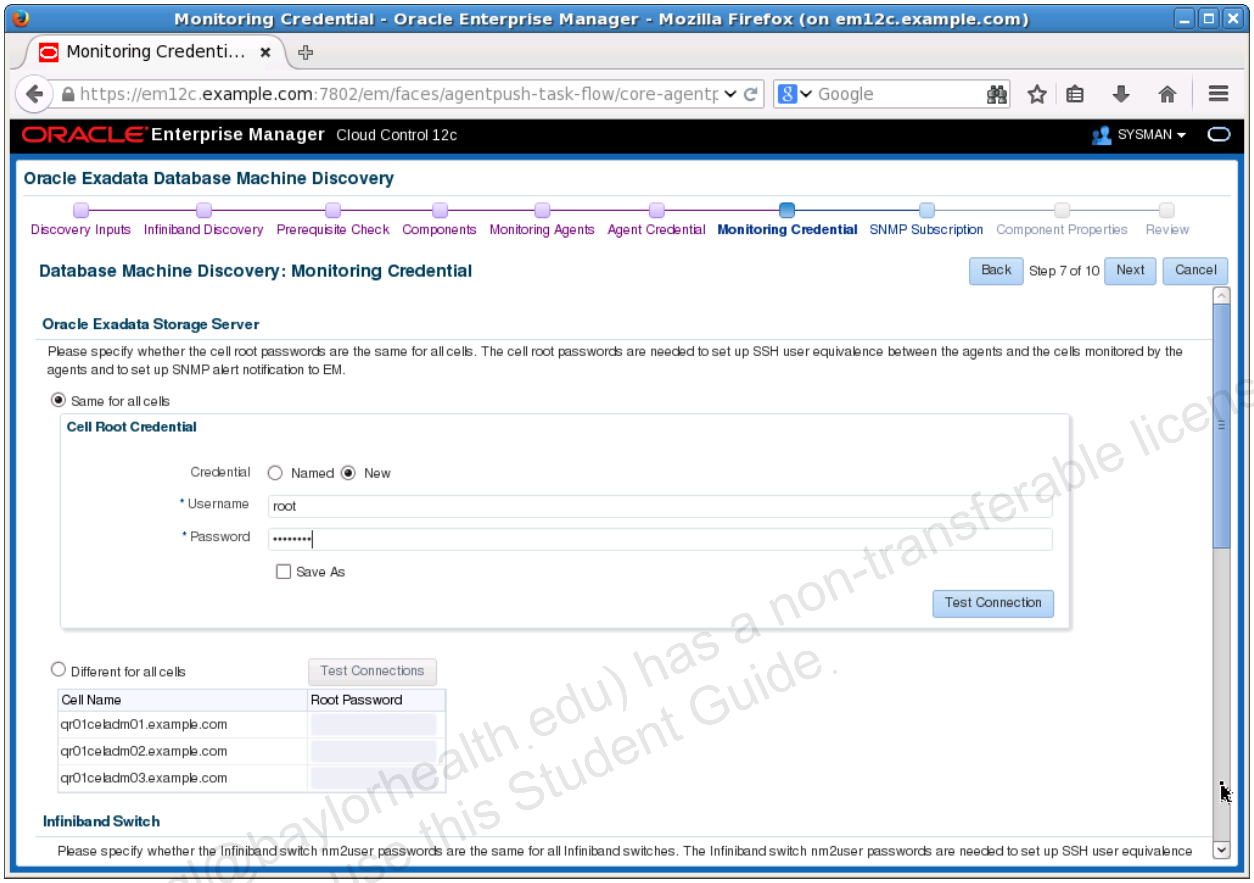
36. Inside Enterprise Manager, Database Machine components are represented as management targets. Each target is monitored by one of the available agents, and each target is assigned a backup agent in case the primary agent is unavailable. The Monitoring Agents page allows you to assign primary and backup monitoring agents for each management target. By default, the targets are spread across all of the available agents. Alternatively, you can select specific agents to monitor each target. For this practice, accept the default assignments and click Next to proceed.



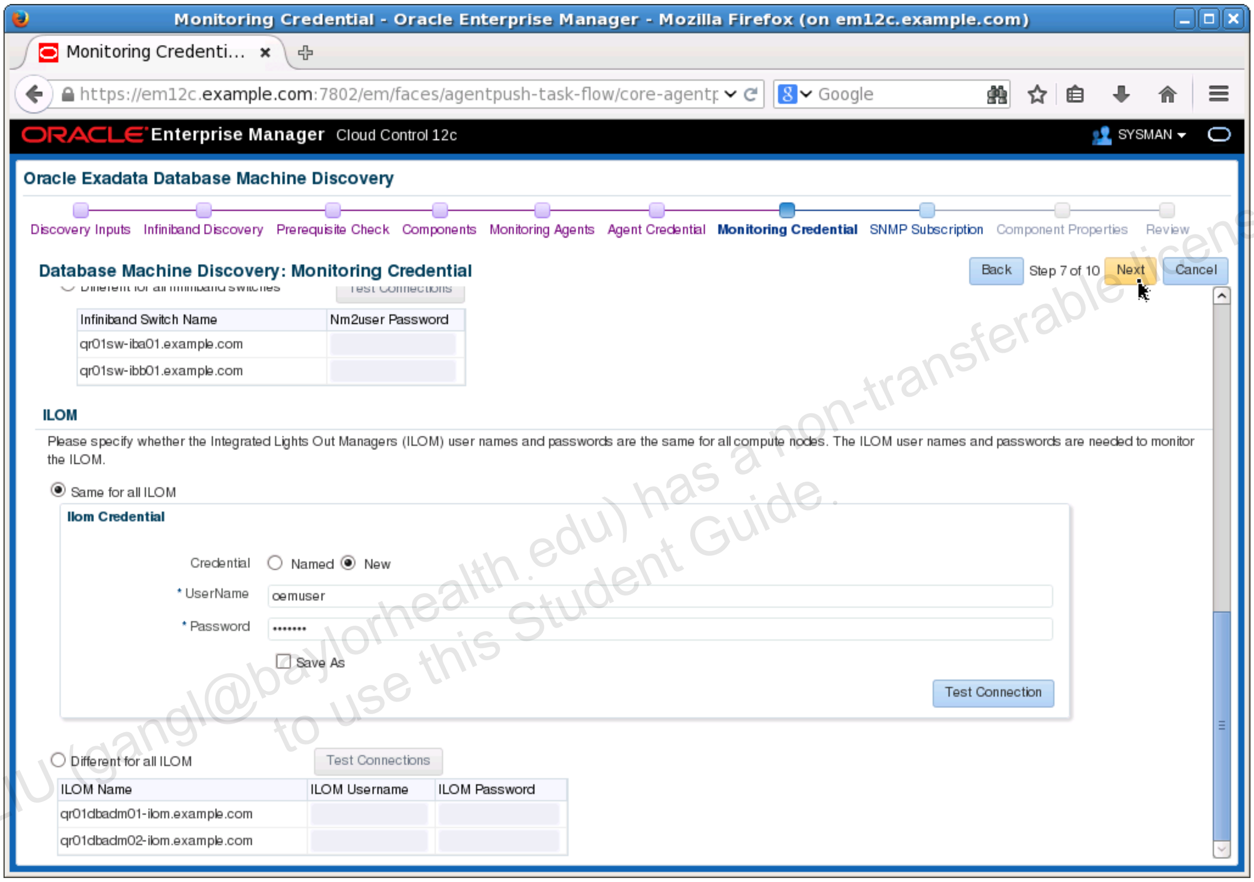
37. The Agent Credential page enables you to specify the credentials used to access and control the management agents on each Exadata database server. Because you earlier used Enterprise Manager to configure the agents, the required credentials are already specified on the page. Click Next to proceed.



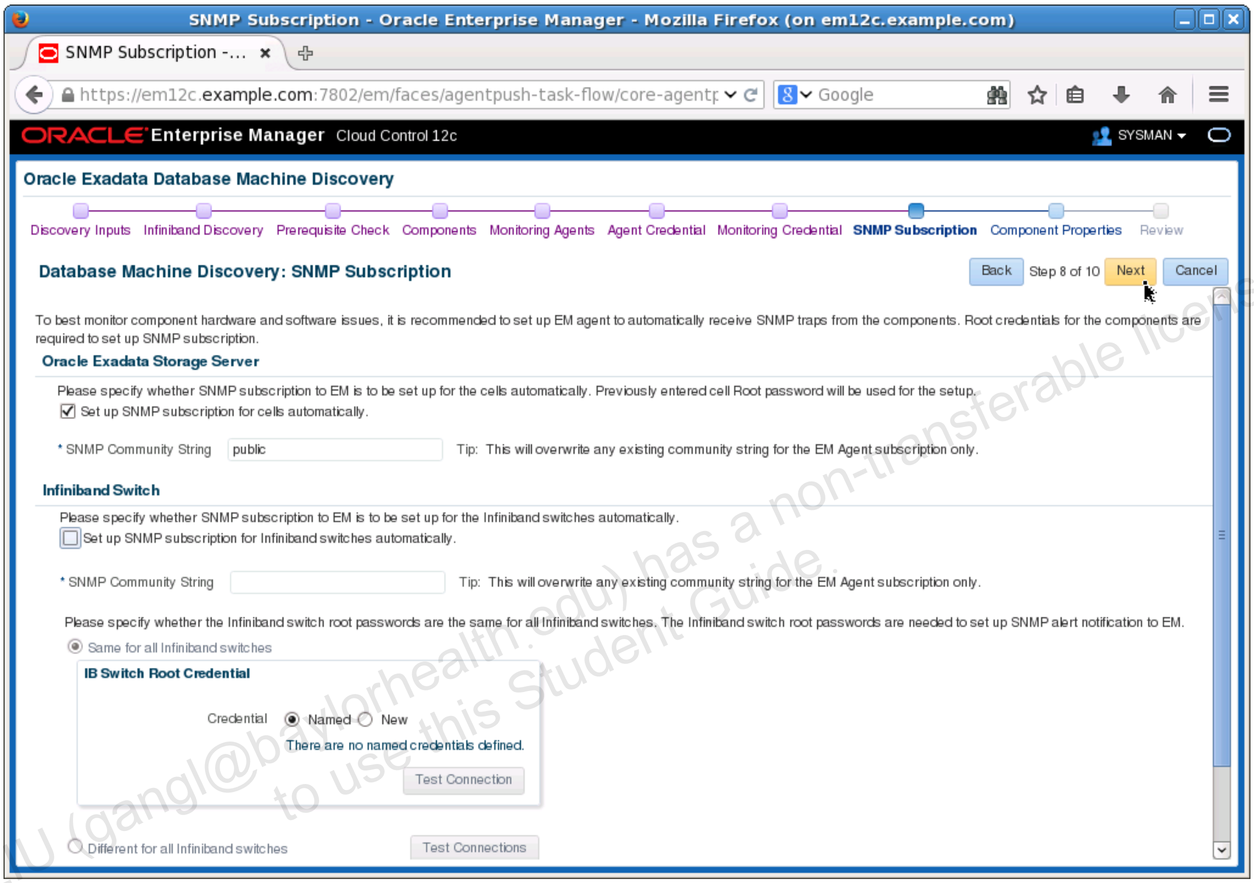
38. The Monitoring Credential page enables you to specify the required credentials for Exadata Storage Servers, InfiniBand switches, and Oracle ILOM Servers.
Specify the Exadata Storage Server `root` user credentials.



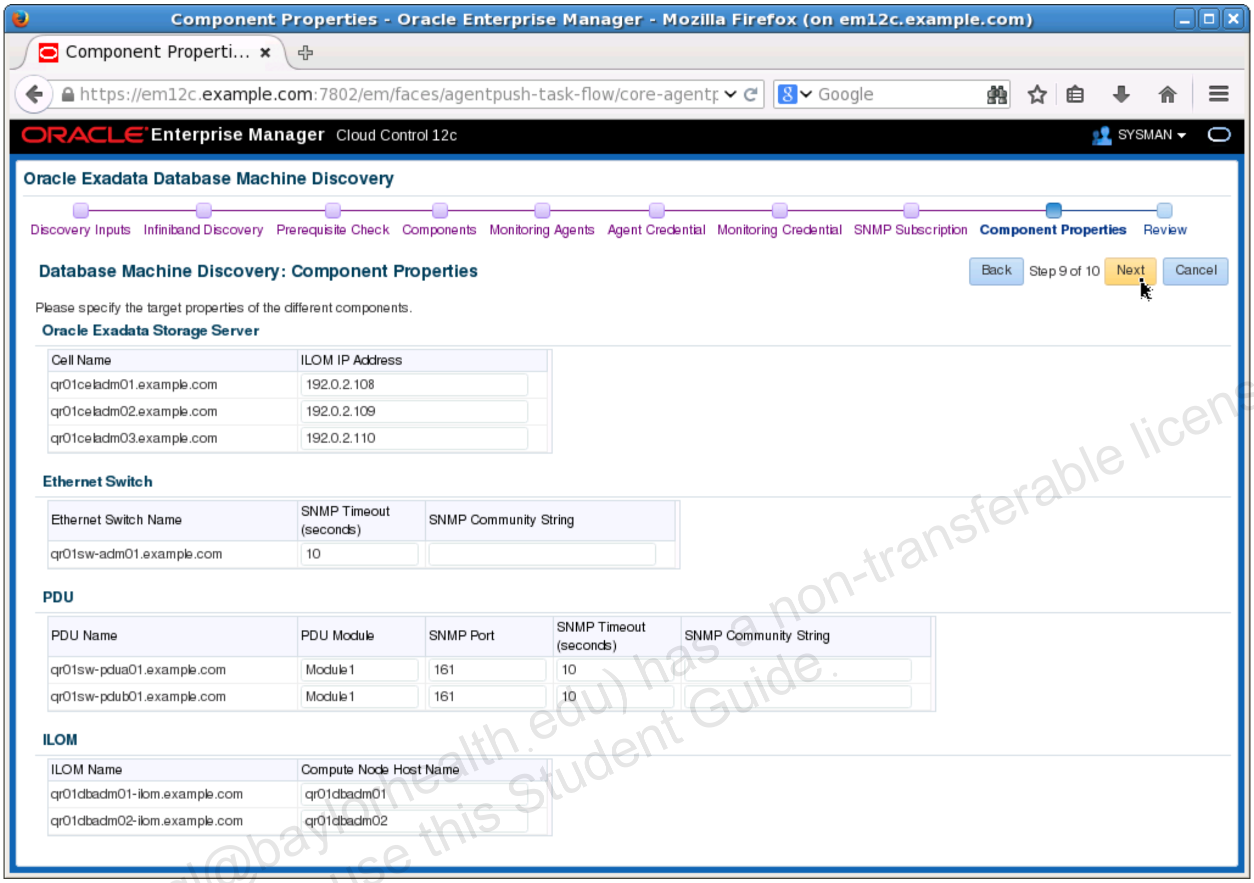
39. Scroll to the bottom of the Monitoring Credential page. Note that the InfiniBand Switch monitoring credentials are already populated with the details you supplied on the InfiniBand Discovery page. Next, specify the credentials for the ILOM user (`oemuser`).
    Note that the ILOM user specified in this step must normally be created on each ILOM server before Database Machine discovery. However, because the ILOM interfaces in your laboratory environment are not fully functioning ILOM servers, you are unable to create the ILOM user, and instead the required user has been pre-created.
    Click Next to proceed.



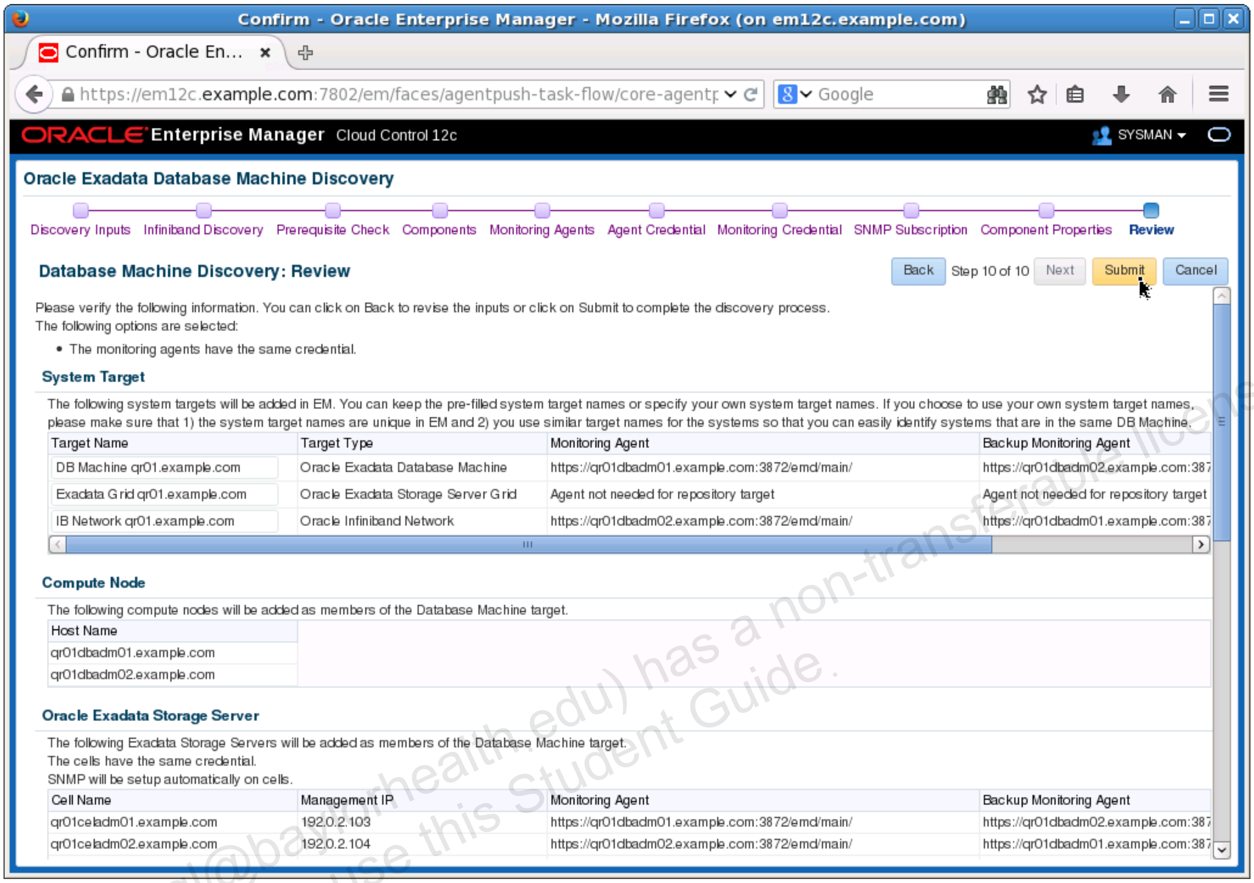
40. The SNMP Subscription page enables you to automatically configure SNMP subscriptions for Enterprise Manager on the Exadata Storage Servers and the InfiniBand switches. Specify `public` for the Exadata Storage Server SNMP Community String. Because your laboratory environment does not contain fully functioning InfiniBand switches, deselect the option to automatically set up SNMP subscriptions for the InfiniBand switches. Finally, click Next to proceed.



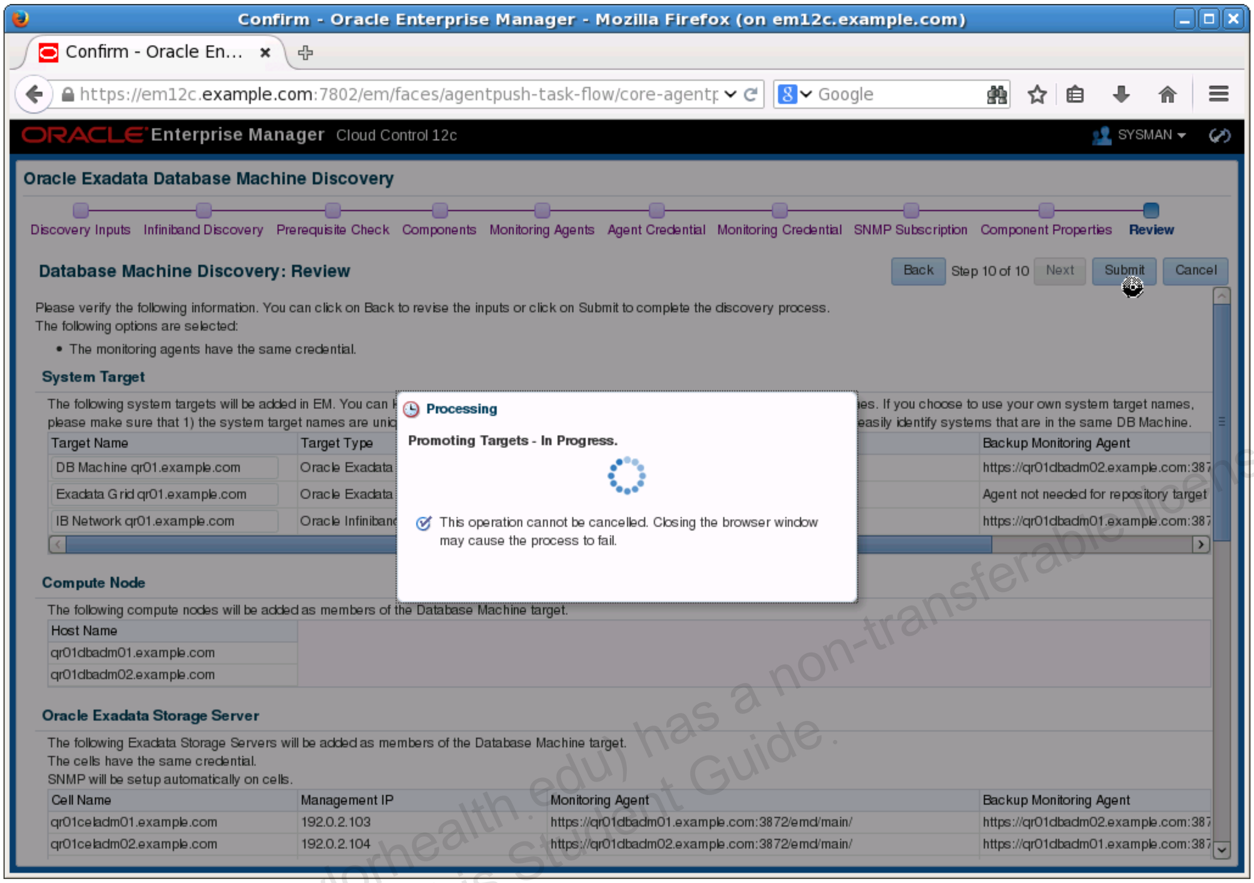
41. The Component Properties page enables you to specify various component property settings. In your laboratory environment, the default settings will suffice. So, click Next to proceed.



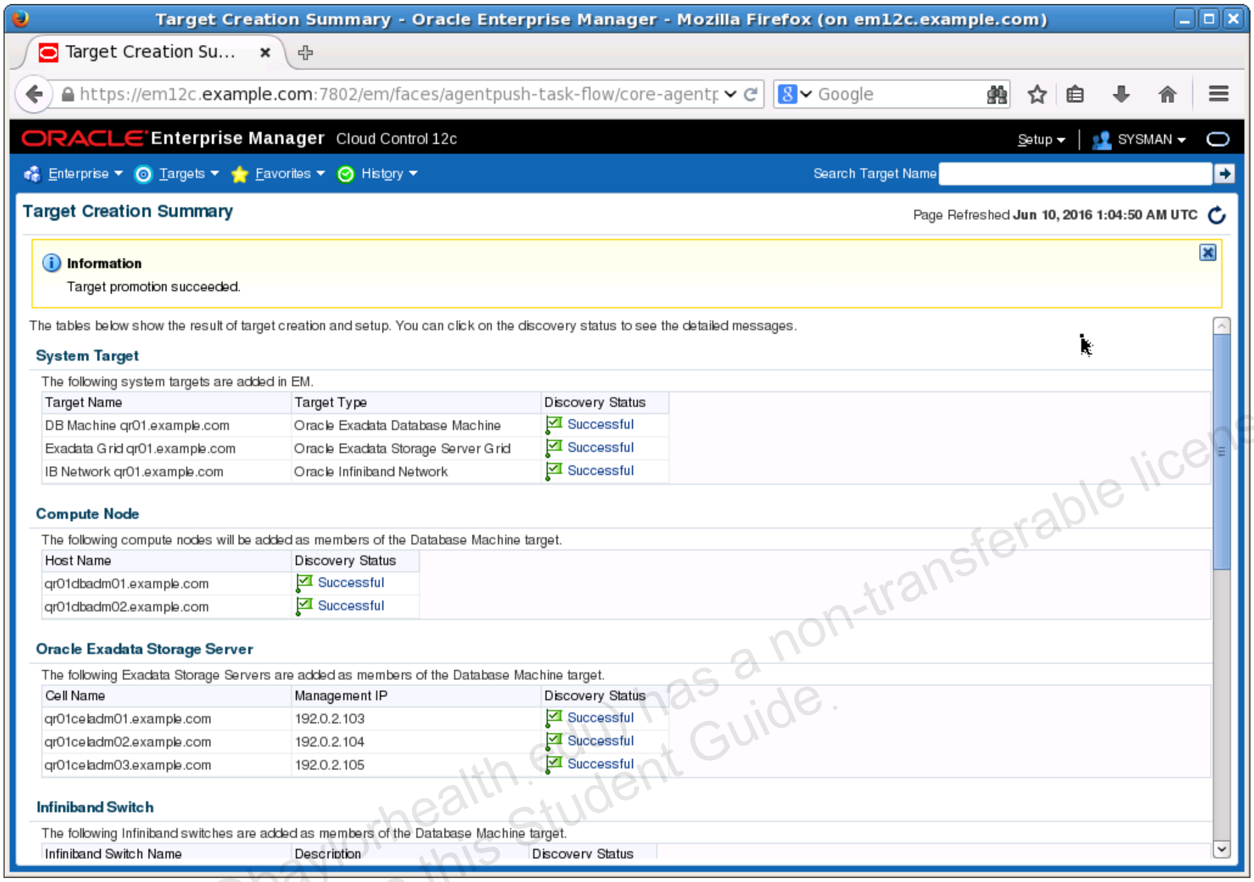
42. The Review page displays the summary of the discovery session. Also, you can optionally specify customized Target Names in the System Target section. When you are satisfied, click Submit to create the Database Machine monitoring targets.



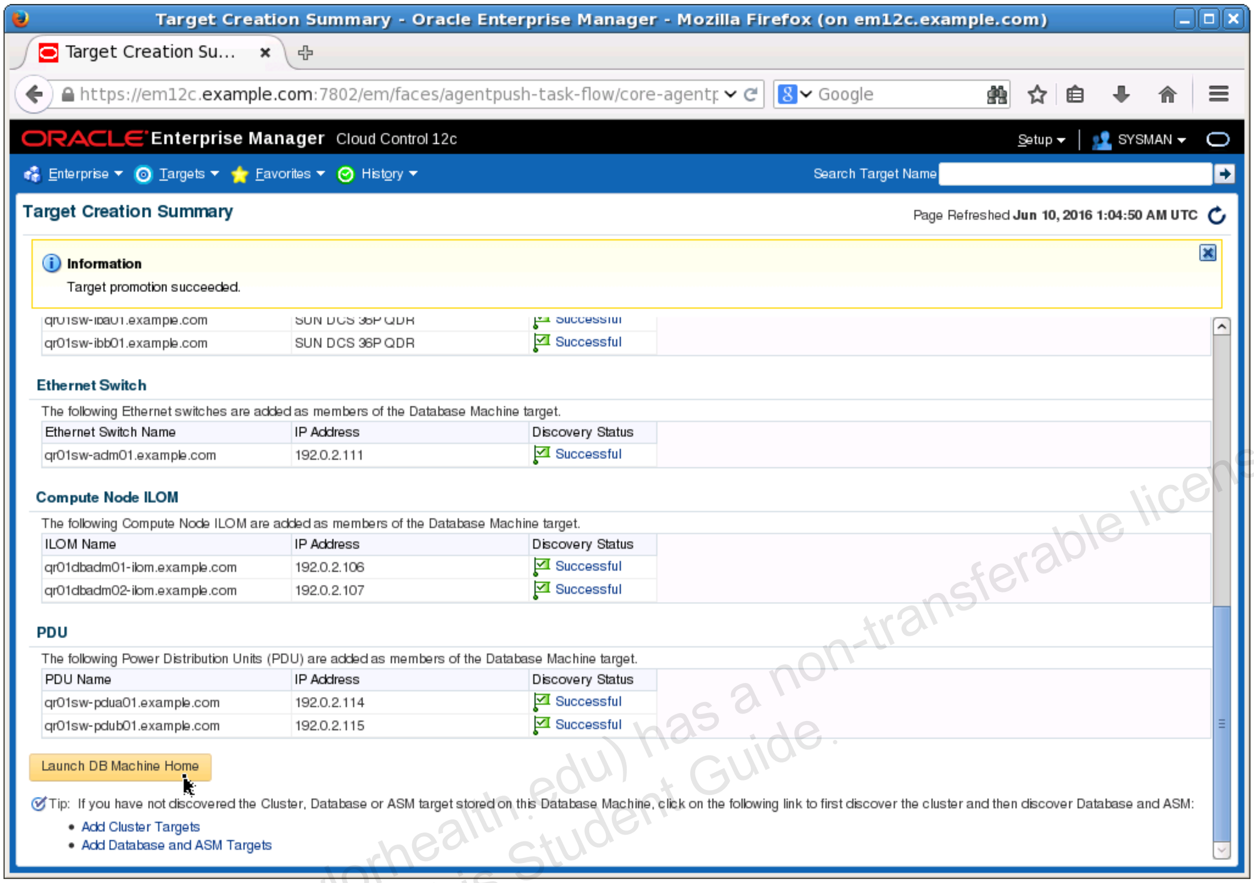
43. Wait while the Processing dialog box is displayed.



44. When the Target Creation Summary page appears, examine it to ensure that all the targets were successfully added to Enterprise Manager.

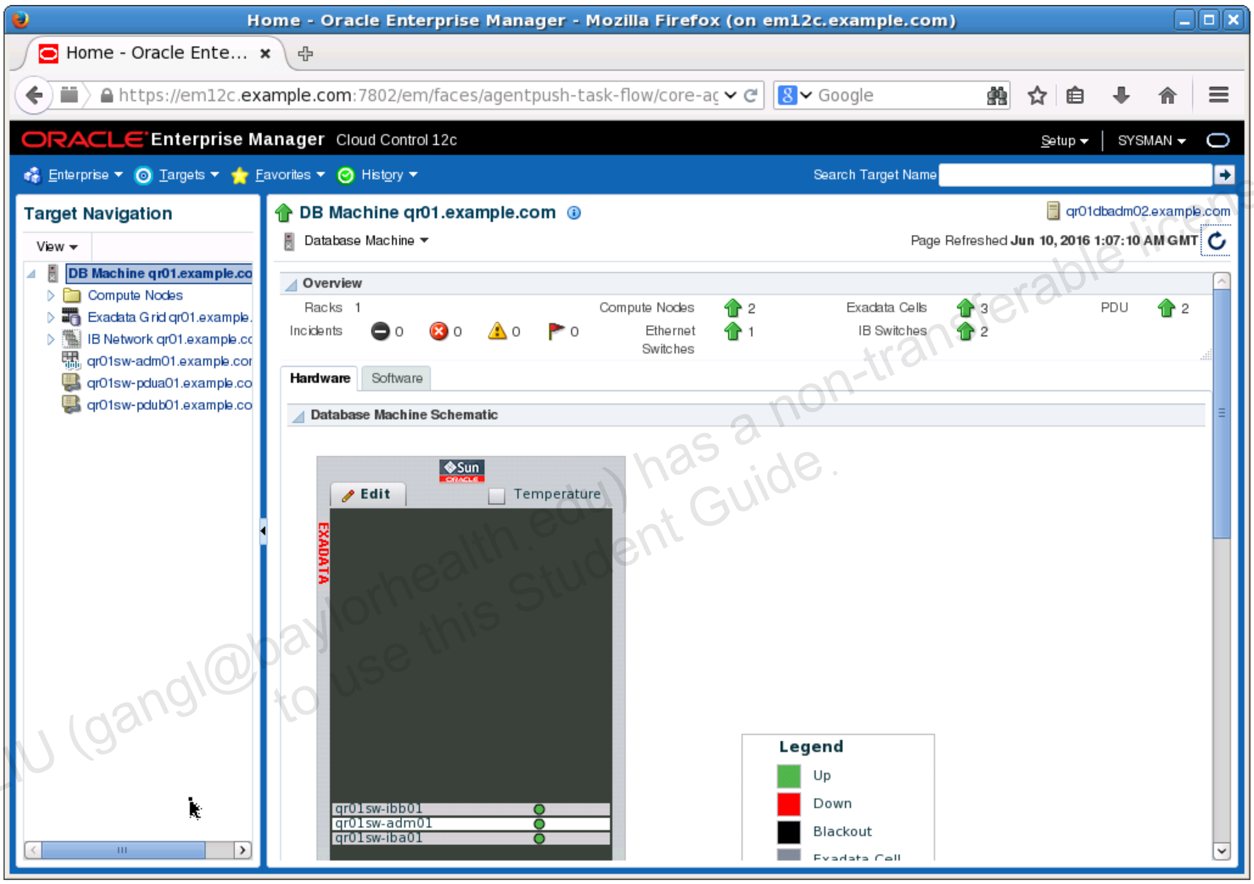


45. At the bottom of the Target Creation Summary page, click Launch DB Machine Home.

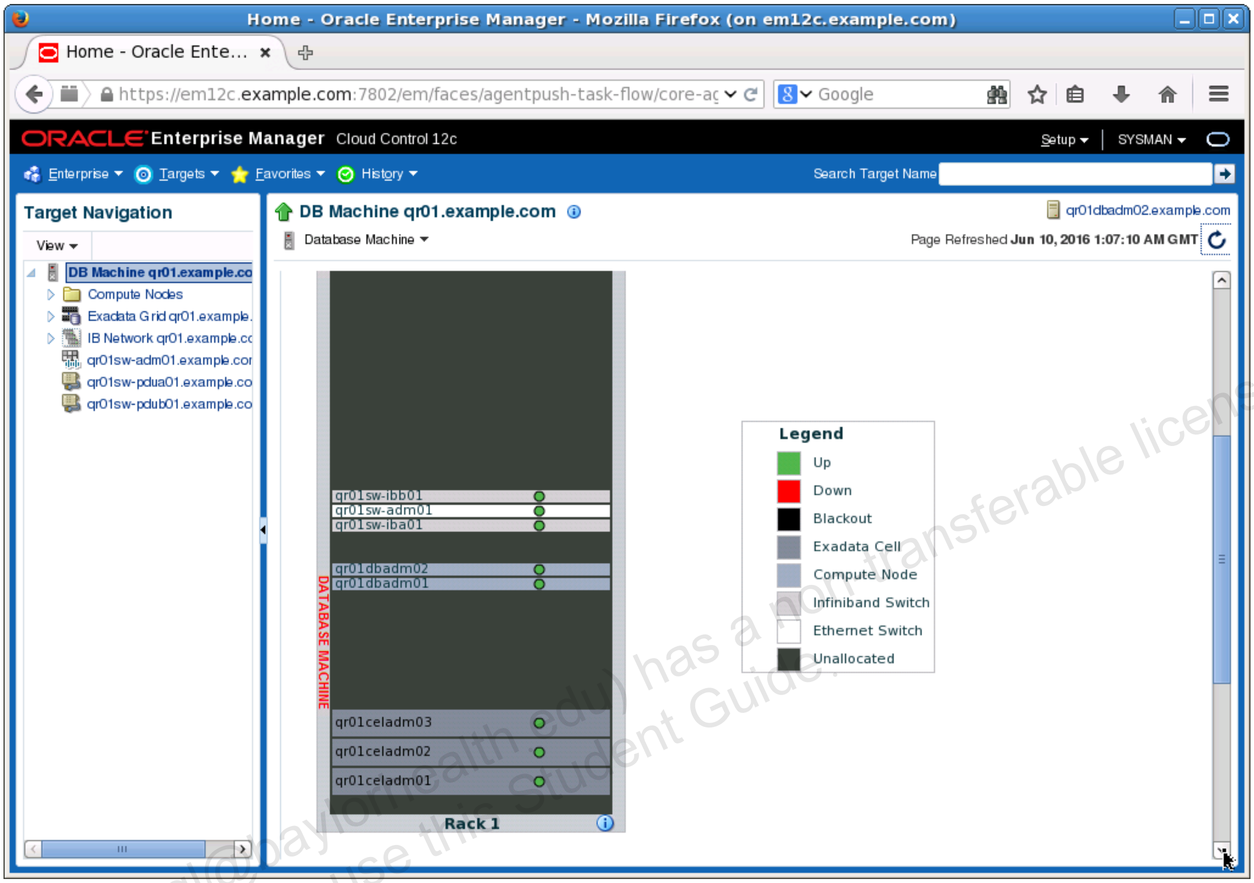


Congratulations! At this point, you have successfully discovered your Database Machine. In the next part of this practice, you will take a brief tour of the Database Machine home page.

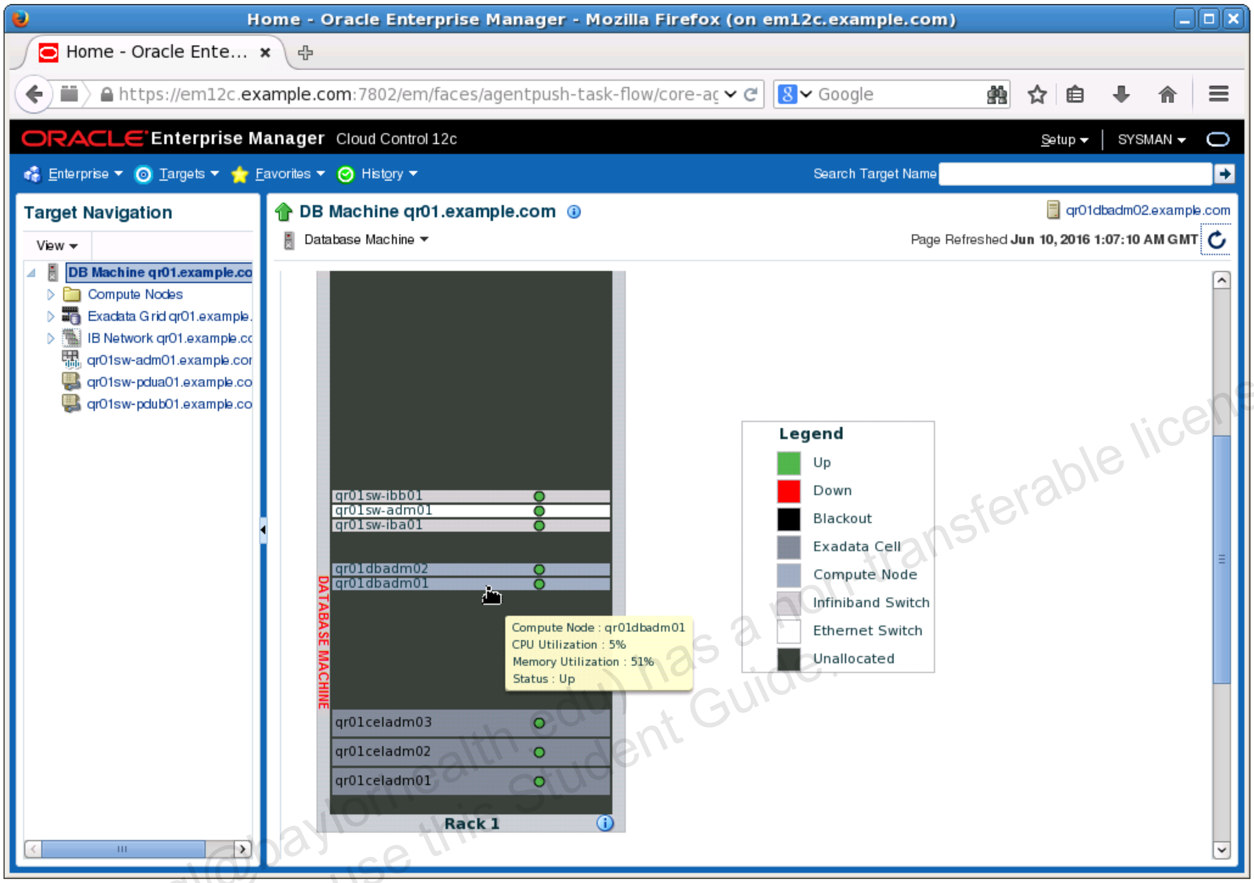
46. Examine the Database Machine home page. Notice that the various Database Machine components are organized in the Target Navigation pane. At the top of the main DB Machine pane, an Overview area summarizes any incidents and the component status. Below the overview, you will find the Database Machine Schematic, which is a diagrammatic representation of the Database Machine.
Note that it will take a few minutes for Enterprise Manager to fully update the status of the Database Machine components. Therefore, if any of the components are initially listed as down, periodically refresh the page until the status of the entire Database Machine is displayed as up.



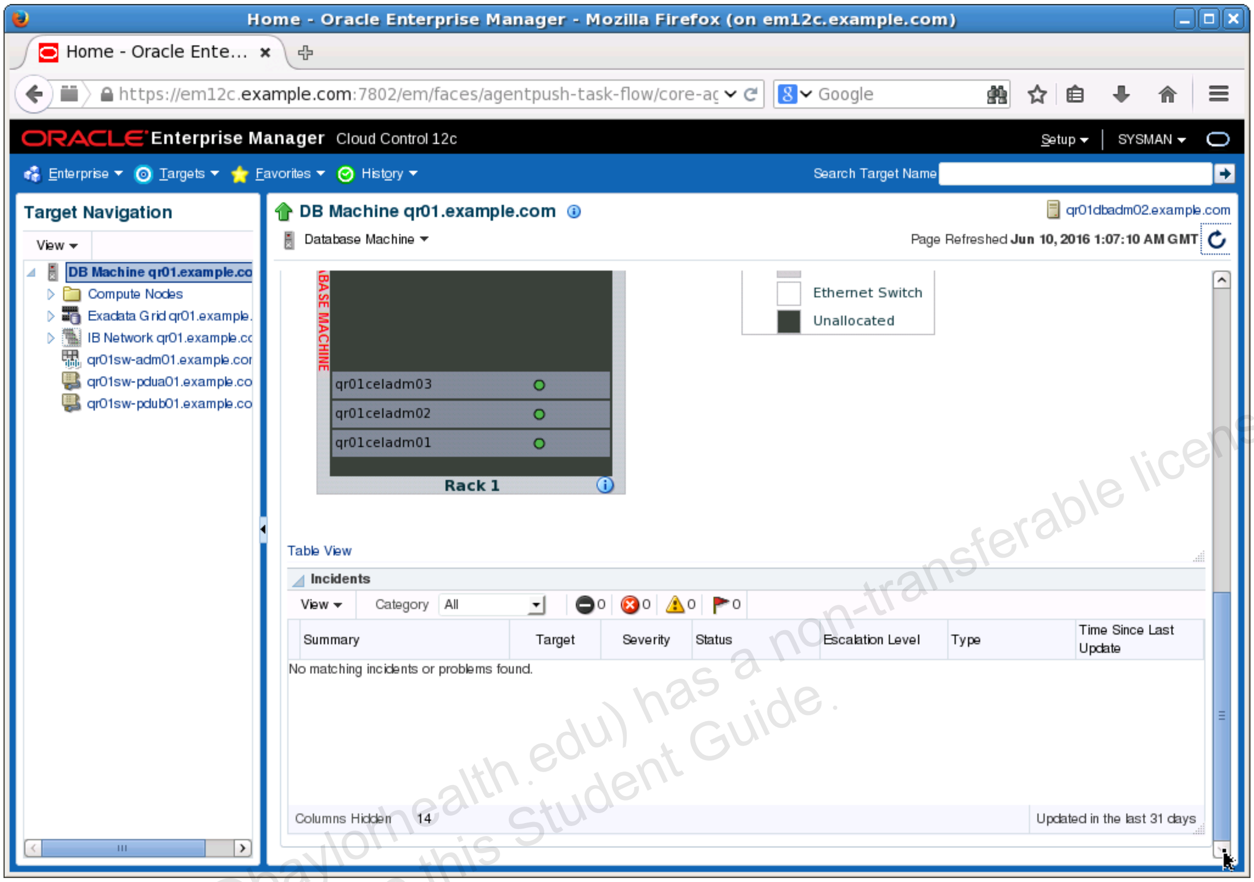
47. Scroll down until the lower half of the Database Machine Schematic diagram is displayed. Here you will see a representation of the Database Machine hardware components arranged just as they are inside the rack.



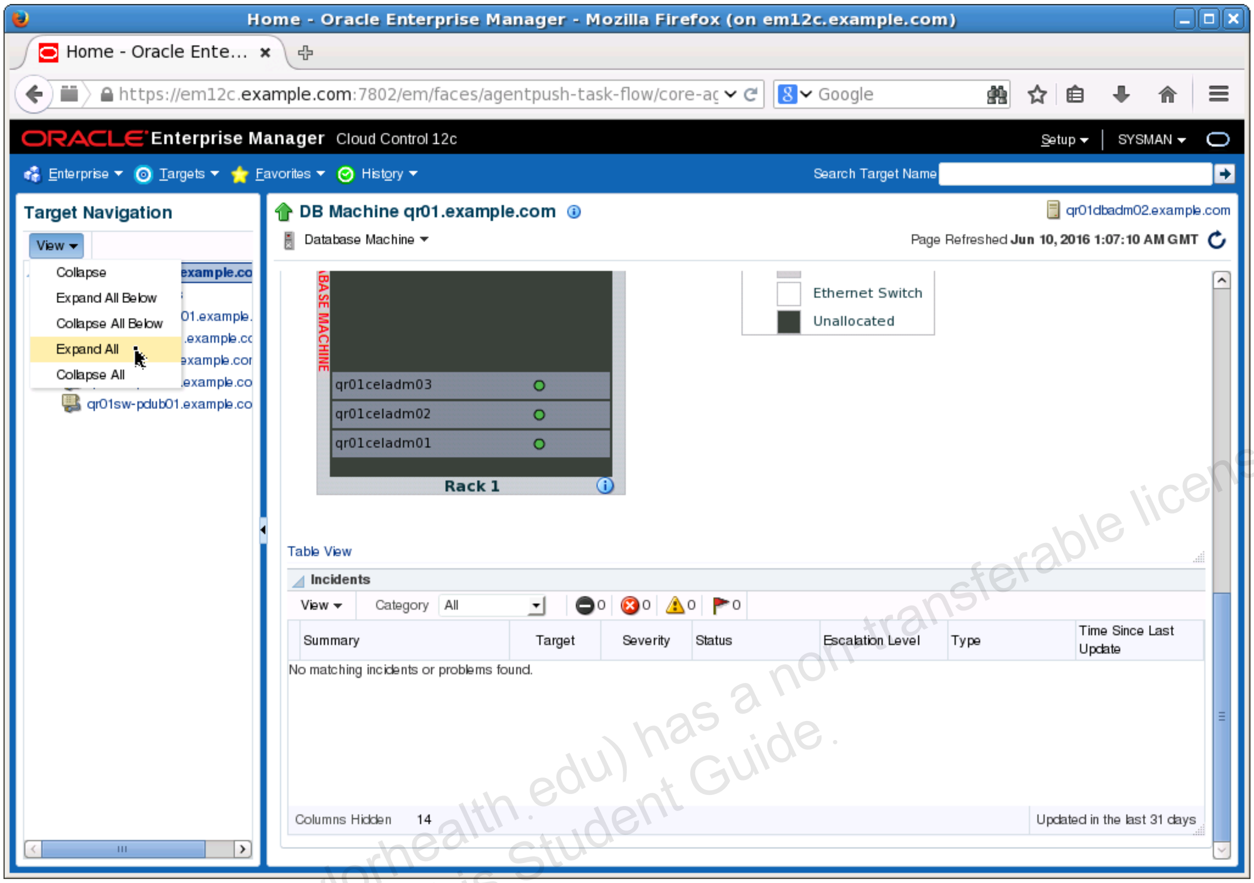
48. Hold the cursor over each component in the diagram to display key status information. Note that you may initially see zero values for the utilization metrics. This will change over time as the metric values are collected.



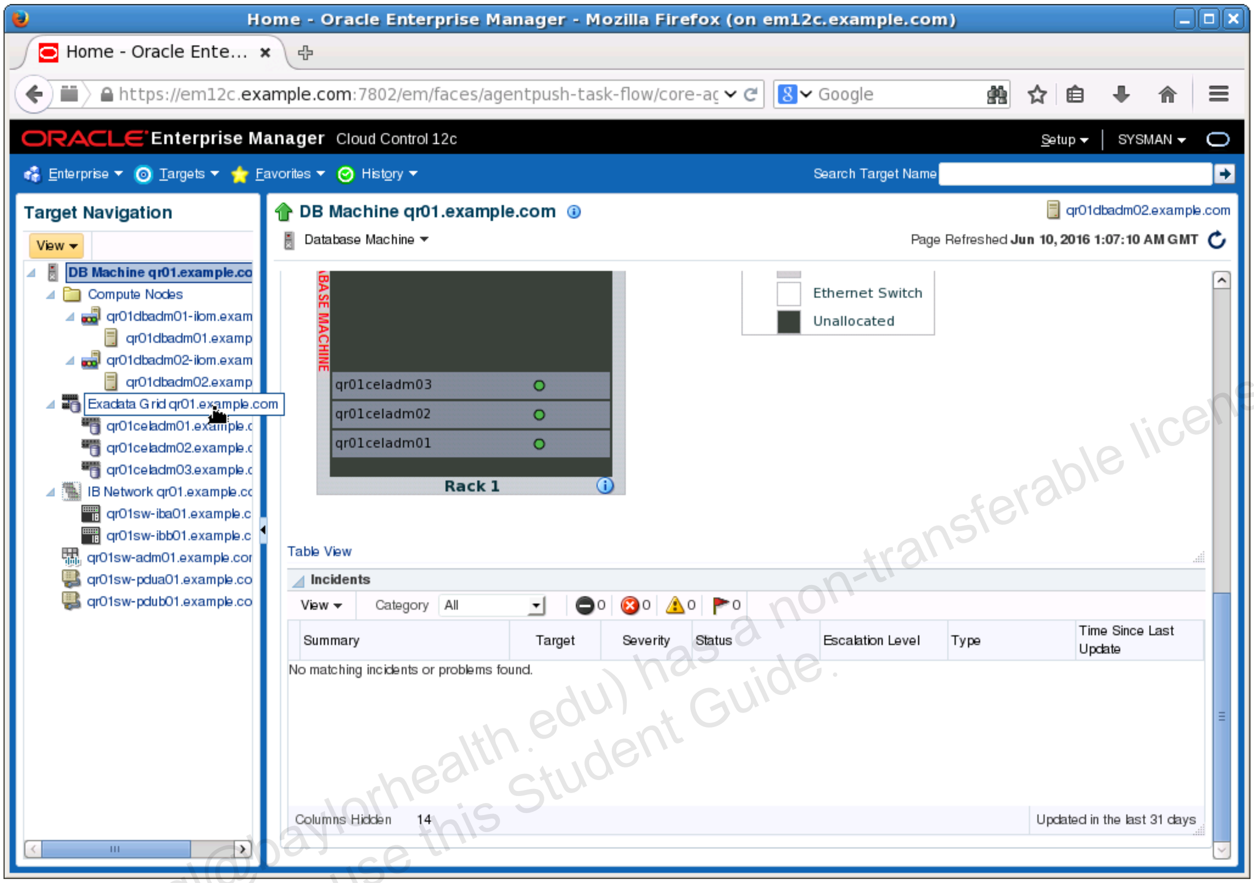
49. Scroll further down the page to reveal the Incidents area. Use the summary shown here to quickly ascertain if there are any incidents relating to the Database Machine.



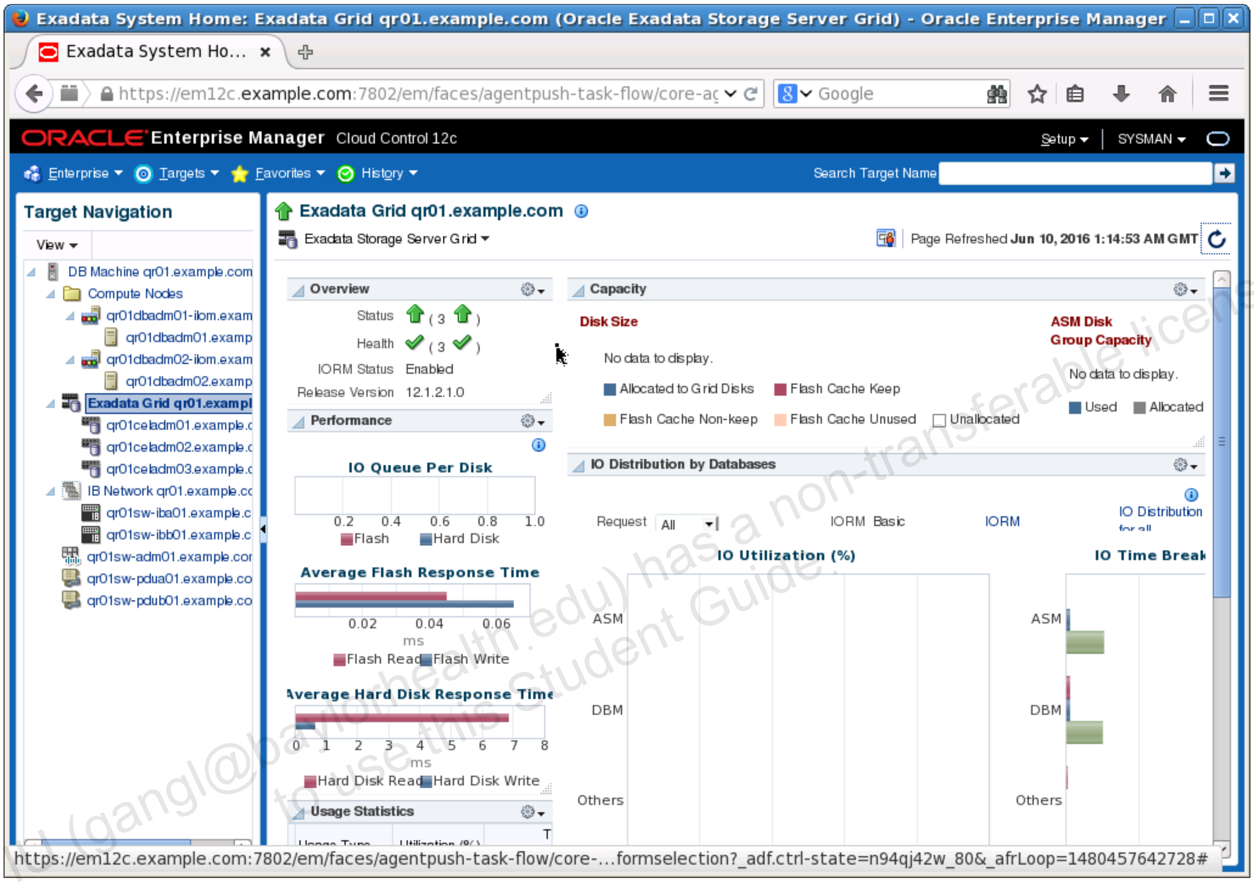
50. In the Target Navigation pane, select the View > Expand All menu command.



51. Now the Target Navigation pane lists all the monitoring targets associated with the Database Machine. At any time you can navigate to the target home page by clicking any of the listed targets. Click “Exadata Grid qr01.example.com.”

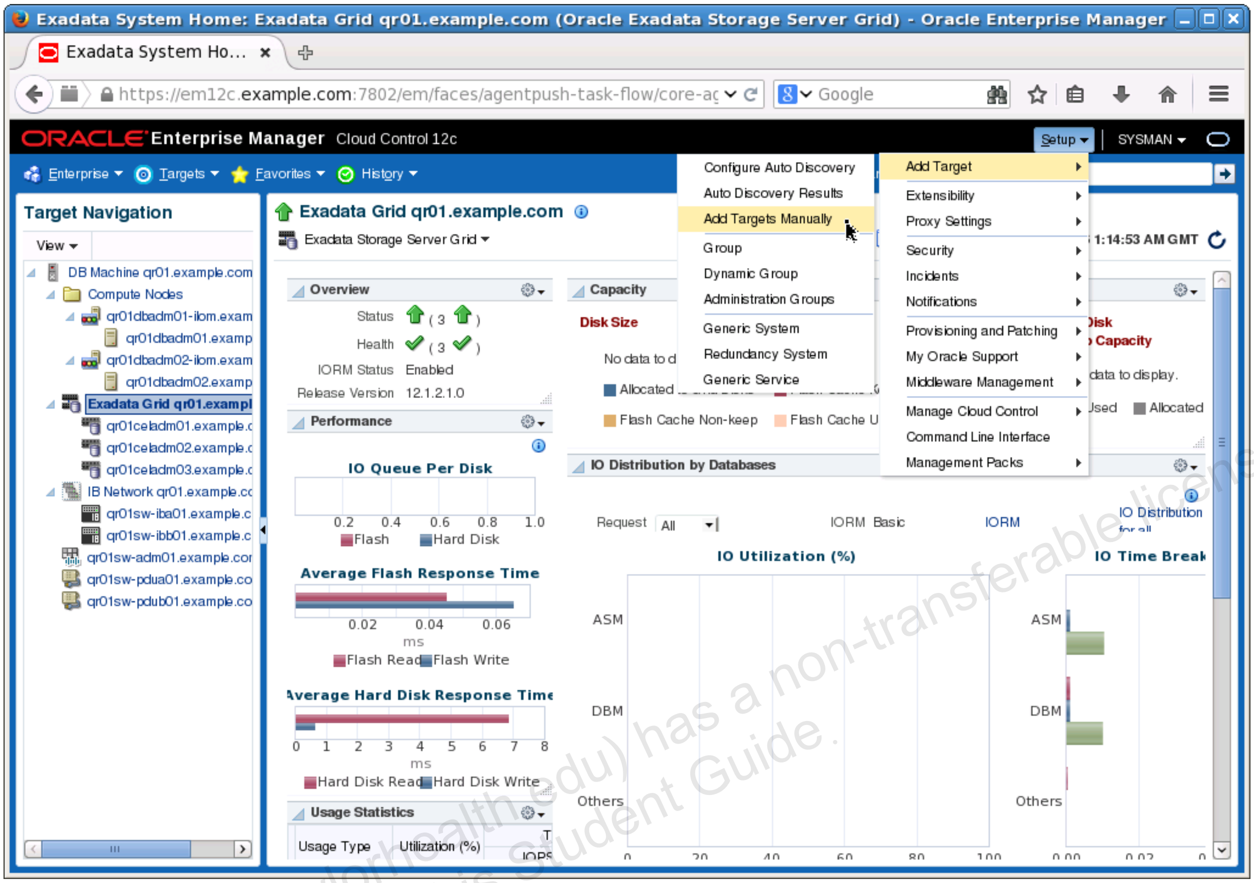


52. Now the Exadata Grid home page is displayed. This page summarizes key information about all the Exadata Storage Servers in the Database Machine. At this point, you can see that the storage grid consists of three Exadata Storage Servers. You can also notice that many of the summaries on the page are currently empty or contain zero values. Over time this will change as Enterprise Manager collects the underlying metrics and computes the summary values.

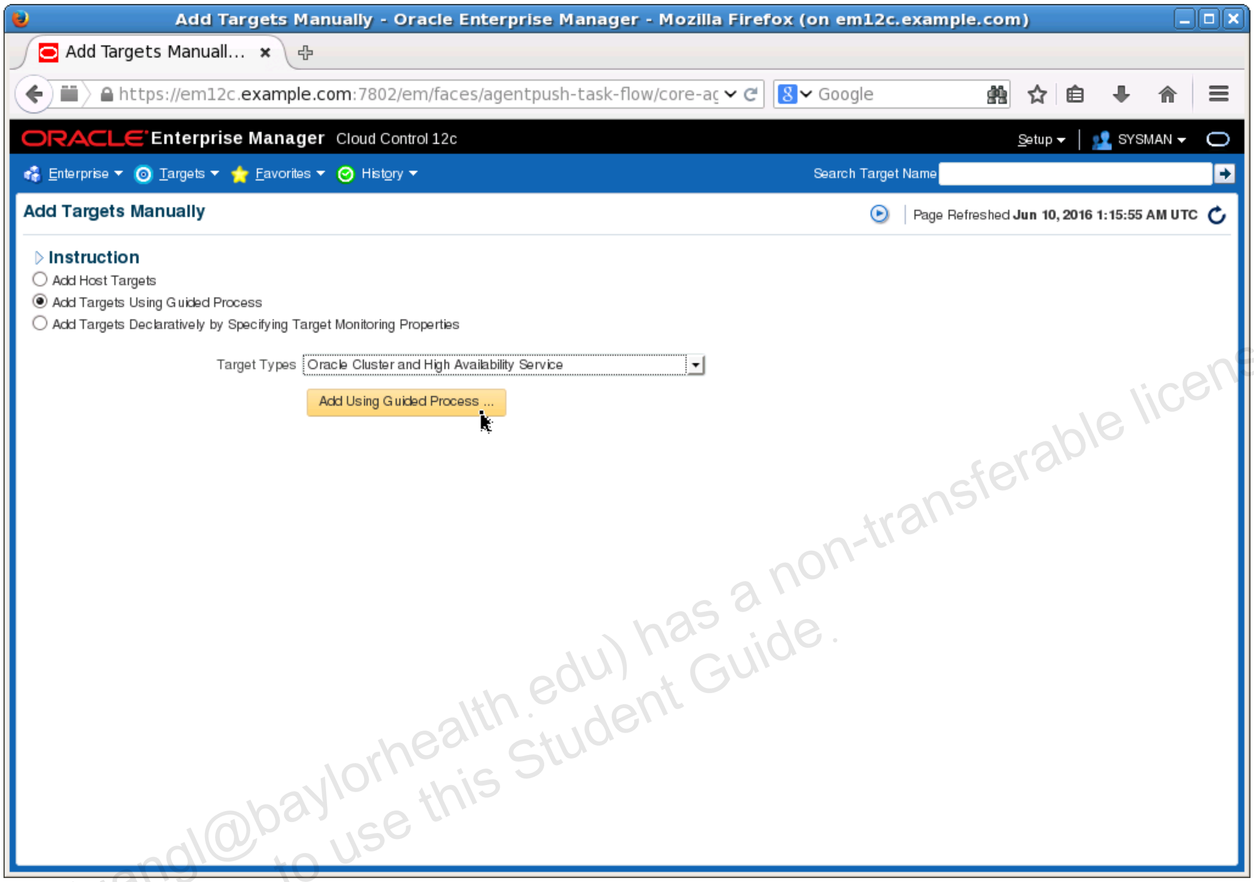


You may have noticed that the Database Machine home page contains no information about the databases and other software services running on the Database Machine. You will discover these separately in the next part of this practice.

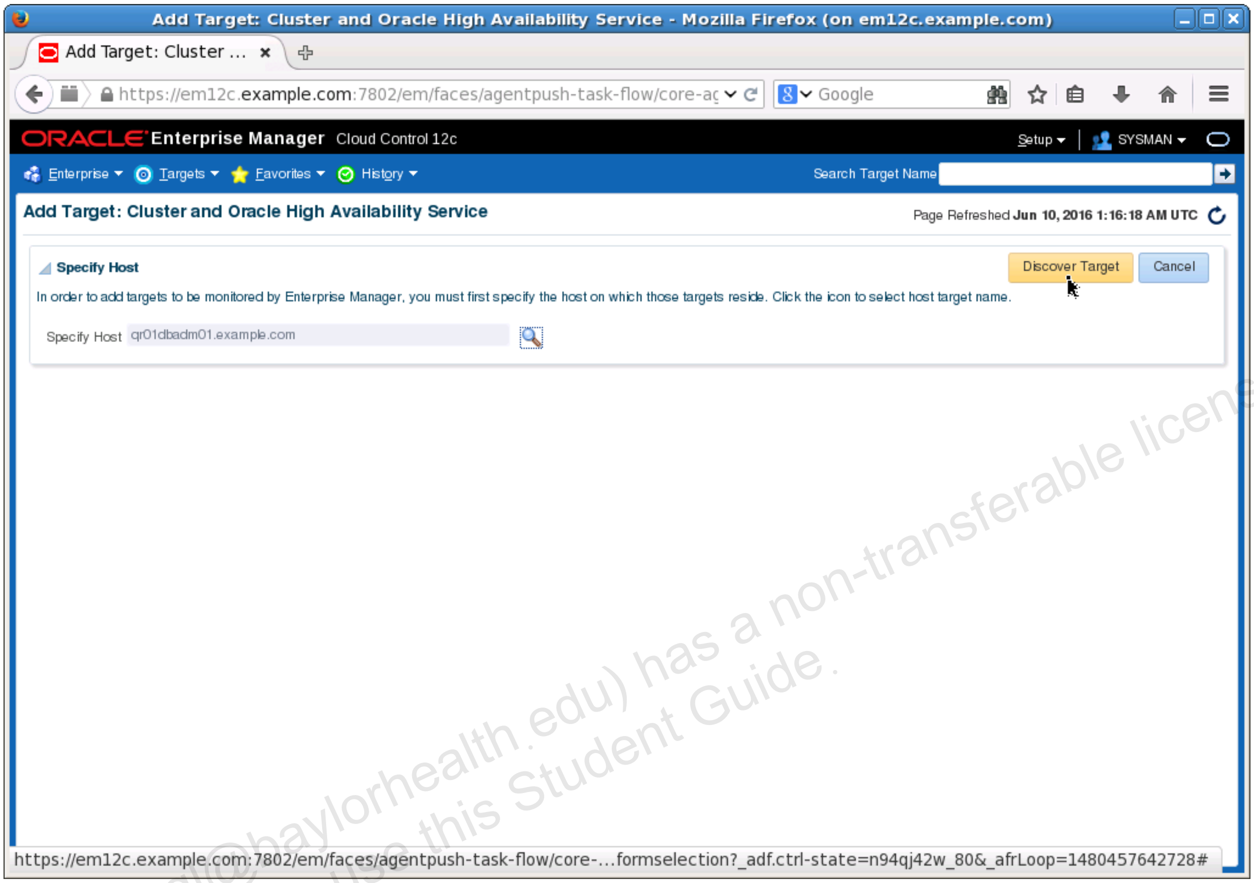
53. Select the Setup > Add Target > Add Targets Manually menu command.



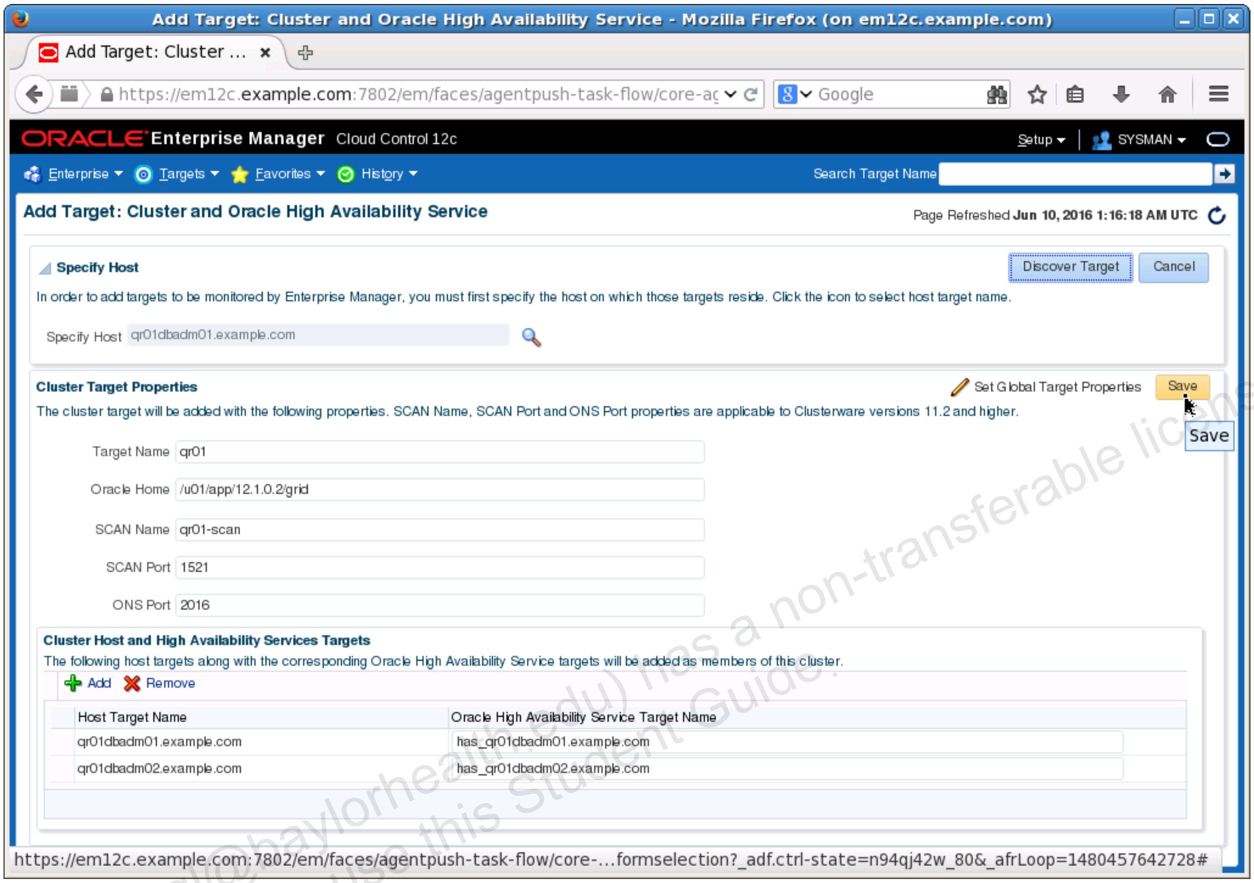
54. On the Add Targets Manually page, select the Add Targets Using Guided Process option. Then select “Oracle Cluster and High Availability Service” from the Target Types drop-down list. Finally, click Add Using Guided Process to start the cluster discovery process. Note that you must discover the cluster before you can discover any databases.



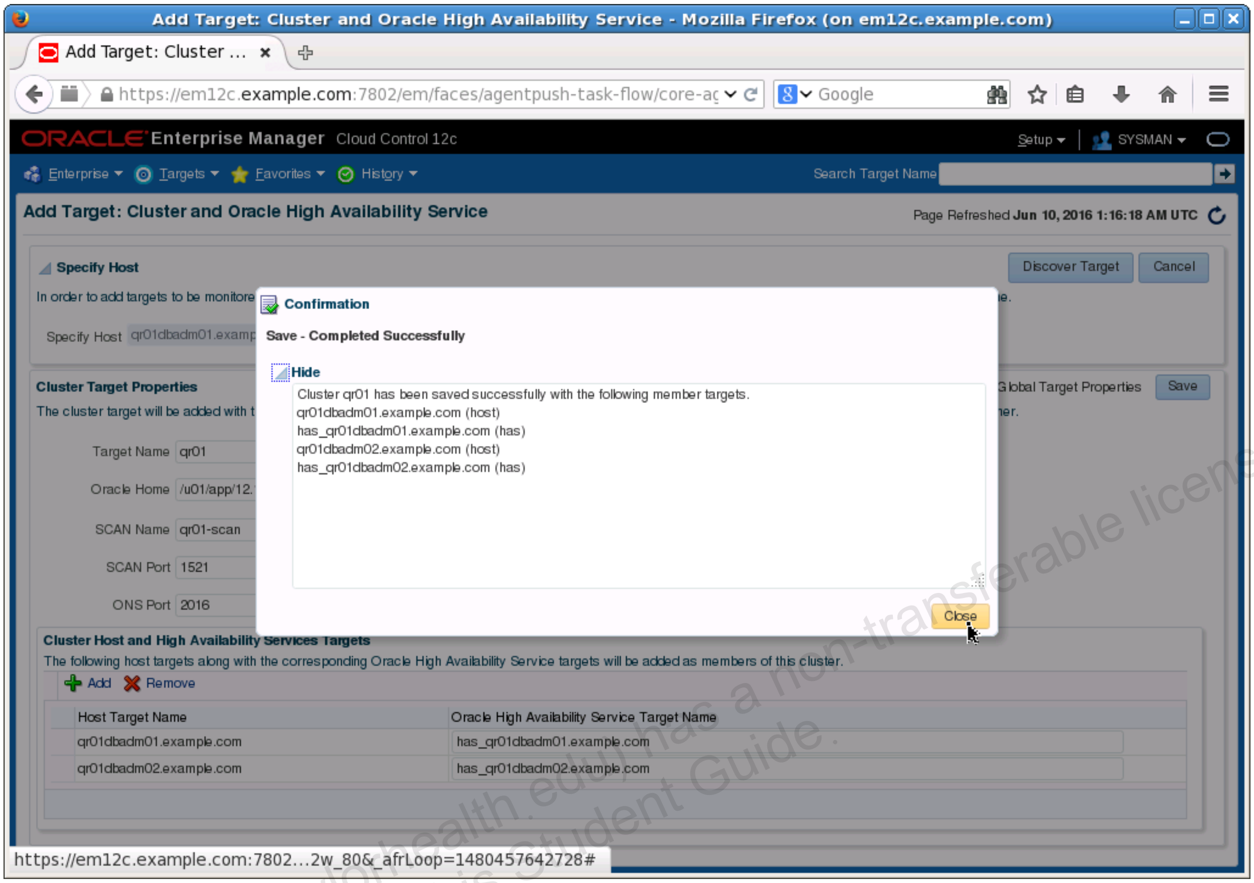
55. Enter the name of one of the database server hosts (`qr01dbadm01.example.com`) by clicking the magnifying glass icon and using the resulting dialog box to specify the host. Then click Discover Target to proceed.



56. The Add Target: Cluster and Oracle High Availability Service page should display various attributes of the cluster. Verify the accuracy of the cluster attributes and click Save to proceed.

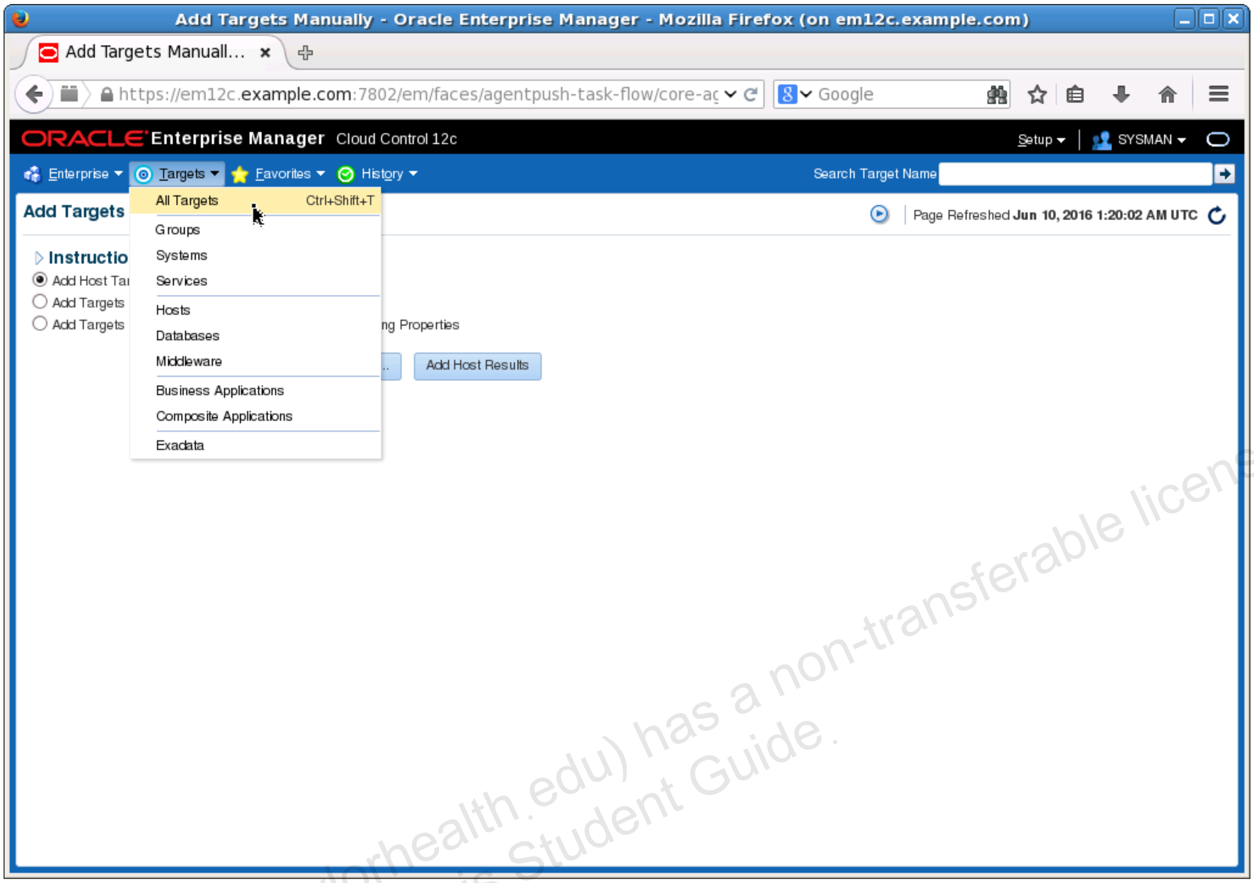


57. When the results page is displayed. Ensure that there are no errors and click Close.

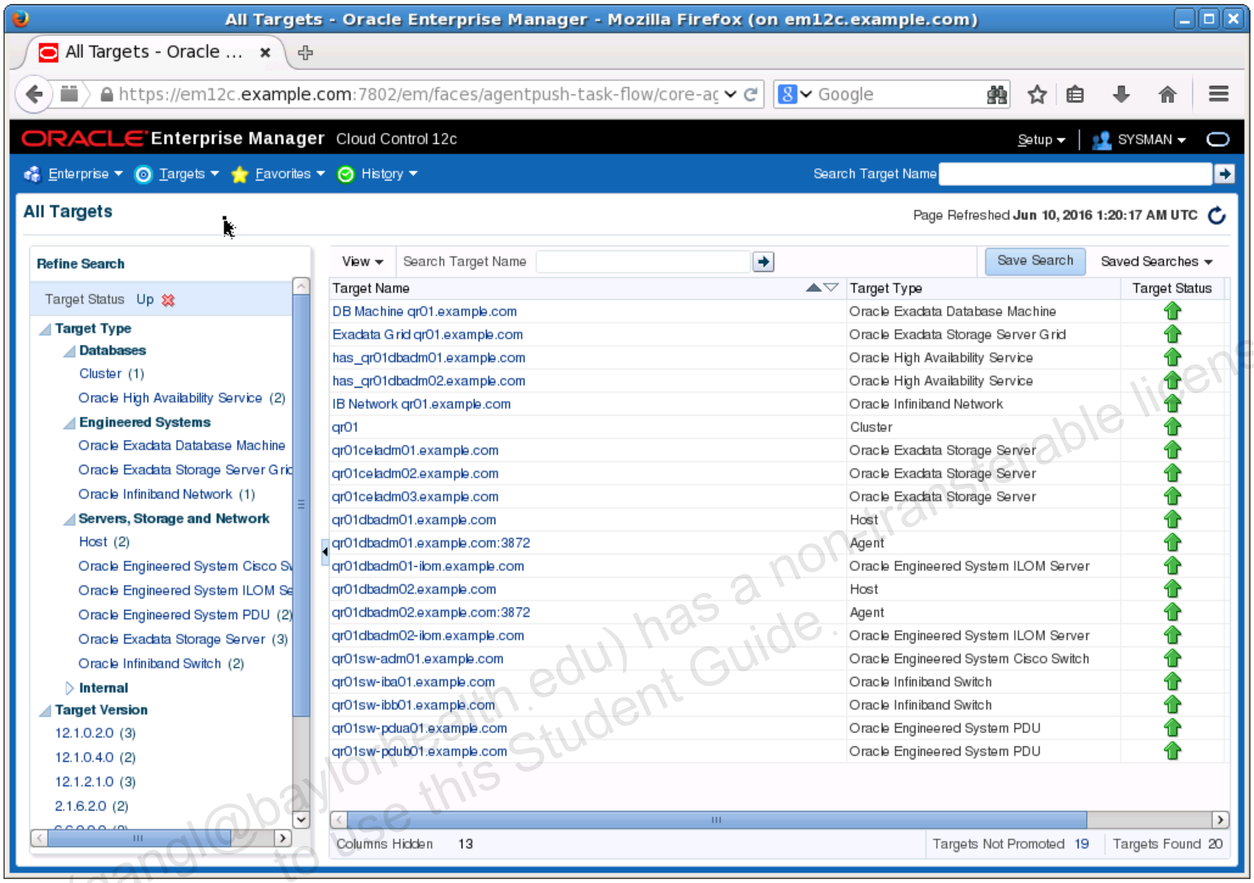


At this point, the cluster has been added to Enterprise Manager. Next, you will validate that all the cluster targets are up and available.

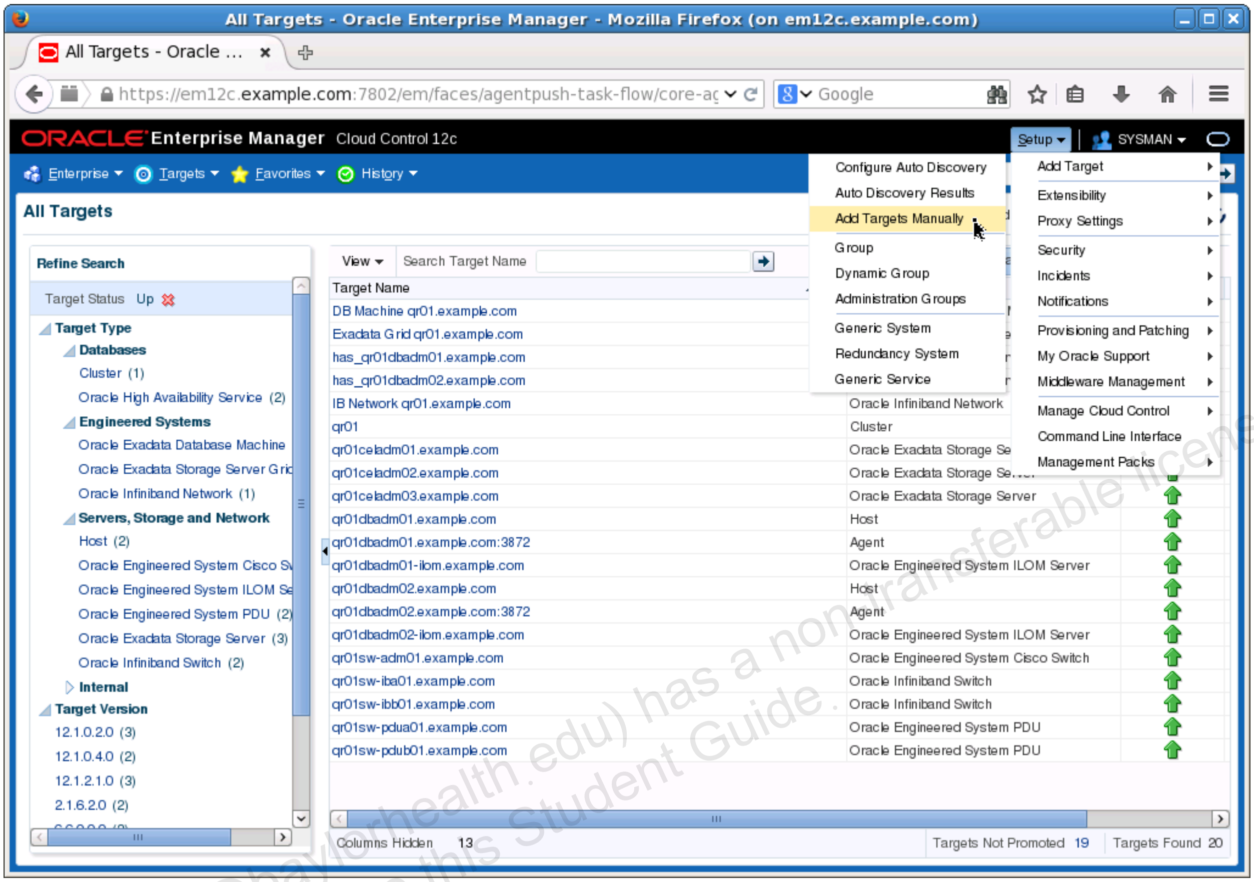
58. Select the Targets > All Targets menu command.



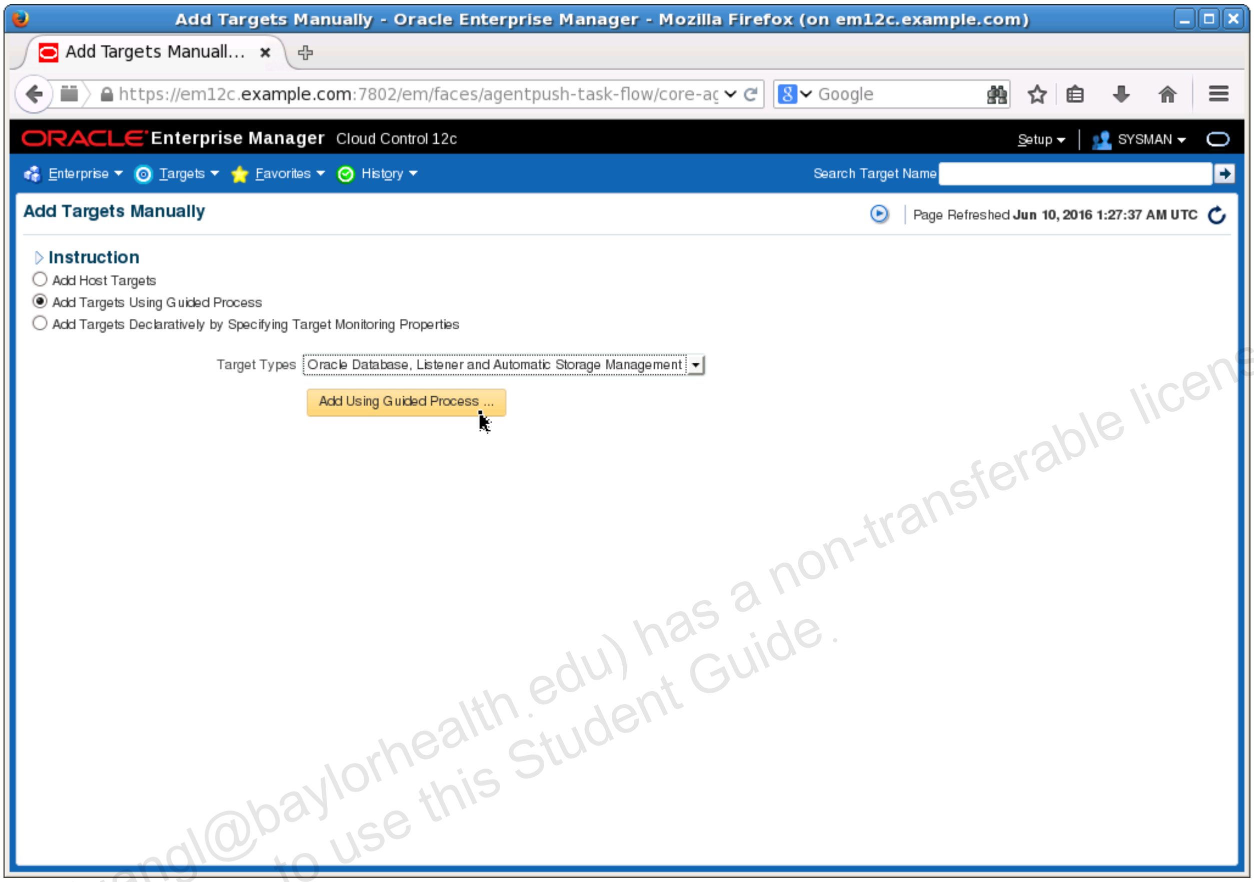
59. Examine the target list and confirm that the cluster target (`qr01`) and two high availability service targets (`has_qr01dbadm01.example.com` and `has_qr01dbadm02.example.com`) are up. If any of the targets are not up, periodically refresh the page until they appear in the list.



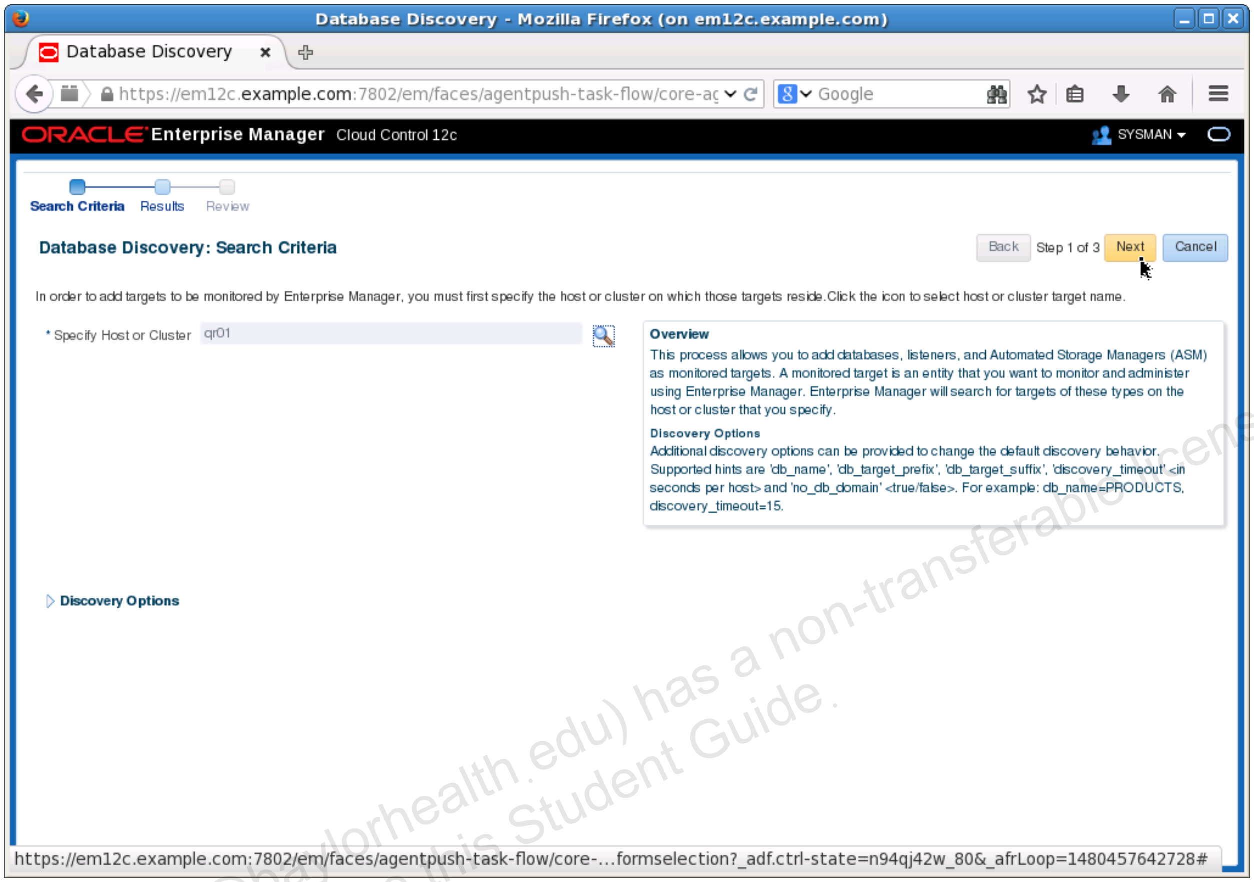
60. Now you are ready to discover the databases running on your Database Machine. To start this process, select the Setup > Add Target > Add Targets Manually menu command.



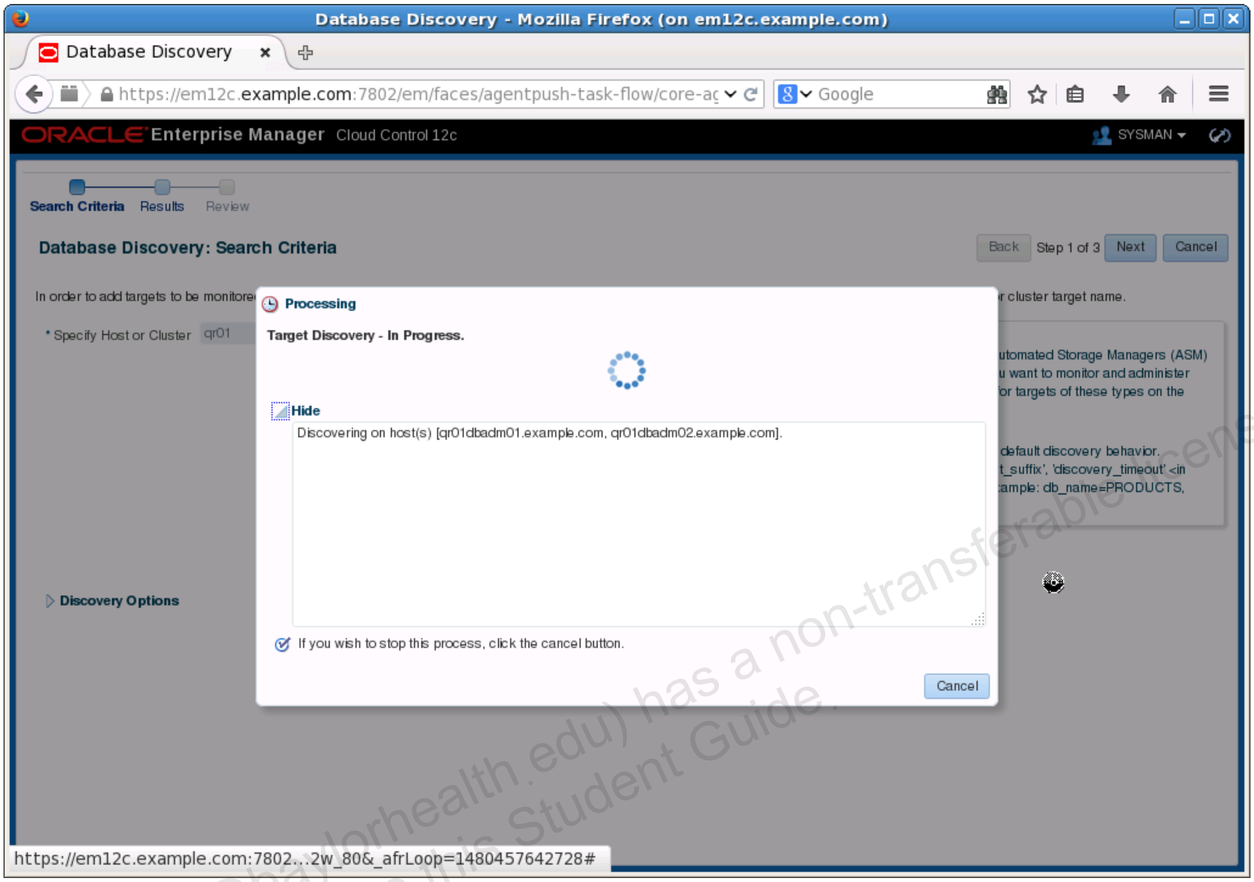
61. On the Add Targets Manually page, select the Add Targets Using Guided Process option. Then select “Oracle Database, Listener and Automatic Storage Management” from the Target Types drop-down list. Finally, click Add Using Guided Process to start the discovery process.



62. Enter the name of the cluster (`qr01`) by clicking the magnifying glass icon and selecting the cluster in the dialog box. Then click Next to proceed.



63. While target discovery is in progress proceed to the next steps to configure the monitoring credentials on your database.



64. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

65. Connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0...

SQL>
```

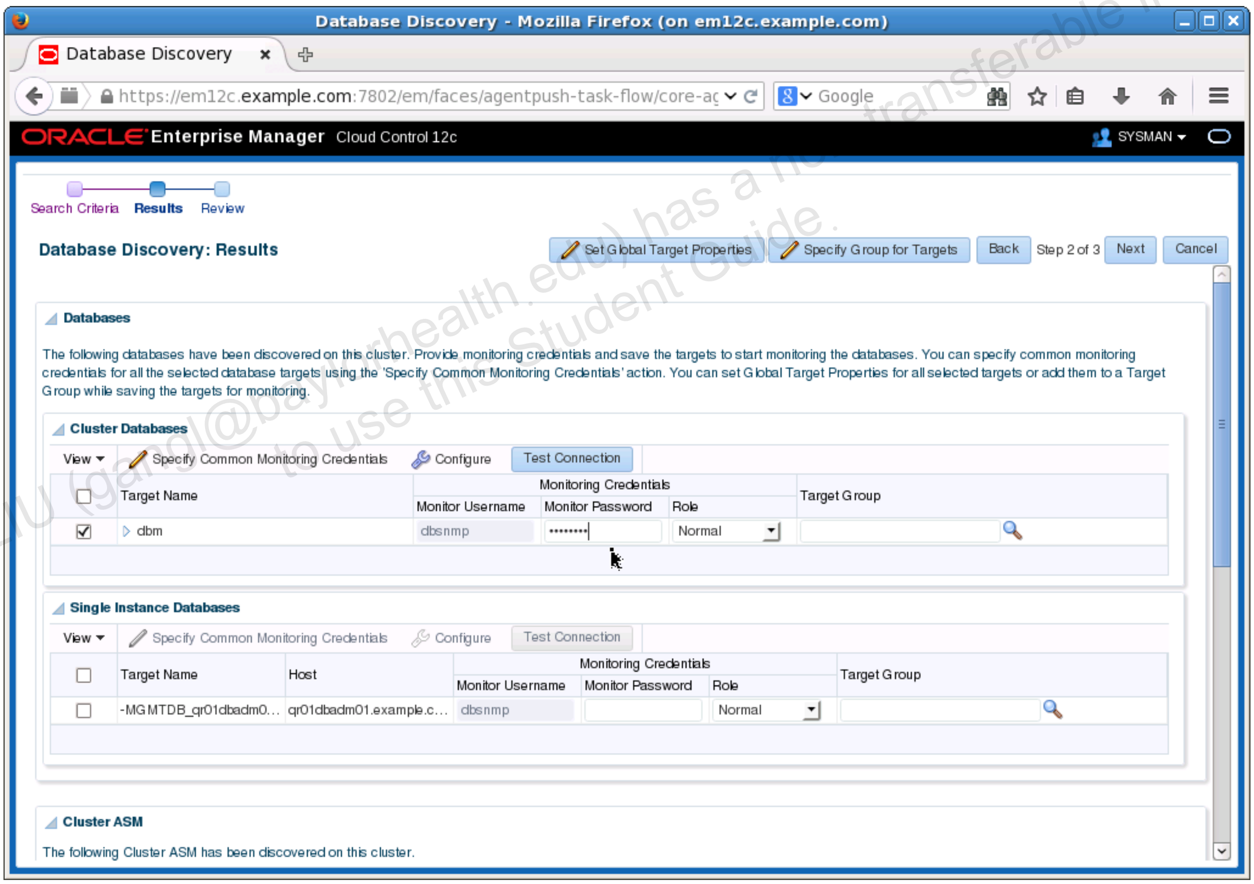
66. Execute the `ALTER USER` command to configure the monitoring credentials for the database and remember your specified password.

```
SQL> alter user dbsnmp account unlock identified by ????????;

User altered.

SQL>
```

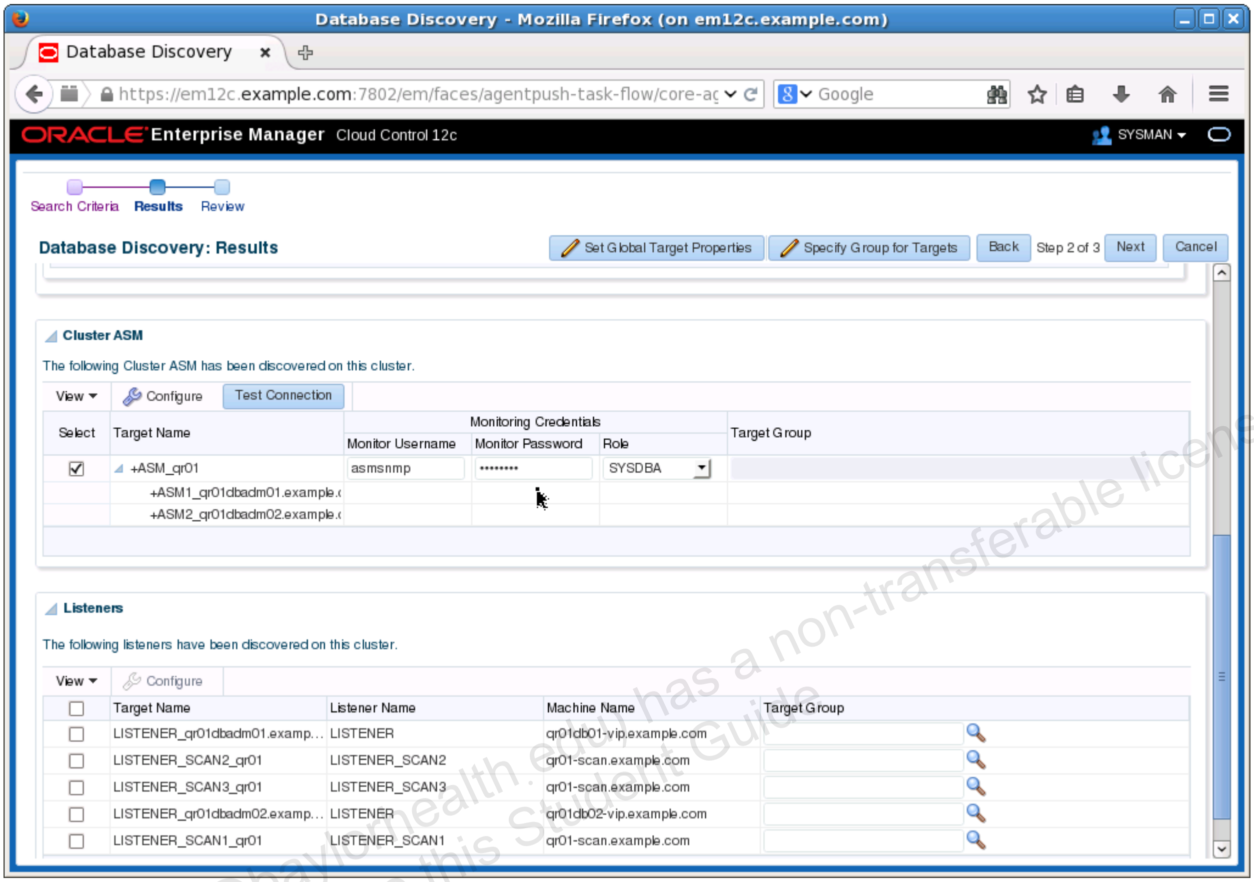
67. Exit SQL*Plus.

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition ...
[oracle@qr01dbadm01 ~]$
```

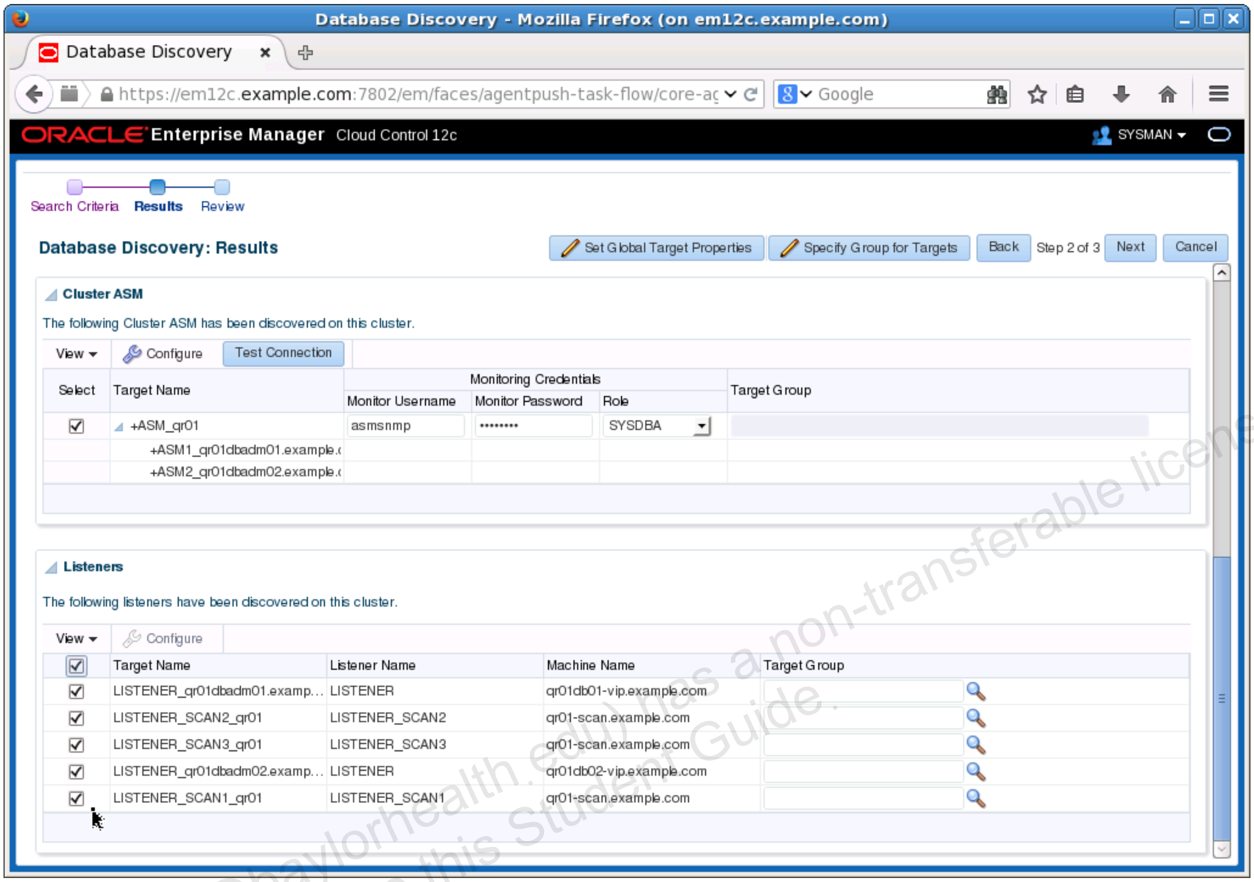
68. On the Database Discovery: Results page, click the check box beside the `dbm` database and enter the Monitor Password that you set in step 66.



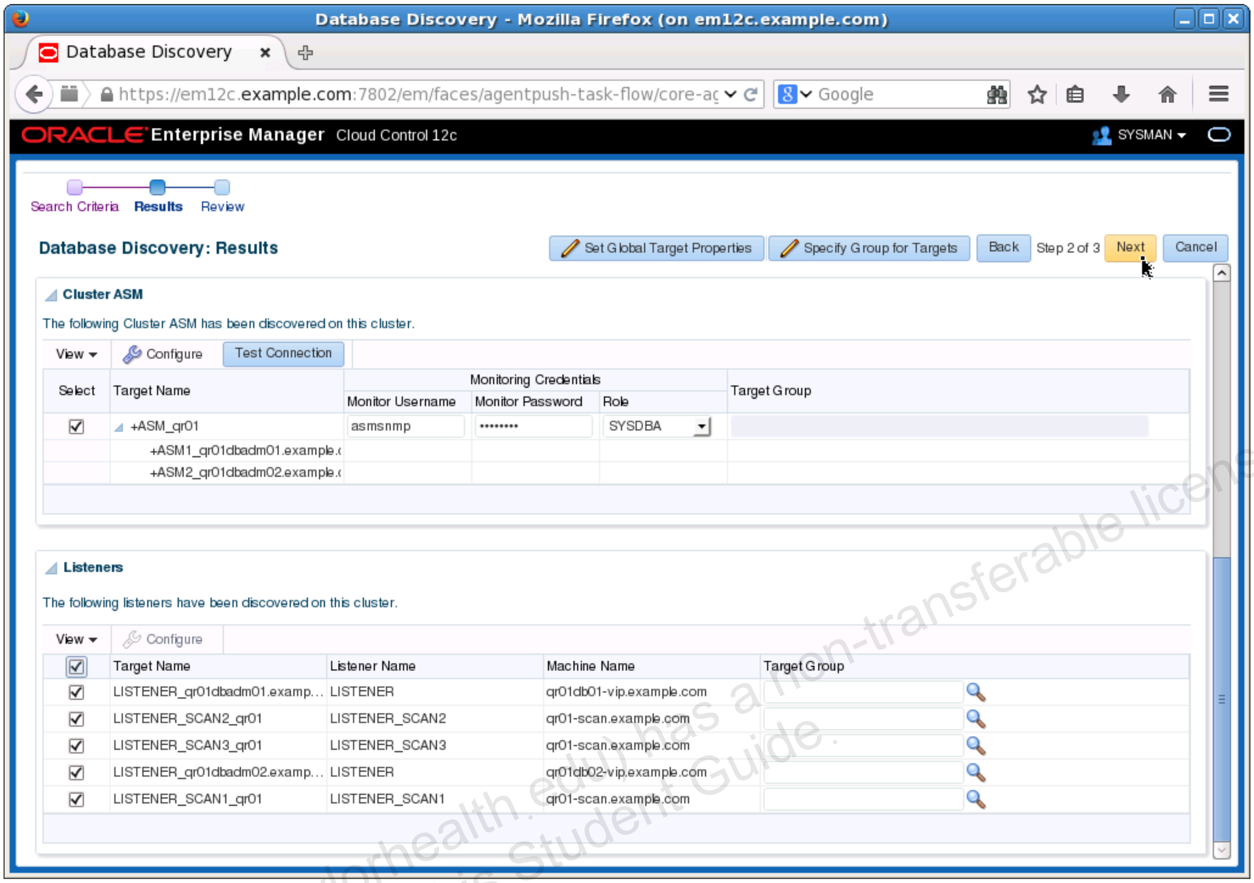
69. Scroll down the Database Discovery: Results page, click the check box beside the ASM target and enter the ASM Monitor Password.



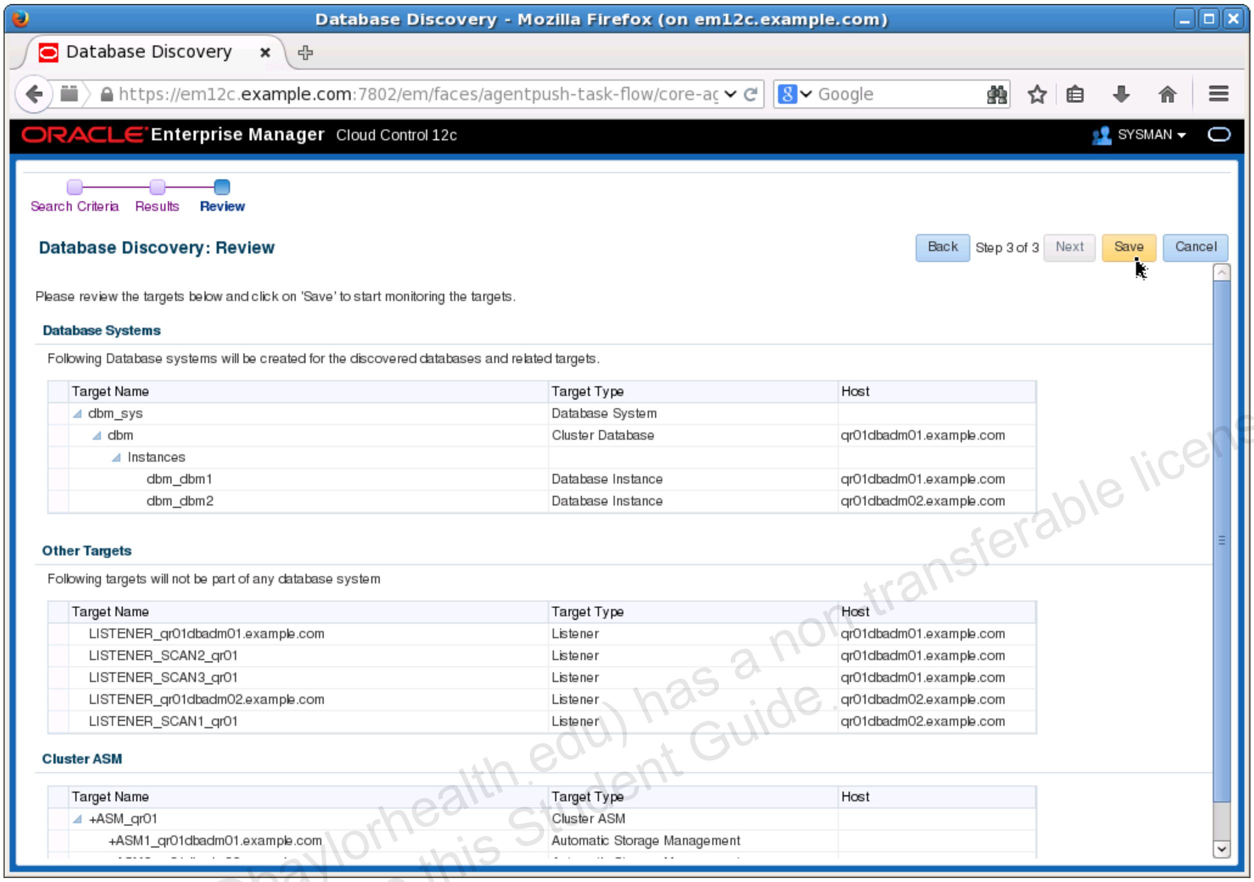
70. Scroll to the bottom of the Database Discovery: Results page, click all of the check boxes beside all of the Listeners.



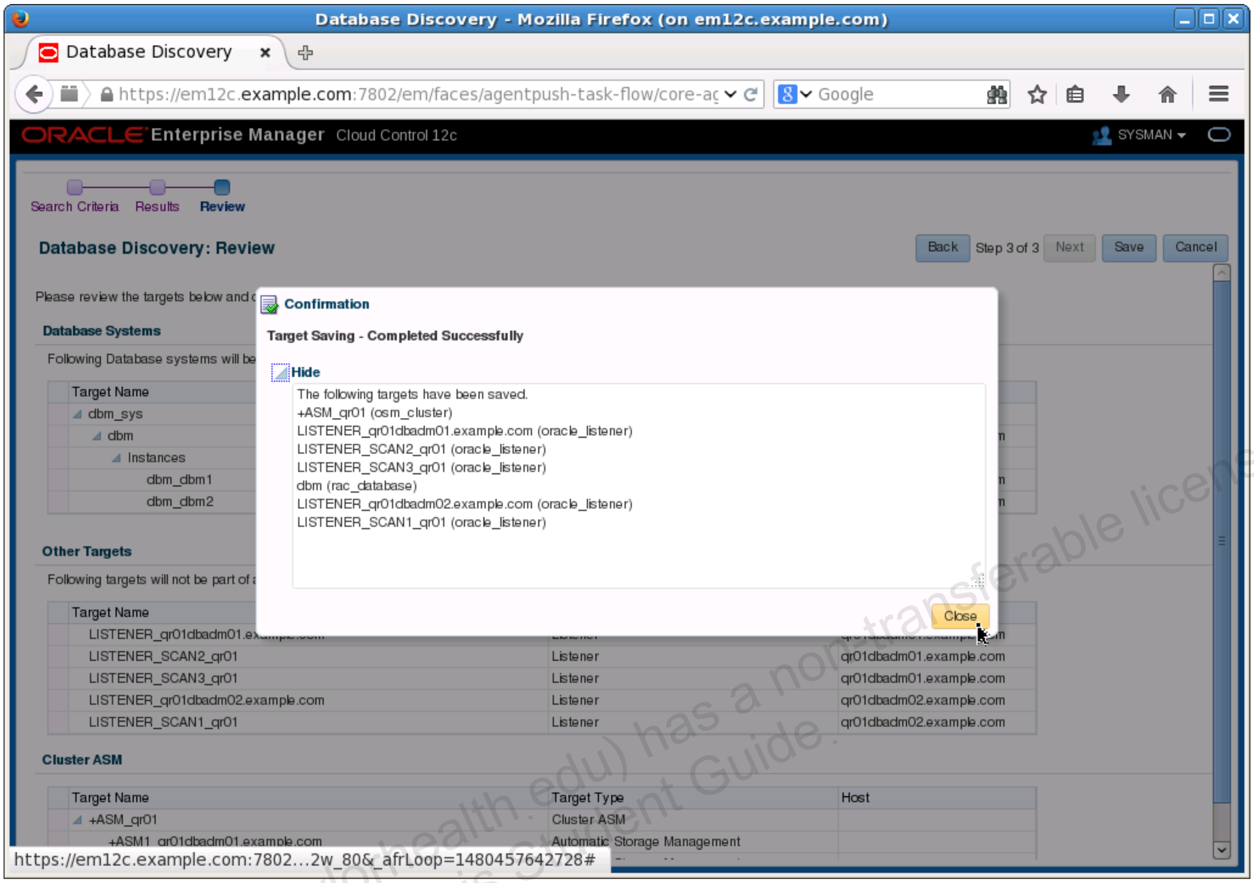
71. Click Next to continue.



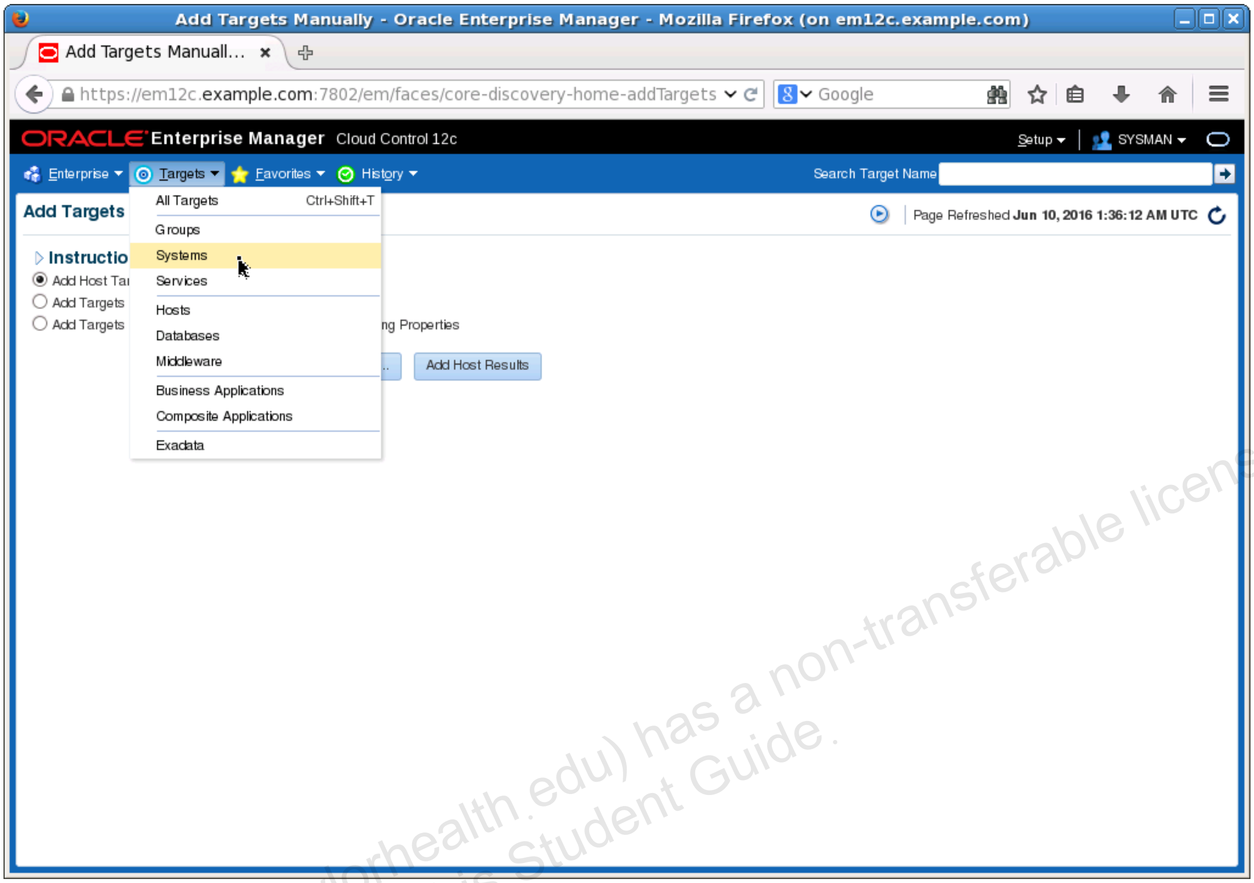
72. On the summary page that appears, click Save to add the database, listener, and ASM targets to Enterprise Manager.



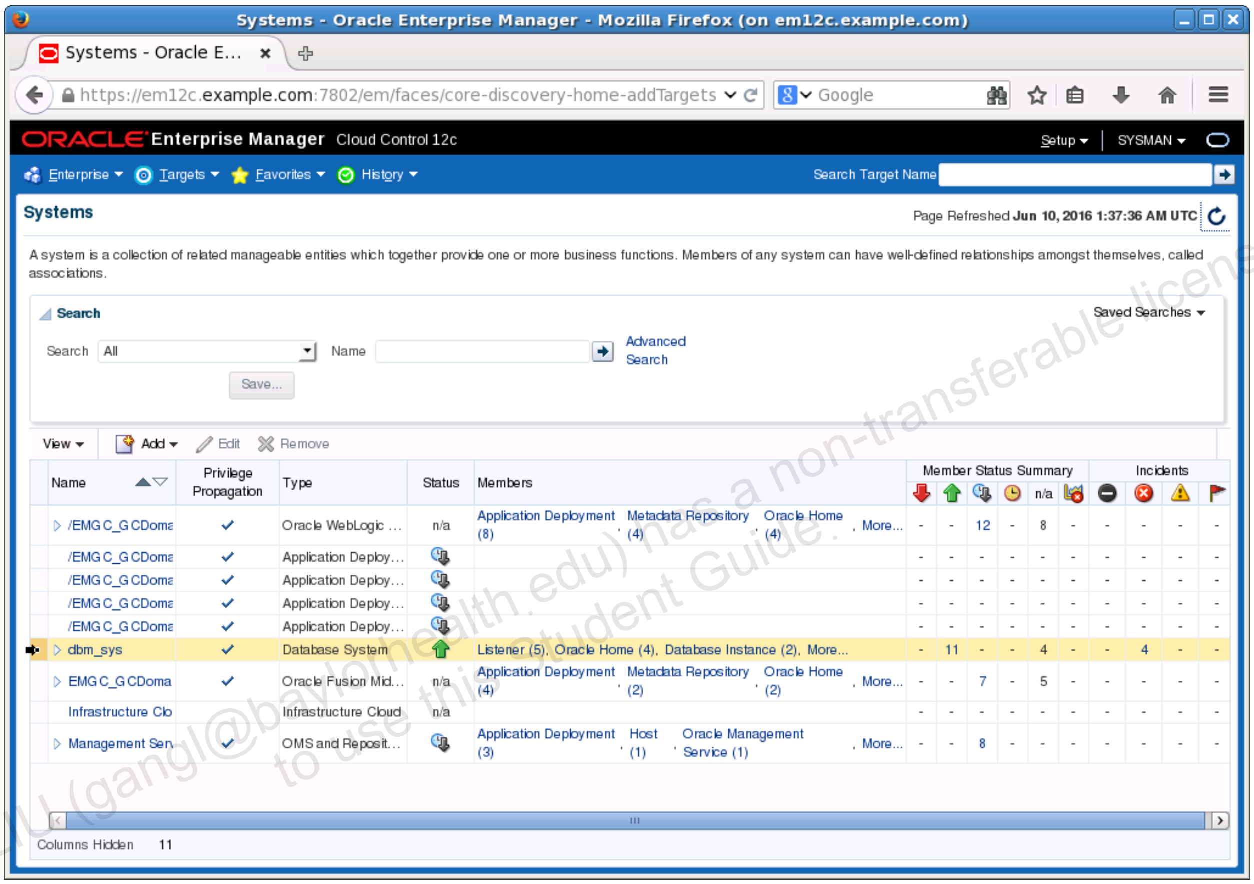
73. Examine the results page to ensure that no errors are reported. Click Close to proceed.



74. Select the Targets > Systems menu command.

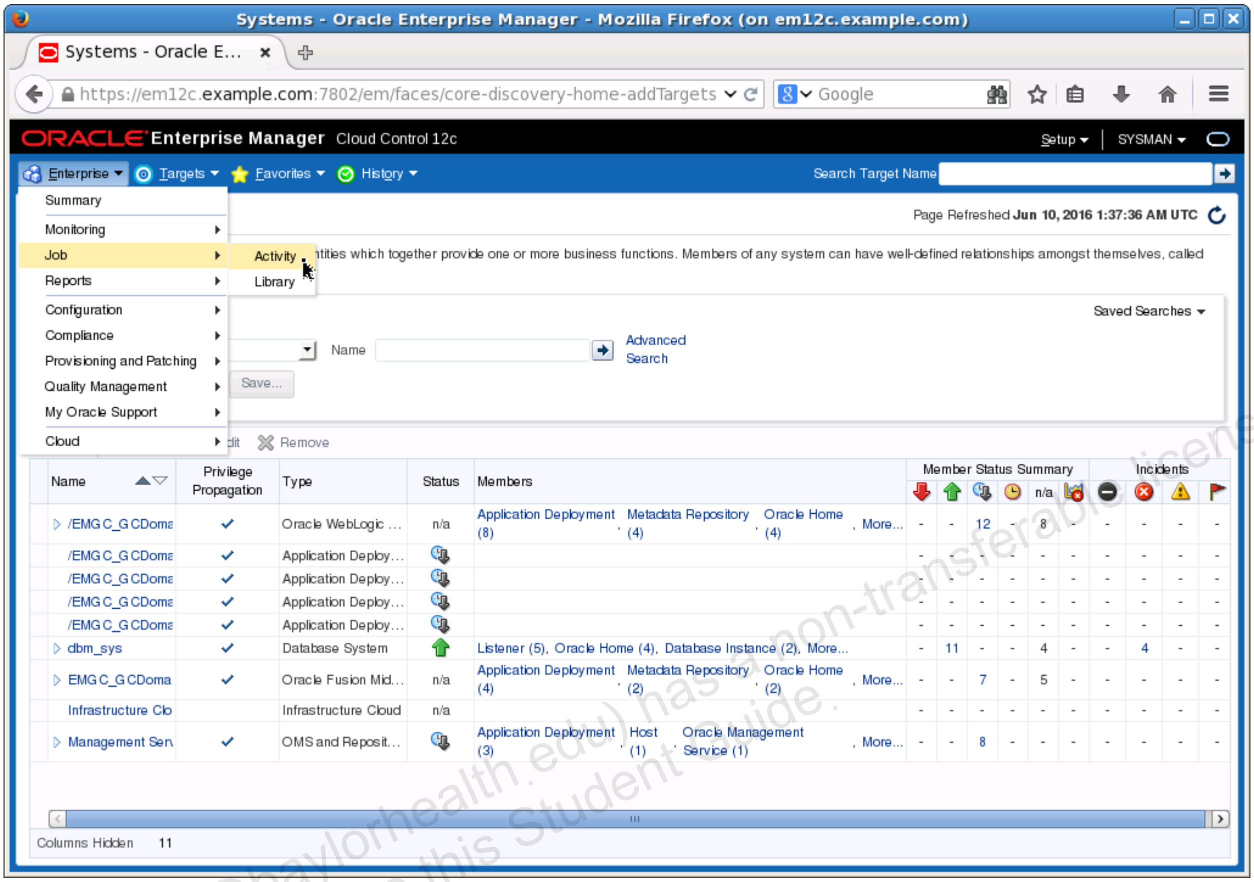


75. In Enterprise Manager, a System is a collection of related components. When you discovered the cluster database earlier in this practice, a Database System was implicitly created for you. Initially, the Database System will be associated with the cluster database and its database instances. Over time, Enterprise Manager will automatically add associations to other related entities, such as listeners and ASM. Examine the status of the `dbm_sys` system. Periodically refresh the page until the status is shown as up (displayed as a green upward pointing arrow).

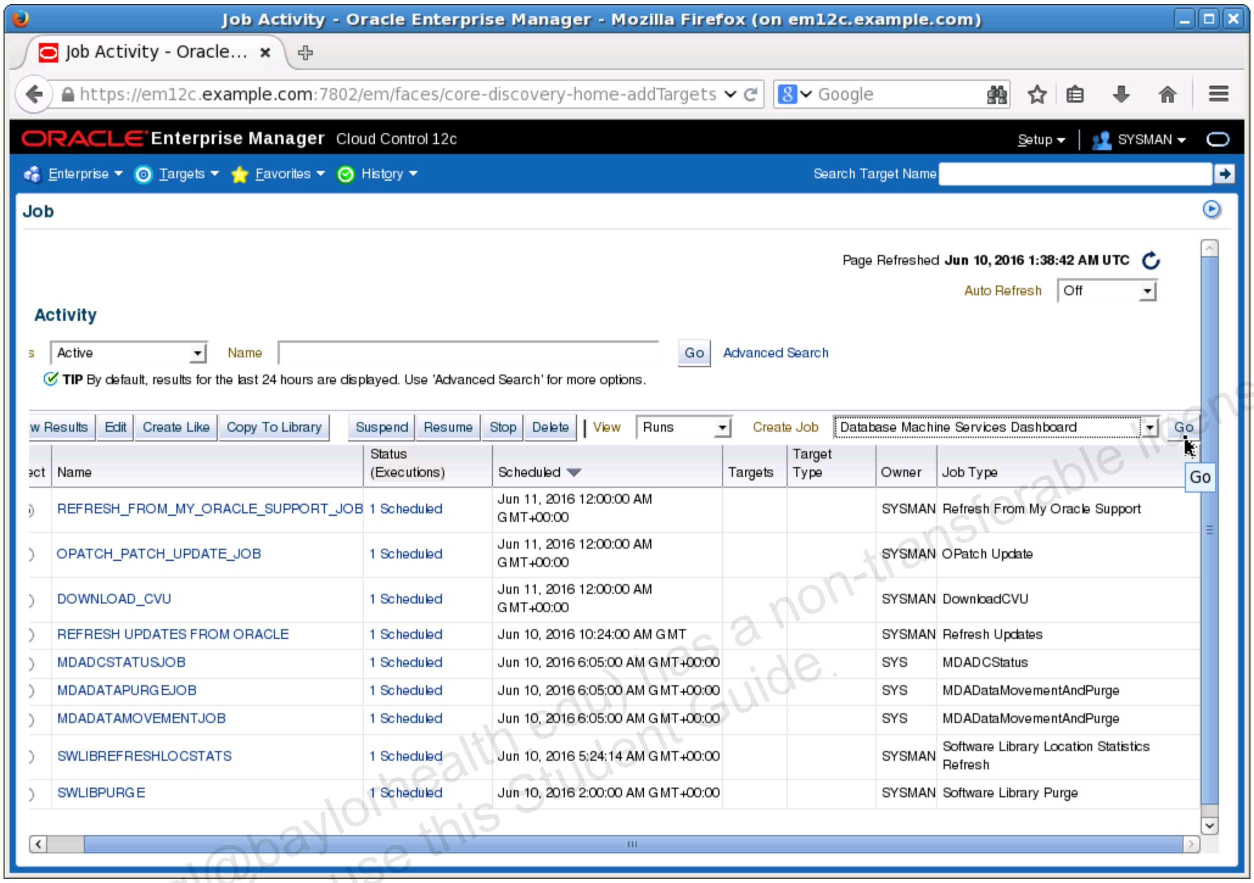


At this point, you have configured your Database Machine and related software services in Enterprise Manager. In the final part of this practice, you will configure a Database Machine Services Dashboard. The dashboard provides a one-stop overview that enables you to quickly and easily monitor the key availability and performance metrics for your Database Machine and the related software services on one page.

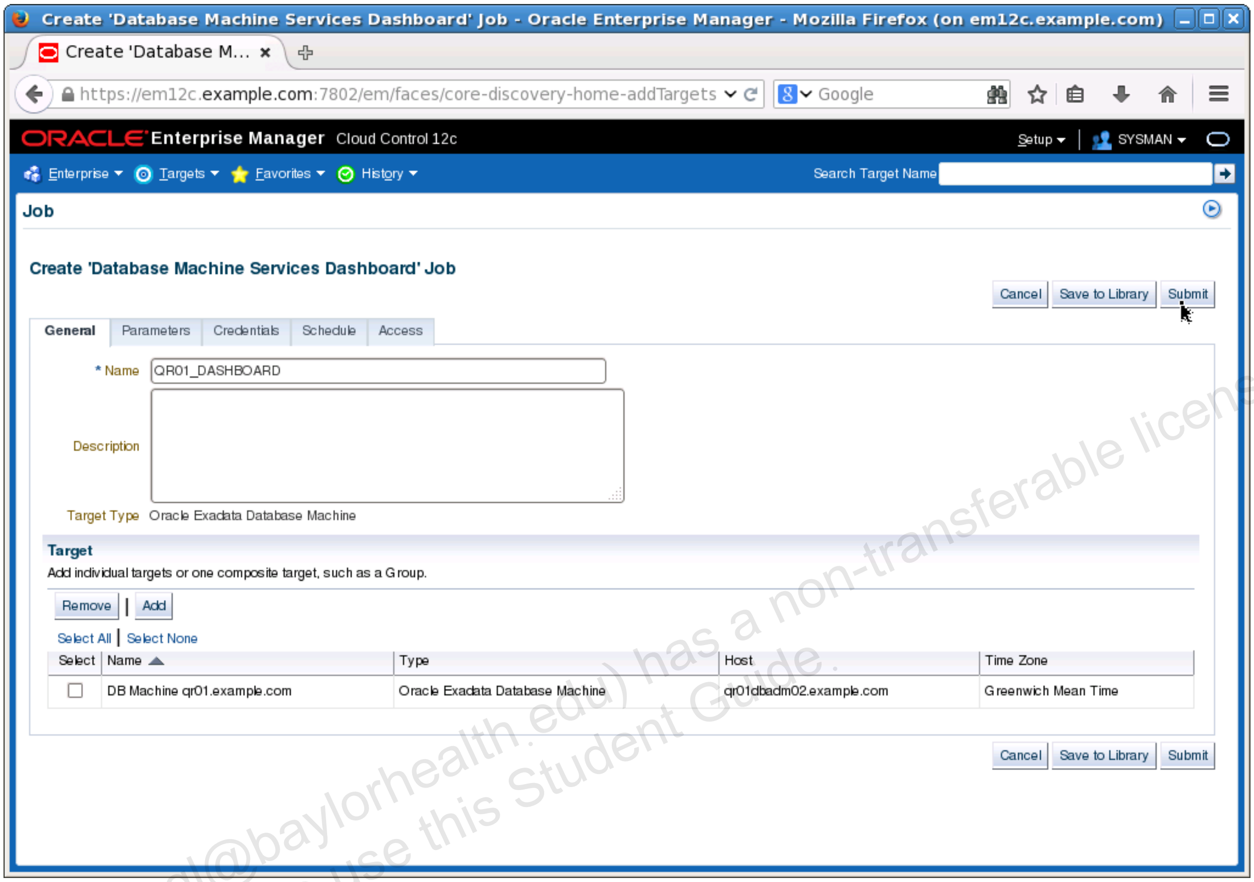
76. To commence the dashboard creation process, select the Enterprise > Job > Activity menu command.



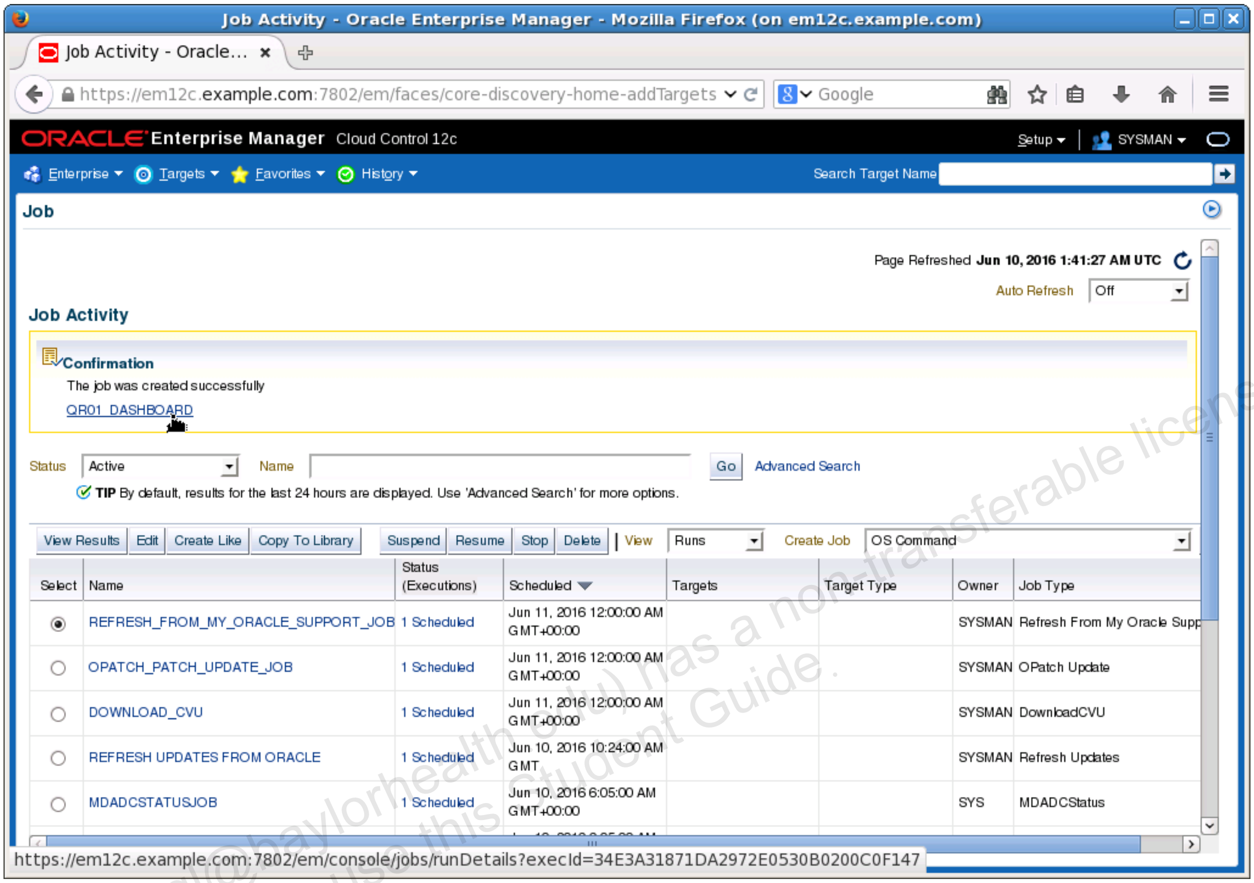
77. In the Create Job list, select the option to create a Database Machine Services Dashboard and click Go to proceed. Note that you may need to use the scroll bar along the bottom of the page to reveal the Go button.



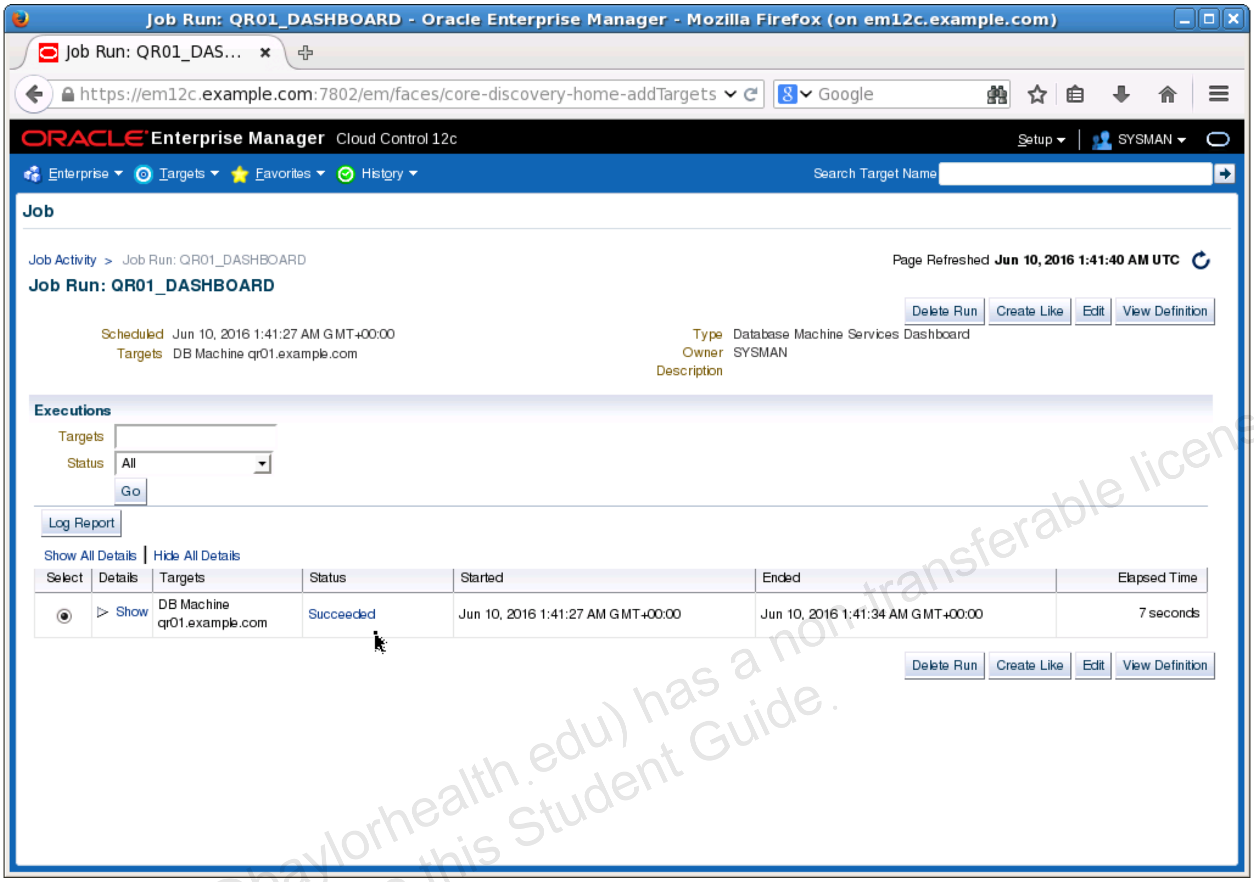
78. On the resulting page, specify `QR01_DASHBOARD` as the job name. Also, add the `DB Machine qr01.example.com` target by clicking Add and selecting your Database Machine from the resulting dialog. Finally, click Submit to proceed.



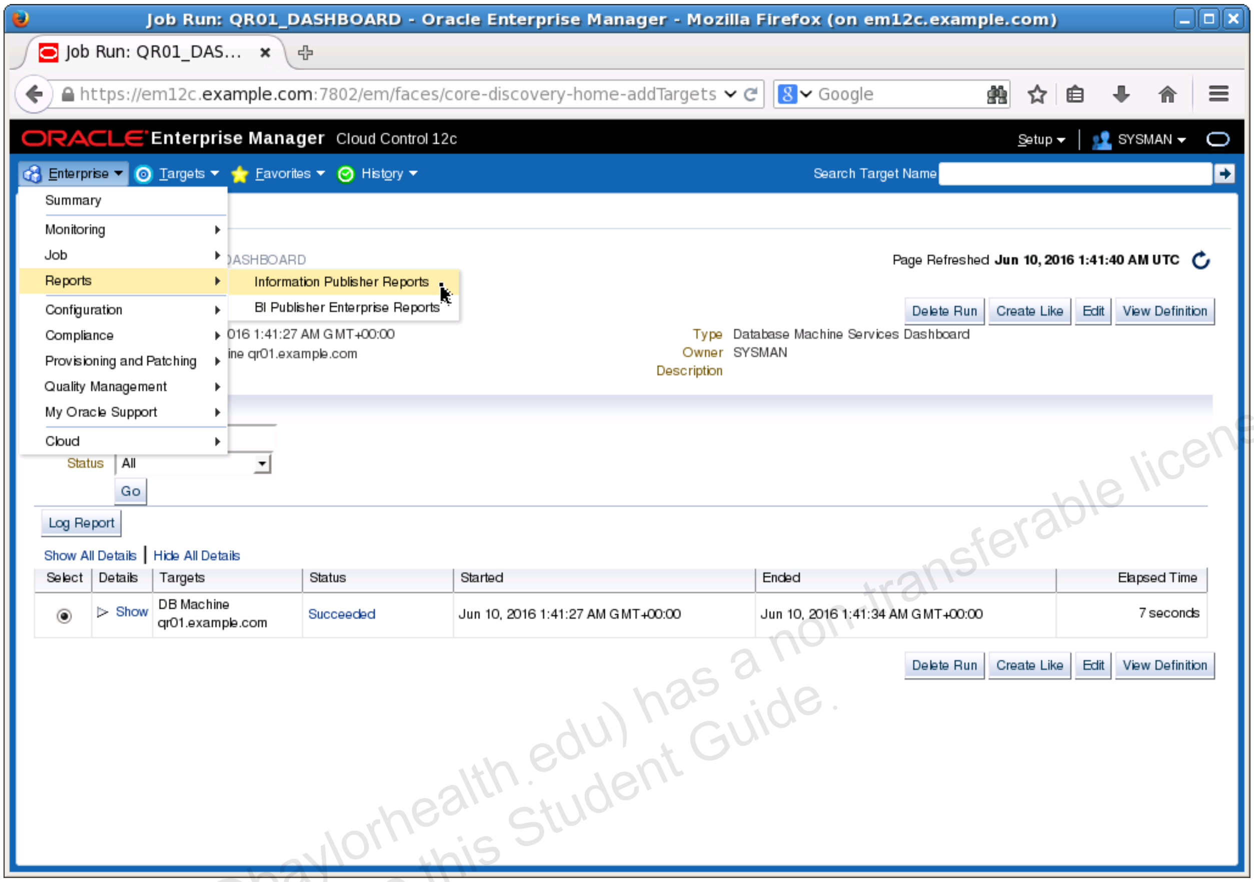
79. On the Job Activity page, you should see a notification confirming that your dashboard creation job was submitted successfully. Click the QR01_DASHBOARD link associated with the confirmation message.



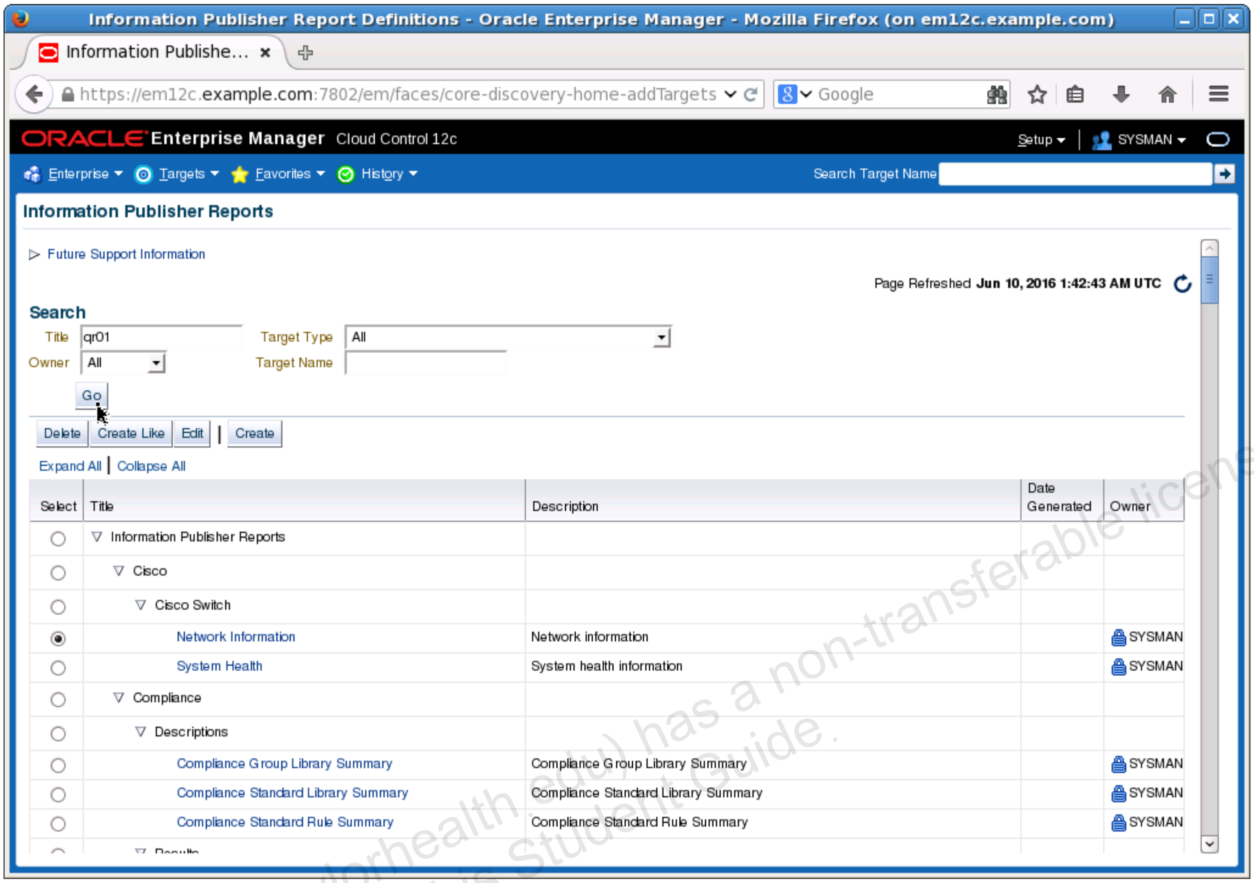
80. Examine the job status and periodically refresh the page until the job status is shown as Succeeded. Now the Database Machine Services Dashboard is created.



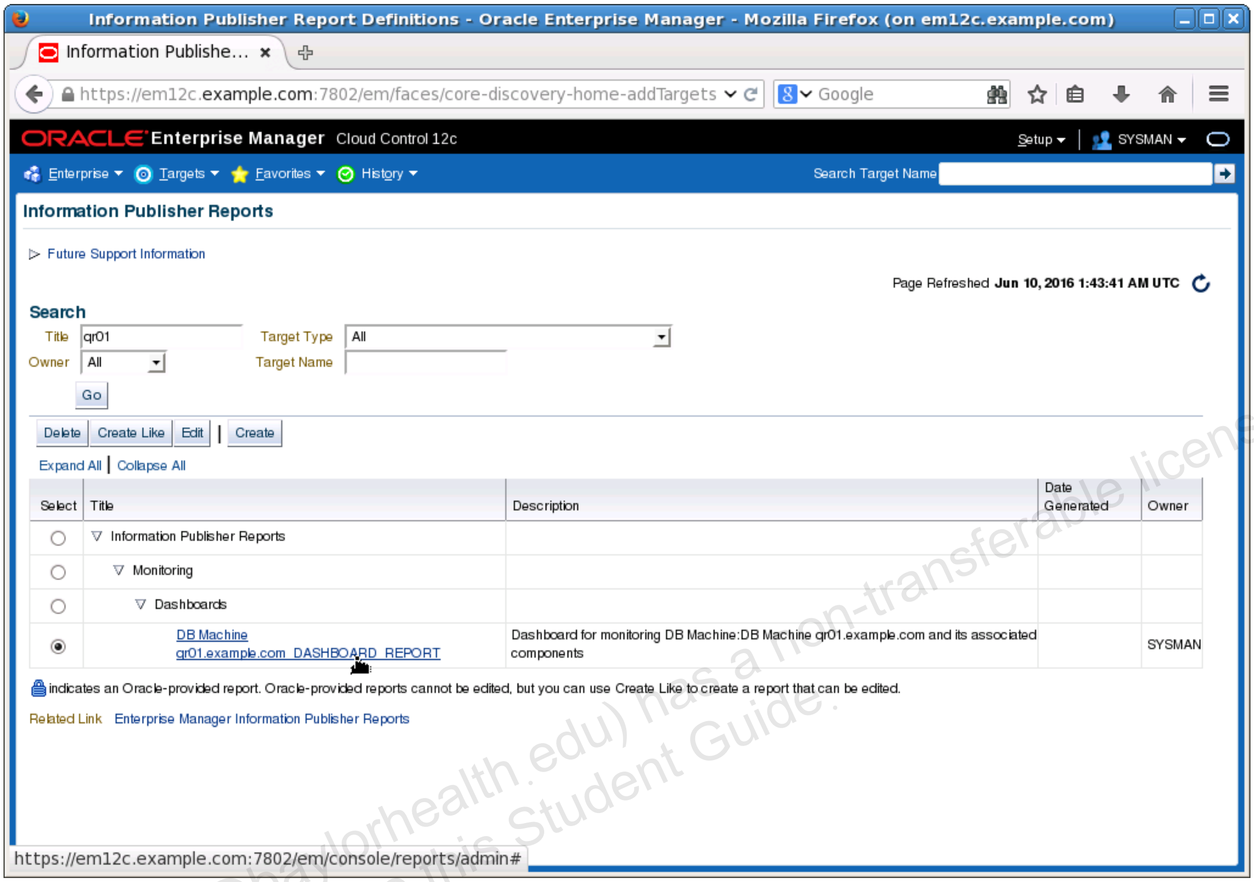
81. To examine the Database Machine Services Dashboard, select the Enterprise > Reports > Information Publisher Reports menu command.



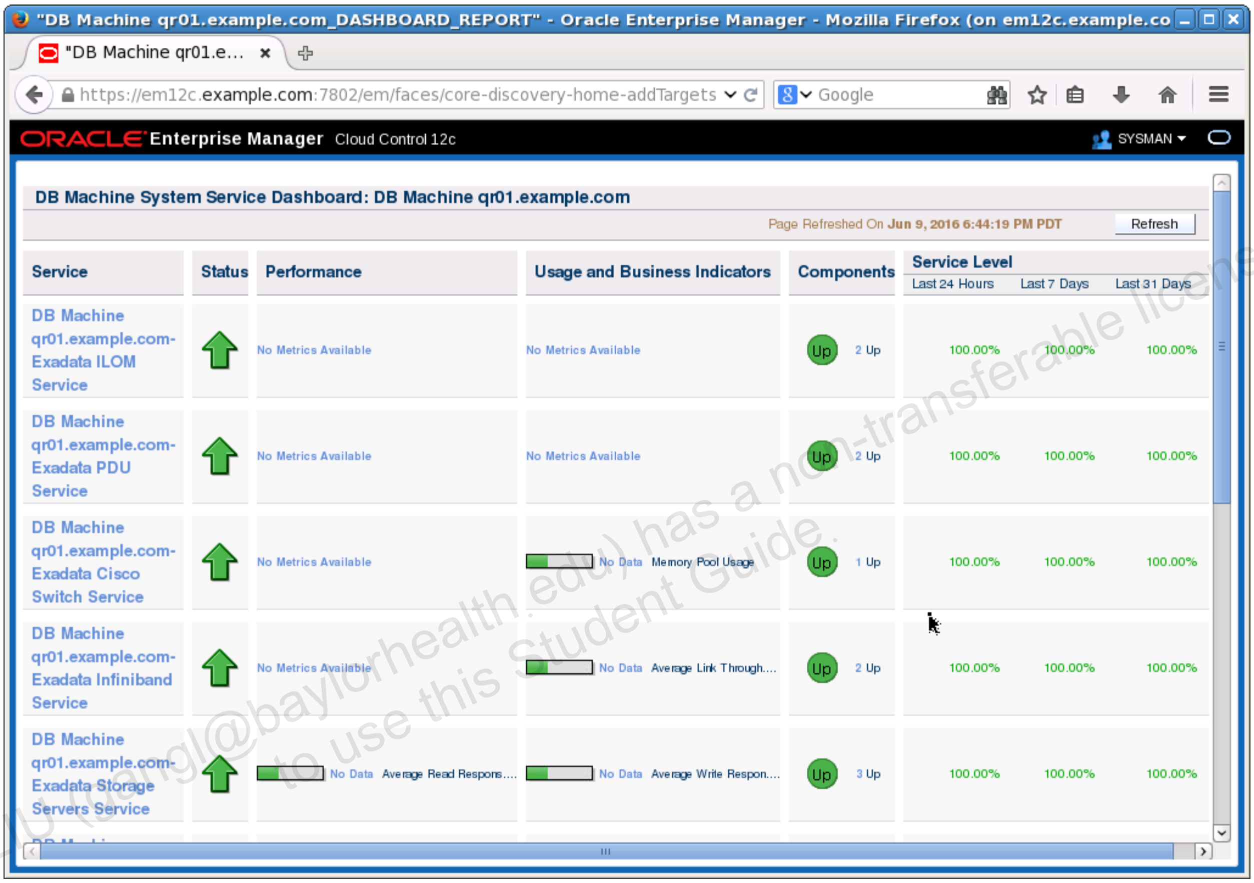
82. Enter `qr01` in the Title search field and click Go.



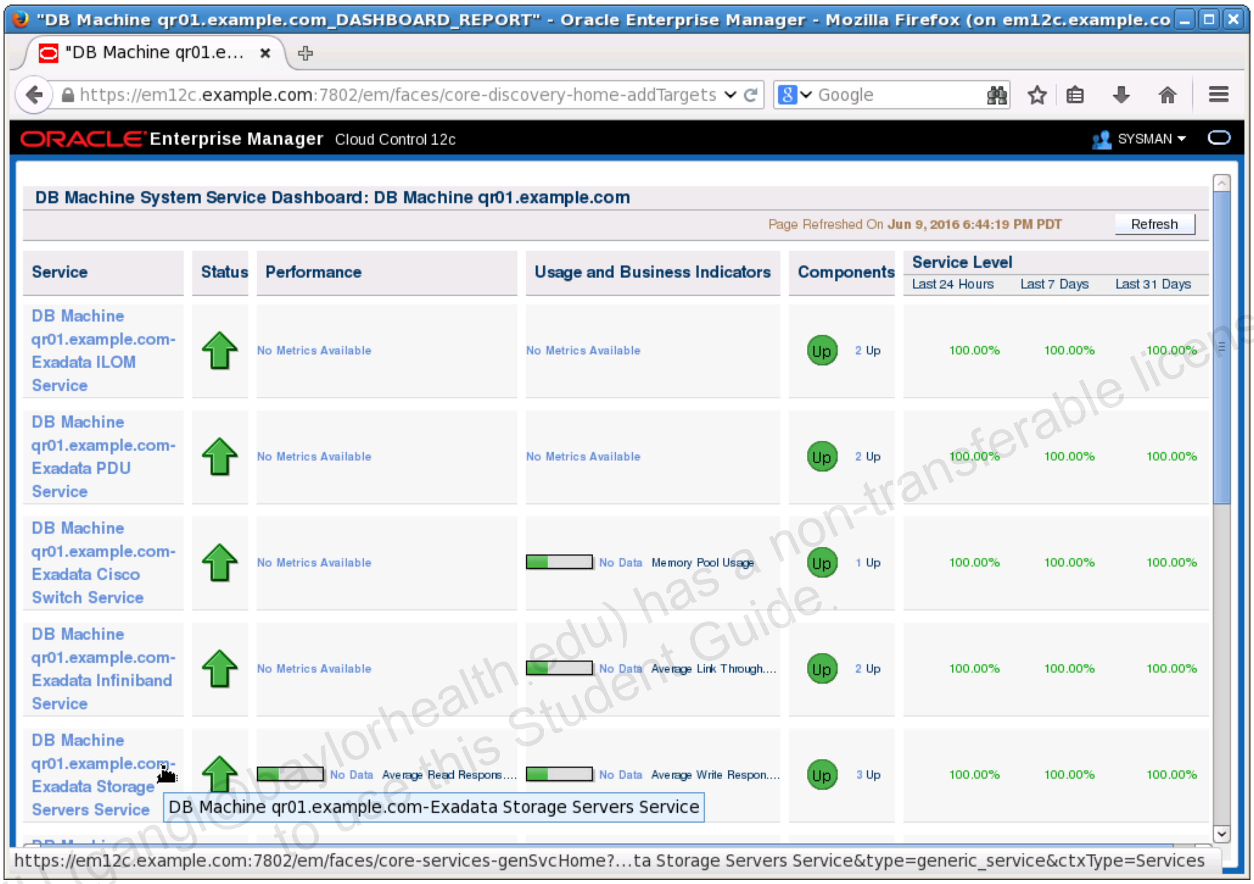
83. Click the “DB Machine qr01.example.com_DASHBOARD_REPORT” link to run the Database Machine Services Dashboard.



84. Examine the Database Machine Services Dashboard. You should be able to quickly confirm that all the Database Machine hardware and software components are currently operational. You may also notice that currently many of the metrics are empty. In a production environment, this would quickly change as the metrics are gathered and displayed by Enterprise Manager. In this laboratory environment, some of the metrics will never be displayed because the components that would normally generate the metrics are not fully functional.



85. Along with providing a one-stop monitoring overview of all the Database Machine hardware and software components, the Database Machine Services Dashboard also enables you to drill into further details by clicking various links across the entire dashboard. Click “DB Machine qr01.example.com-Exadata Storage Servers Service” to examine further details about the Exadata Database Machine storage servers.



The resulting page shows further summary information about the Database Machine storage service.

Congratulations! You have now configured Enterprise Manager Cloud Control 12*c* to monitor and manage Exadata Database Machine.

86. Exit your terminal sessions and log out of Enterprise Manager.

Practice 14-2: Post-Discovery Configuration and Verification

Overview

In this practice, you perform some of the recommended post-discovery configuration and verification tasks. Specifically, you will:

- Configure SNMP forwarding on the Database Machine servers that host the Enterprise Manager Agent.
- Verify the SNMP configuration for the Exadata Storage Servers.

Note that other recommended post-discovery tasks cannot be performed in your laboratory environment. For more information about all of the recommended post-discovery tasks, refer to chapter 4 of the *Oracle Enterprise Manager Exadata Management Getting Started Guide*.

Assumptions

This practice relies on the configuration activities performed in the previous practice.

Tasks

SNMP is used extensively throughout Exadata Database Machine so that the various components can report monitoring information and alerts to network management systems, such as Enterprise Manager Cloud Control 12*c*. Enterprise Manager Agents listen for SNMP events (also known as traps) and propagate the trap information to the Oracle Management Server (OMS) so that it can be processed and displayed to administrators. Some Exadata components, such as the Cisco Ethernet switch, have no documented way to send SNMP traps to ports other than UDP 162. However, the Oracle Management Agent runs as the `oracle` user and cannot listen on UDP ports under 1024. Because of these restrictions, a trap forwarder must be configured so that the SNMP traps sent to port 162 are forwarded to Enterprise Manager through the Oracle Management Agent.

In the first part of this practice, you will configure SNMP forwarding on the Database Machine servers that host the Enterprise Manager Agent.

1. Establish a terminal session connected to `qr01dbadm01` by using the `root` OS user.

```
$ ssh root@qr01dbadm01
root@qr01dbadm01's password: ????????
[root@qr01dbadm01 ~]#
```

2. Confirm the port number used by the Enterprise Manager Agent. One way to do this is to examine the `EMD_URL` property in the `<AGENT_INST>/sysman/config/emd.properties` file. By default, the agent will listen on port 3872.

```
[root@qr01dbadm01 ~]# grep EMD_URL=
/u01/app/oracle/product/12.1.0/agent/agent_inst/sysman/config/em
d.properties
EMD_URL=https://qr01dbadm01.example.com:3872/emd/main/
[root@qr01dbadm01 ~]#
```

3. Using the following command, confirm that the port you identified in the previous step is currently being used to listen for UDP messages. This is an indication that the agent is listening for SNMP traps.

```
[root@qr01dbadm01 ~]# netstat -an | grep 3872 | grep udp
udp        0      0 :::3872                     :::*
[root@qr01dbadm01 ~]#
```

4. Ensure that the `snmptrapd` OS service is currently stopped.

```
[root@qr01dbadm01 ~]# service snmptrapd status
snmptrapd is stopped
[root@qr01dbadm01 ~]#
```

5. Using the following command, configure the `snmptrapd` OS service so that SNMP traps received on the default port (162) are forwarded to the Enterprise Manager Agent. Note that in a production environment, if you are using SNMP community values in your network, you should alter the configuration in this step to reflect your community values rather than using the default `public` community. Also, ensure that the port number at the end of the `forward` directive matches your agent port.

```
[root@qr01dbadm01 ~]# cat > /etc/snmp/snmptrapd.conf << END
> authcommunity log,execute,net public
> forward default udp:localhost:3872
> END
[root@qr01dbadm01 ~]#
```

6. Configure the `snmptrapd` OS service to make sure that it is automatically started on server reboot.

```
[root@qr01dbadm01 ~]# chkconfig snmptrapd on
[root@qr01dbadm01 ~]#
```

7. Start the `snmptrapd` OS service.

```
[root@qr01dbadm01 ~]# service snmptrapd start
Starting snmptrapd:                                    [  OK  ]
[root@qr01dbadm01 ~]#
```

At this point, you have configured and started SNMP forwarding on `qr01dbadm01`.

8. Next, repeat the process on qr01dbadm02.

```
$ ssh root@qr01dbadm02
root@qr01dbadm02's password: ????????
[root@qr01dbadm02 ~]# grep EMD_URL=
/u01/app/oracle/product/12.1.0/agent/agent_inst/sysman/config/em
d.properties
EMD_URL=https://qr01dbadm02.example.com:3872/emd/main/
[root@qr01dbadm02 ~]# netstat -an | grep 3872 | grep udp
udp        0      0 :::3872                     :::*

[root@qr01dbadm02 ~]# service snmptrapd status
snmptrapd is stopped
[root@qr01dbadm02 ~]# cat > /etc/snmp/snmptrapd.conf << END
> authcommunity log,execute,net public
> forward default udp:localhost:3872
> END
[root@qr01dbadm02 ~]# chkconfig snmptrapd on
[root@qr01dbadm02 ~]# service snmptrapd start
Starting snmptrapd:                              [  OK  ]
[root@qr01dbadm02 ~]#
```

Earlier, when you performed the Database Machine discovery process, your Exadata Storage Servers were configured to send alerts to Enterprise Manager by using SNMP. In the final part of this practice, you will confirm the SNMP configuration on your Exadata Storage Servers.

9. Establish a terminal session connected to qr01celadm01 by using the celladmin OS user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

10. Using the following command, examine the notificationMethod storage server attribute. Seeing snmp in the notification method, confirms that SNMP notifications are enabled.

```
[celladmin@qr01celadm01 ~]$ dcli -c
qr01celadm01,qr01celadm02,qr01celadm03 cellcli -e list cell
attributes notificationMethod
qr01celadm01: mail,snmp
qr01celadm02: snmp
qr01celadm03: snmp
[celladmin@qr01celadm01 ~]$
```

11. Using the following command, examine the `snmpSubscriber` storage server attribute. As shown in the example output below, your storage servers should be configured to send SNMP traps directly to all the Enterprise Manager agents. This eliminates the possibility of alert messages getting lost due to the loss or unavailability of a single agent.

```
[celladmin@qr01celadm01 ~]$ dcli -c
qr01celadm01,qr01celadm02,qr01celadm03 cellcli -e list cell
attributes snmpSubscriber
qr01celadm01:
((host=qr01dbadm02.example.com,port=3872,community=public),(host
=qr01dbadm01.example.com,port=3872,community=public))
qr01celadm02:
((host=qr01dbadm02.example.com,port=3872,community=public),(host
=qr01dbadm01.example.com,port=3872,community=public))
qr01celadm03:
((host=qr01dbadm02.example.com,port=3872,community=public),(host
=qr01dbadm01.example.com,port=3872,community=public))
[celladmin@qr01celadm01 ~]$
```

12. Using the following command, examine the `notificationPolicy` storage server attribute. Your storage server should be configured to deliver critical alerts, warning alerts, and alert clearance notifications, as shown in the example below.

```
[celladmin@qr01celadm01 ~]$ dcli -c
qr01celadm01,qr01celadm02,qr01celadm03 cellcli -e list cell
attributes notificationPolicy
qr01celadm01: critical,warning,clear
qr01celadm02: critical,warning,clear
qr01celadm03: critical,warning,clear
[celladmin@qr01celadm01 ~]$
```

13. Exit your terminal sessions.

Practices for Lesson 15: Monitoring Exadata Storage Servers

Chapter 15

Practices for Lesson 15: Overview

Practices Overview

In these practices, you will monitor Exadata Storage Server by using metrics, alerts, and active requests. You will also perform a variety of Exadata Storage Server monitoring and administration tasks by using Enterprise Manager Cloud Control 12*c*.

Practice 15-1: Metrics, Alerts, and Active Requests

Overview

In this practice, you will monitor Exadata Storage Server by using metrics, alerts, and active requests.

Tasks

1. Establish a terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

2. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

Metrics are recorded observations of important run-time properties or internal instrumentation values of the storage cell and its components, such as cell disks or grid disks.

3. Execute the `LIST METRICDEFINITION` command to list every metric.

```
CellCLI> list metricdefinition
         CD_BY_FC_DIRTY
         CD_IO_BY_R_LG
         CD_IO_BY_R_LG_SEC
         CD_IO_BY_R_SCRUB
         CD_IO_BY_R_SCRUB_SEC
...
         SIO_IO_WR_HD_SEC
         SIO_IO_WR_RQ_FC
         SIO_IO_WR_RQ_FC_SEC
         SIO_IO_WR_RQ_HD
         SIO_IO_WR_RQ_HD_SEC

CellCLI>
```

4. Execute `LIST METRICDEFINITION DETAIL` to view more comprehensive information about all the metrics.

```
CellCLI> list metricdefinition detail
...

         name:                      SIO_IO_WR_RQ_HD
         description:               "Cumulative number of write IO requests to
hard disk by smart IO"
         metricType:                Cumulative
         objectType:                SMARTIO
         unit:                      "IO requests"

         name:                      SIO_IO_WR_RQ_HD_SEC
         description:               "Number of write IO requests per second to
hard disk by smart IO"
         metricType:                Rate
         objectType:                SMARTIO
         unit:                      IO/sec

CellCLI>
```

5. Add a `WHERE` condition to view information about specific metrics. Note the metrics `CL_MEMUT` and `CL_FSUT`, which you will use later in this practice.

```
CellCLI> list metricdefinition where name like 'CL_.*UT' detail
         name:                      CL_CPUT
         description:               "Percentage of time over the previous minute
that the system CPUs were not idle."
         metricType:                Instantaneous
         objectType:                CELL
         unit:                      %

         name:                      CL_FSUT
         description:               "Percentage of total space on this file
system that is currently used"
         metricType:                Instantaneous
         objectType:                CELL_FILESYSTEM
         unit:                      %

         name:                      CL_MEMUT
         description:               "Percentage of total physical memory on this
server that is currently used"
         metricType:                Instantaneous
         objectType:                CELL
         unit:                      %

CellCLI>
```

6. Execute `LIST METRICCURRENT` to view the most current metric observations.

```
CellCLI> list metriccurrent
...
         SIO_IO_RV_OF_SEC               SMARTIO          0.000 MB/sec
         SIO_IO_SI_SV                   SMARTIO          0.000 MB
         SIO_IO_SI_SV_SEC               SMARTIO          0.000 MB/sec
         SIO_IO_WR_FC                   SMARTIO          8,073 MB
         SIO_IO_WR_FC_SEC               SMARTIO          0.000 MB/sec
         SIO_IO_WR_HD                   SMARTIO          2,280 MB
         SIO_IO_WR_HD_SEC               SMARTIO          0.000 MB/sec
         SIO_IO_WR_RQ_FC                SMARTIO          8,409 IO requests
         SIO_IO_WR_RQ_FC_SEC            SMARTIO          0.0 IO/sec
         SIO_IO_WR_RQ_HD                SMARTIO          2,332 IO requests
         SIO_IO_WR_RQ_HD_SEC            SMARTIO          0.0 IO/sec

CellCLI>
```

7. Add a `WHERE` condition to view the most current metric observations for a subset of metrics.

```
CellCLI> list metriccurrent where objecttype = 'CELL'
         CL_CPUT                  qr01celadm01    1.8 %
         CL_CPUT_CS               qr01celadm01    10.4 %
         CL_CPUT_MS               qr01celadm01    0.1 %
         CL_FANS                  qr01celadm01    0
         CL_IO_RQ_NODATA          qr01celadm01    0 IO requests
         CL_IO_RQ_NODATA_SEC      qr01celadm01    0.0 IO/sec
         CL_MEMUT                 qr01celadm01    98 %
         CL_MEMUT_CS              qr01celadm01    38.8 %
         CL_MEMUT_MS              qr01celadm01    12.7 %
         CL_RUNQ                  qr01celadm01    0.2
         CL_SWAP_IN_BY_SEC        qr01celadm01    0.0 KB/sec
         CL_SWAP_OUT_BY_SEC       qr01celadm01    0.0 KB/sec
         CL_SWAP_USAGE            qr01celadm01    3 %
         CL_VIRTMEM_CS            qr01celadm01    5,868 MB
         CL_VIRTMEM_MS            qr01celadm01    2,161 MB
         IORM_MODE                qr01celadm01    2
         N_HCA_MB_RCV_SEC         qr01celadm01    0.000 MB/sec
         N_HCA_MB_TRANS_SEC       qr01celadm01    0.000 MB/sec
         N_NIC_KB_RCV_SEC         qr01celadm01    0.1 KB/sec
         N_NIC_KB_TRANS_SEC       qr01celadm01    0.1 KB/sec

CellCLI>
```

8. Examine a specific metric by specifying the metric name.

```
CellCLI> list metriccurrent cl_memut
         CL_MEMUT         qr01celadm01     98 %

CellCLI>
```

9. Use the following command to determine if there are any metrics that are currently in an abnormal state. Normally, you should not expect any output from this command.

```
CellCLI> list metriccurrent where alertState != normal

CellCLI>
```

10. Show the current space utilization on the root (/) cell file system. Note the reported utilization value because you will need this value later in the practice.

```
CellCLI> list metriccurrent cl_fsut where metricObjectName = '/'
         CL_FSUT          "/"      21 %

CellCLI>
```

The `LIST METRICHISTORY` command shows historical metric observations that are maintained in an internal repository on the cell. The retention period for metric history observations is specified by the `metricHistoryDays` cell attribute. The default retention period is 7 days and you can modify this setting with the CellCLI `ALTER CELL` command. You can display all the retained history for all the metrics by using `LIST METRICHISTORY`; however, doing so will output vast amounts of data so you should always use a filter to return more specific output.

11. Use the following command to determine if, and when, any metrics were in an abnormal state during the retention period. Normally, you should not expect any output from this command. Note that it may take quite a few seconds for this command to complete.

```
CellCLI> list metrichistory where alertState != normal

CellCLI>
```

Alerts represent events of importance occurring within Exadata, typically indicating that cell functionality is either compromised or in danger of failure.

12. Use the `LIST ALERTHISTORY` command to view all the alerts maintained in the alert repository. Normally you should expect to see no alerts.

```
CellCLI> list alerthistory

CellCLI>
```

13. Use the `DROP ALERTHISTORY` command to clear out old or unwanted alerts. If you want to focus on future alerts, you can clear the entire alert history by using `DROP ALERTHISTORY ALL`. Execute the `DROP ALERTHISTORY ALL` command.

```
CellCLI> drop alerthistory all
No alerts qualified for this drop operation

CellCLI>
```

An alert is automatically triggered when a predefined hardware or software issue is detected, or when a metric exceeds a threshold. In the next few steps, you will define a threshold, and then create a condition which will cause an alert.

14. List the thresholds currently defined on the Exadata cell. By default, there are no thresholds defined.

```
CellCLI> list threshold

CellCLI>
```

15. Create a warning threshold for file system utilization on the root (/) file system. Set the warning level to a value slightly larger than the utilization you observed in step 10.

```
CellCLI> create threshold cl_fsut."/" comparison='>', warning=22
Threshold cl_fsut."/" successfully created

CellCLI>
```

16. View the newly created threshold definition.

```
CellCLI> list threshold detail
         name:                    cl_fsut./
         comparison:              >
         warning:                 22.0

CellCLI>
```

17. Exit CellCLI.

```
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

18. Execute the following command inside the cell operating system. It creates a 1 GB file on the root file system, which will increase the utilization metric. After the metric crosses the threshold you defined in step 15, an alert will be generated.

```
[celladmin@qr01celadm01 ~]$ dd if=/dev/zero of=/tmp/file.out
bs=1024 count=1000000
1000000+0 records in
1000000+0 records out
1024000000 bytes (1.0 GB) copied, 7.47261 s, 137 MB/s
[celladmin@qr01celadm01 ~]$
```

19. Relaunch CellCLI.

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

20. Execute the `LIST ALERTHISTORY` command. Do you see an alert? If you do not see any alerts, re-execute `LIST ALERTHISTORY` periodically until the alert appears. Usually an alert is generated within a few minutes of a threshold being exceeded.

```
CellCLI> list alerthistory
         1_1     2016-06-10T02:28:18+00:00       warning
"The warning threshold for the following metric has been
crossed. Metric Name        : CL_FSUT  Metric Description :
Percentage of total space on this file system that is currently
used  Object Name        : /  Current Value      : 23.0 %
Threshold Value    : 22.0 %  "


CellCLI>
```

21. List the detailed alert information associated with the alert. Note that the `examinedBy` attribute is empty.

```
CellCLI> list alerthistory detail
         name:                    1_1
         alertDescription:        "The warning threshold for
metric CL_FSUT has been crossed: current value 23.0 %, threshold
value 22.0 %"
         alertMessage:            "The warning threshold for the
following metric has been crossed. Metric Name        : CL_FSUT
Metric Description : Percentage of total space on this file
system that is currently used  Object Name        : /  Current
Value      : 23.0 %  Threshold Value    : 22.0 %  "
         alertSequenceID:         1
         alertShortName:          CL_FSUT
         alertType:               Stateful
         beginTime:               2016-06-10T02:28:18+00:00
         endTime:
         examinedBy:
         metricObjectName:        "/"
         metricValue:             23.0
         notificationState:       1
         sequenceBeginTime:       2016-06-10T02:28:18+00:00
         severity:                warning
         alertAction:             "Examine the metric value that
is violating the specified threshold, and take appropriate
actions if needed."


CellCLI>
```

22. Modify the alert to indicate that you have examined it.

```
CellCLI> alter alerthistory 1_1 examinedby='student'
Alert 1_1 successfully altered

CellCLI>
```

23. List the detailed alert information associated with the alert. Note that the `examinedBy` attribute is now set as you specified in the previous step.

```
CellCLI> list alerthistory detail
         name:                   1_1
         alertDescription:       "The warning threshold for
metric CL_FSUT has been crossed: current value 23.0 %, threshold
value 22.0 %"
         alertMessage:           "The warning threshold for the
following metric has been crossed. Metric Name        : CL_FSUT
Metric Description : Percentage of total space on this file
system that is currently used  Object Name        : /  Current
Value      : 23.0 %  Threshold Value    : 22.0 %  "
         alertSequenceID:        1
         alertShortName:         CL_FSUT
         alertType:              Stateful
         beginTime:              2016-06-10T02:28:18+00:00
         endTime:
         examinedBy:             student
         metricObjectName:       "/"
         metricValue:            23.0
         notificationState:      1
         sequenceBeginTime:      2016-06-10T02:28:18+00:00
         severity:               warning
         alertAction:            "Examine the metric value that
is violating the specified threshold, and take appropriate
actions if needed."

CellCLI>
```

24. Exit CellCLI.

```
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

25. Delete the file you created in step 18. Be careful not to delete any other files.

```
[celladmin@qr01celadm01 ~]$ rm /tmp/file.out
[celladmin@qr01celadm01 ~]$
```

26. Relaunch CellCLI.

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

27. Examine the file system utilization and confirm that the root (`/`) file system utilization has fallen back below the warning threshold. If the metric still exceeds the warning threshold, re-execute the command periodically until the metric value is updated.

```
CellCLI> list metriccurrent cl_fsut where metricObjectName = '/'
         CL_FSUT         "/"     24 %

CellCLI> list metriccurrent cl_fsut where metricObjectName = '/'
         CL_FSUT         "/"     21 %

CellCLI>
```

28. Re-execute `LIST ALERTHISTORY`. Note that the alert is now listed as cleared.

```
CellCLI> list alerthistory
         1_1     2016-06-10T02:28:18+00:00       warning
"The warning threshold for the following metric has been
crossed. Metric Name        : CL_FSUT  Metric Description :
Percentage of total space on this file system that is currently
used  Object Name        : /  Current Value      : 23.0 %
Threshold Value    : 22.0 %  "
         1_2     2016-06-10T02:30:18+00:00       clear
"The warning threshold for the following metric has been
cleared. Metric Name        : CL_FSUT  Metric Description :
Percentage of total space on this file system that is currently
used  Object Name        : /  Current Value      : 21.0 %
Threshold Value    : 22.0 %  "

CellCLI>
```

29. View the alert details to determine the period of the alert.

```
CellCLI> list alerthistory detail
         name:                   1_1
         alertDescription:       "The warning threshold for metric CL_FSUT has
been crossed: current value 23.0 %, threshold value 22.0 %"
         alertMessage:           "The warning threshold for the following
metric has been crossed. Metric Name        : CL_FSUT  Metric Description :
Percentage of total space on this file system that is currently used  Object
Name        : /  Current Value      : 23.0 %  Threshold Value    : 22.0 %  "
         alertSequenceID:        1
         alertShortName:         CL_FSUT
         alertType:              Stateful
         beginTime:              2016-06-10T02:28:18+00:00
         endTime:                2016-06-10T02:30:18+00:00
         examinedBy:             student
         metricObjectName:       "/"
         metricValue:            23.0
         notificationState:      1
         sequenceBeginTime:      2016-06-10T02:28:18+00:00
         severity:               warning
         alertAction:            "Examine the metric value that is violating
the specified threshold, and take appropriate actions if needed."


         name:                   1_2
         alertMessage:           "The warning threshold for the following
metric has been cleared. Metric Name        : CL_FSUT  Metric Description :
Percentage of total space on this file system that is currently used  Object
Name        : /  Current Value      : 21.0 %  Threshold Value    : 22.0 %  "
         alertSequenceID:        1
         alertShortName:         CL_FSUT
         alertType:              Stateful
         beginTime:              2016-06-10T02:30:18+00:00
         endTime:                2016-06-10T02:30:18+00:00
         examinedBy:
         metricObjectName:       "/"
         metricValue:            21.0
         notificationState:      1
         sequenceBeginTime:      2016-06-10T02:28:18+00:00
         severity:               clear
         alertAction:            "The threshold value is no longer violated.
No further action is required for threshold CL_FSUT."/"."

CellCLI>
```

An active request provides a view of I/O requests that are currently being processed by a cell. In the next few steps, you will execute a database update and analyze the associated active requests.

30. Establish a separate terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance). Leave your CellCLI terminal session running in the background.

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

31. Connect to your database with SQL*Plus. Log in as the `sh` user and configure your new session to force parallel DML.

```
[oracle@qr01dbadm01 ~]$ sqlplus sh/??

SQL*Plus: Release 12.1.0.2.0 Production...

SQL> alter session force parallel dml;

Session altered.

SQL>
```

32. Execute the following long-running update statement. (Use the SQL script `/home/oracle/labs/lab15-01-32.sql` if you prefer.) **Continue to the next step while the update statement is processing.** If the update completes before you gather the information in the next step, commit the changes and re-execute the update statement.

```
SQL> update customers
  2  set cust_credit_limit=0.9*cust_credit_limit
  3  where cust_id < 500000;
```

33. Switch back to your original terminal session and execute the CellCLI `LIST ACTIVEREQUEST DETAIL` command. If necessary, re-execute the command until you see an active request that exhibits the following attributes:
    - The `ioGridDisk` matches one of your `DATA` grid disks.
    - The `objectNumber` is a positive value.
    - The `sqlID` is a non-zero value.

    Note that you may see different types of active requests including many that relate to internal database I/O, such as recursive SQL or writes to the redo log files.

```
CellCLI> list activerequest detail
...
         name:                      445
         asmDiskGroupNumber:        1
         asmFileIncarnation:        874117569
         asmFileNumber:             273
         consumerGroupID:           17326
         consumerGroupName:         OTHER_GROUPS
         dbID:                      2080757153
         dbName:                    DBM
         dbRequestID:               445
         fileType:                  Datafile
         id:                        445
         instanceNumber:            2
         ioBytes:                   491520
         ioBytesSofar:              0
         ioGridDisk:                DATA_QR01_CD_00_qr01celadm01
         ioOffset:                  646096
         ioReason:                  "BufferCache Read"
         ioType:                    CacheGet
         objectNumber:              20473
         parentID:                  445
         pdbID:                     0
         requestState:              "Queued for Disk Read"
         sessionID:                 817
         sessionSerNumber:          52242
         sqlID:                     6fkmg917x69xs
         tableSpaceNumber:          7

CellCLI>
```

34. Switch back to your SQL*Plus session. After the update from step 32 finishes, commit the transaction. Then use the object number and tablespace number from the output in step 33 to determine the identity of the database object associated with the active request. (Use the SQL script `/home/oracle/labs/lab15-01-34.sql` if you prefer.) Verify that the active request information you saw earlier is associated with an action on the `CUSTOMERS` table.

```
499999 rows updated.

SQL> commit;

Commit complete.

SQL> select distinct object_name, tablespace_name
  2  from v$segment_statistics
  3  where dataobj#=20473 and ts#=7;

OBJECT_NAME
--------------------------------------------------------------
TABLESPACE_NAME
------------------------------
CUSTOMERS
SH



SQL>
```

35. Query `V$SQL` and use the `sqlID` from step 33 to verify that the active request information is in fact associated with the update you executed in step 32. (Use the SQL script `/home/oracle/labs/lab15-01-35.sql` if you prefer.)

```
SQL> select distinct sql_text, sql_id
  2  from v$sql where sql_id = '6fkmg917x69xs';

SQL_TEXT
--------------------------------------------------------------
SQL_ID
-------------
update customers set cust_credit_limit=0.9*cust_credit_limit
where cust_id < 500000
6fkmg917x69xs



SQL>
```

36. Exit all of your SQL*Plus and CellCLI sessions.

Practice 15-2: Exadata Storage Server Monitoring with Enterprise Manager

Overview

In this practice, you will exercise a variety of Exadata Storage Server monitoring and administration capabilities provided by Enterprise Manager Cloud Control 12*c*.

Assumptions

The practice relies on the configurations performed in Practice 14-1.

Tasks

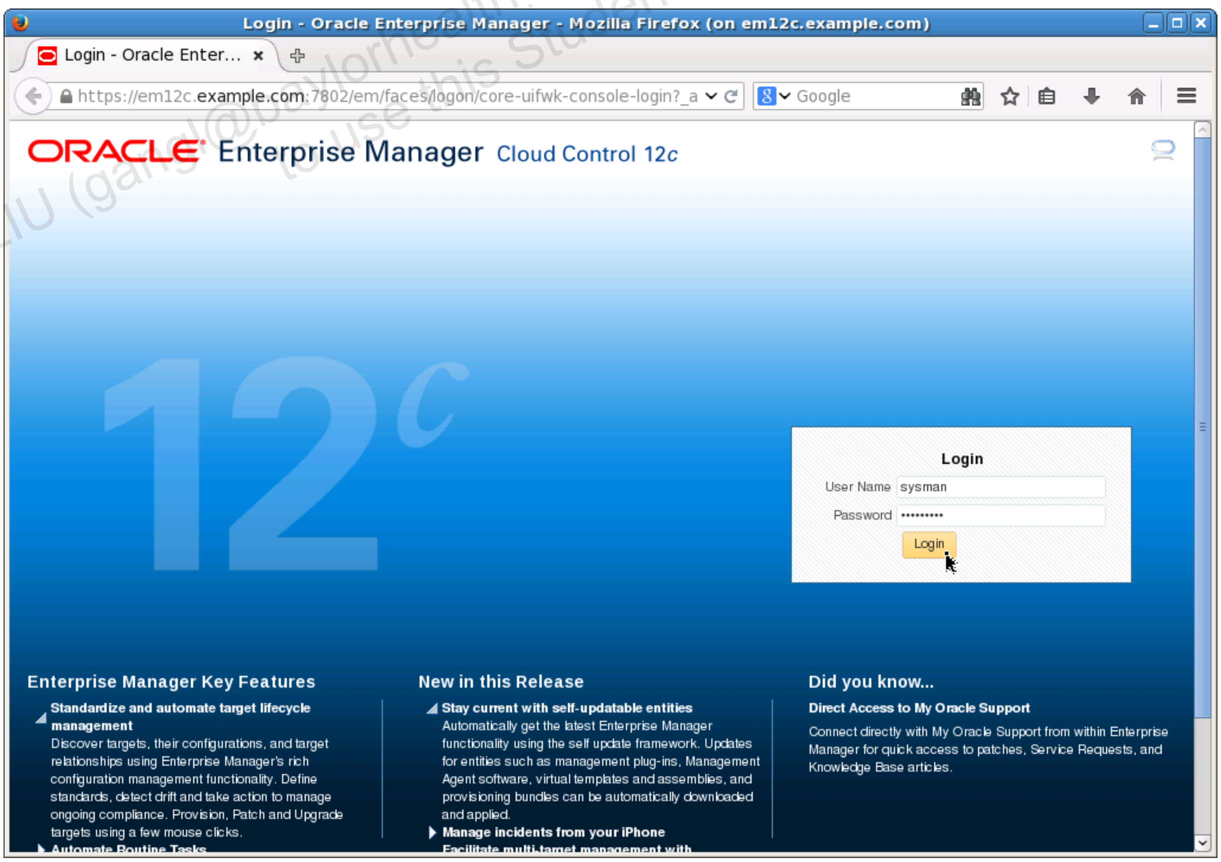
1. Establish a terminal session connected to `em12c` by using the `oracle` OS user. Ensure that you specify the `-X` option for `ssh`.

```
$ ssh -X oracle@em12c
oracle@em12c password: ????????
[oracle@em12c ~]$
```

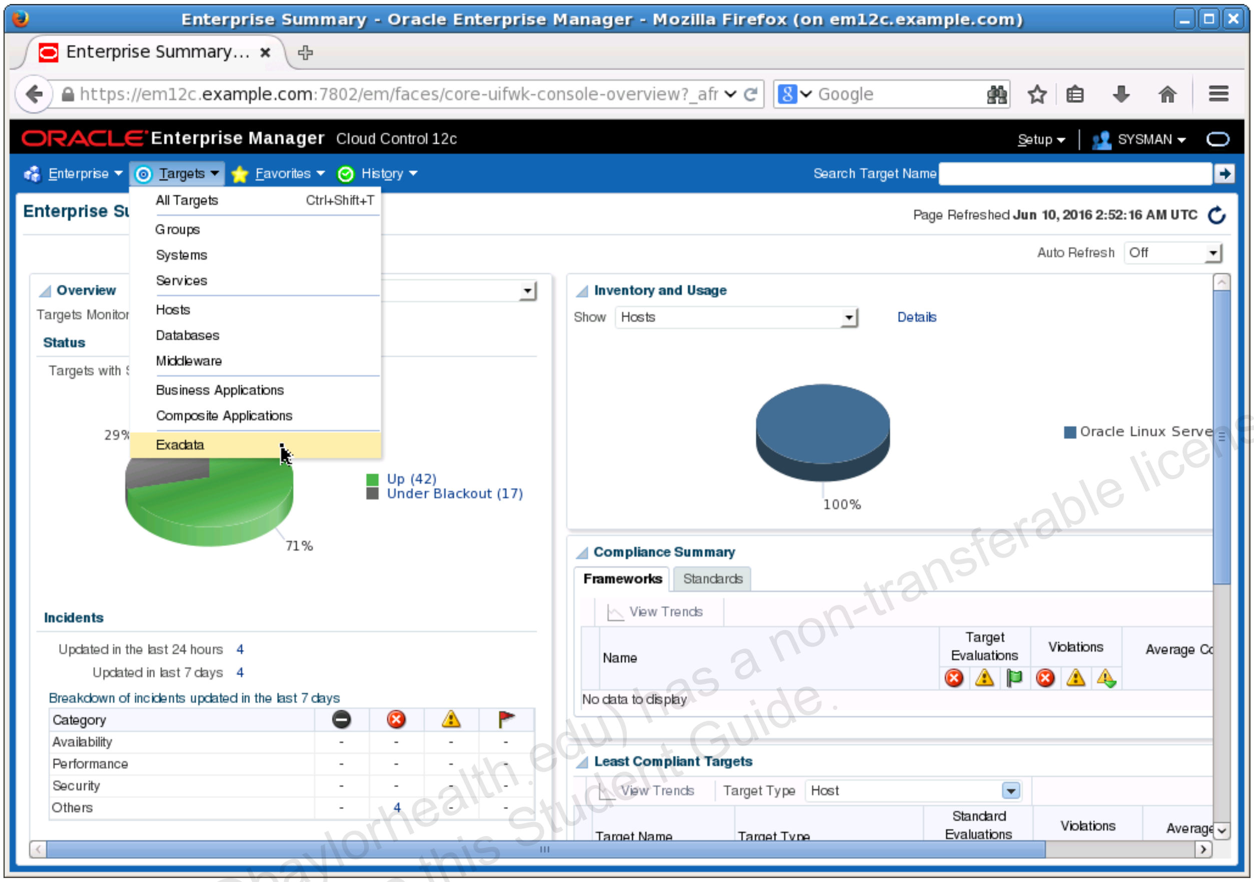
2. Start the Firefox web browser.

```
[oracle@em12c ~]$ firefox &
[1] 30053
```

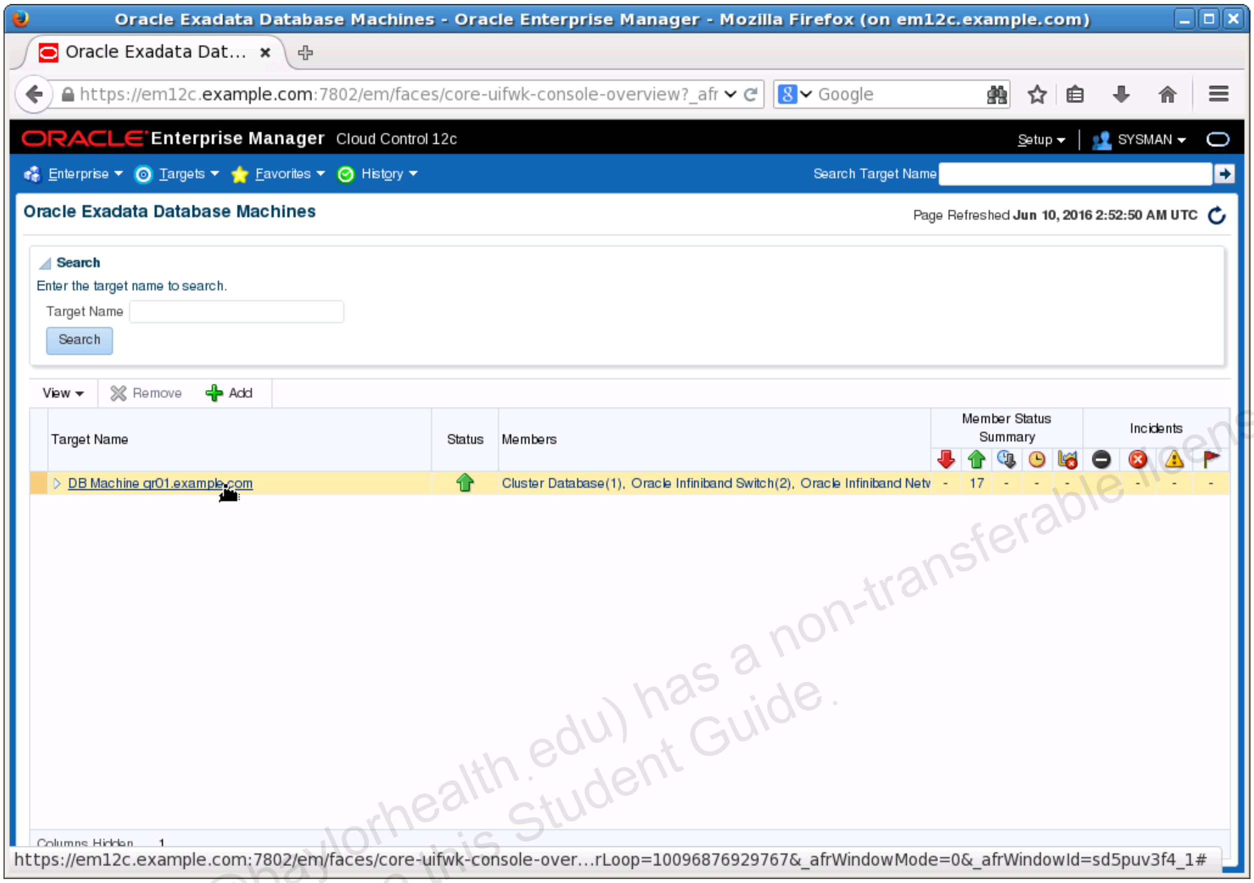
3. Log in to Enterprise Manager Cloud Control 12*c* as the `sysman` user.



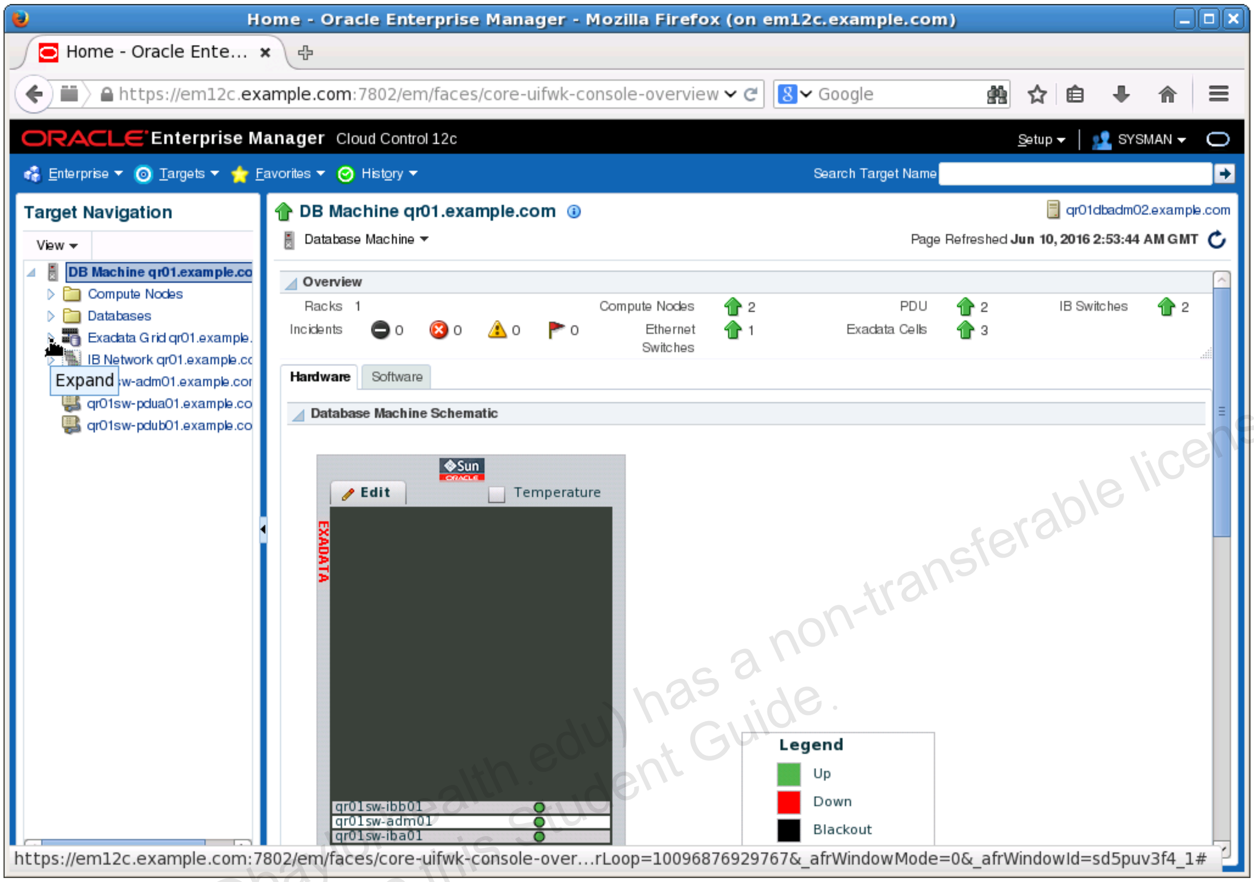
4. To begin Exadata Storage Server monitoring, select the Targets > Exadata menu command.



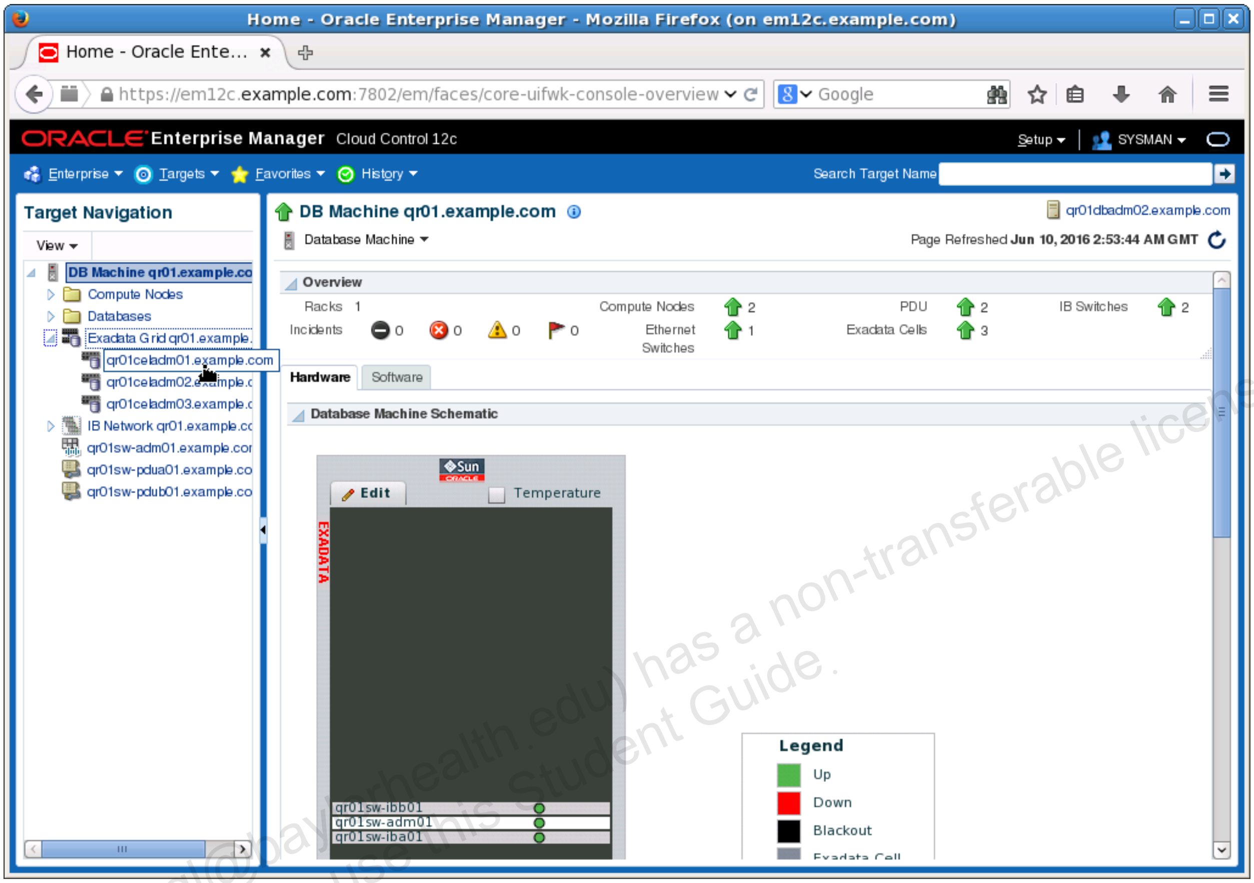
5. Click the link corresponding to your Exadata Database Machine (`DB Machine qr01.example.com`).



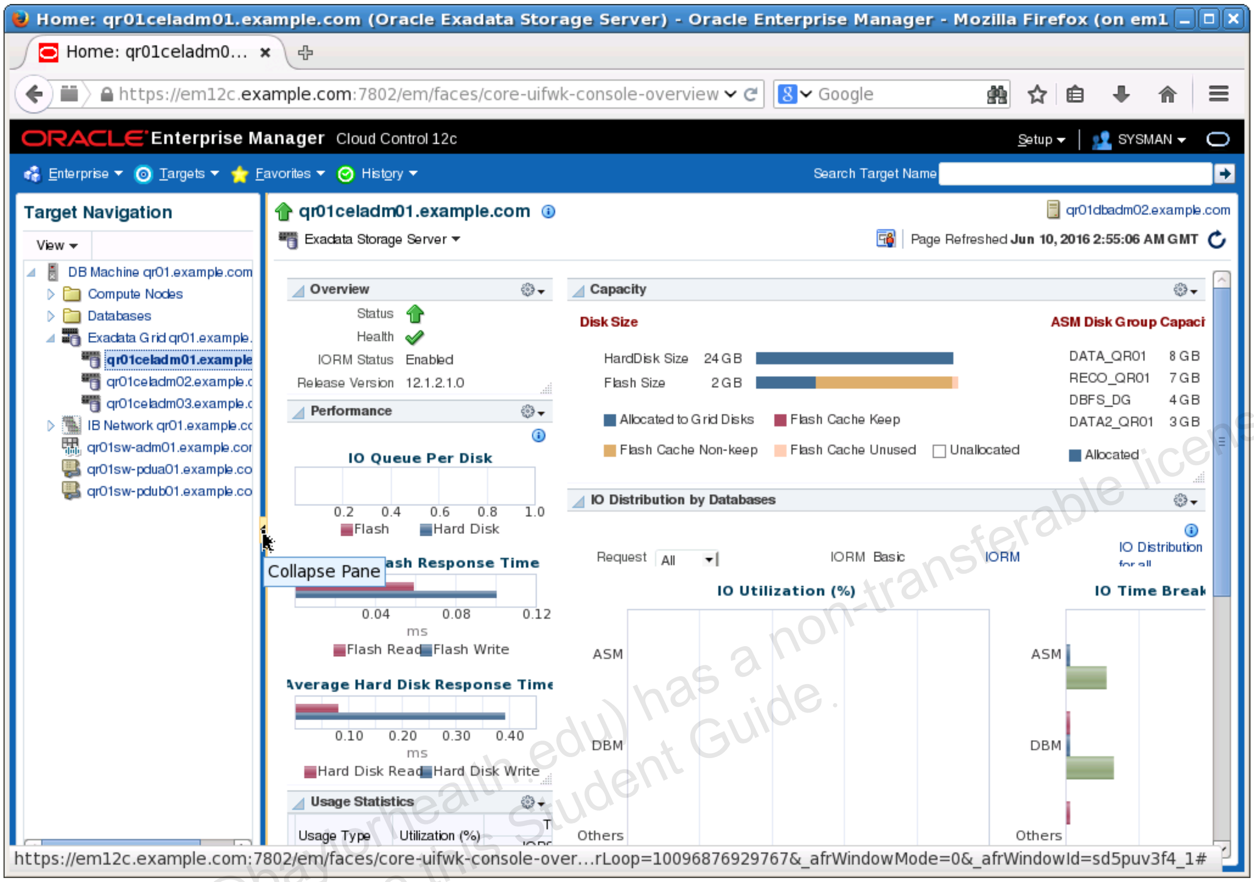
6. On the Database Machine home page, expand the Exadata Grid node in the Target Navigation pane.



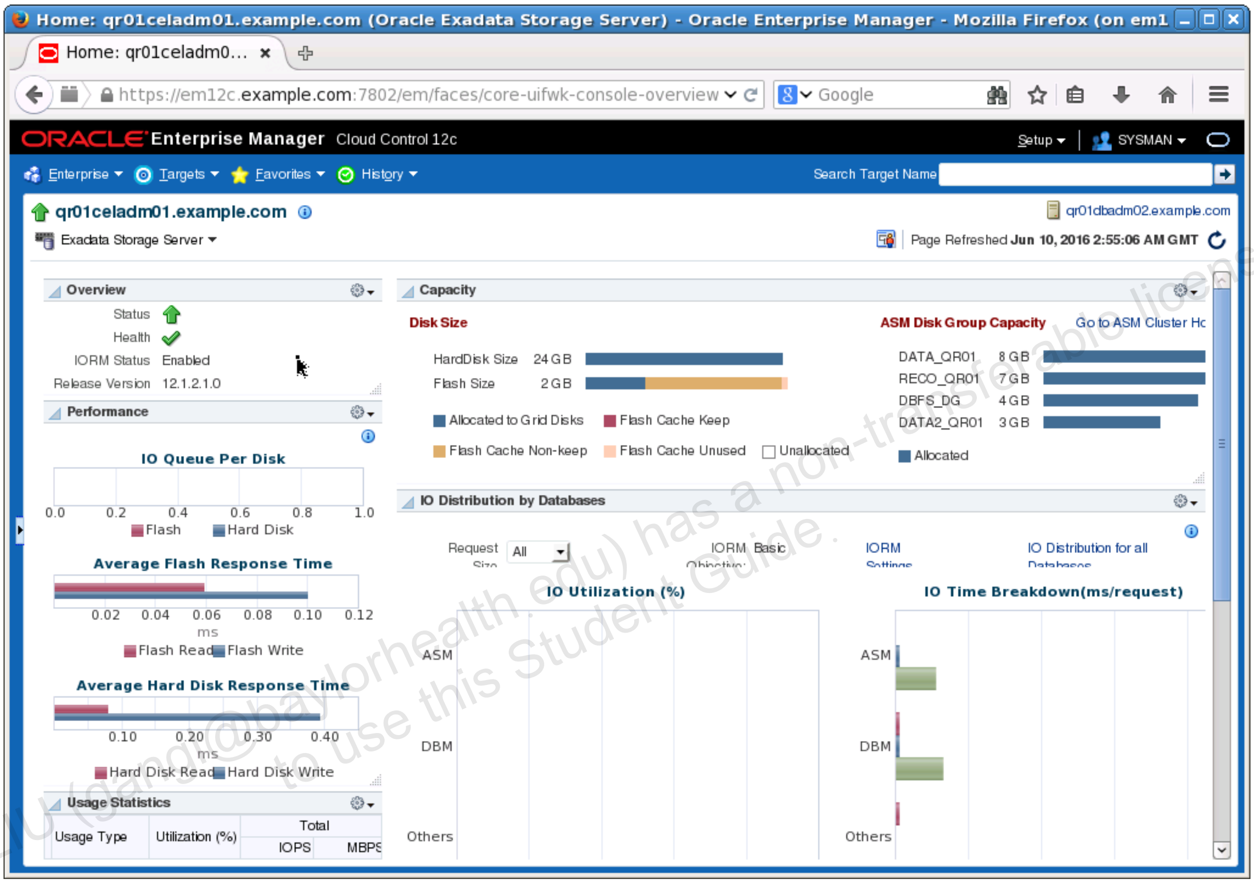
7. Now the Target Navigation pane shows the three Exadata Storage Servers contained in this Database Machine. Click the name of the first storage server (`qr01celadm01.example.com`).



8. On the Exadata Storage Server home page for `qr01celadm01`, collapse the Target Navigation pane.

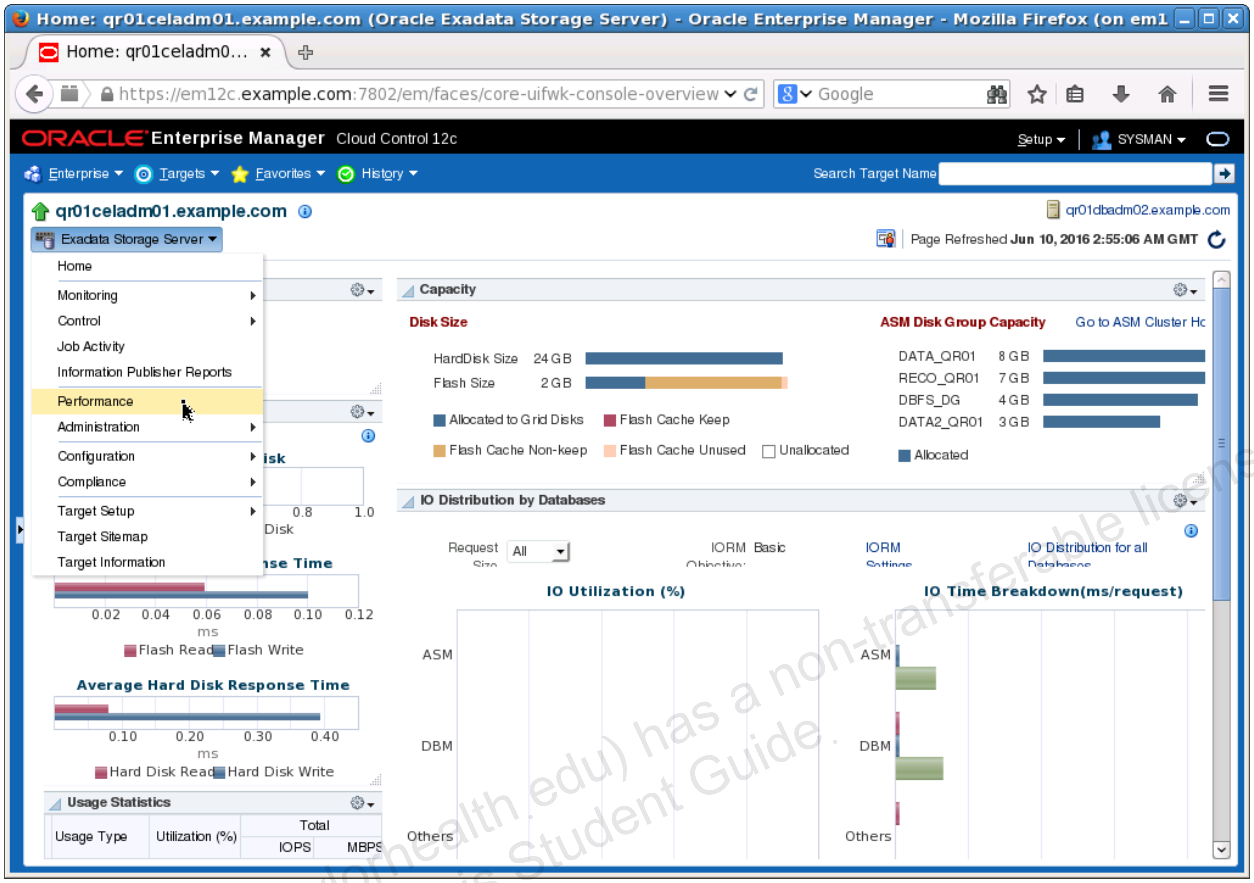


9. Take a moment to examine the Exadata Storage Server home page. Notice that the Overview area provides a quick visual indication of the status and health of the storage server. The other areas provide summary information relating to capacity, performance, and utilization of the cell. At the bottom of the page, the Incidents area displays alerts relating to the cell. You will investigate this area later in the practice. Note that the Size values reported on this page are accurate for your laboratory environment; however, they do not reflect the capacity of a real Exadata Storage Server.

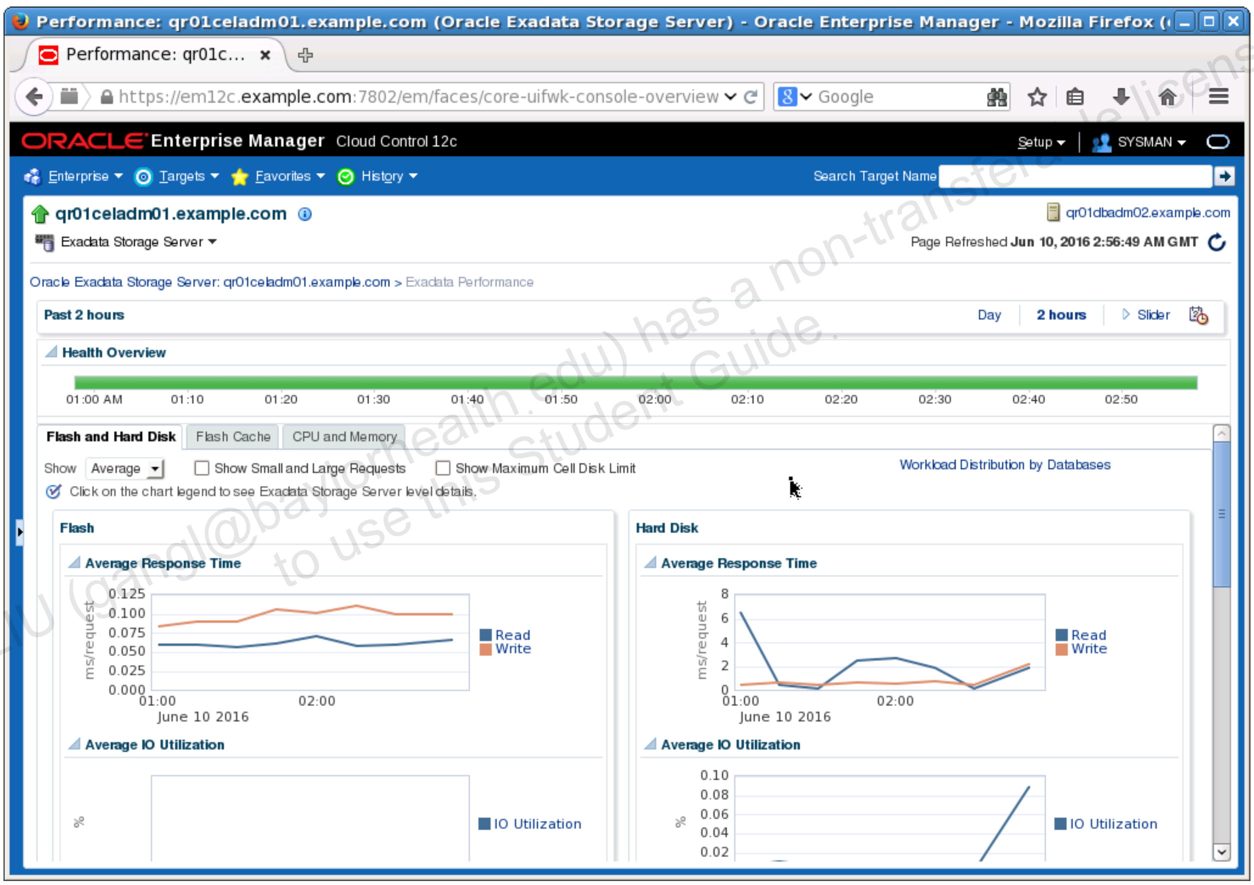


The Exadata Storage Server menu provides access to the other monitoring and administration functions provided by Enterprise Manager. In the next part of this practice, you will explore a number of these capabilities, starting with performance monitoring.

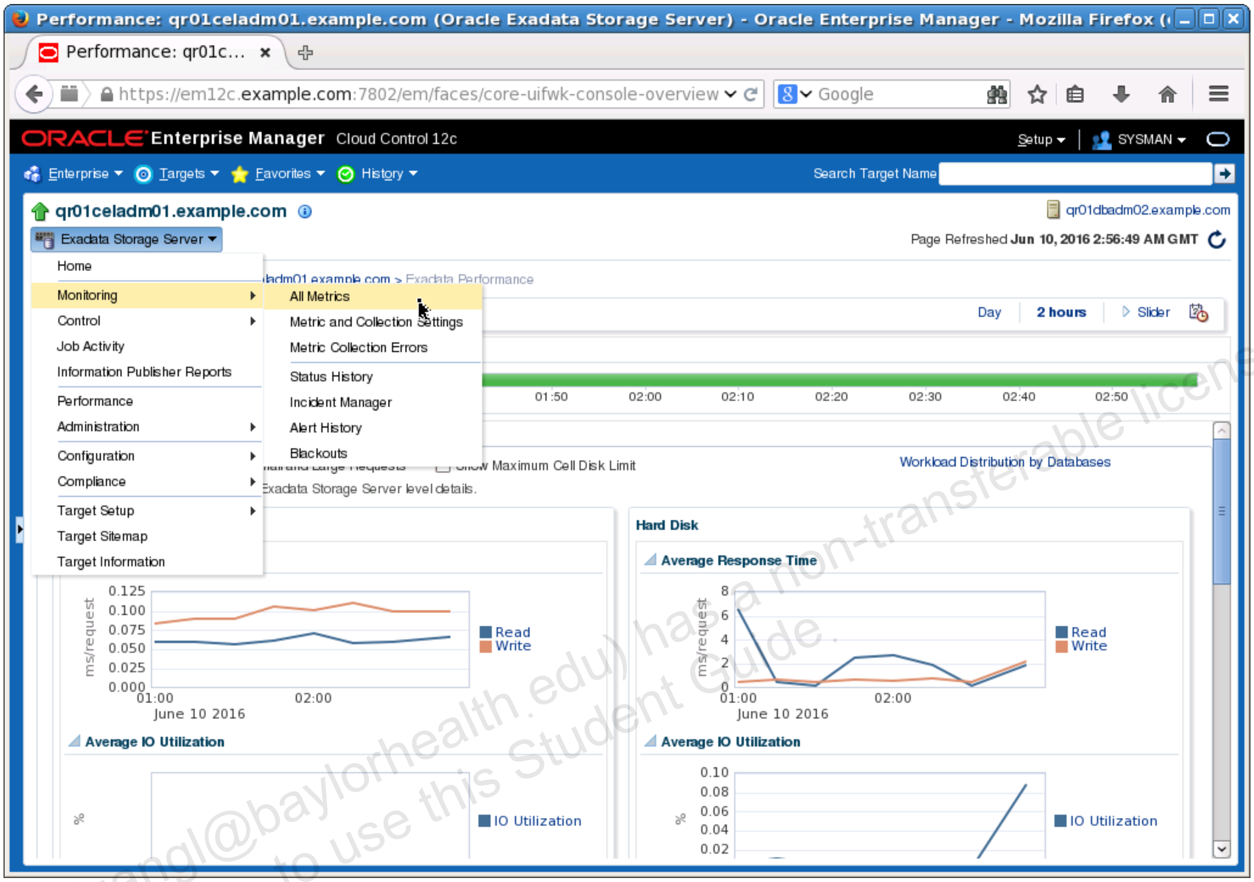
10. Select the Exadata Storage Server > Performance menu command.



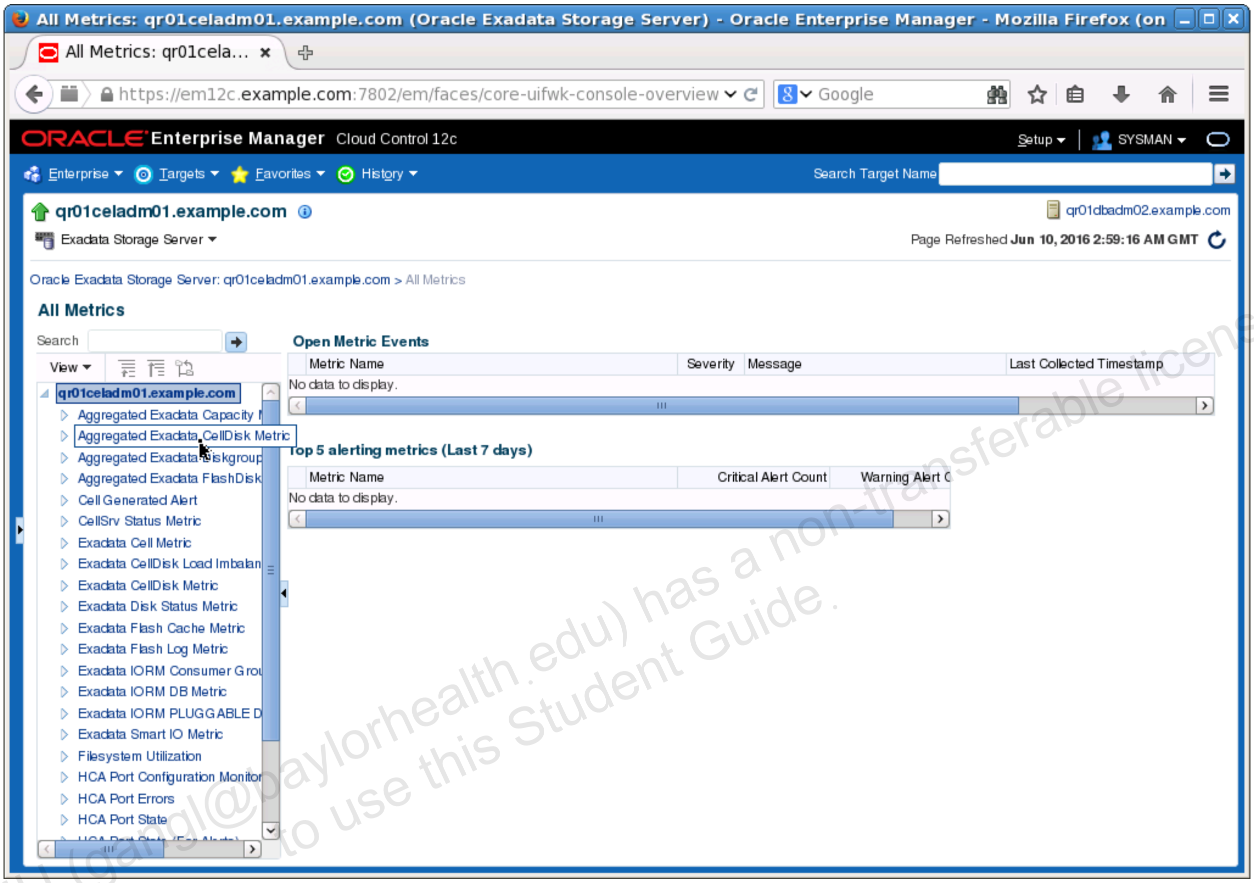
11. You should now see the Exadata Storage Server Performance page for `qr01celadm01`.
    This page shows various performance metrics leading up to the current time.
    By default, you will see performance information relating to the past two hours.
    At the top of the page, you will see a visual indication of the cell health. A green band indicates a healthy cell over the corresponding time period.
    Below the Health Overview, a series of charts display key performance metrics associated with the cell. Take a moment to scroll down the page and examine all the performance charts.
    Note that the legend labels associated with each metric are links, which navigate to pages that contain more detailed information about the metric.
    Typically, administrators will gather an understanding of what this page should look like during normal system operation. Then they will check this page for abnormalities that could point to the underlying cause of an issue.



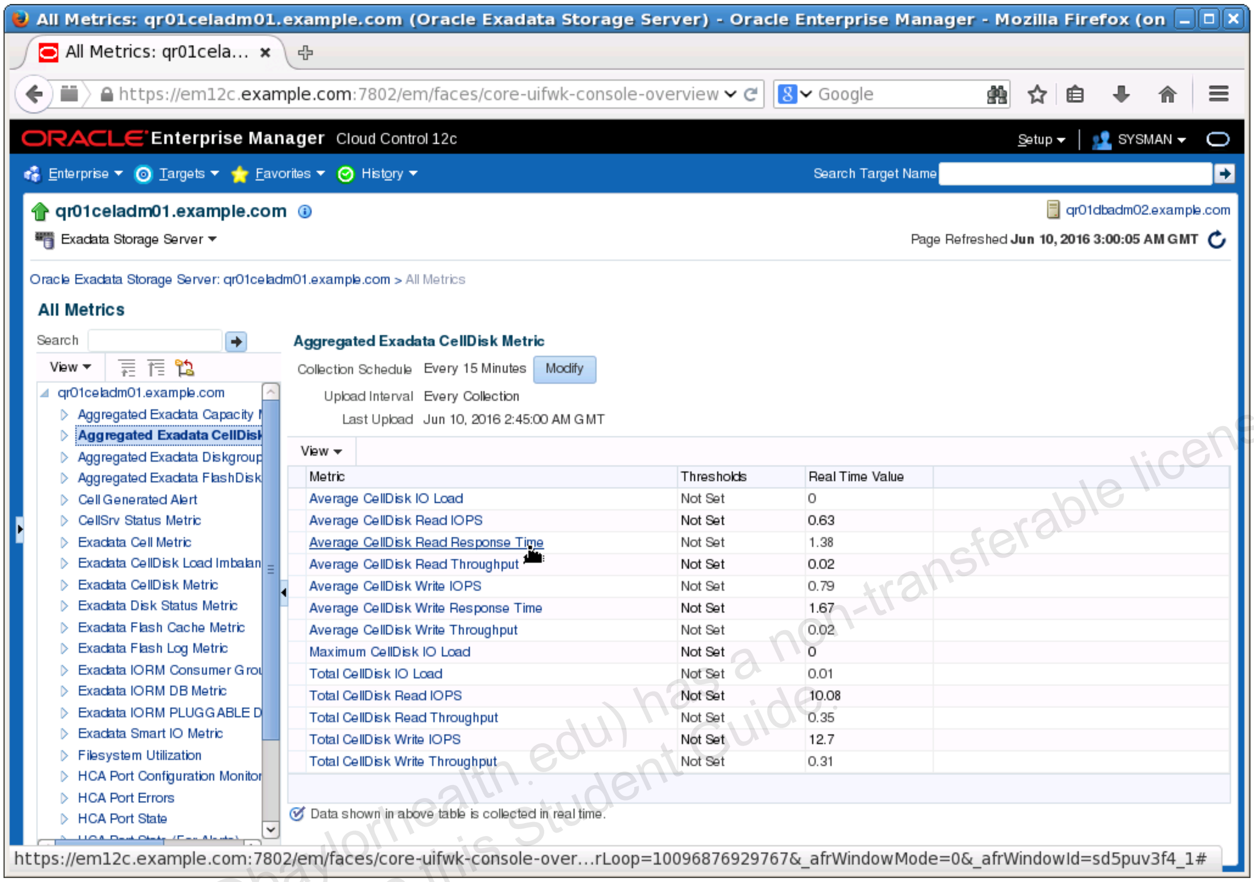
12. The Performance page gathers key cell performance metrics on a single page. However, these are not all the cell metrics gathered by Enterprise Manager. Select the Exadata Storage Server > Monitoring > All Metrics menu command to navigate to the All Metrics page.



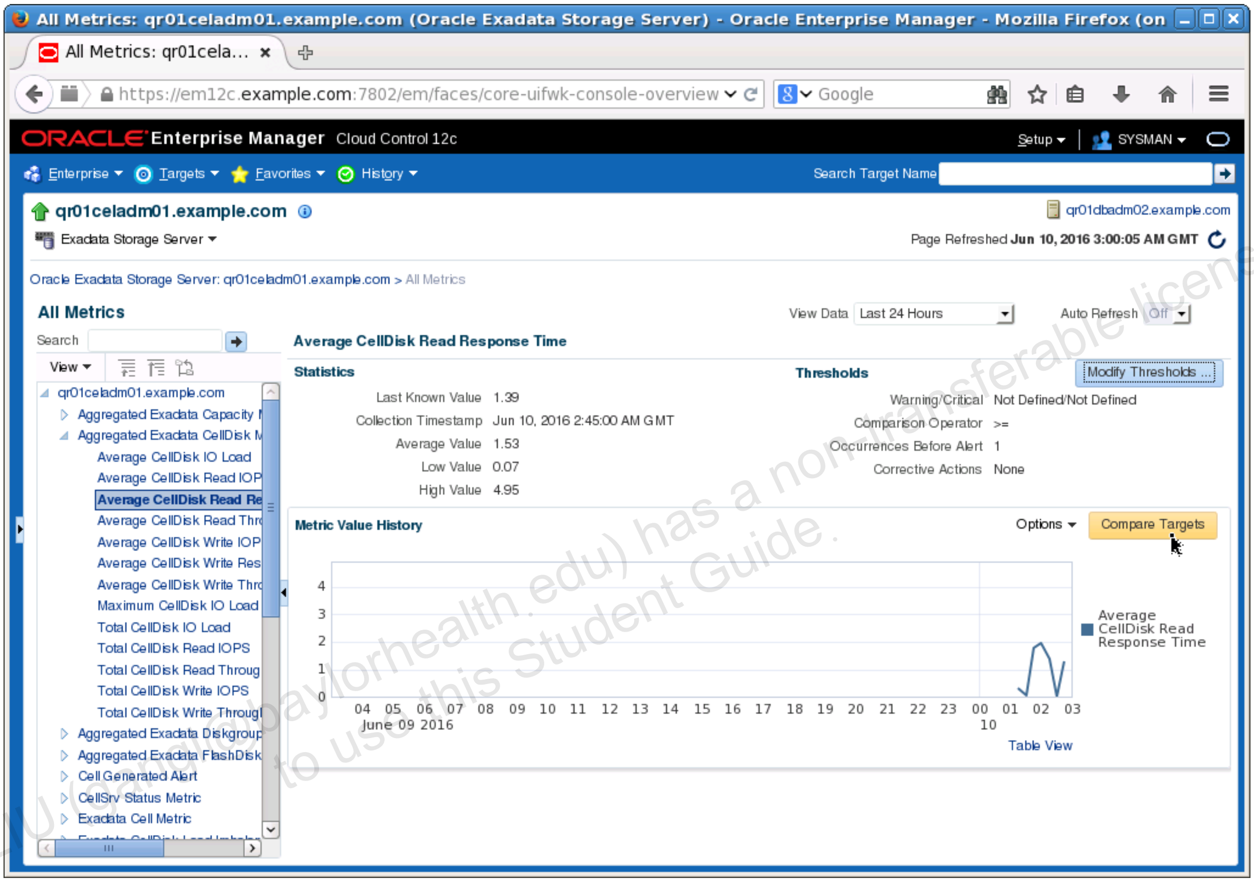
13. The All Metrics page provides an interface for administrators to examine the metrics associated with a monitoring target. For Exadata Storage Servers, this page provides access to more than 100 metrics associated with each cell. The metrics are organized into groups, which are displayed in a hierarchical list on the left side of the page. Click the Aggregated Exadata CellDisk Metric group.



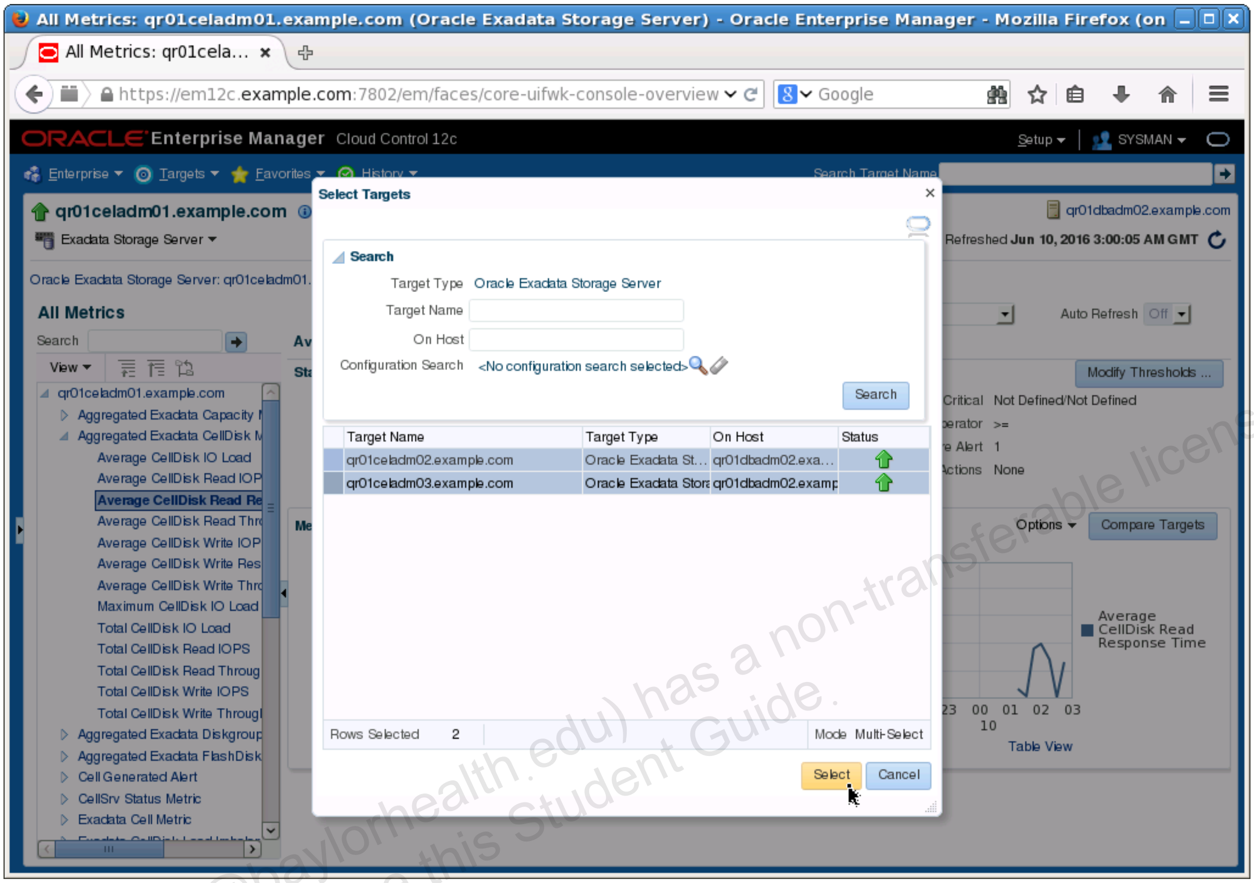
14. Now you see the aggregated cell disk metrics. Notice that these metrics are collected every 15 minutes by default. Click the Average CellDisk Read Response Time metric.



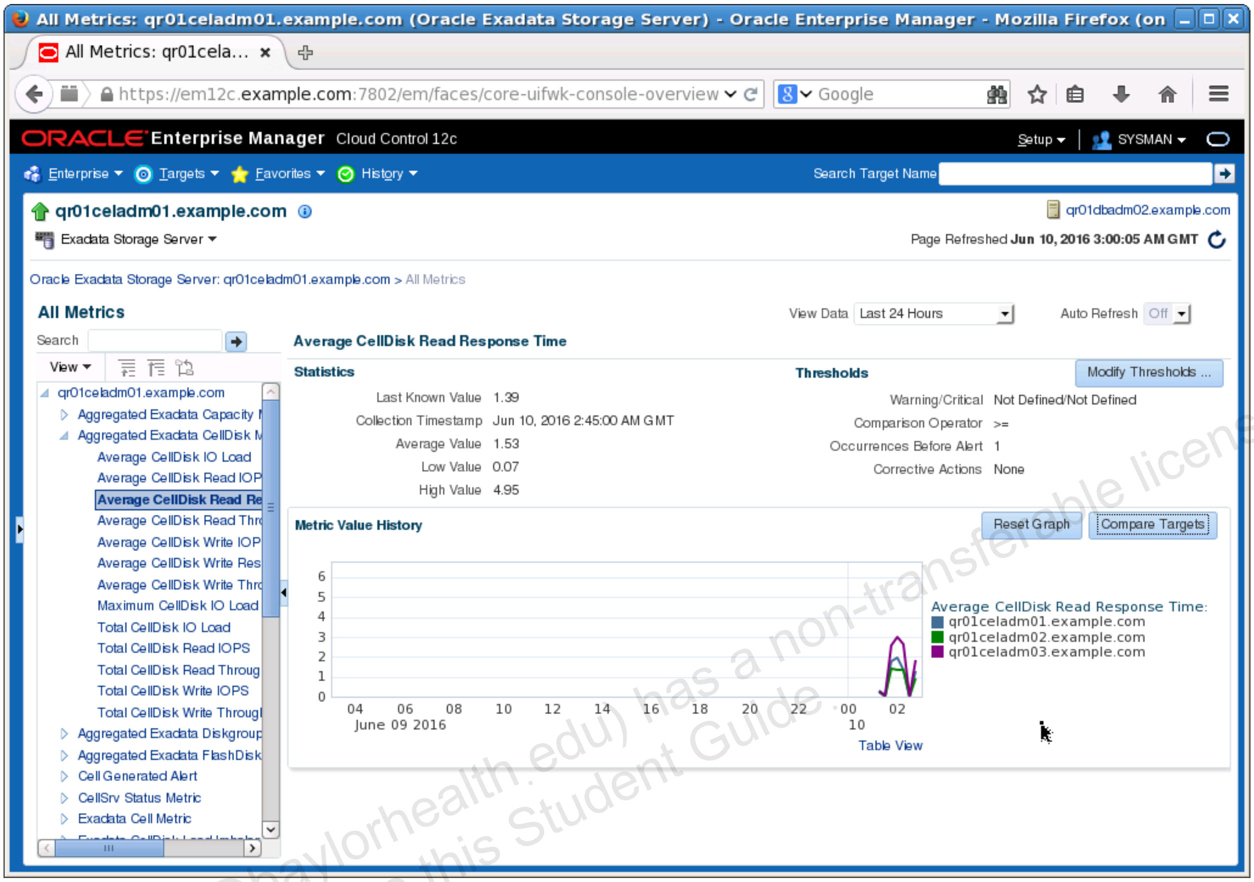
15. When you select an individual metric, you are provided with detailed information about the metric. For Average CellDisk Read Response Time, you can see various statistics along with the thresholds relating to the metric. From this page, you can also modify the threshold settings. A chart, graphing the metric value over time, is also provided. This chart also contains a useful function that enables administrators to compare the metric observation on this cell with the observations on other cells. Click Compare Targets to initiate the comparison.



16. In the dialog box that appears, select both the cells (shift-click `qr01celadm02` and `qr01celadm03`) and click Select.

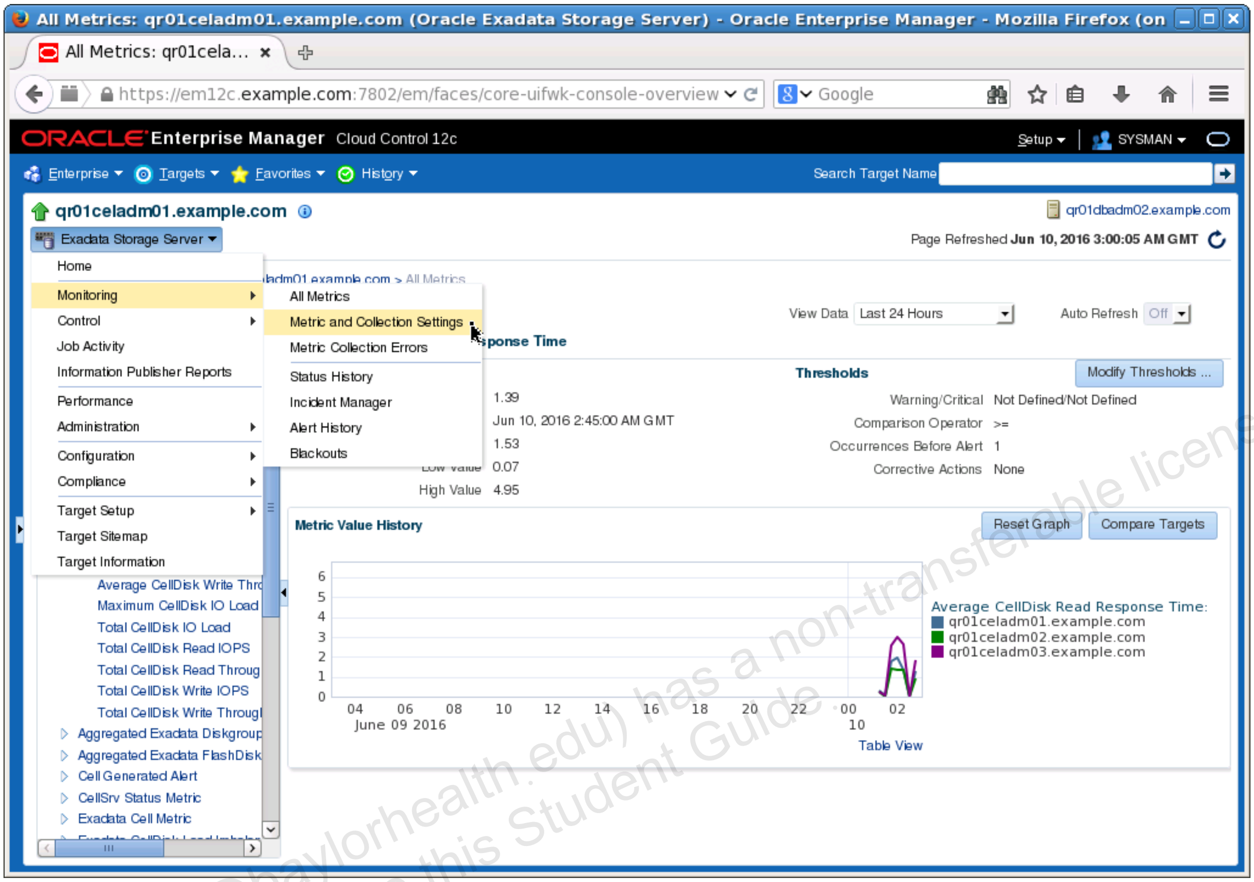


17. Now the chart displays the metric observations for all three cells. This can be very useful for highlighting cell-specific metric anomalies.

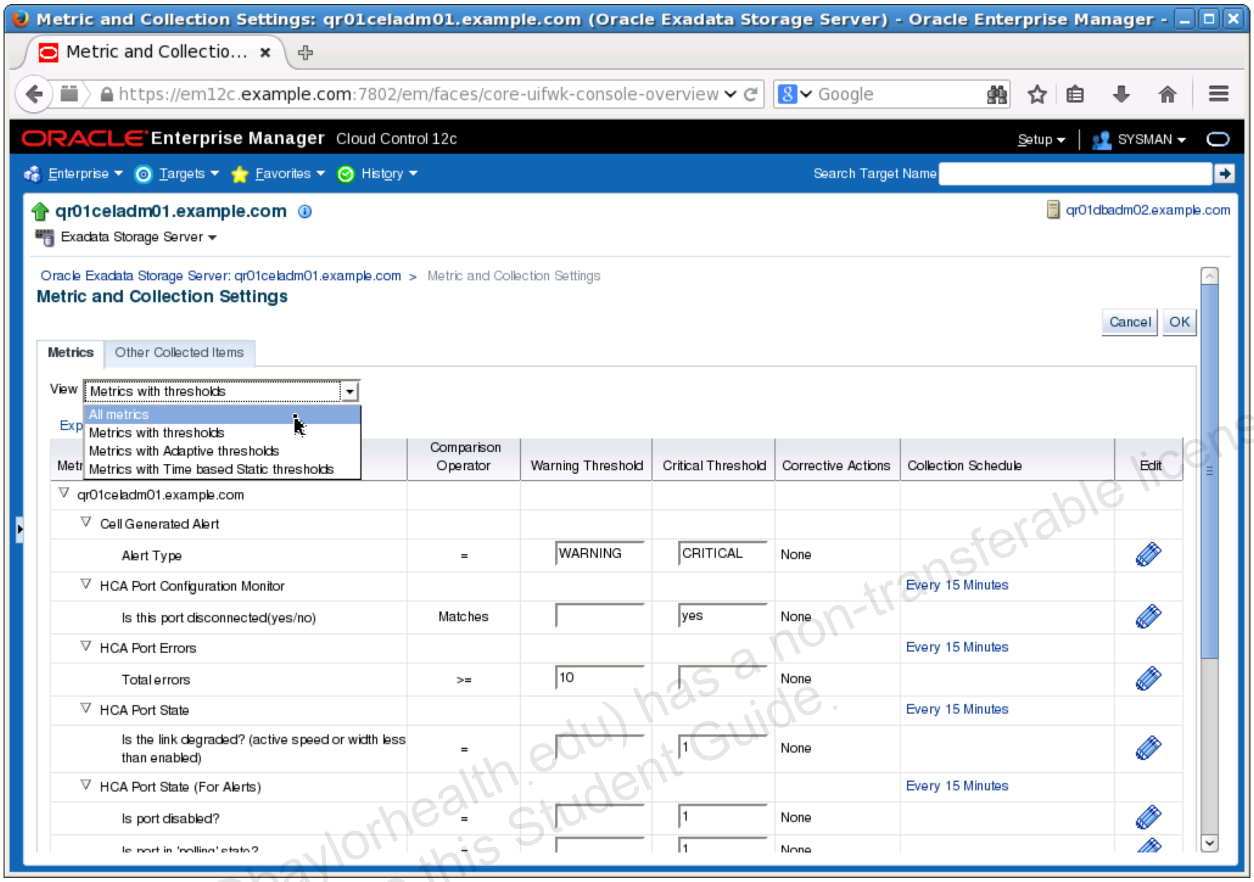


So far, you have seen the storage server performance page and interacted with storage server metrics. In the next part of this practice, you will modify some metric threshold settings.

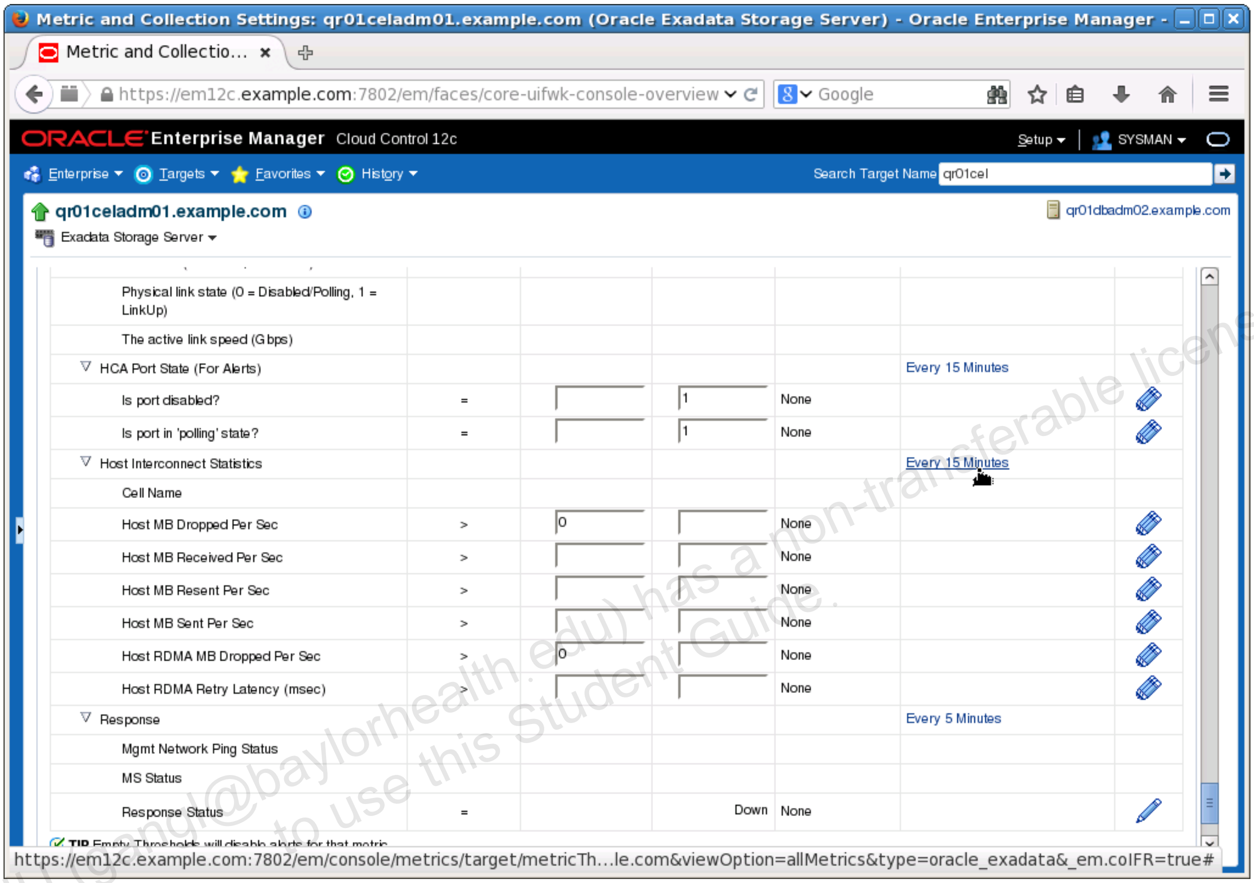
18. Select the Exadata Storage Server > Monitoring > “Metric and Collection Settings” menu command.



19. By default, the “Metric and Collection Settings” page shows the metrics that have associated threshold definitions. Select the option to view all metrics.

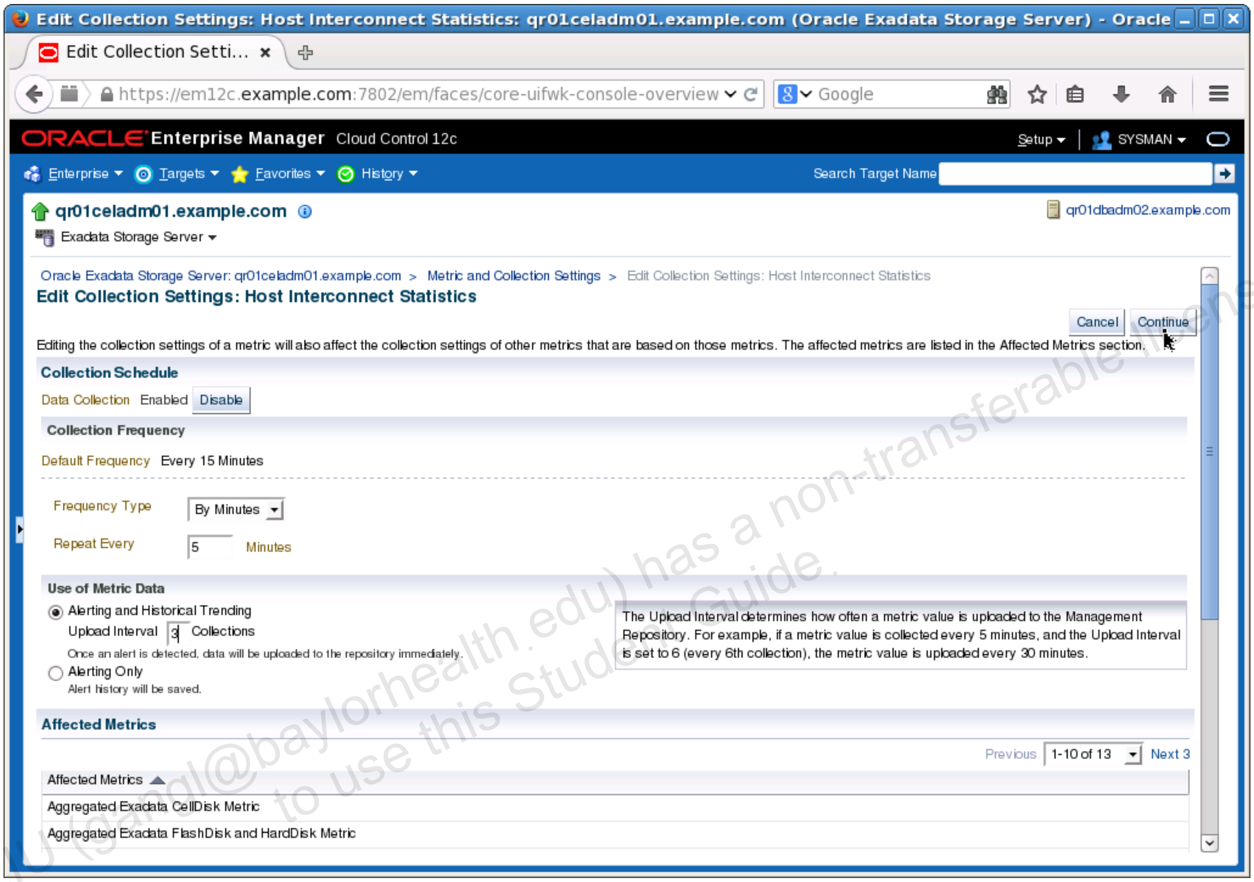


20. Scroll down until you see the group of metrics associated with Host Interconnect Statistics. Set the warning threshold to zero (0) for Host MB Dropped Per Sec and Host RDMA MB Dropped Per Sec. This setting will cause a warning to be generated if packets are dropped on the InfiniBand network. Click Every 15 Minutes to adjust the collection schedule for the Host Interconnect Statistics group of metrics.

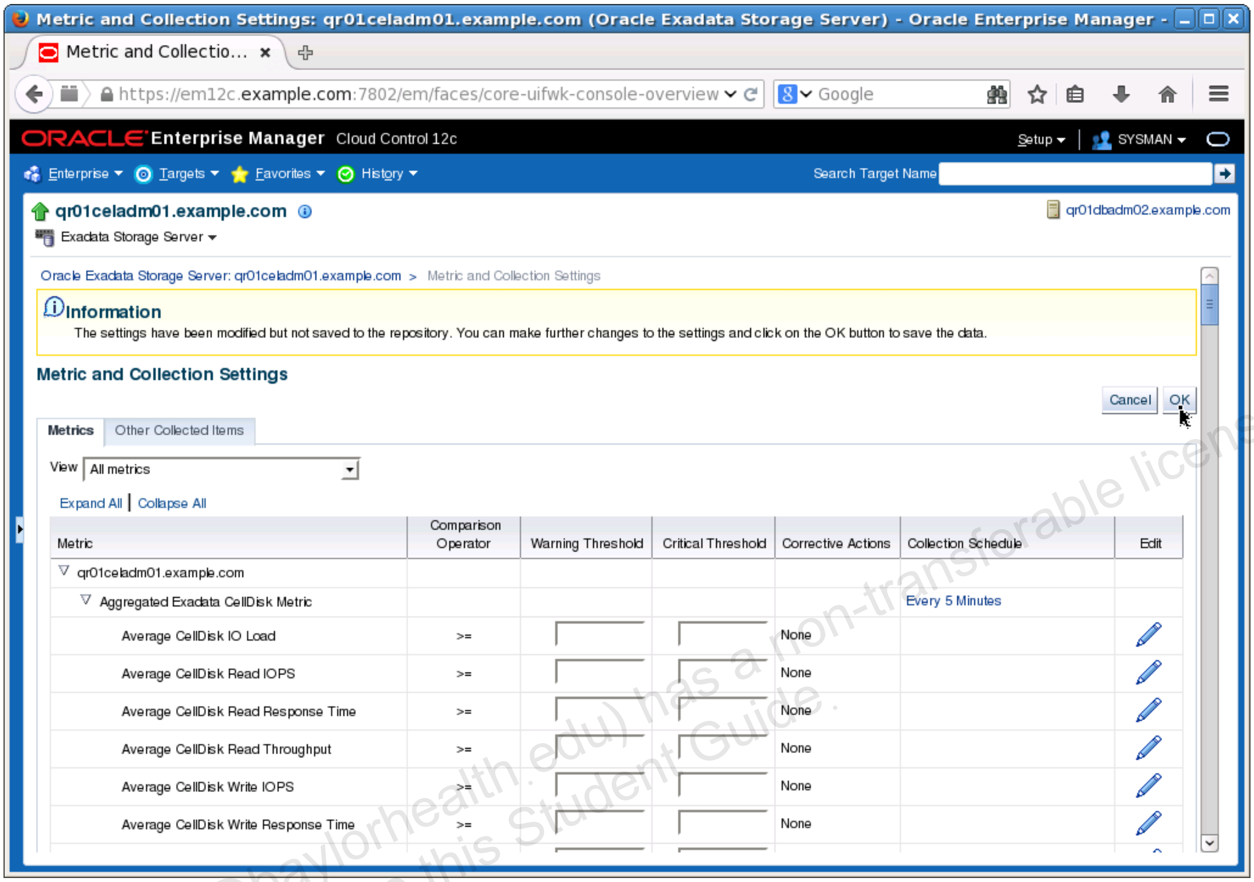


21. Make the following adjustments to the collection schedule:
    - Repeat Every `5` Minutes
    - Upload Interval `3` Collections

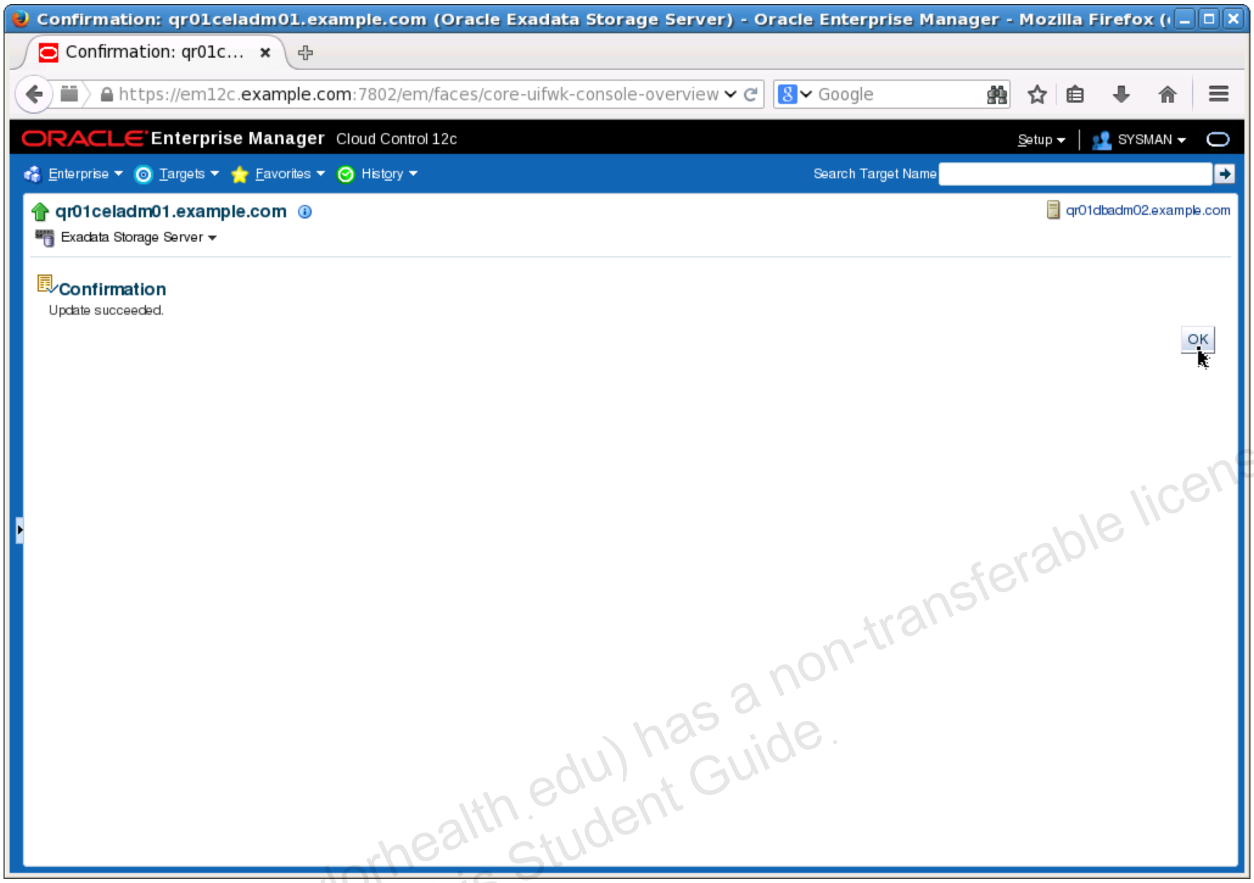
    These settings will cause the metrics to be collected every 5 minutes and uploaded to the Management Repository every 15 minutes. Finally, click Continue to proceed.



22. Click OK to save the thresholds and collection schedule settings, which you made in the previous steps.

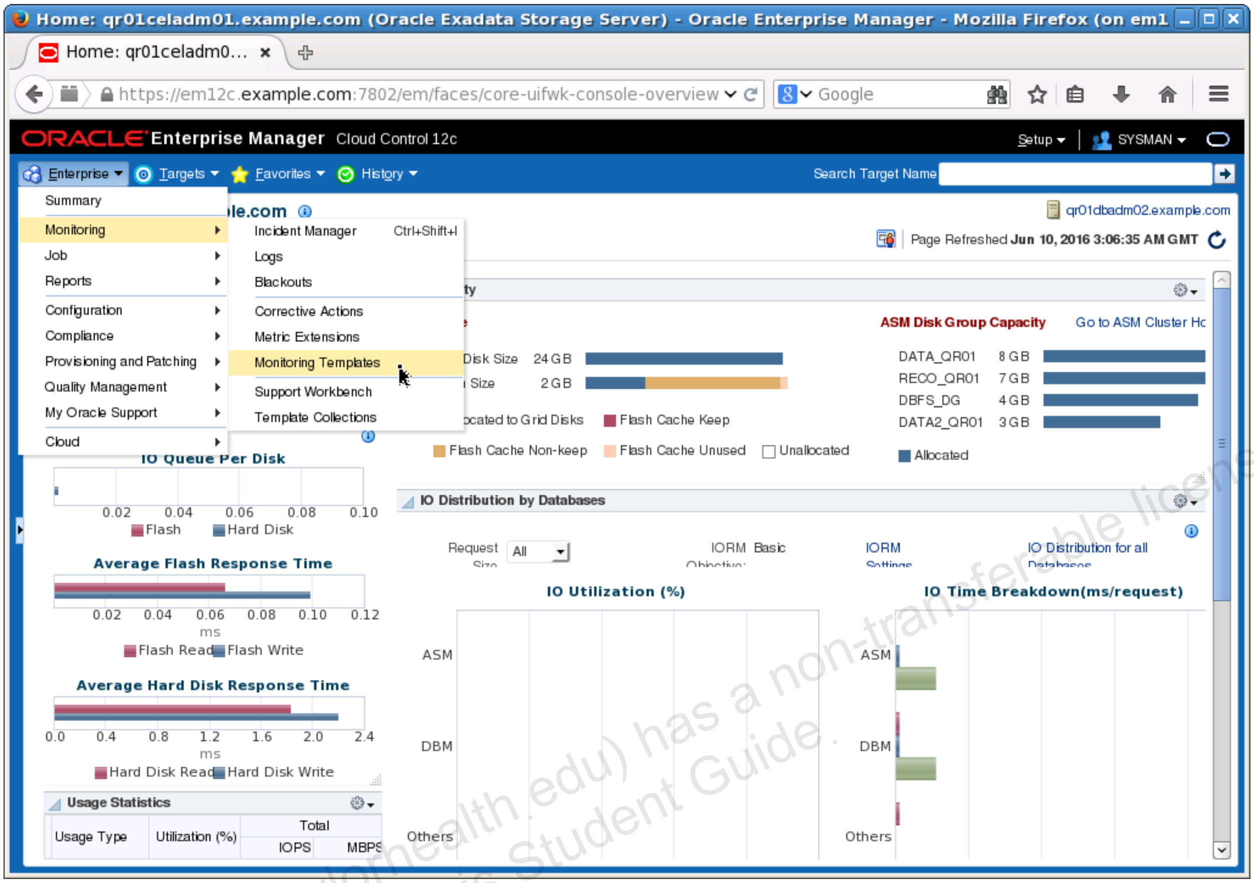


23. Click OK to acknowledge the update confirmation.

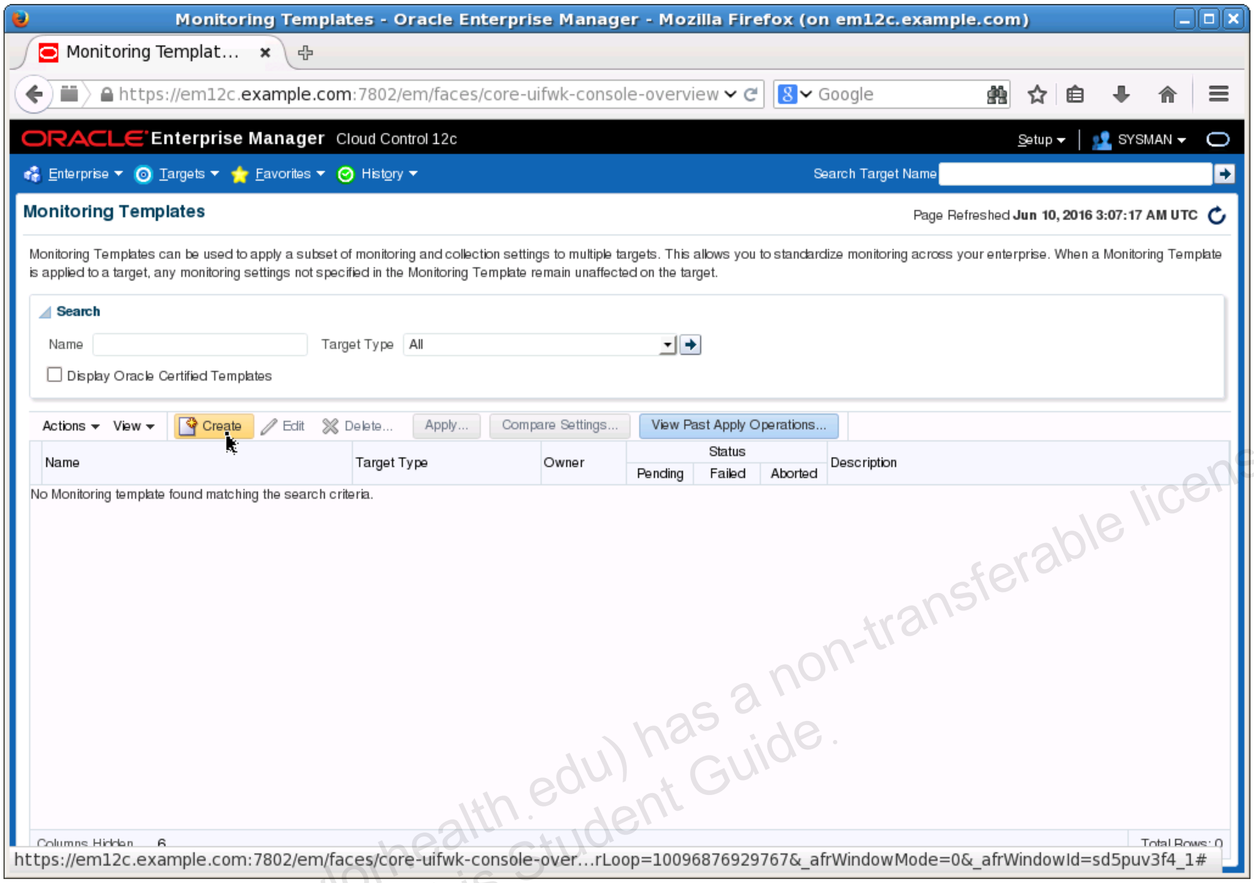


You have just seen how to adjust the Enterprise Manager thresholds and metric collection settings for one Exadata Storage Server. However, what if you have to replicate these settings across numerous Exadata Storage Servers? That is where Monitoring Templates can help. In the next part of this practice, you will examine Monitoring Templates and how they relate to Exadata Storage Servers. You will first create a monitoring template based on one of your existing cells, and then you will apply the template to your other cells.

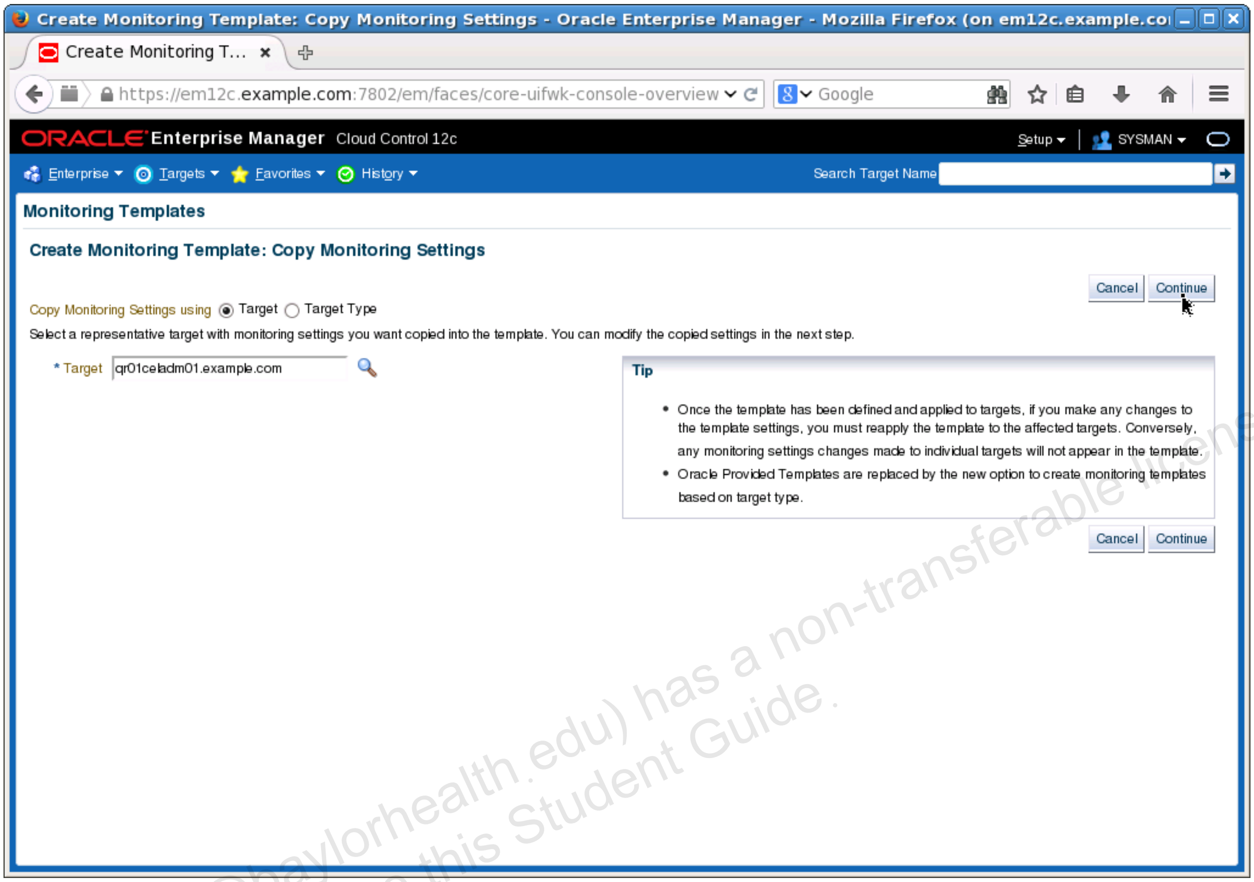
24. Select the Enterprise > Monitoring > Monitoring Templates menu command.



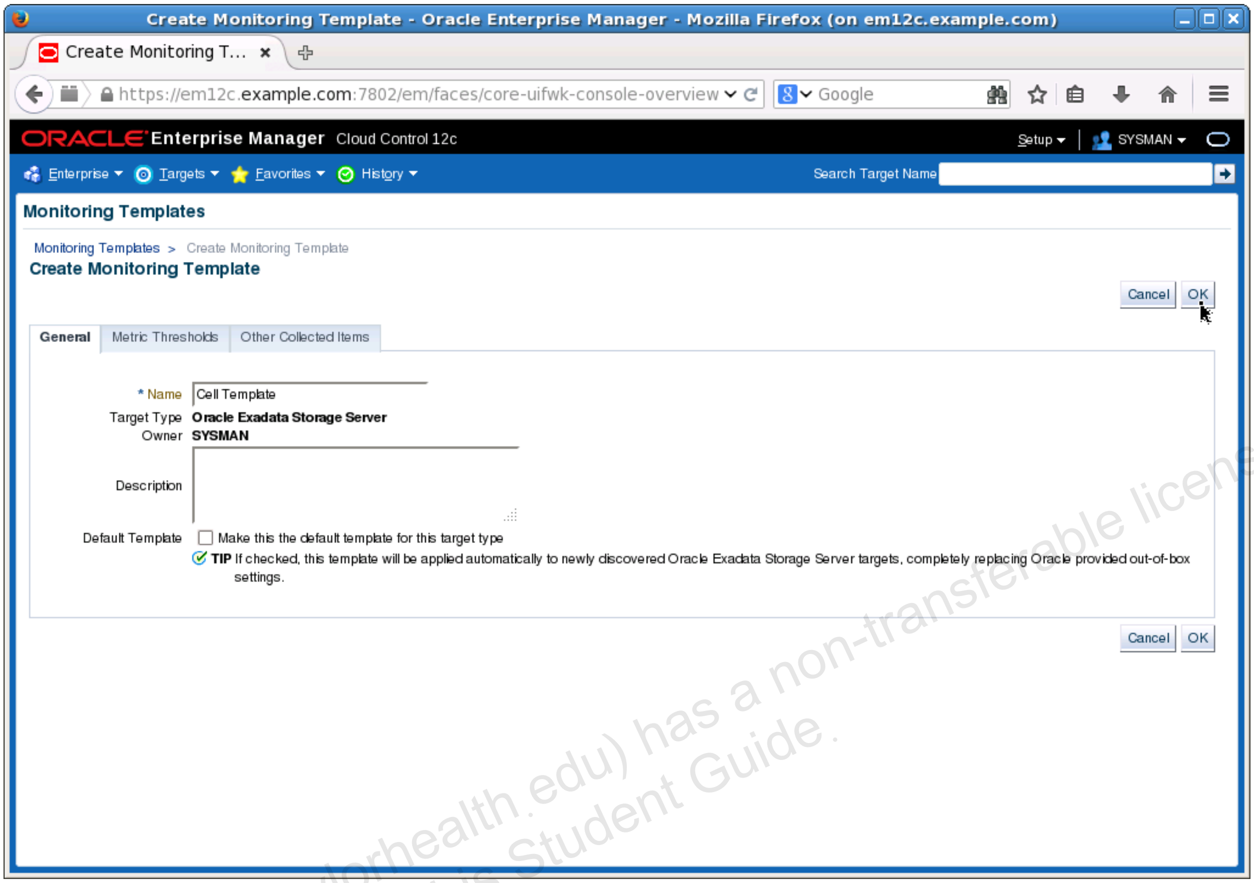
25. Click Create.



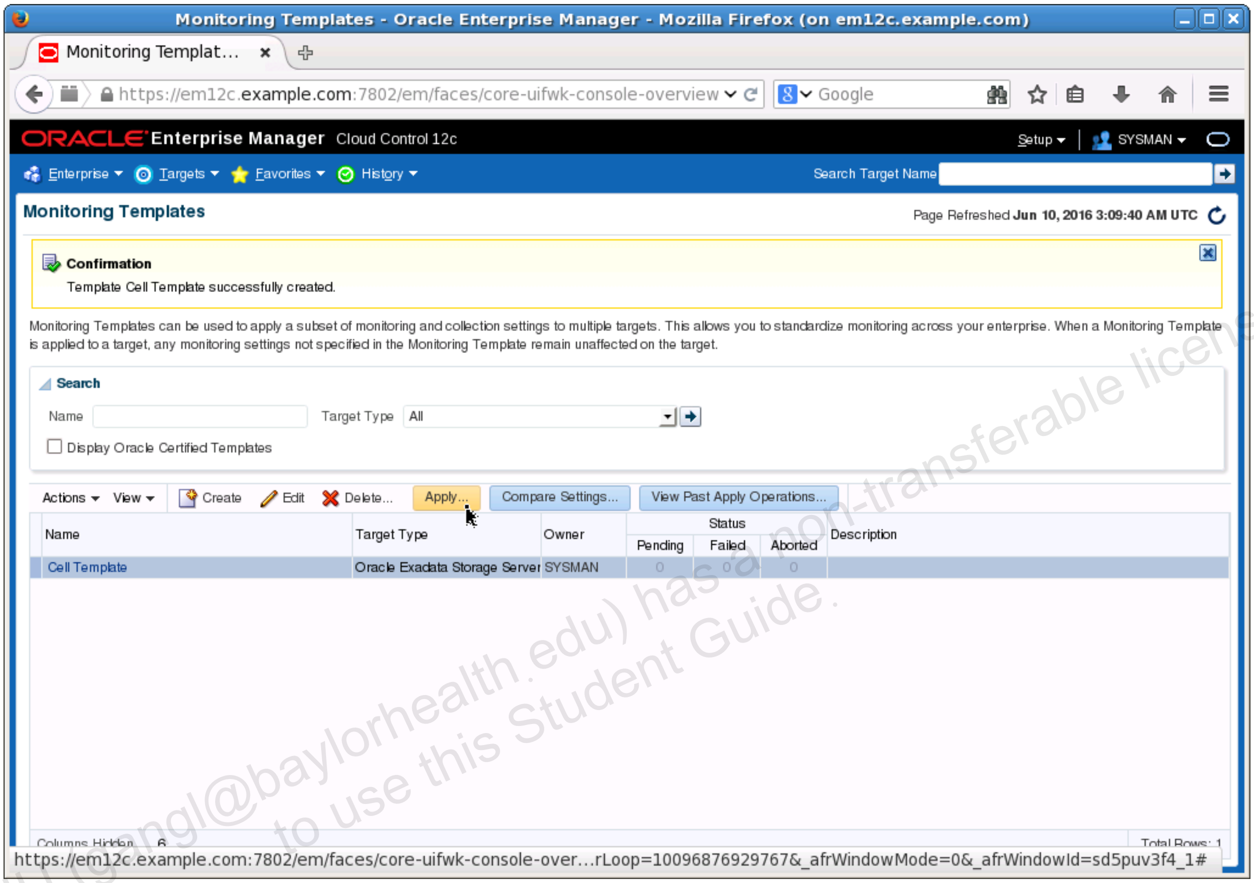
26. Specify the Exadata Storage Server `qr01celadm01.example.com` as the target and click Continue.



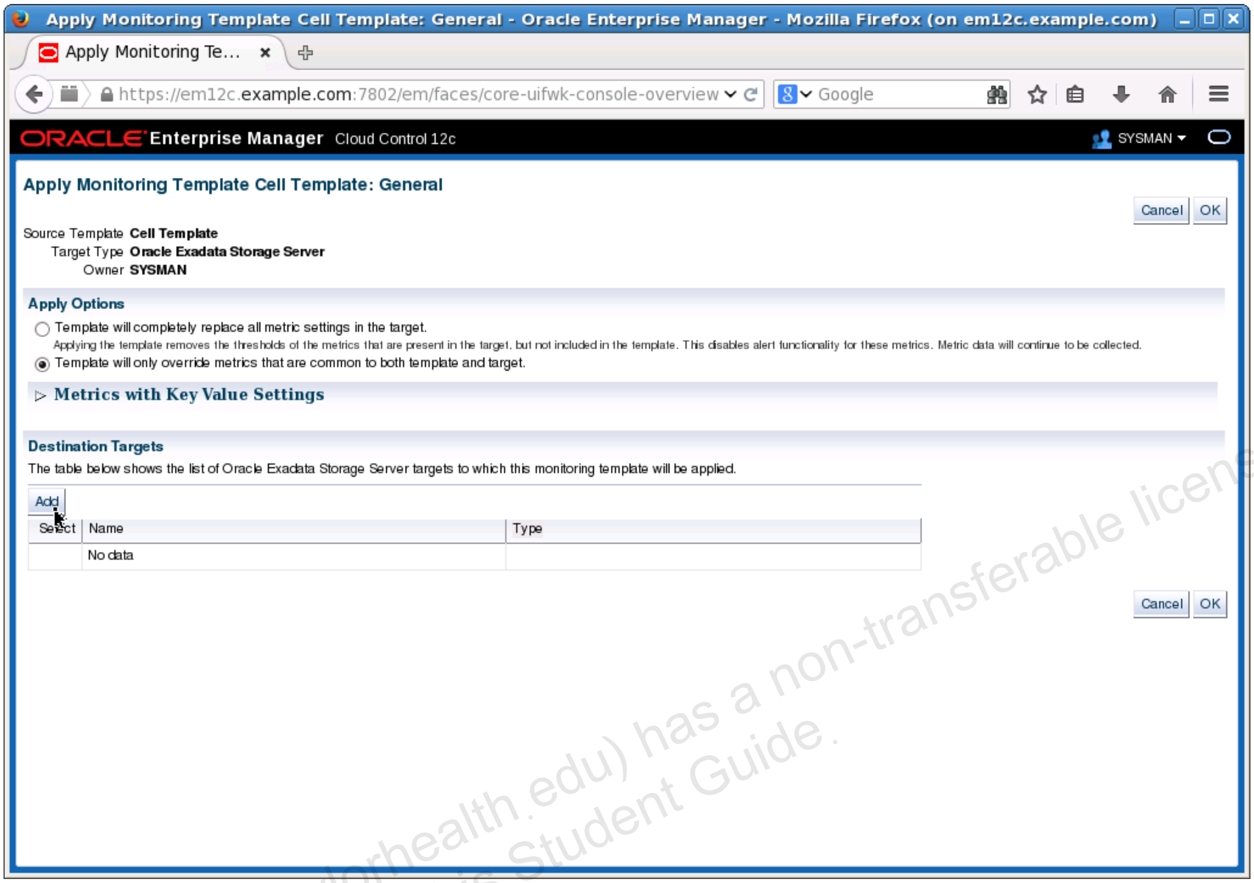
27. Specify `Cell Template` as the template name and click OK.



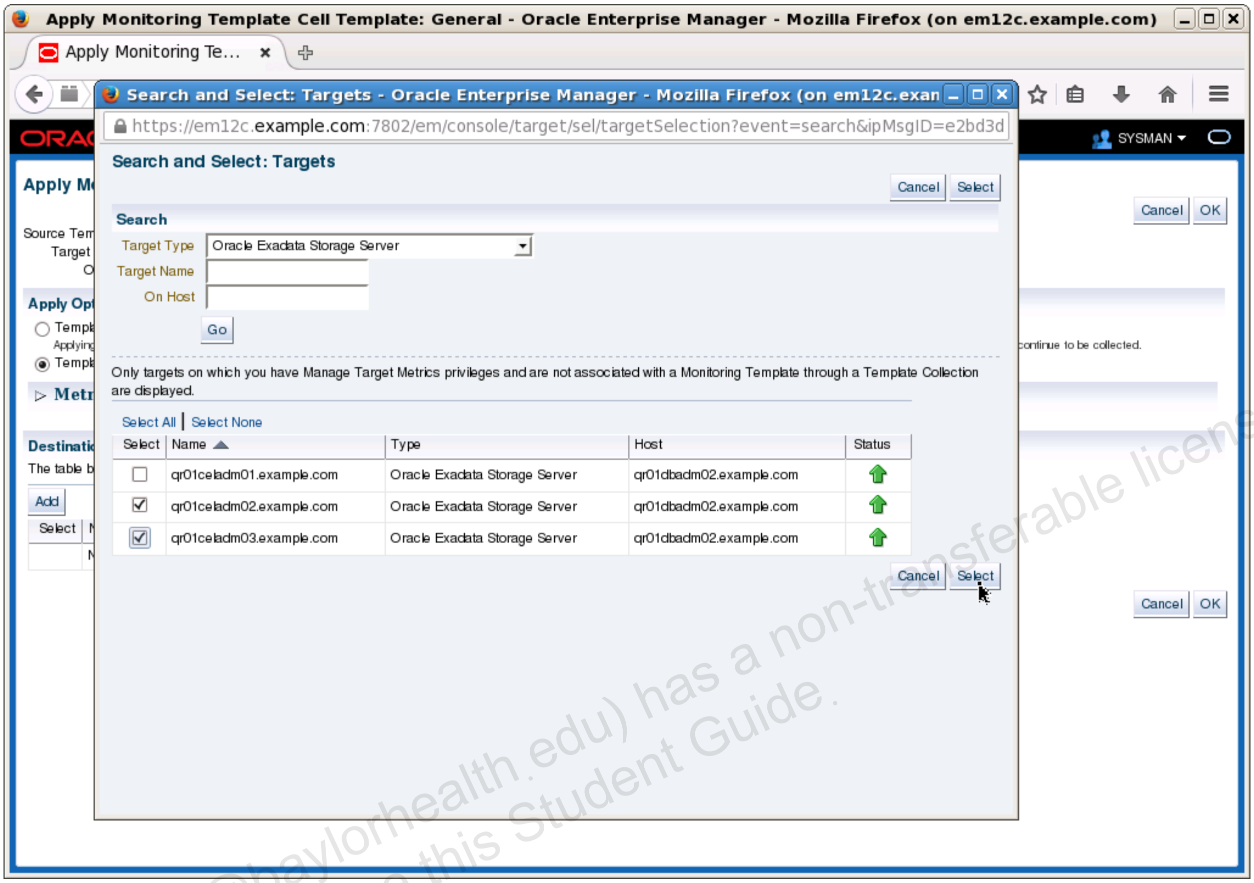
28. At this point, a confirmation message appears indicating the successful creation of your monitoring template. Based on the Exadata Storage Server `qr01celadm01`, this template includes the thresholds and metric collection settings that you customized earlier in the practice. Select the template and click Apply to start the process of applying this template to your other storage servers.



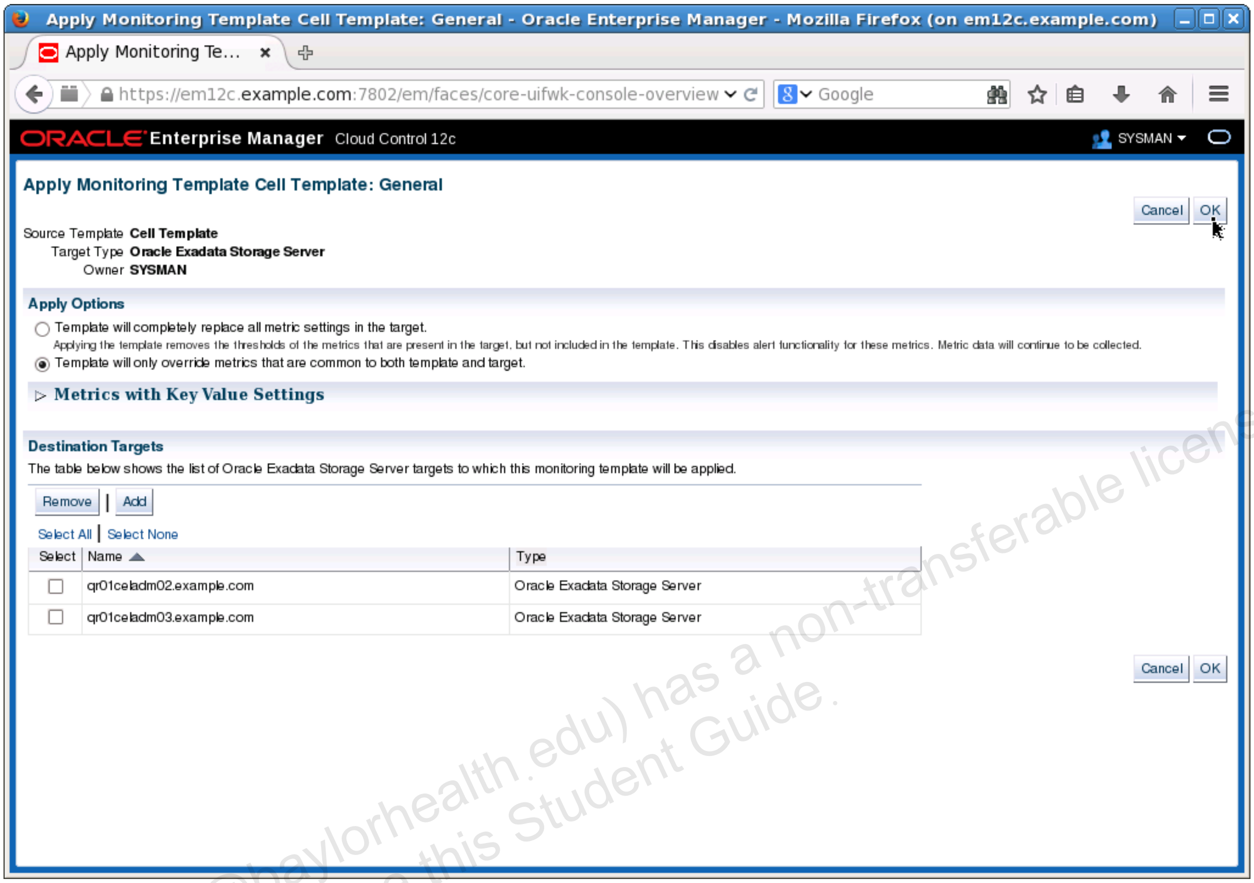
29. Click Add to specify the destination targets.



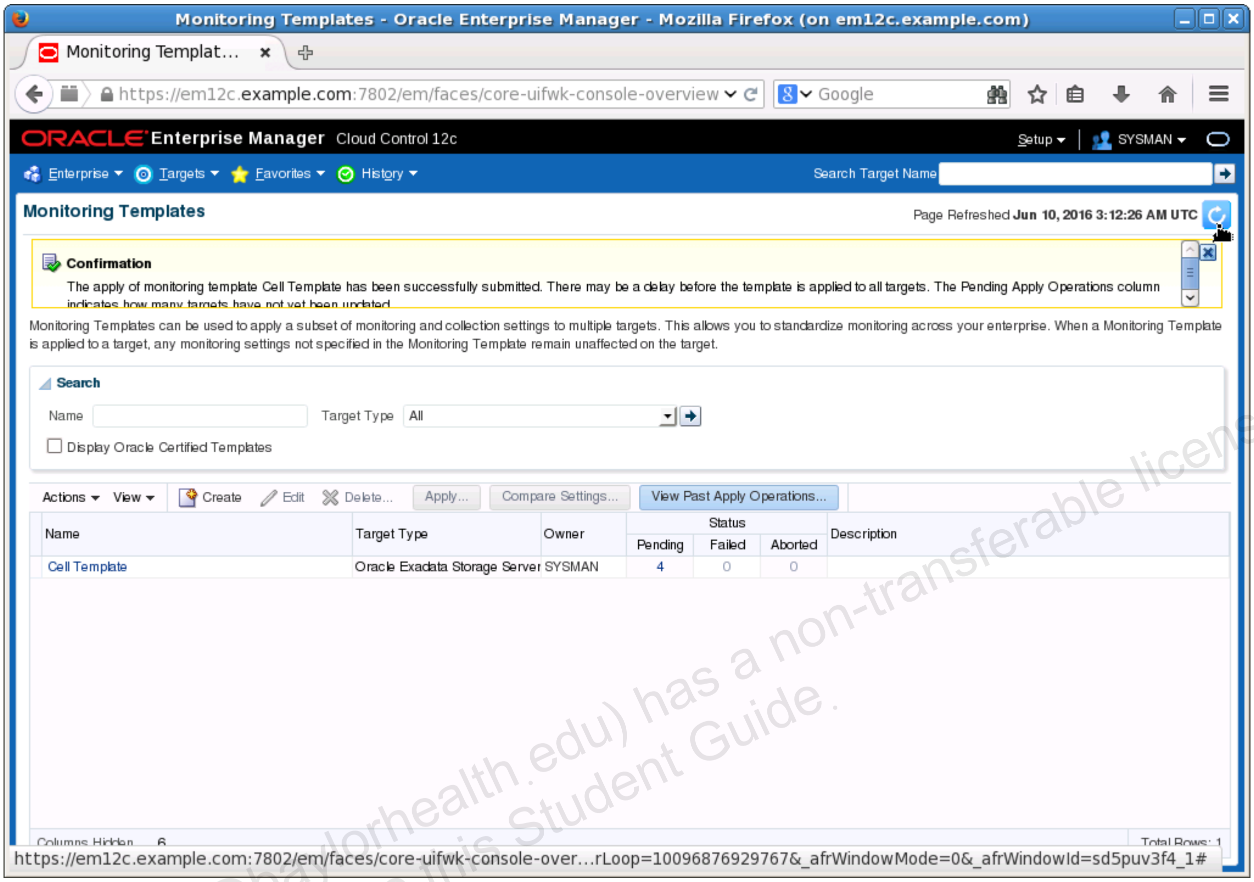
30. Select the Exadata Storage Servers `qr01celadm02.example.com` and `qr01celadm03.example.com`, and then click Select.



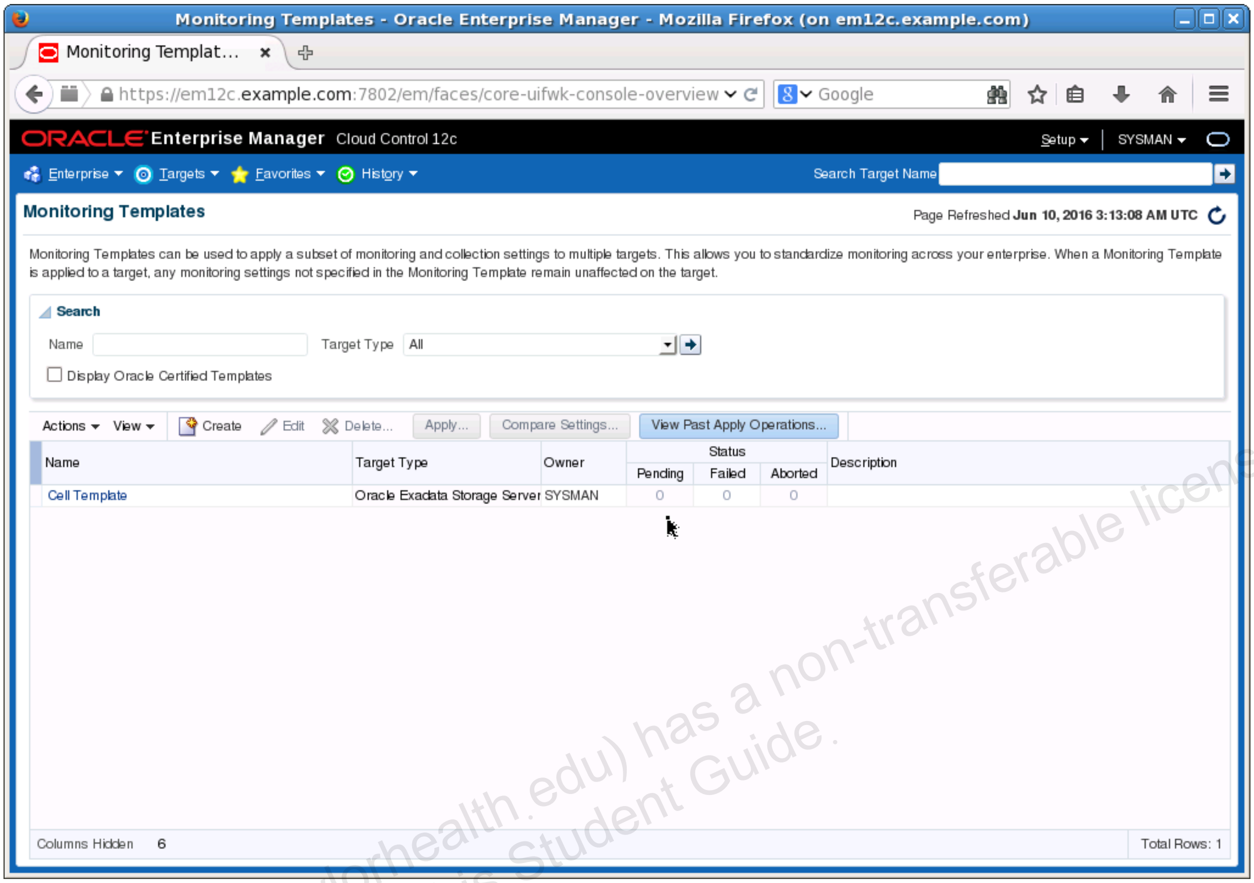
31. Click OK to apply your monitoring template (based on `qr01celadm01`) to `qr01celadm02` and `qr01celadm03`.



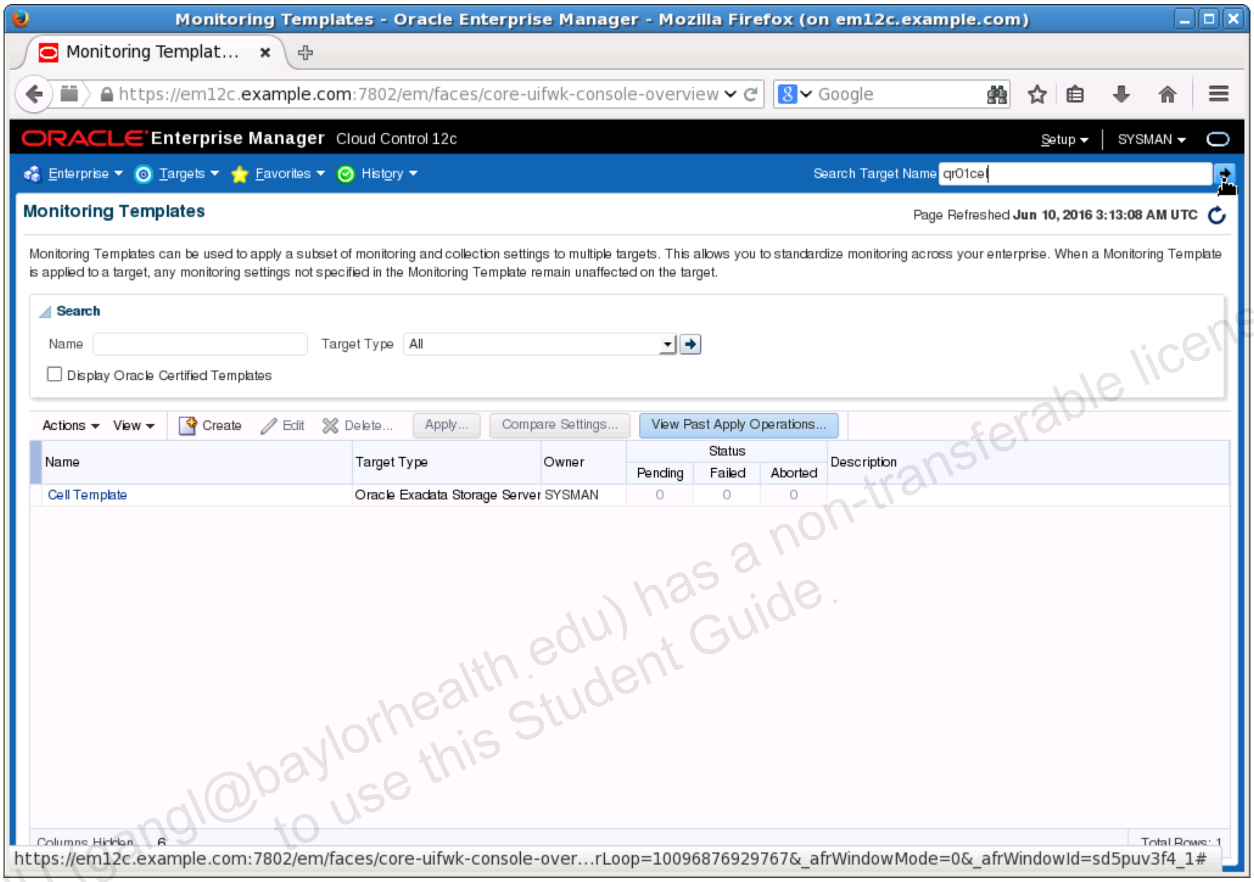
32. Application of the monitoring template occurs as a background task. Periodically, refresh the page while Pending shows a value greater than zero.



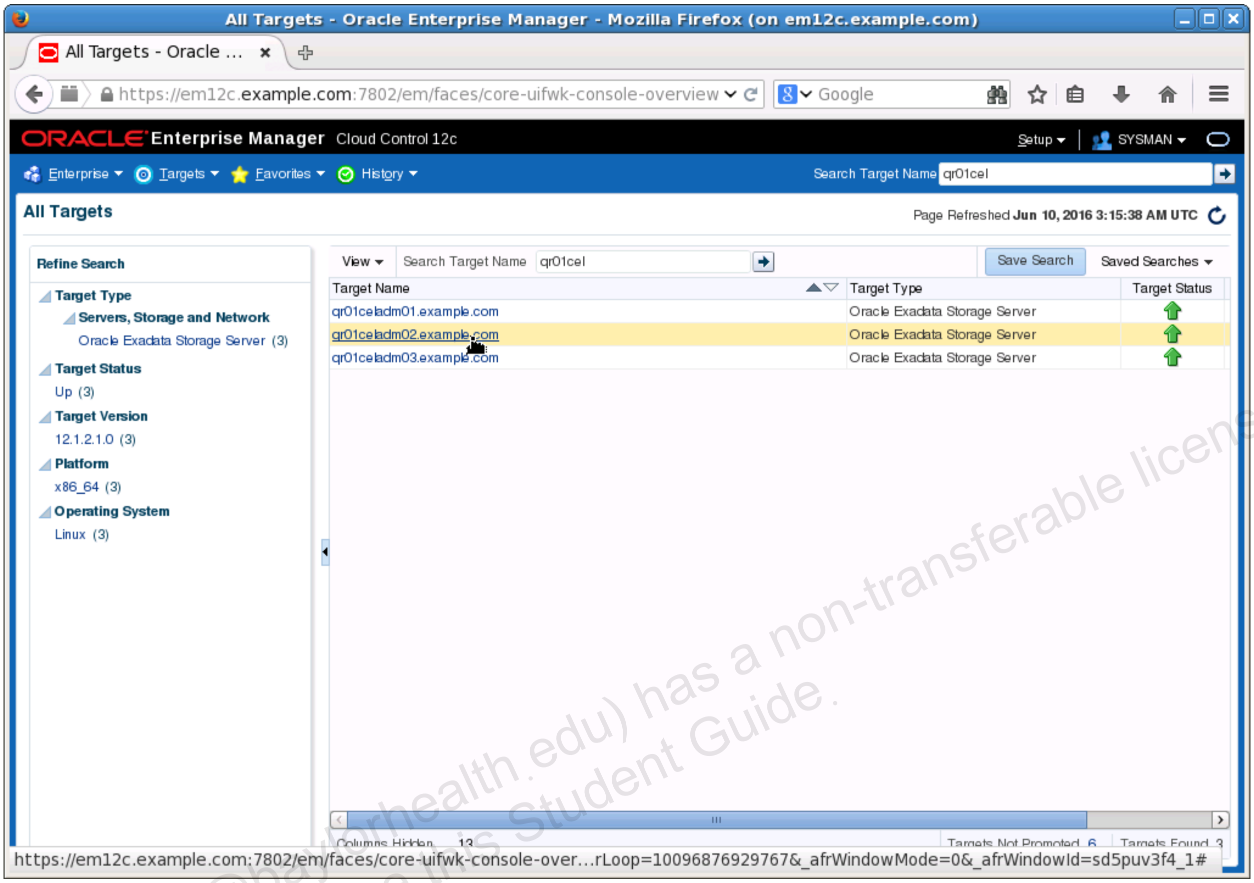
33. When Pending shows zero, the application process is finished.



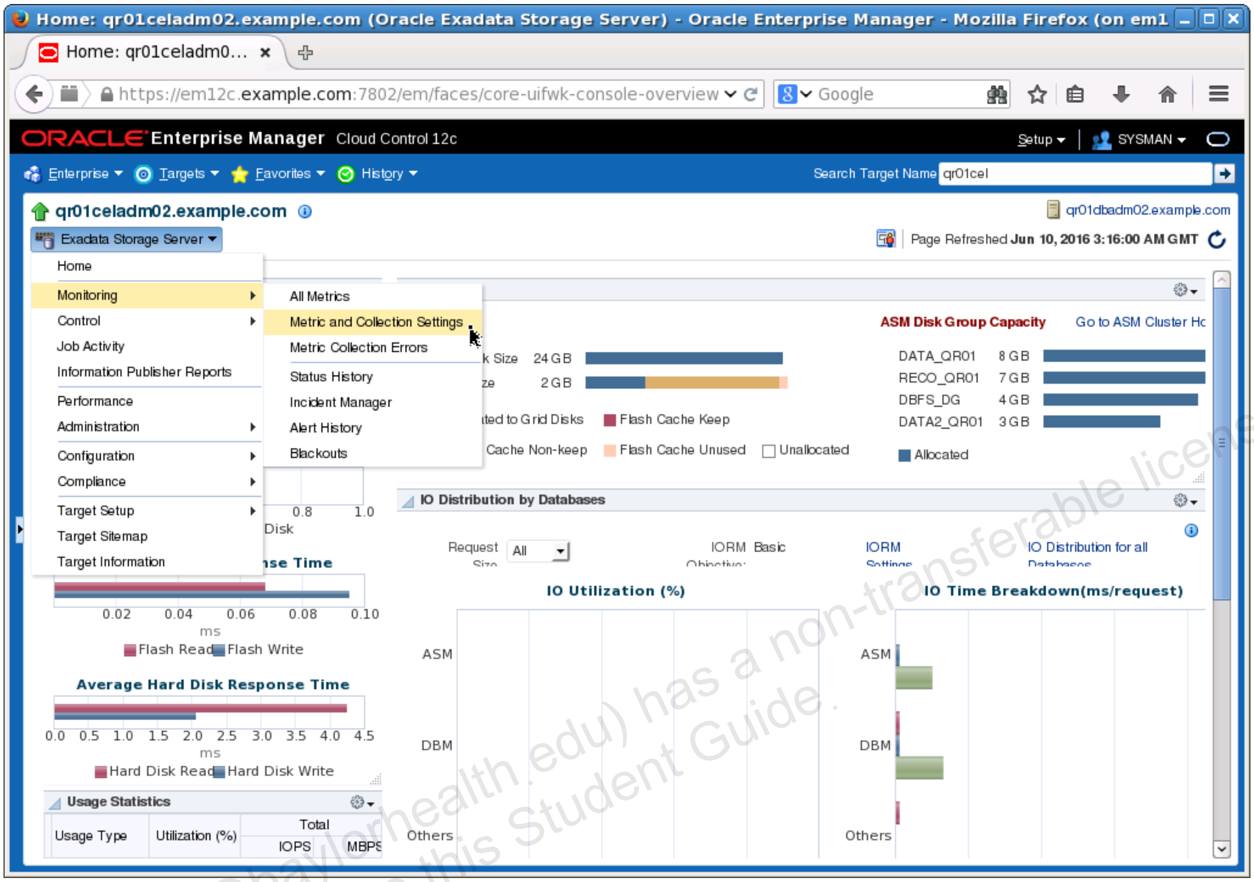
34. To confirm the application of the monitoring template, you can examine the cell threshold settings on `qr01celadm02` and verify that they match the settings from `qr01celadm01`. To commence this process, enter `qr01cel` in the Search Target Name field and click the Search button. Ensure that you use the search facility located on the right side of the menu bar at the top of the page and not the Search field inside the page.



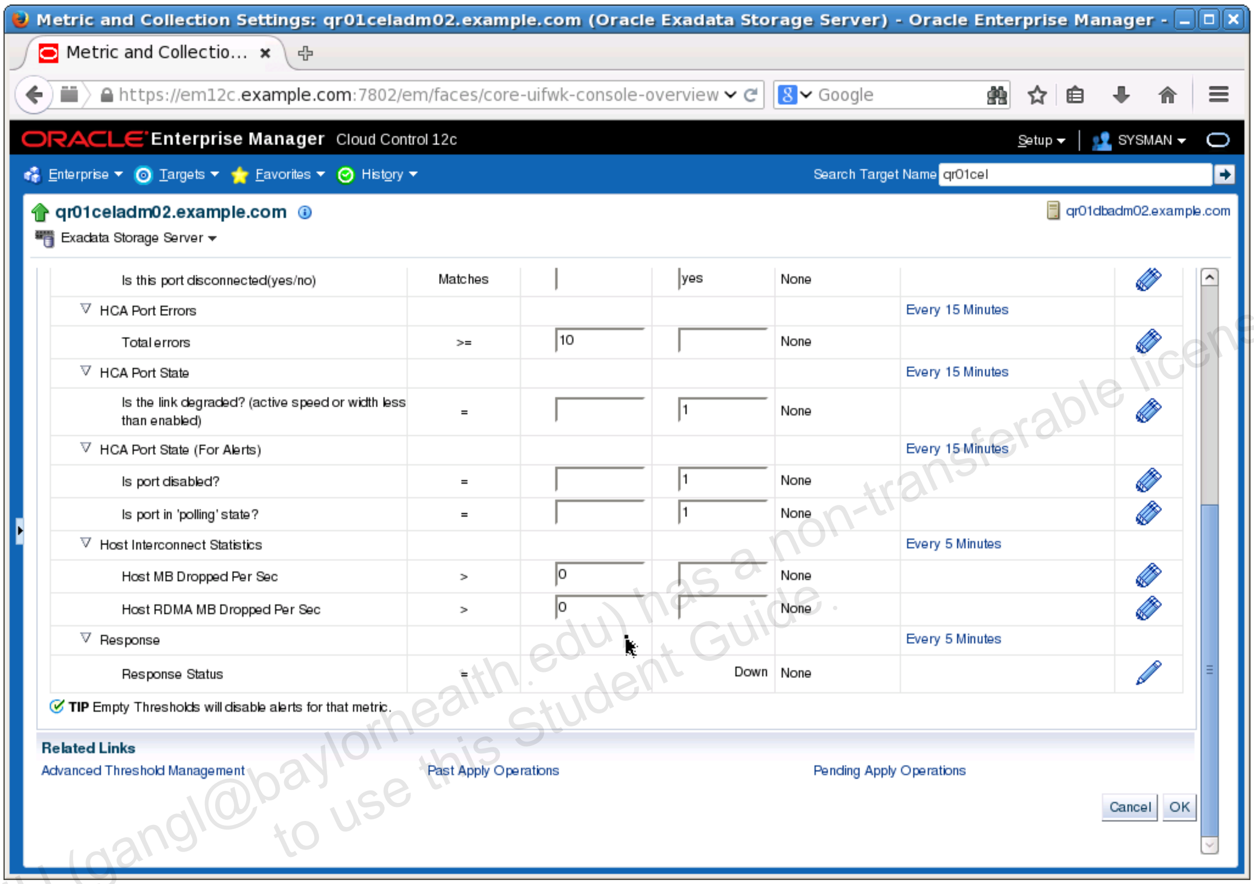
35. Click qr01celadm02.example.com to navigate to the Exadata Storage Server home page for qr01celadm02.



36. Select the Exadata Storage Server > Monitoring > “Metric and Collection Settings” menu command.



37. Scroll down the page until you see the thresholds associated with the metrics Host MB Dropped Per Sec and Host RDMA MB Dropped Per Sec. These threshold settings were created when the monitoring template was applied to this cell. Notice also that the collection schedule for these metrics is Every 5 Minutes, rather than the default (Every 15 Minutes). This is another consequence of applying the monitoring template.



In the next part of this practice, you will examine the difference between thresholds set in the Exadata Storage Server and thresholds set in Enterprise Manager. You will also examine the different alerts generated when these thresholds are crossed. First, you will examine cell-based thresholds and how the associated alerts are propagated to Enterprise Manager.

38. Establish a terminal connection to the `qr01celadm01` Exadata cell as the `celladmin` user.

```
$ ssh celladmin@qr01celadm01
celladmin@qr01celadm01's password: ????????
[celladmin@qr01celadm01 ~]$
```

39. Launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

40. Execute the `DROP ALERTHISTORY ALL` command to delete the cell alert history records.

```
CellCLI> drop alerthistory all
Alert 1_1 successfully dropped
Alert 1_2 successfully dropped

CellCLI>
```

41. Examine the current value of the metric relating to storage server memory utilization (`cl_memut`).

```
CellCLI> list metriccurrent cl_memut
         CL_MEMUT        qr01celadm01    92 %

CellCLI>
```

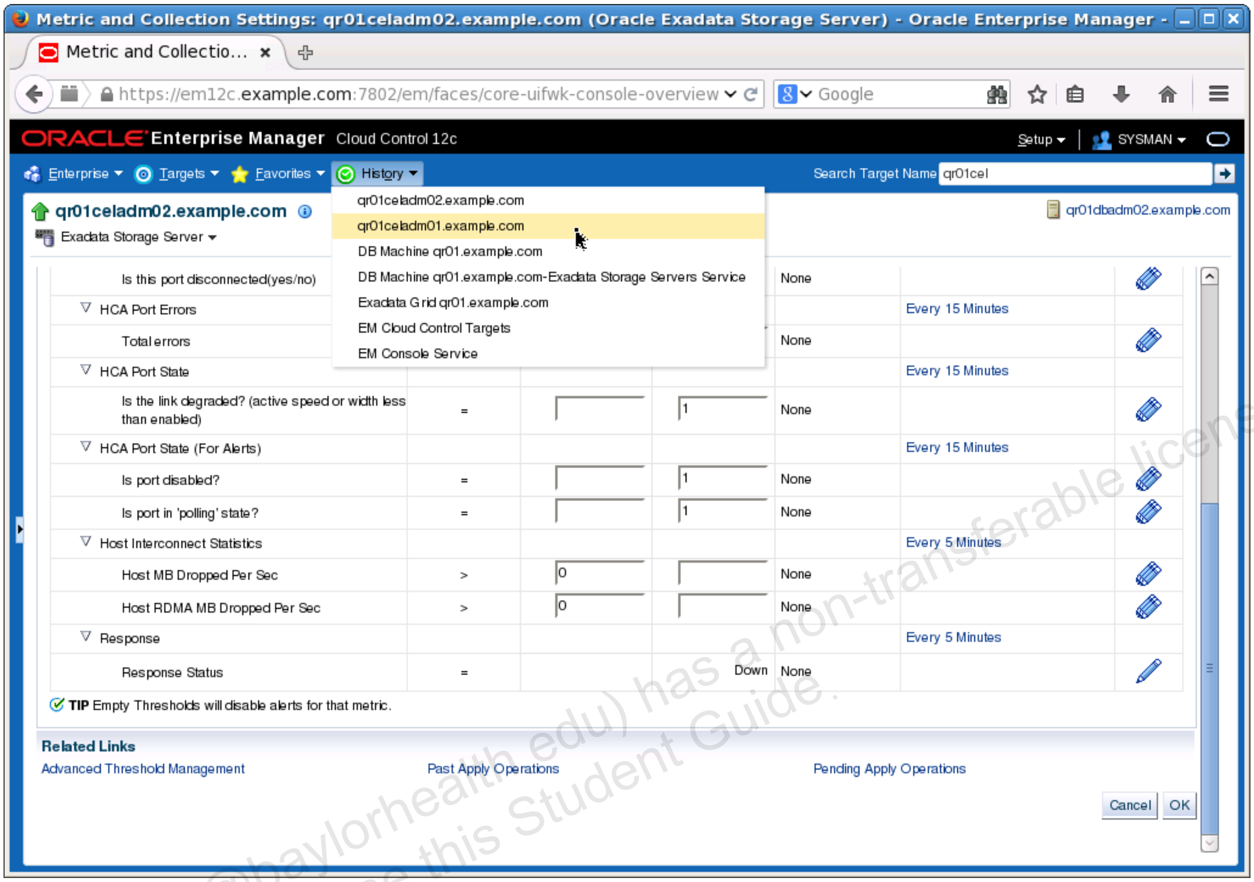
42. Create a cell-based threshold for `cl_memut`. Configure the threshold so that a critical alert is generated when memory utilization exceeds 10% (which will always be the case in your laboratory environment).

```
CellCLI> create threshold cl_memut comparison='>', critical=10
Threshold cl_memut successfully created

CellCLI>
```

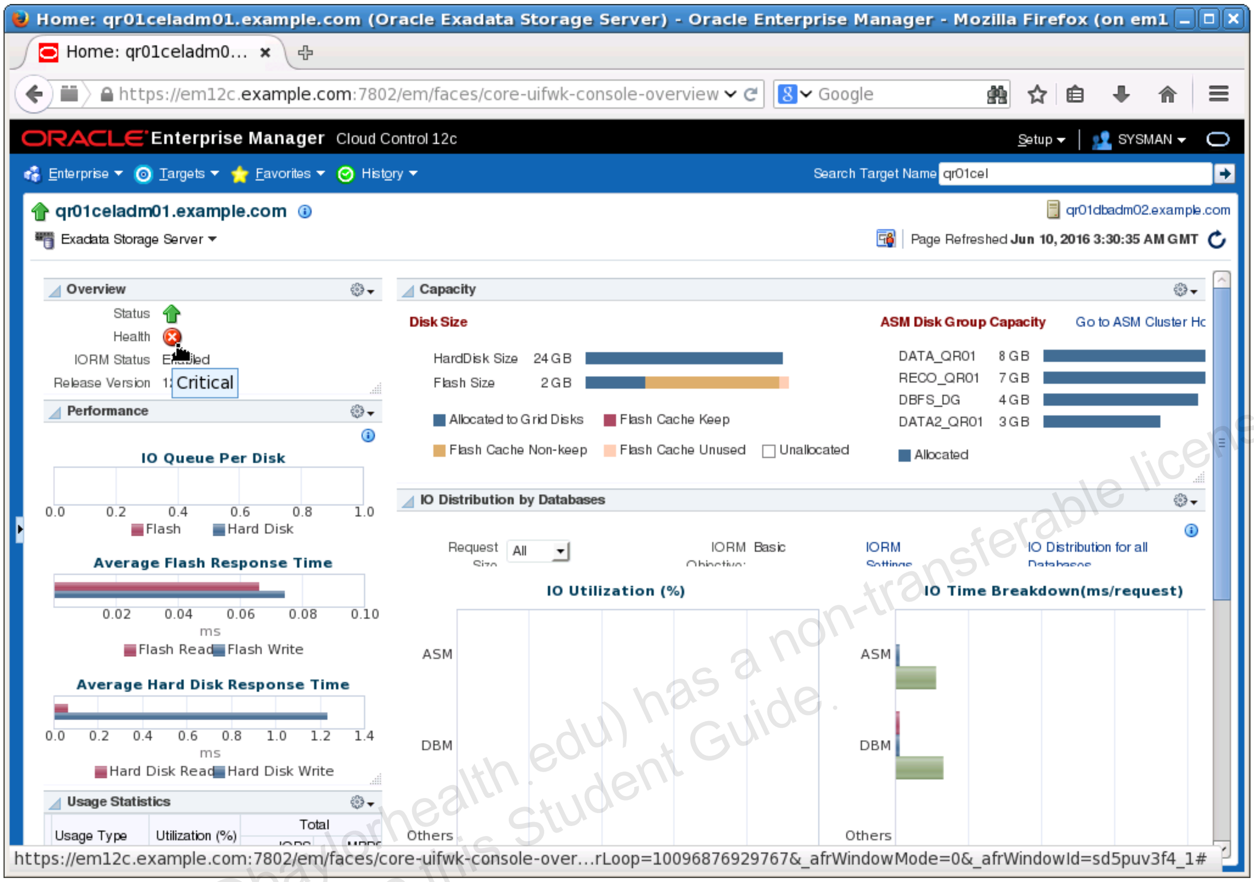
43. Execute the `LIST ALERTHISTORY` command. You should see an alert related to the threshold you created in the previous step. If you cannot see the expected alert, periodically re-execute the `LIST ALERTHISTORY` command until the alert is visible. After the alert becomes visible, leave your CellCLI session running and proceed to the next step.

```
CellCLI> list alerthistory
         1_1     2016-06-10T03:23:18+00:00       critical
"The critical threshold for the following metric has been
crossed. Metric Name        : CL_MEMUT  Metric Description :
Percentage of total physical memory on this server that is
currently used  Object Name        : qr01celadm01  Current Value
: 93.0 %  Threshold Value    : 10.0 %  "

CellCLI>
```

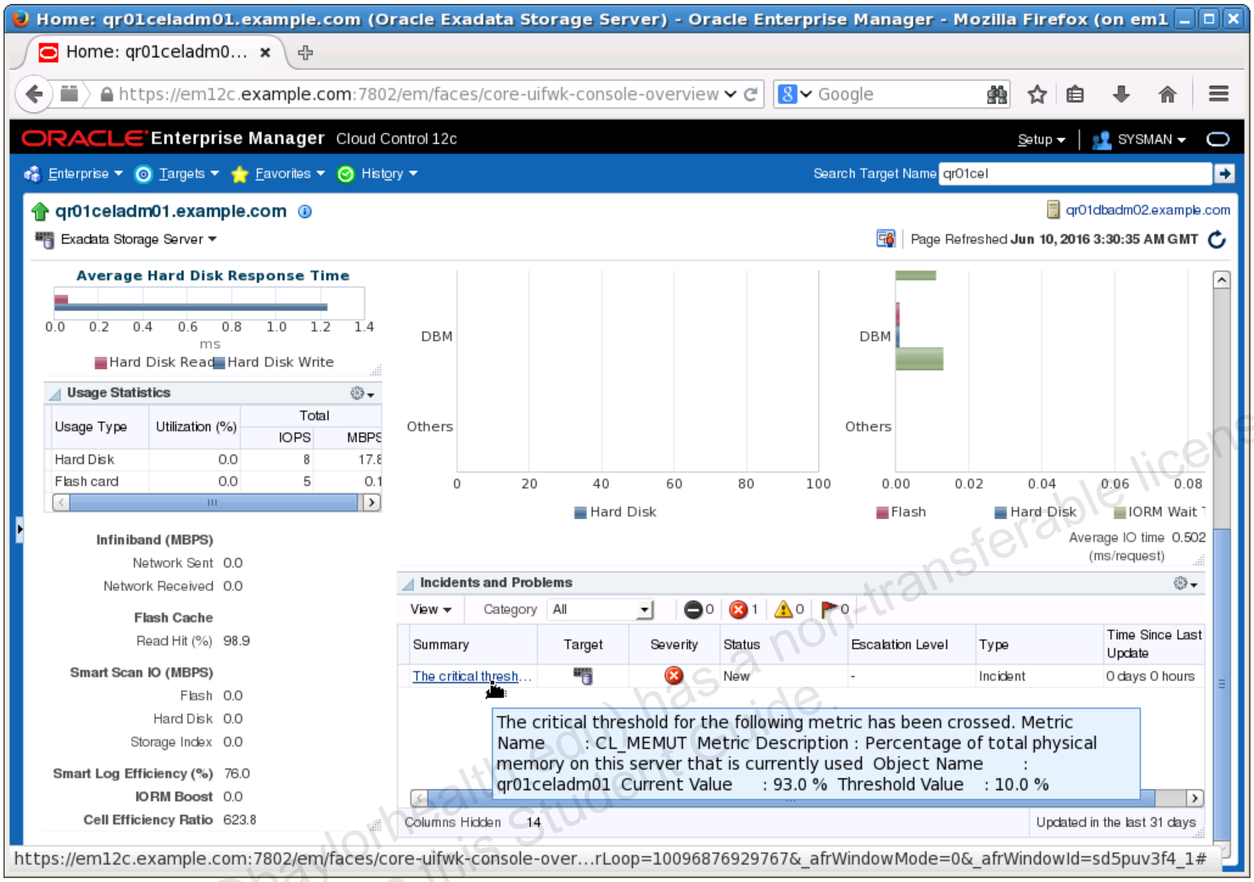
44. Back in your Enterprise Manager session, use the History menu to navigate back to the home page for `qr01celadm01.example.com`.



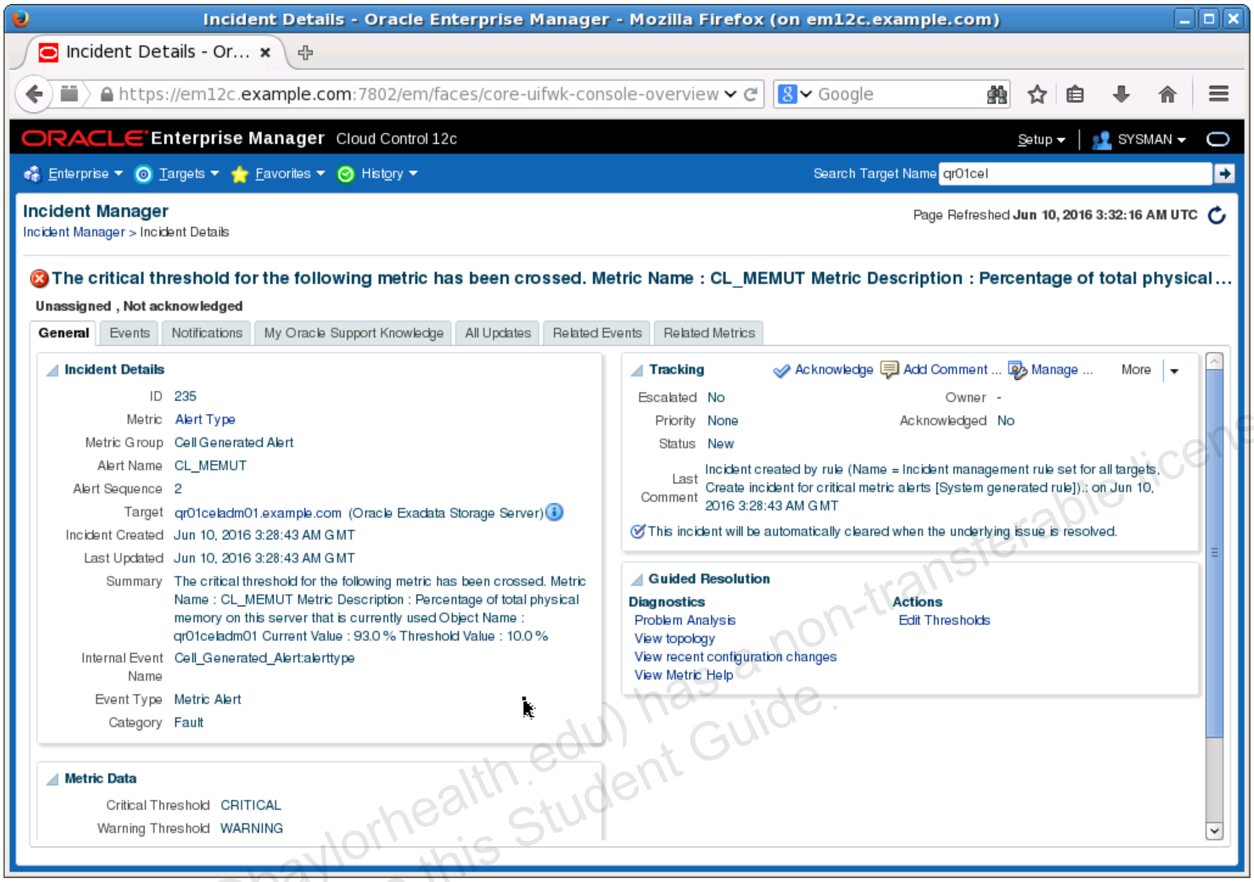
45. You should immediately notice that the Cell Health indicator is now showing a red alert icon.



46. Scroll to the bottom of the page and you should see an entry in the Incidents area. Click the Summary link to examine the incident details.



47. Examine the incident details. Note that the incident is associated with the cell-generated alert that was triggered earlier.



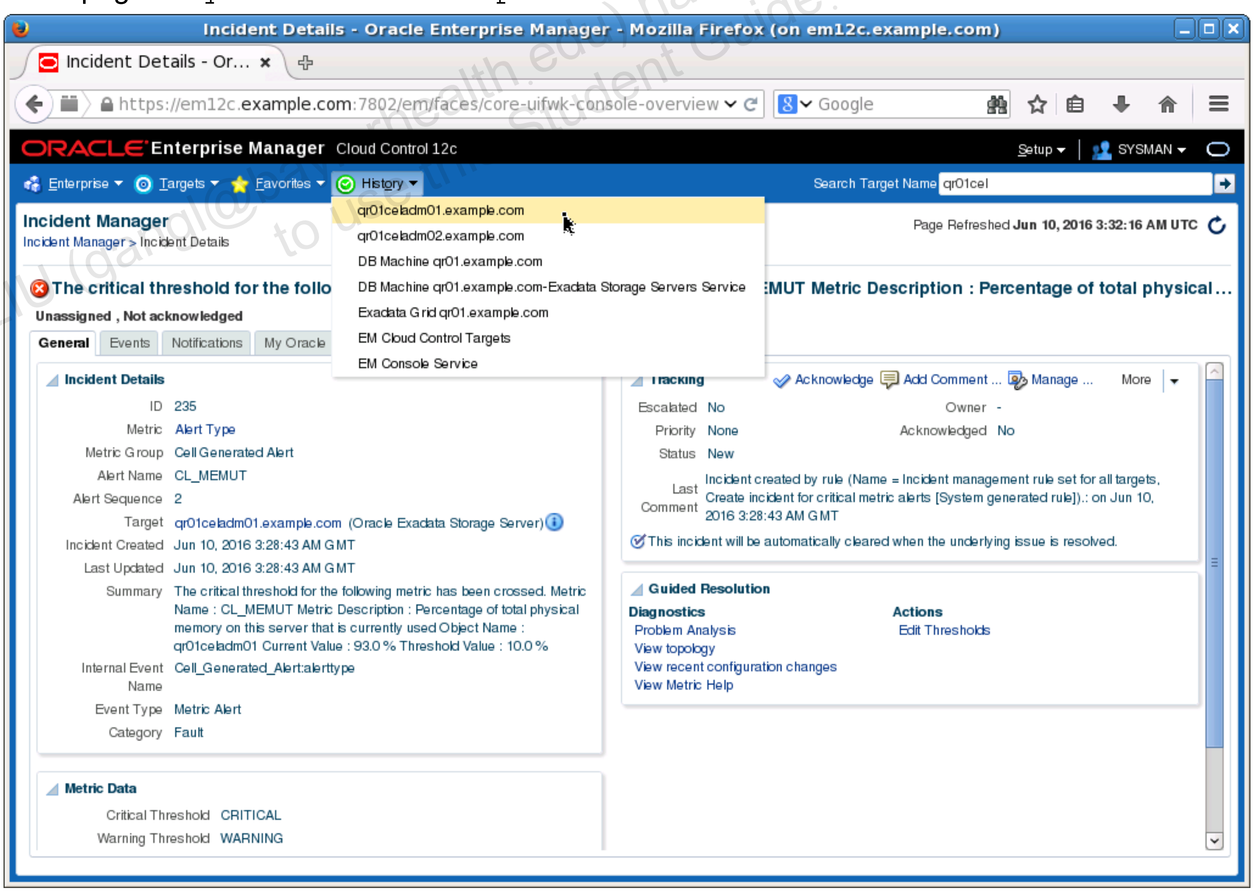
48. Return to your CellCLI session and alter the `cl_memut` threshold. Set the critical threshold to 100%. This will clear the alert associated with the `cl_memut` metric.

```
CellCLI> alter threshold cl_memut comparison='>', critical=100
Threshold cl_memut successfully altered

CellCLI>
```

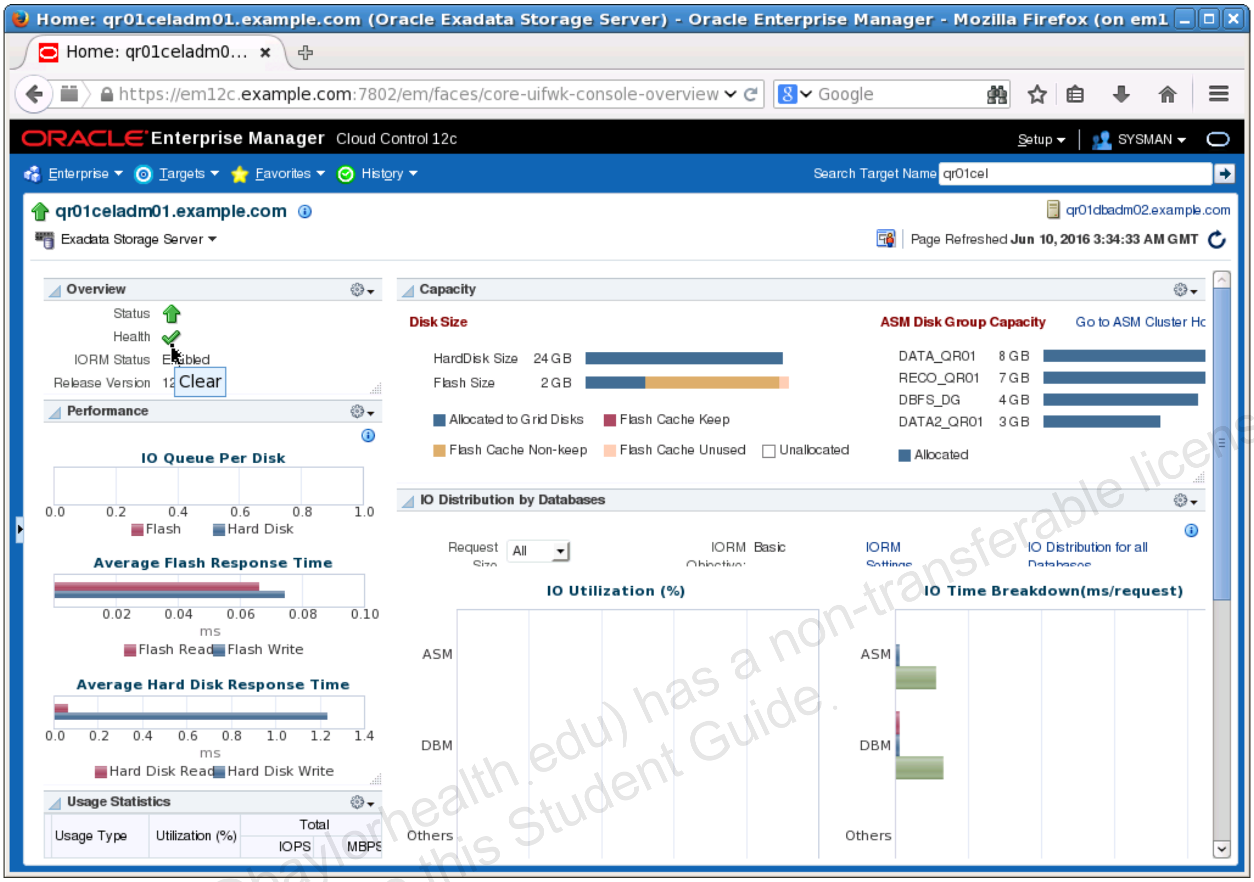
49. Execute the `LIST ALERTHISTORY` command. You should see an entry indicating that the alert has cleared. If you cannot see the expected entry, periodically re-execute the `LIST ALERTHISTORY` command until it is visible. After the alert clearance becomes visible, leave your CellCLI session running and proceed to the next step.

```
CellCLI> list alerthistory
         1_1     2016-06-10T03:23:18+00:00       critical
"The critical threshold for the following metric has been
crossed. Metric Name        : CL_MEMUT  Metric Description :
Percentage of total physical memory on this server that is
currently used  Object Name        : qr01celadm01  Current Value
: 93.0 %  Threshold Value     : 10.0 %  "
         1_2     2016-06-10T03:28:18+00:00       clear
"The critical threshold for the following metric has been
cleared. Metric Name        : CL_MEMUT  Metric Description :
Percentage of total physical memory on this server that is
currently used  Object Name        : qr01celadm01  Current Value
: 93.0 %  Threshold Value     : 100.0 %  "

CellCLI>
```

50. Back in your Enterprise Manager session, use the History menu to again navigate to the home page for `qr01celadm01.example.com`.



51. Now that the alert has cleared, you should notice that the Cell Health indicator shows a green check mark. Also, the Incidents area should be empty.



You have just seen how cell-generated alerts are propagated to Enterprise Manager and how they can appear as incidents in Enterprise Manager. Normally, this process is very reliable; however, various problems could disrupt the process. Because of this, it is recommended that administrators who primarily use Enterprise Manager should also periodically check the cell alert history for undelivered alerts.

52. If an Exadata Storage Server cannot deliver an alert using any of the configured delivery channels, the `notificationState` for the alert will be set to `1`. Back in your CellCLI session, use the following command to list any undelivered and previously unexamined alerts on this cell (`qr01celadm01`).

```
CellCLI> list alerthistory where notificationState != 1 and
examinedBy = ''

CellCLI>
```

53. Exit CellCLI.

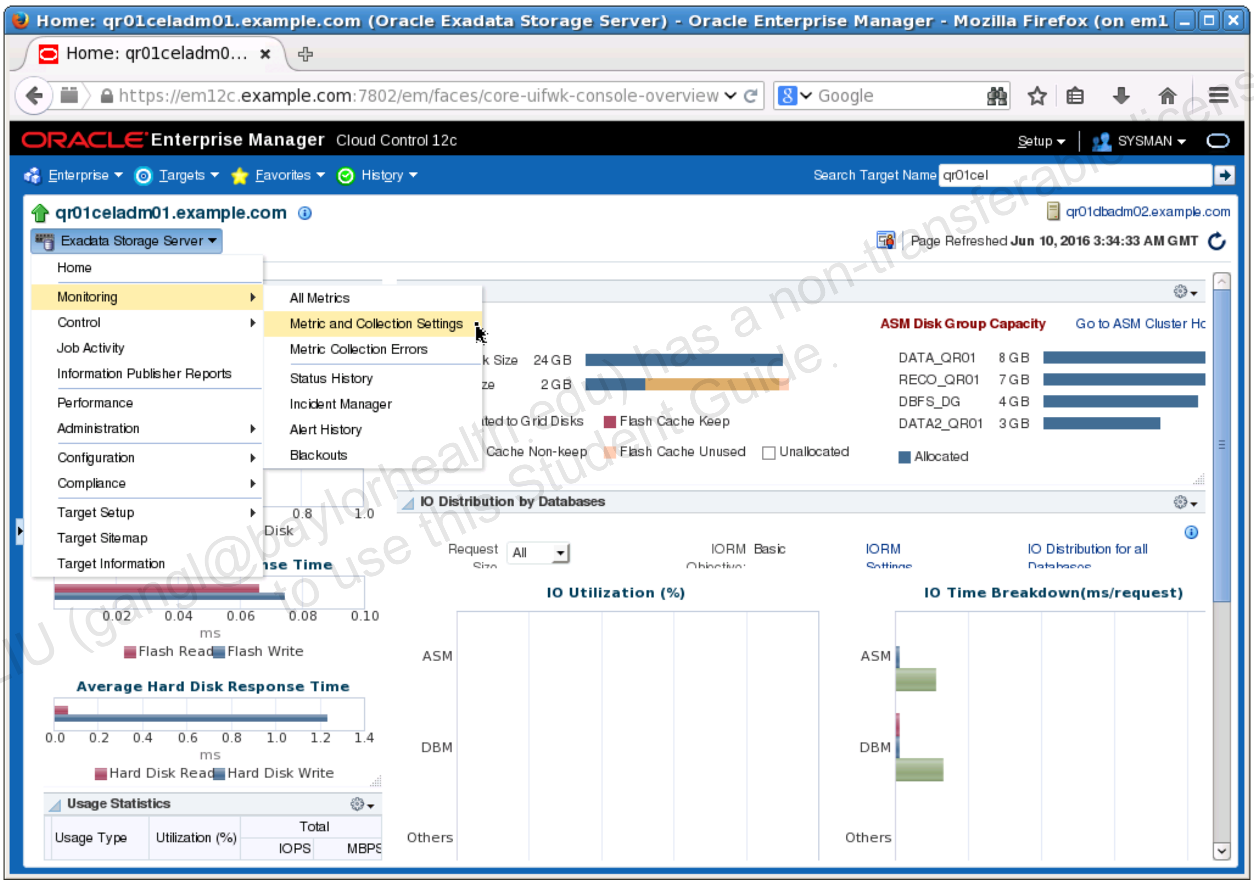
```
CellCLI> exit
quitting

[celladmin@qr01celadm01 ~]$
```

54. Use the following command to check for undelivered and previously unexamined alerts across all your Exadata Storage Servers. Normally, when everything is working correctly, you should not see any output from this command.

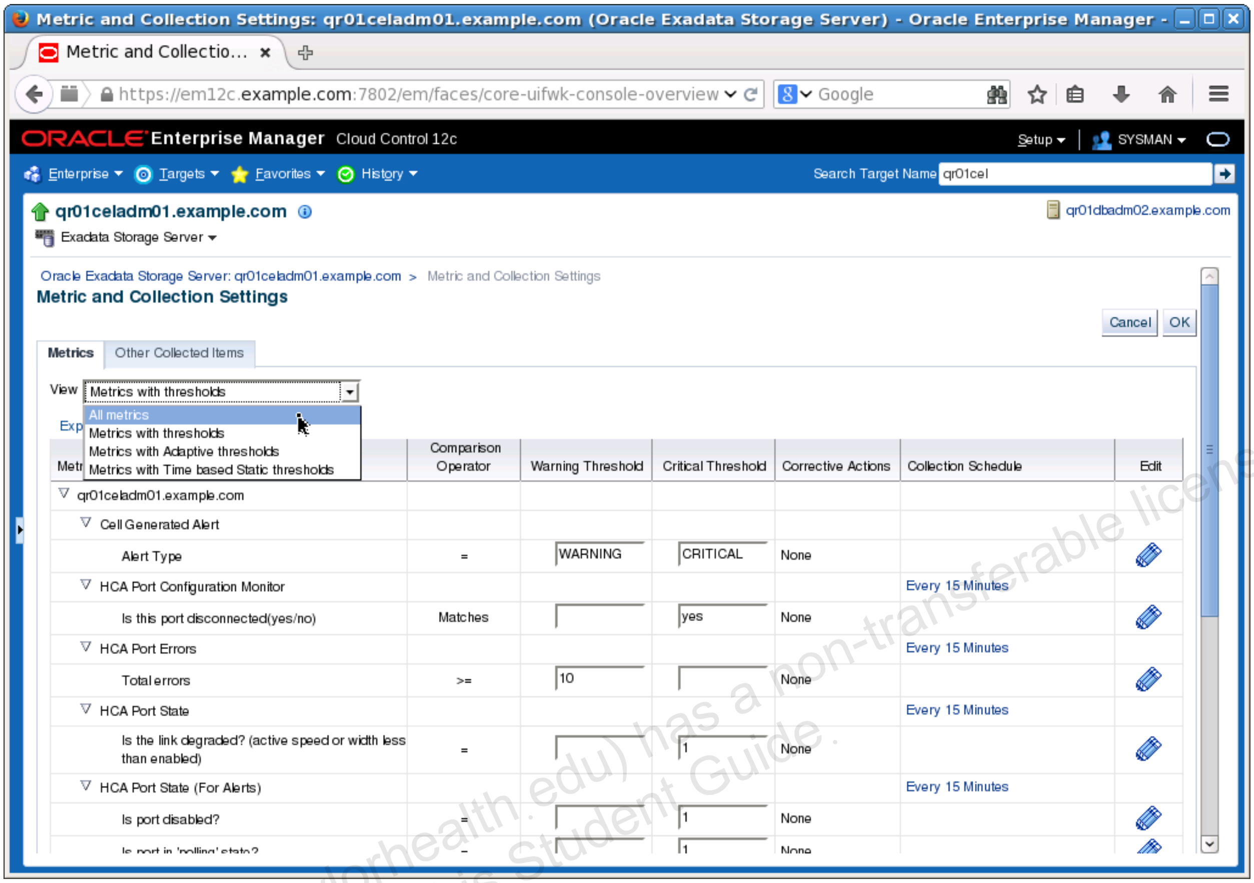
```
[celladmin@qr01celadm01 ~]$ dcli -c
qr01celadm01,qr01celadm02,qr01celadm03 cellcli -e "list
alerthistory where notificationState != 1 and examinedBy = \'\'"
[celladmin@qr01celadm01 ~]$
```

In the next part of this practice, you will configure a threshold in Enterprise Manager and watch as an Incident is generated by using the threshold.

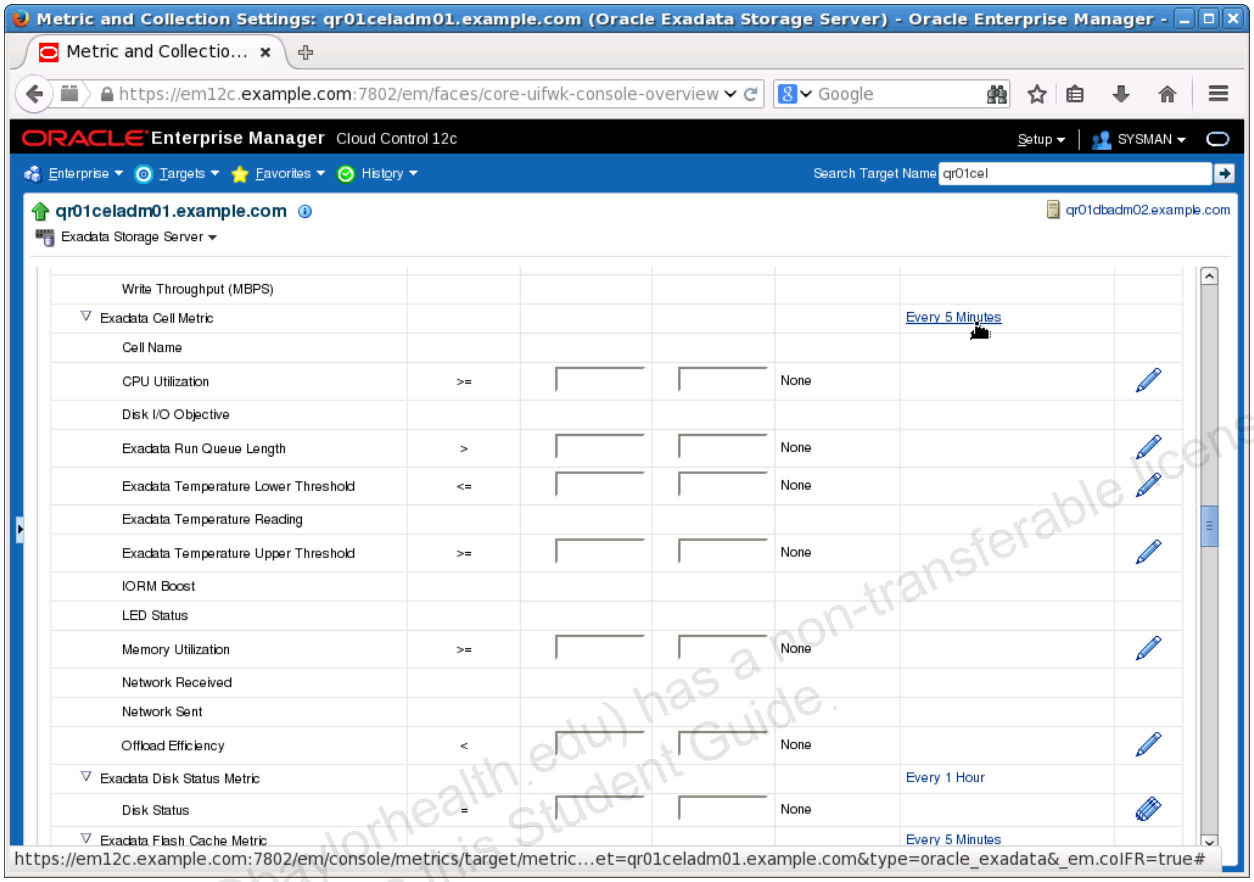
55. Back in your Enterprise Manager session, select the Exadata Storage Server > Monitoring > "Metric and Collection Settings" menu command.



56. Select the option to view all metrics.

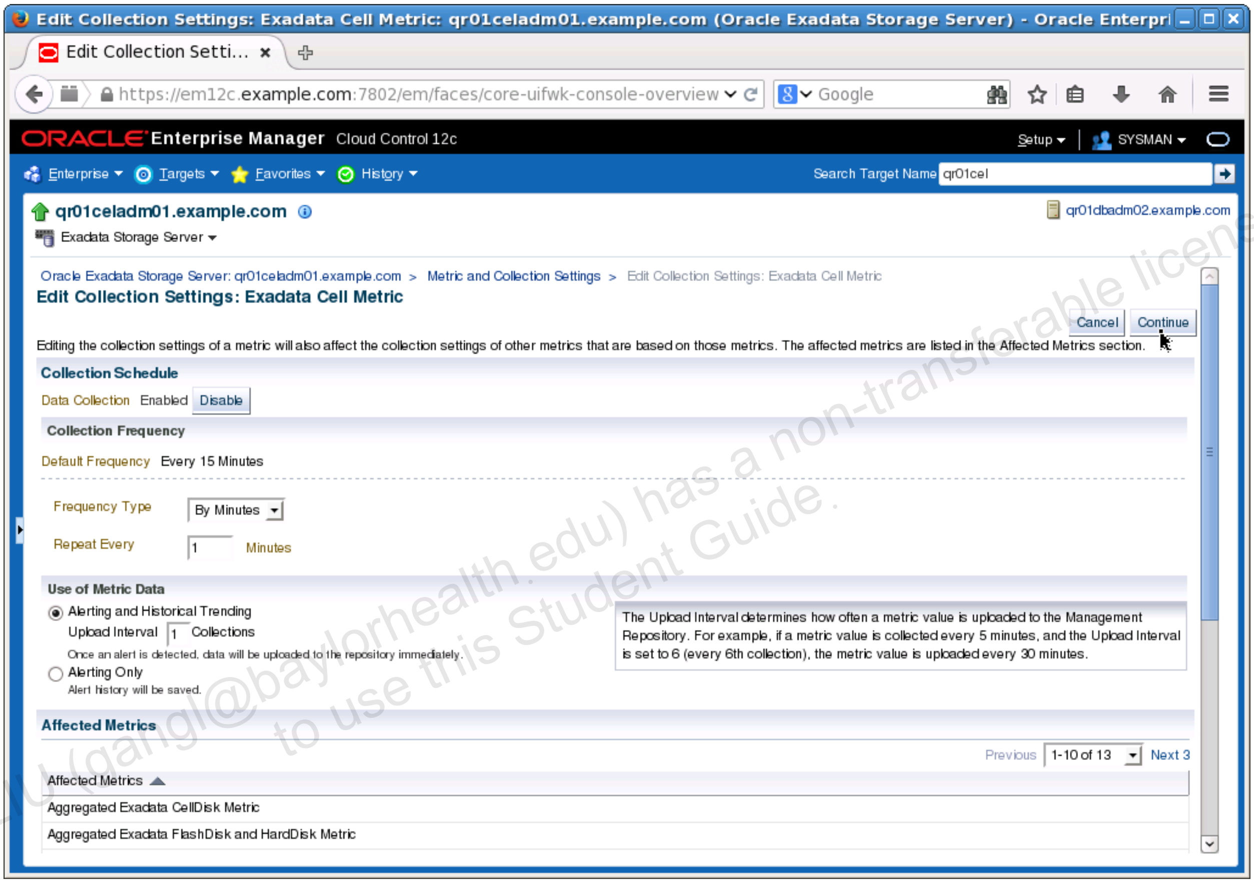


57. Scroll down to the group of Exadata cell metrics. Click Every 5 Minutes to adjust the collection schedule for the Exadata Cell Metric group of metrics.

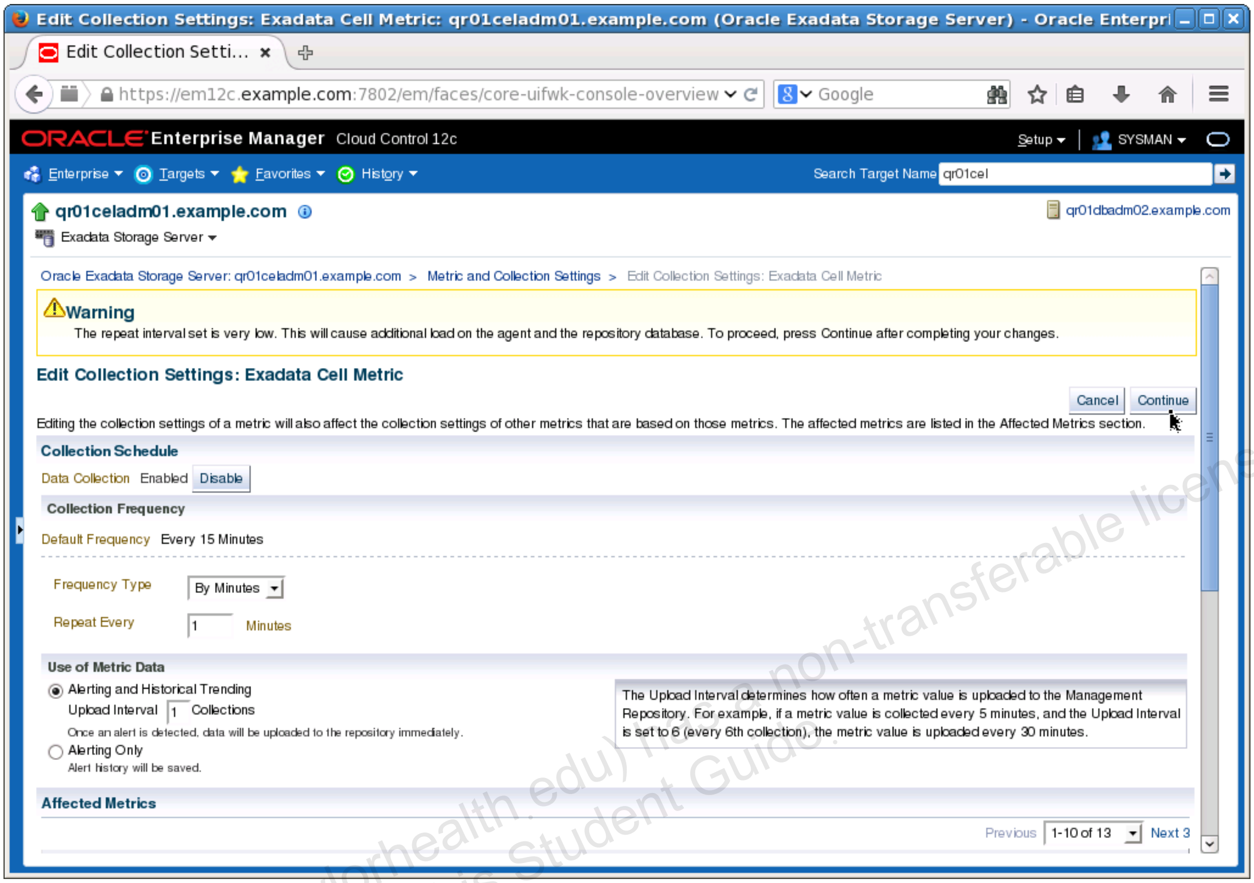


58. Make the following adjustments to the collection schedule:
   - Repeat Every `1` Minutes
   - Upload Interval `1` Collections

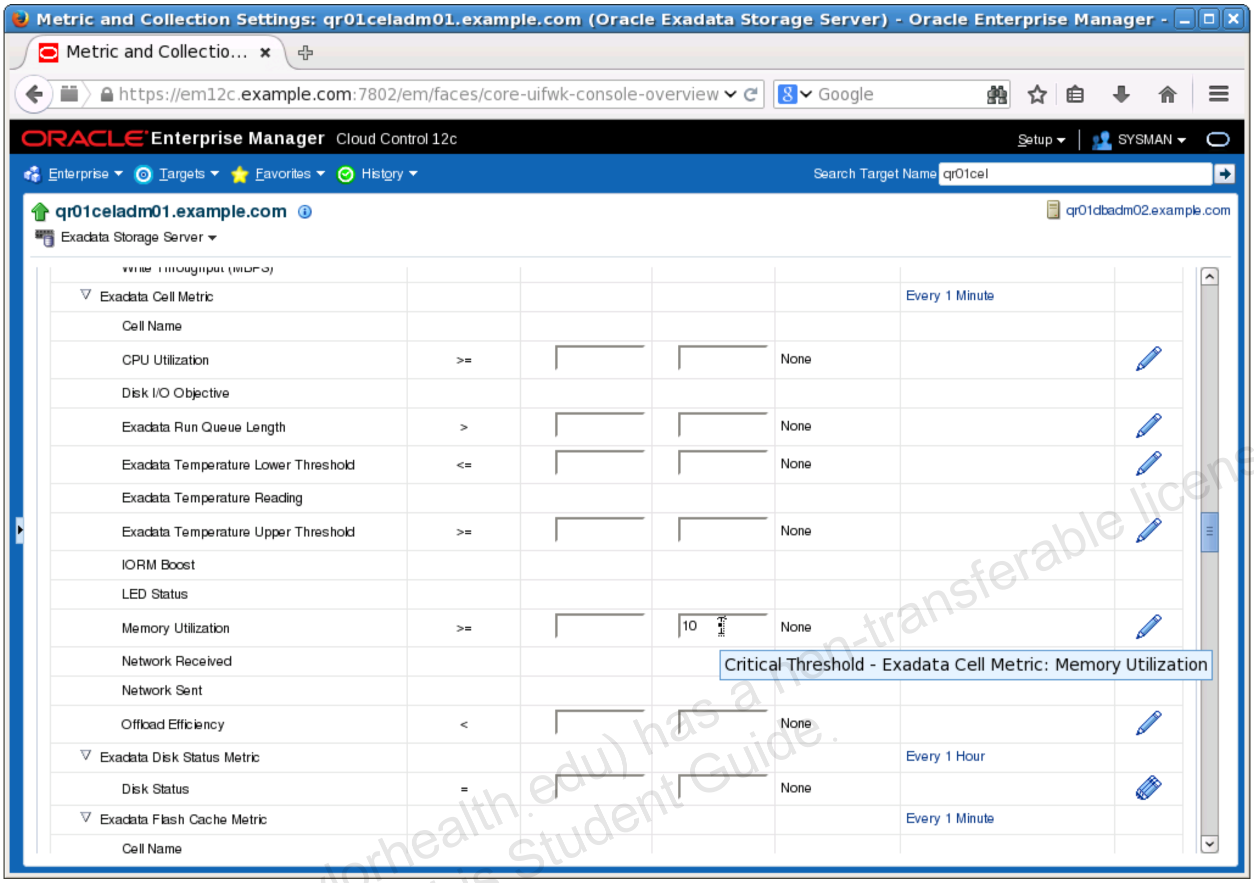
These settings will cause the metrics to be collected and uploaded every minute. Normally this would not be a recommended setting; however, you should use them for this practice to avoid waiting up to 5 minutes for the incident to be generated. Finally, click Continue to proceed.



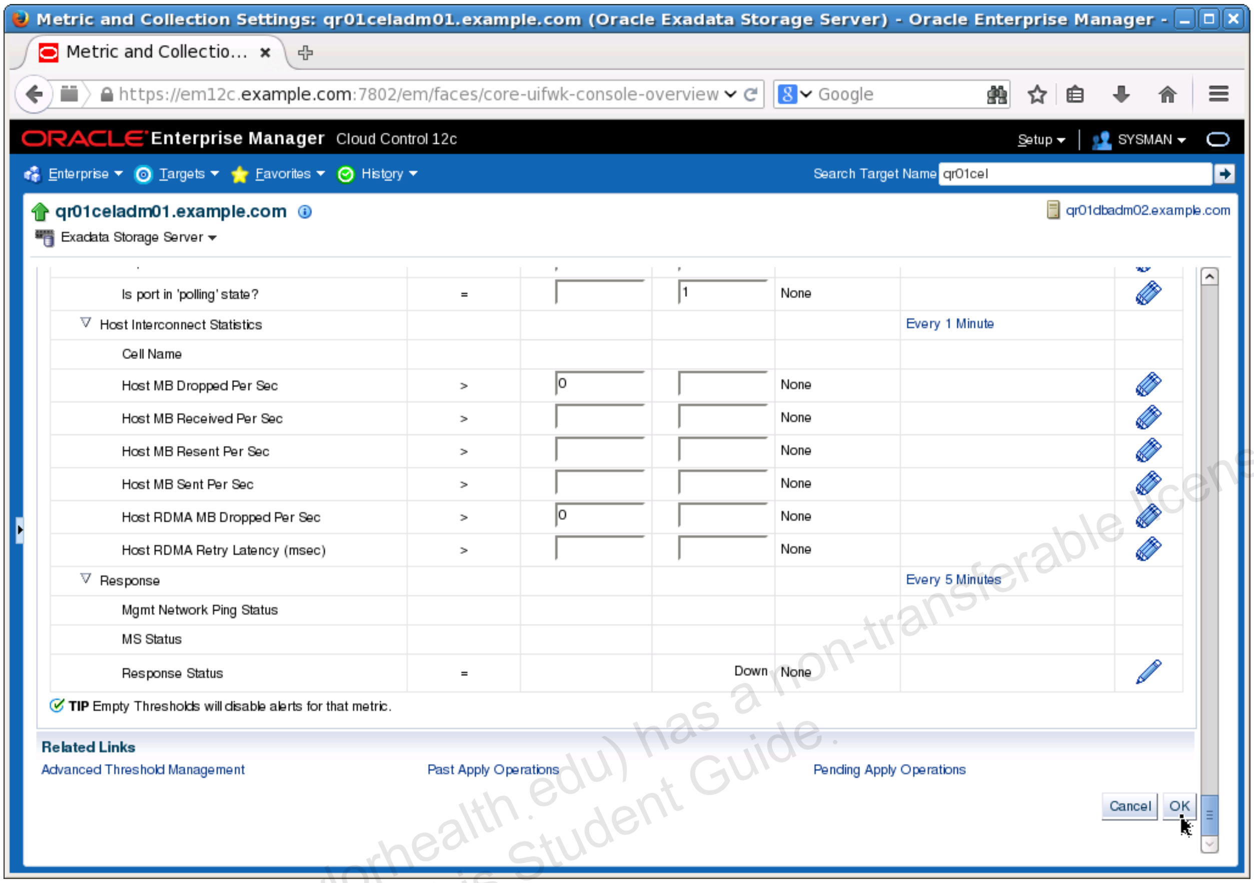
59. Click Continue to acknowledge the warning message and proceed.



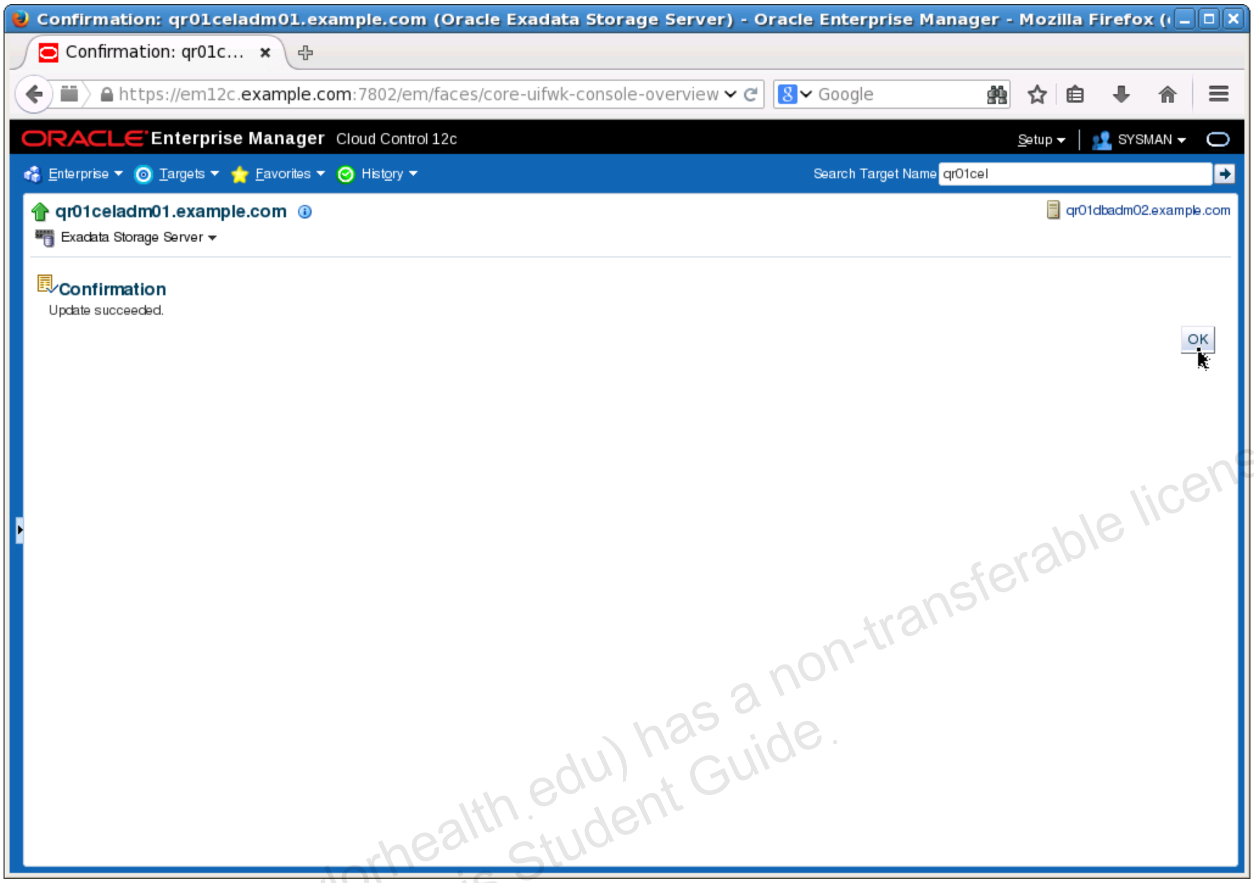
60. Scroll down the page and set the Critical Threshold for the Memory Utilization metric to `10`.



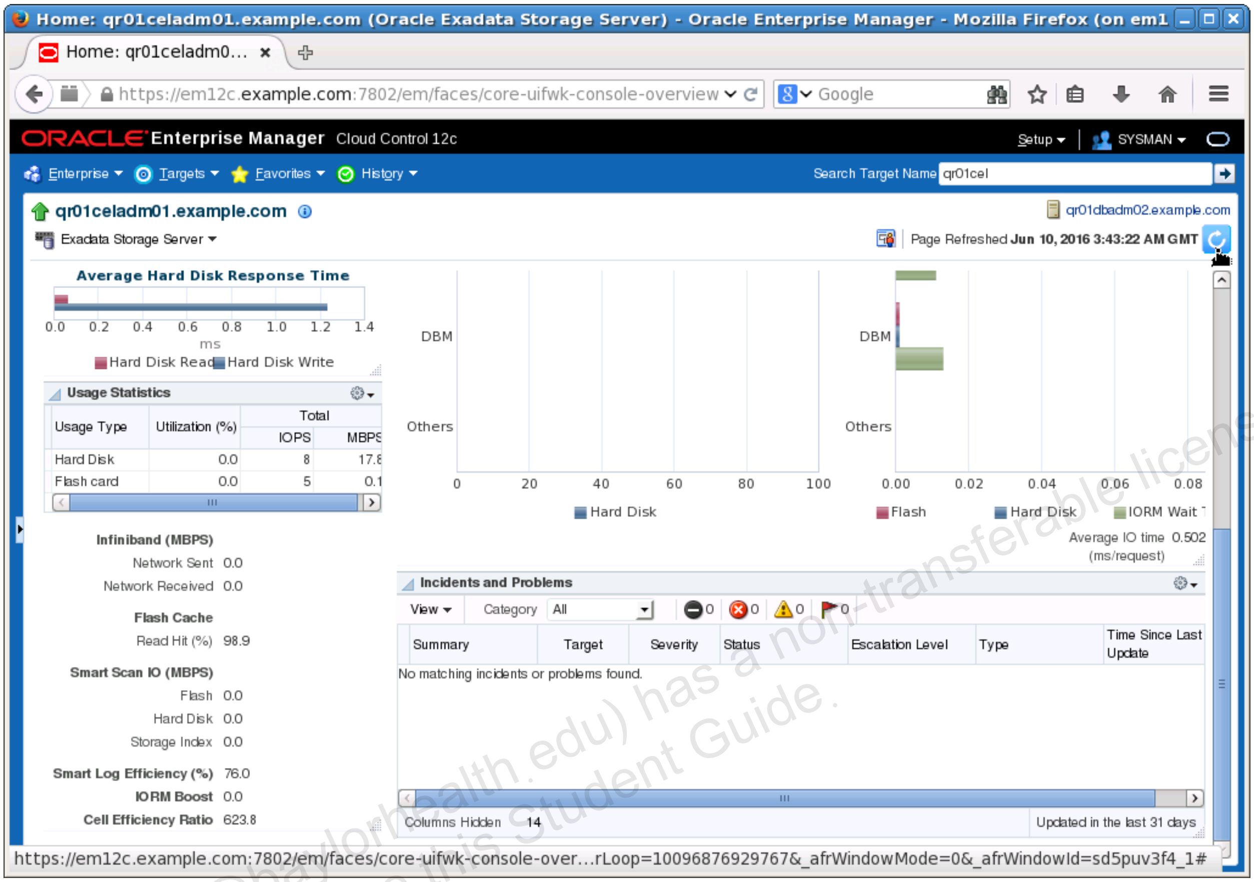
61. Scroll to the bottom of the page and click OK to save the metric and collection settings.



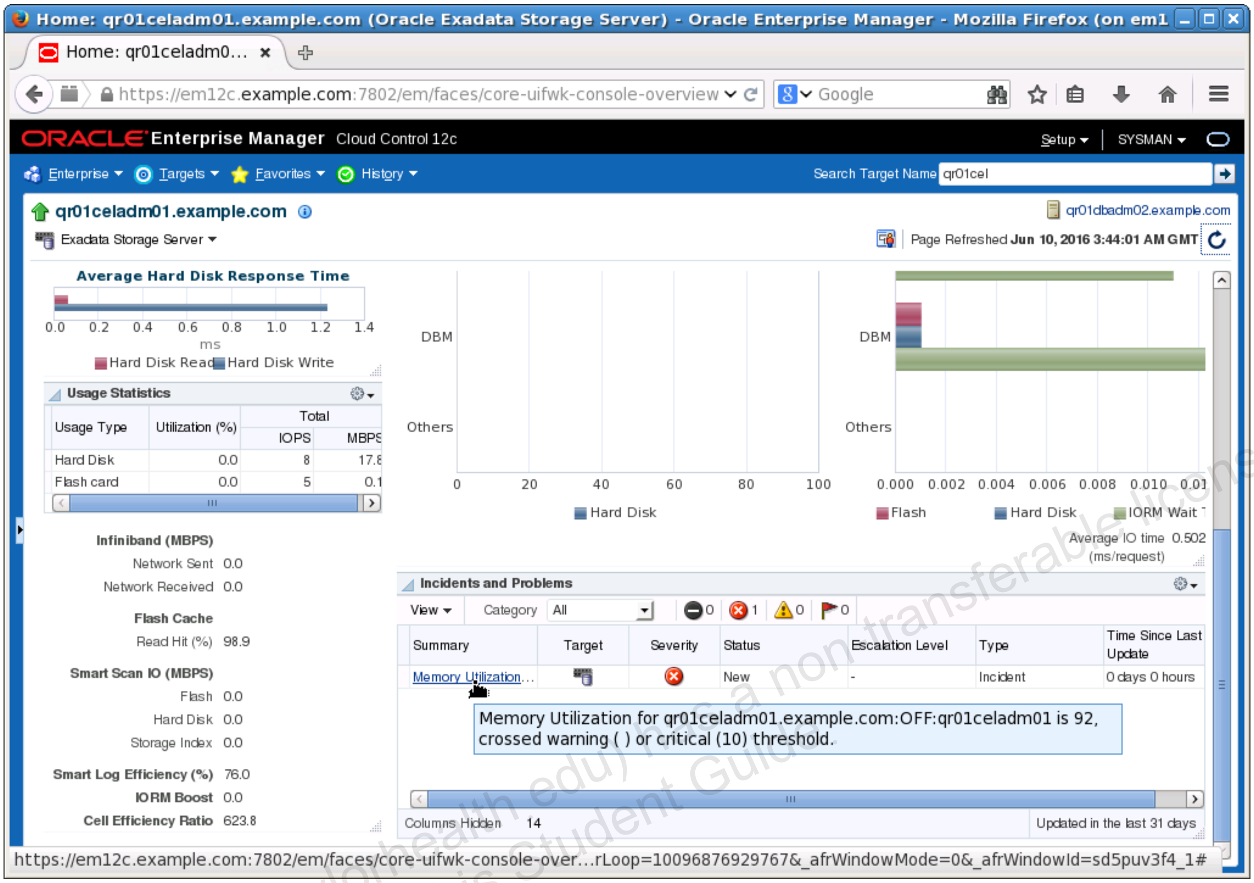
62. Click OK to acknowledge the update confirmation.



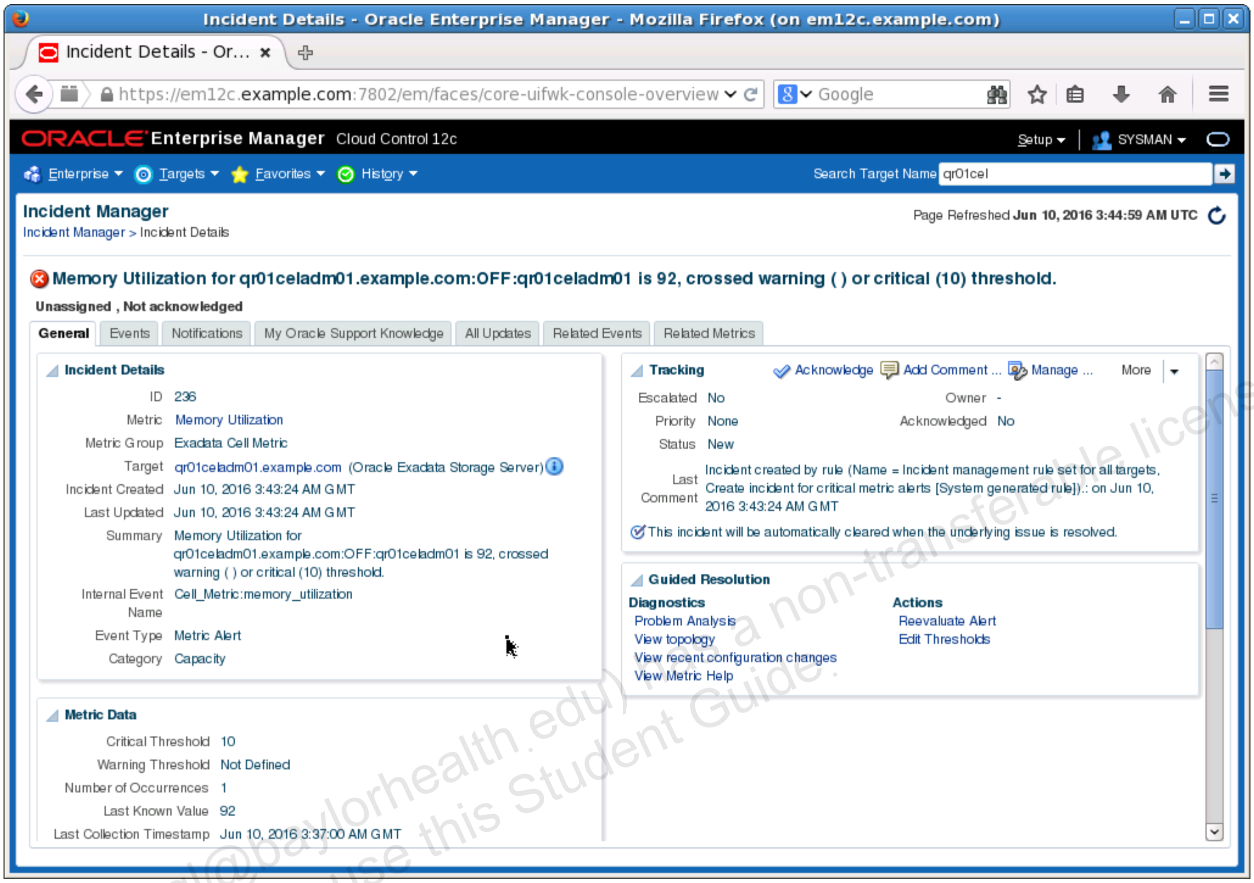
63. Examine the Incidents area on the Exadata Storage Server home page for `qr01celadm01`. Periodically, refresh the page until an incident appears.



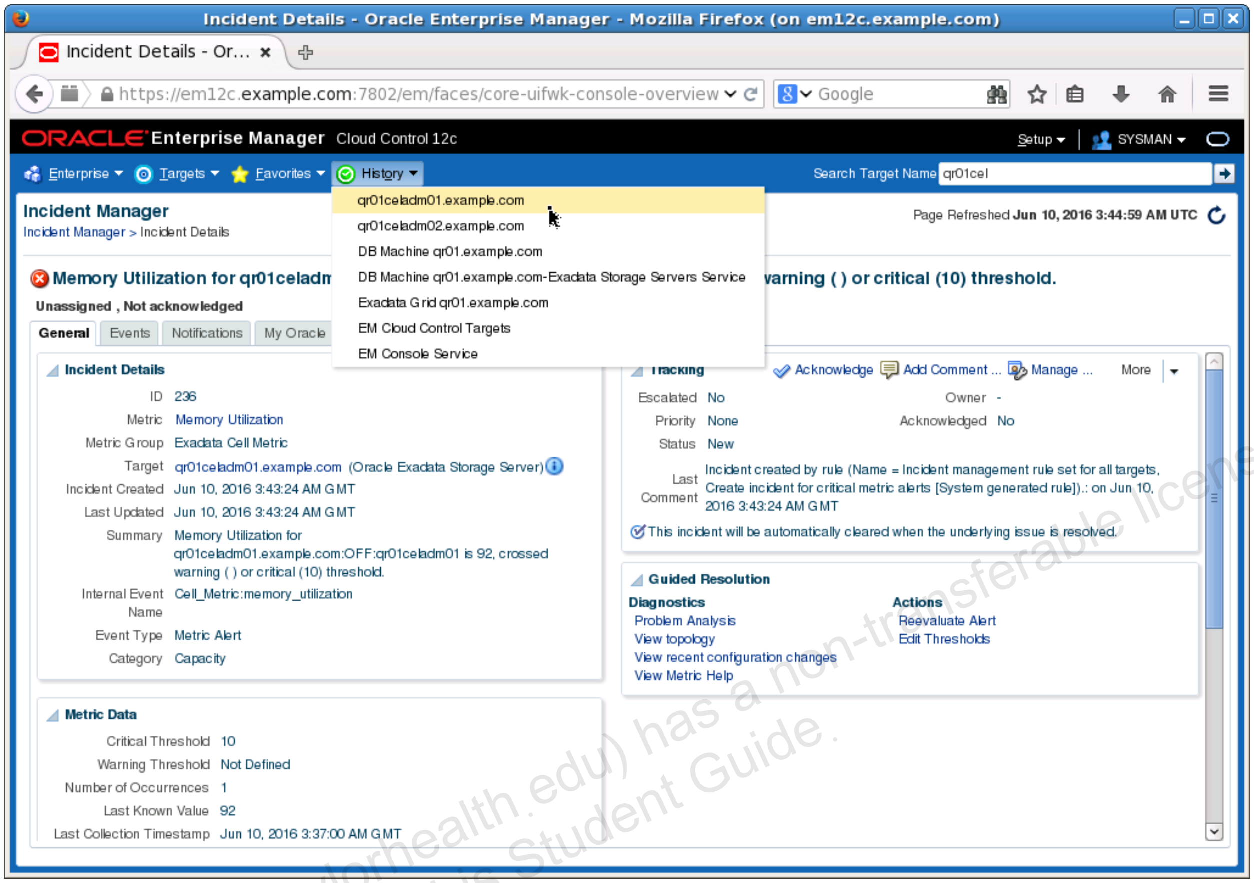
64. Click the Summary link to examine the incident details.



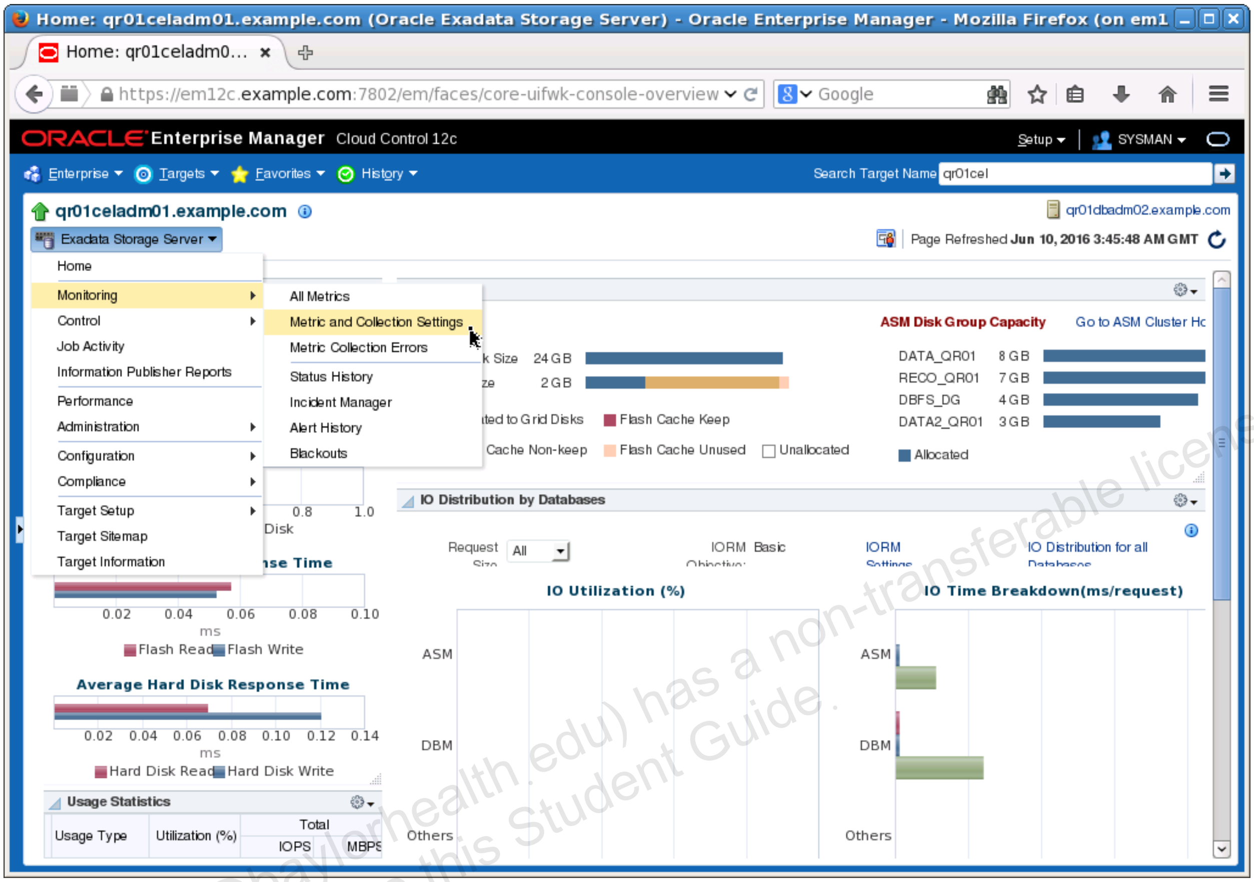
65. Examine the incident details. You should observe that the incident is associated with the metric threshold you just created in Enterprise Manager and that it is very similar to the incident associated with the cell-generated alert that you saw earlier in the practice.



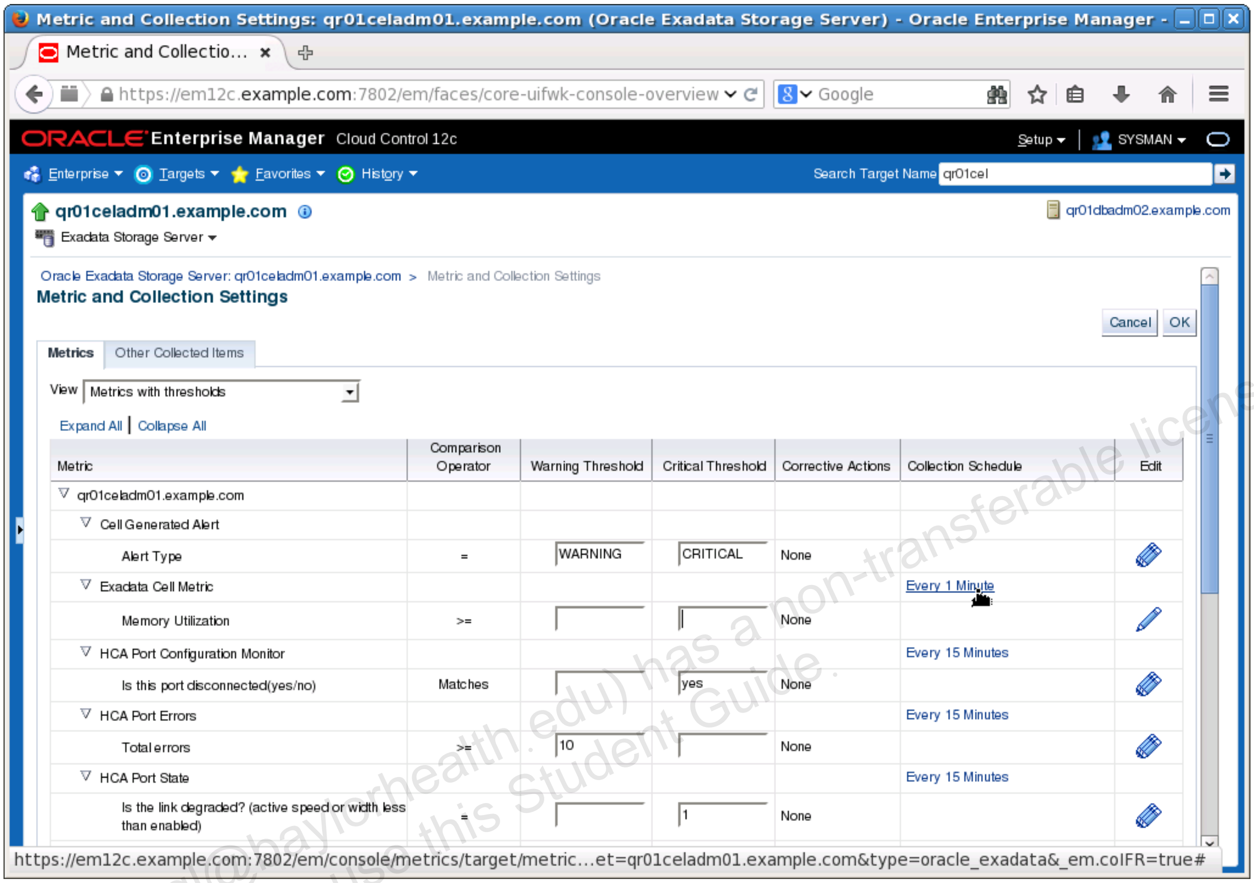
66. Use the History menu to return to the home page for `qr01celadm01.example.com`.



67. Select the Exadata Storage Server > Monitoring > "Metric and Collection Settings" menu command.

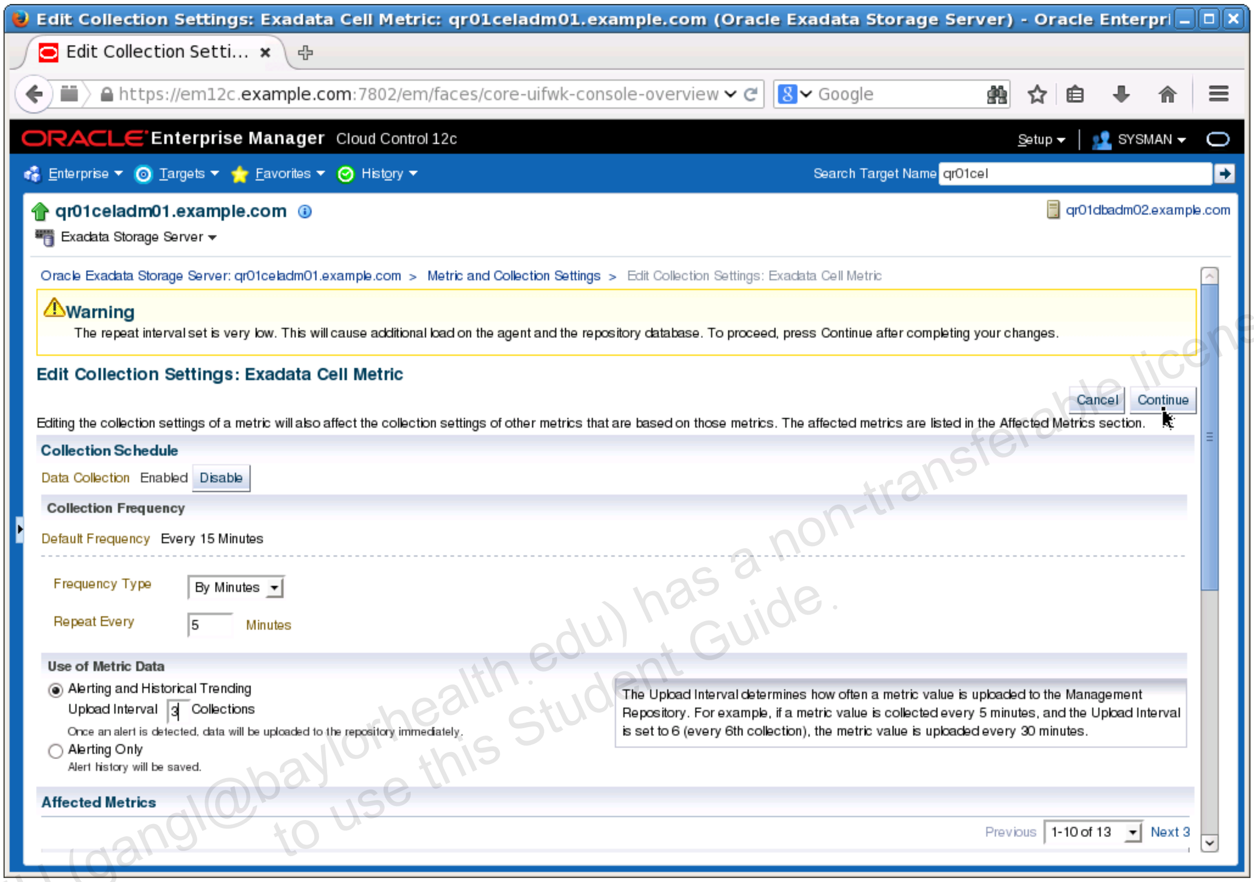


68. Clear the Critical Threshold value associated with the Memory Utilization metric. Then click Every 1 Minute to adjust the collection schedule for the Exadata Cell Metric group of metrics.

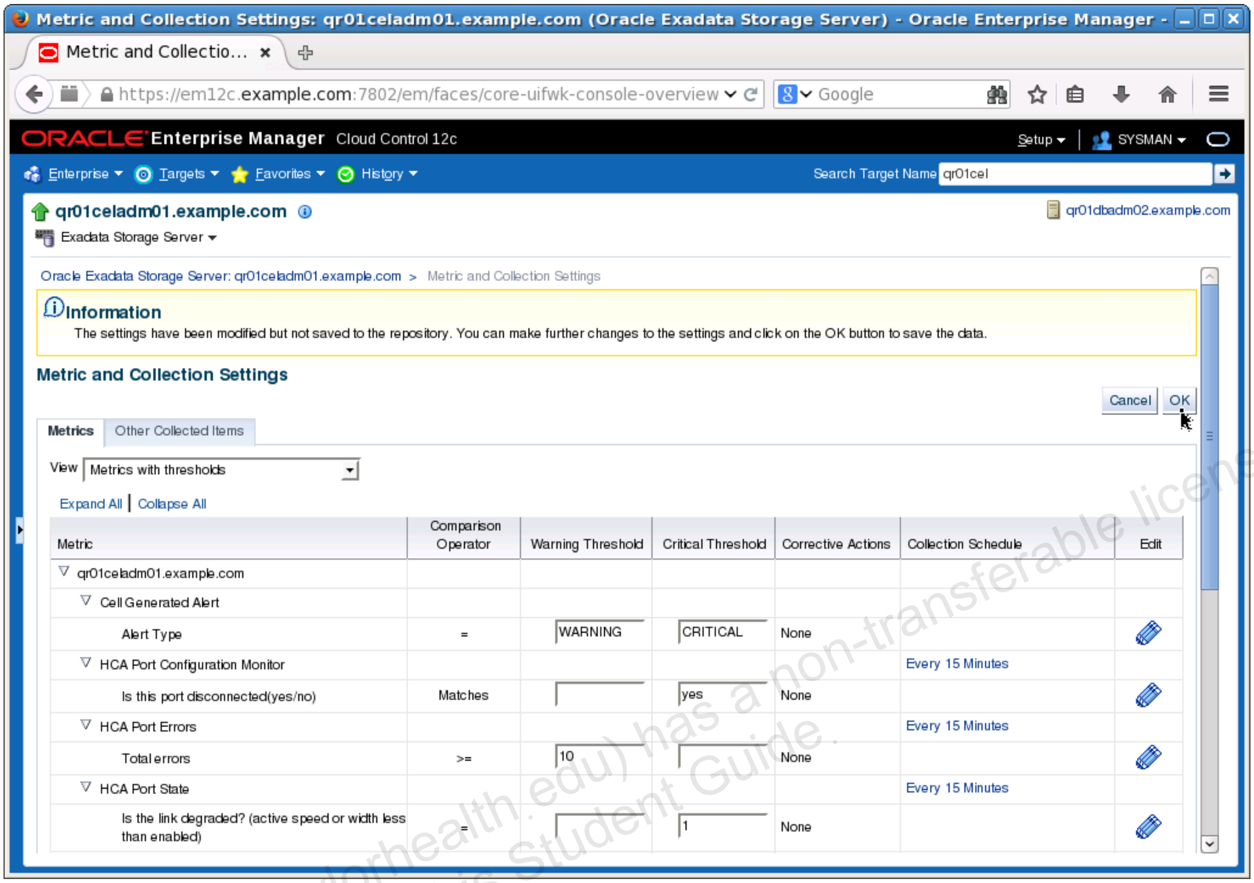


69. Make the following adjustments to revert the collection schedule to the default settings:
   - Repeat Every 5 Minutes
   - Upload Interval 3 Collections

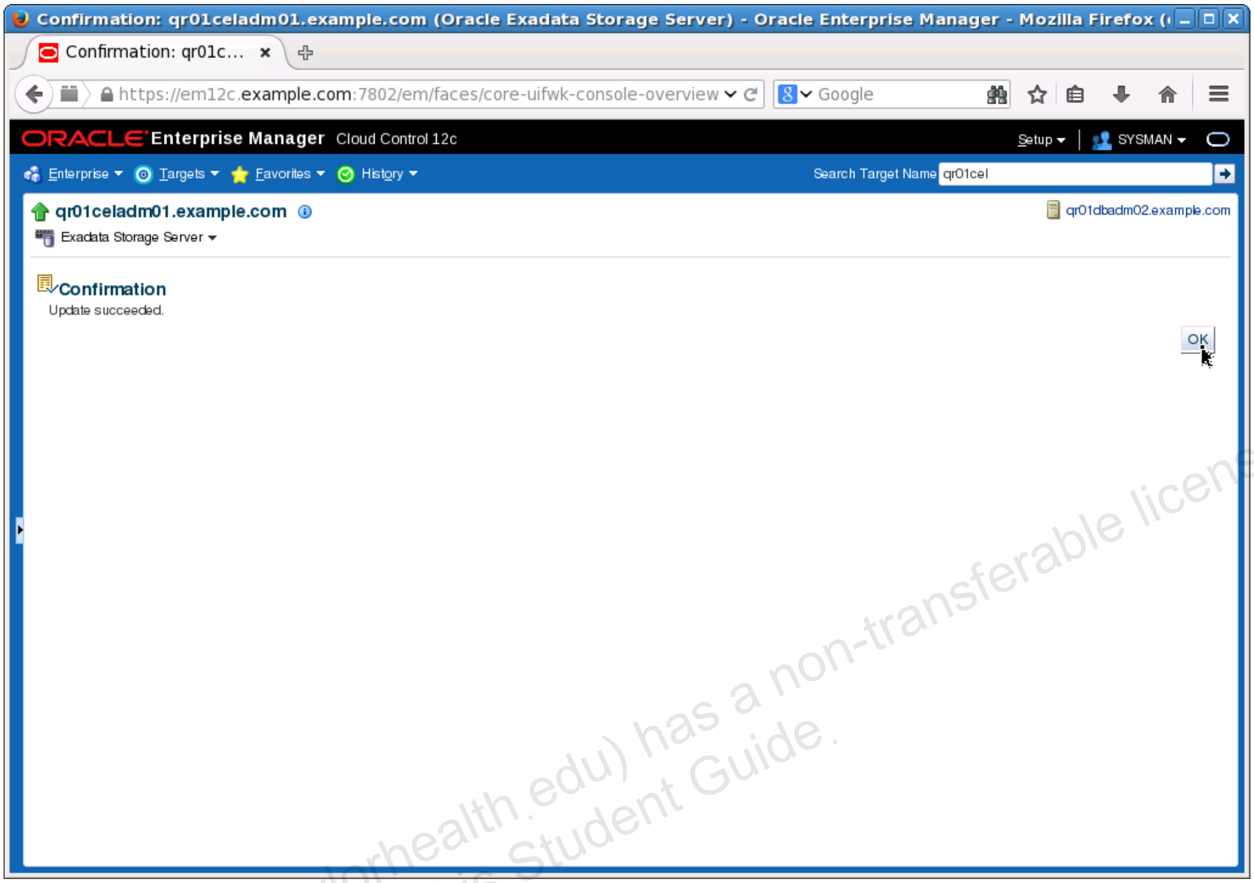
   Then, click Continue to proceed.



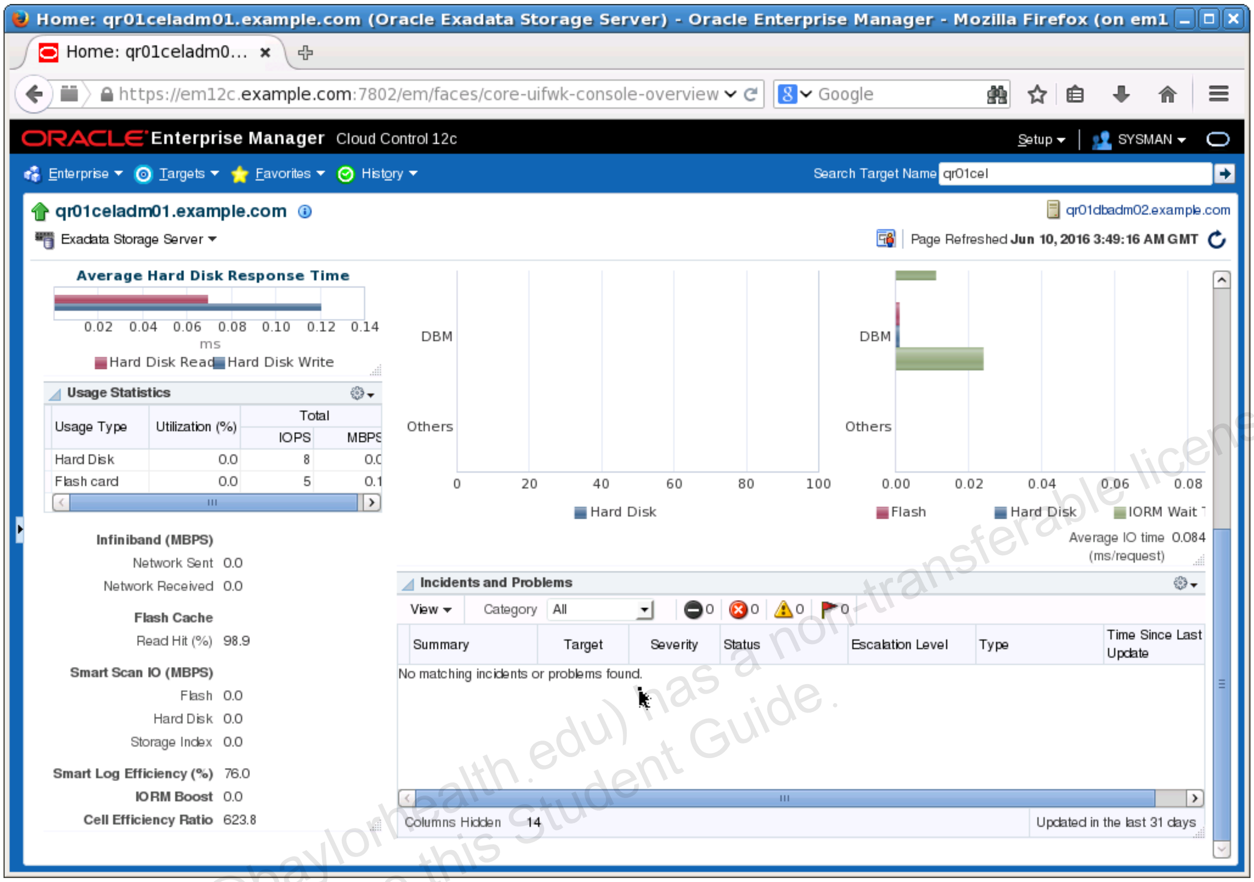
70. Click OK to save the threshold and collection schedule settings.



71. Click OK to acknowledge the update confirmation.



72. When you return to the Exadata Storage Server home page for `qr01celadm01`, you should see that the memory utilization incident is no longer visible.



You have now seen cell-based and Enterprise Manager–based metrics, thresholds, alerts, and incidents in action. In the next part of this practice, you will exercise the configuration management capabilities in Enterprise Manager and you will see how they can be used to ensure that all of your Exadata Storage Servers have consistent configurations.

73. In the `celladmin` terminal session, which you started earlier in the practice, launch the Exadata cell command-line interface (CellCLI).

```
[celladmin@qr01celadm01 ~]$ cellcli
CellCLI: Release 12.1.2.1.0 - Production...

CellCLI>
```

74. Examine the IORM plan on `qr01celadm01`. Notice that by default, the IORM objective is set to `basic`.

```
CellCLI> list iormplan detail
         name:                    qr01celadm01_IORMPLAN
         catPlan:
         dbPlan:
         objective:               basic
         status:                  active

CellCLI>
```

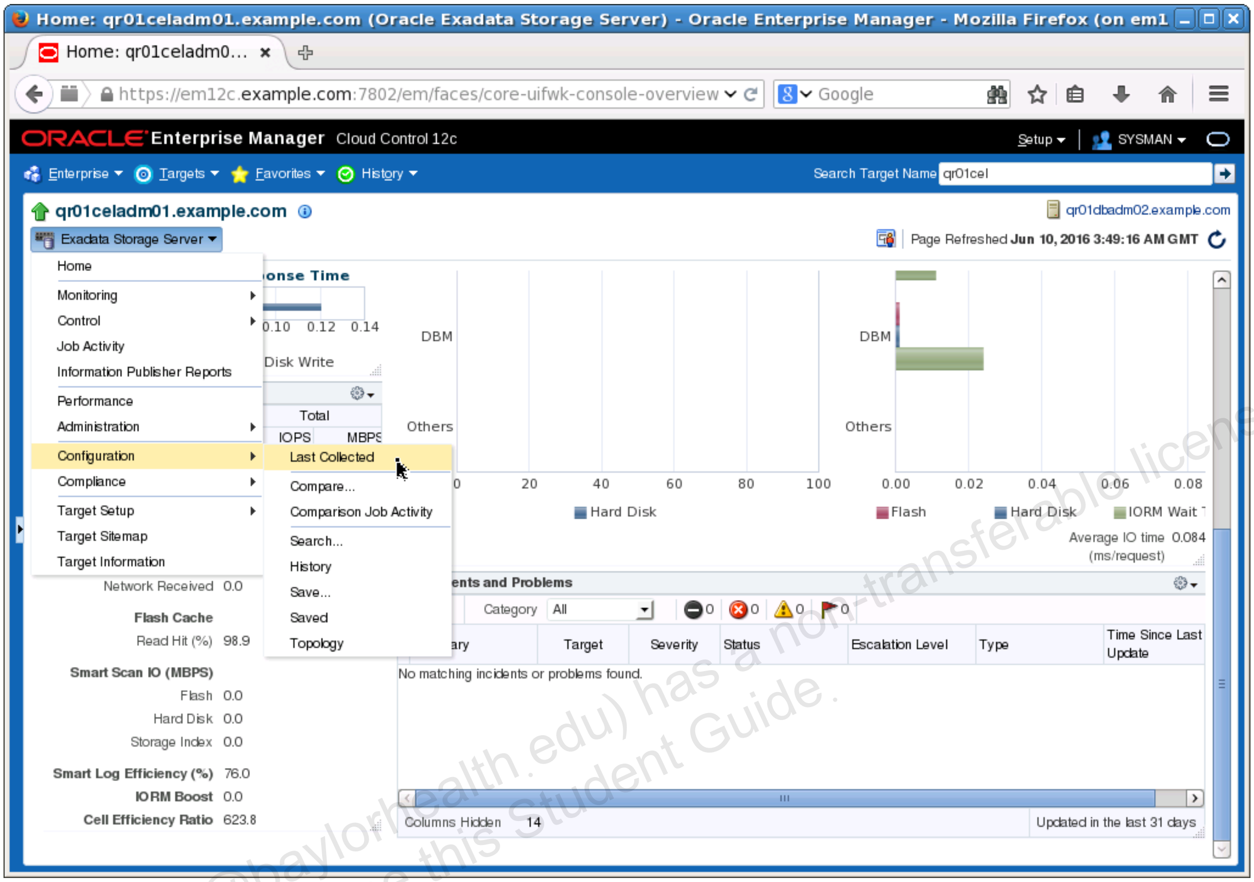
75. Alter the IORM plan and set the IORM object to `balanced`.

```
CellCLI> alter iormplan objective=balanced
IORMPLAN successfully altered

CellCLI>
```

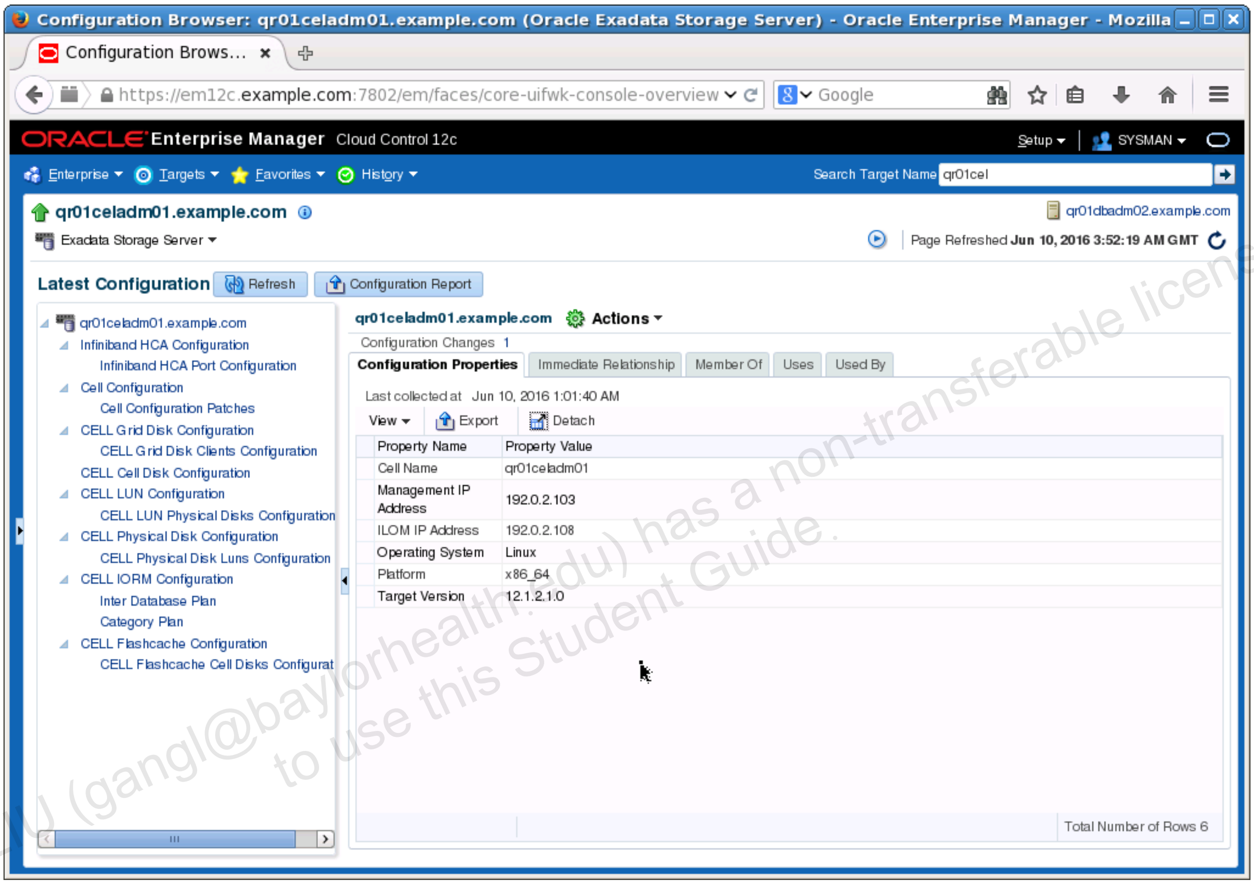
76. Verify that the IORM objective is set to `balanced` on `qr01celadm01`.

```
CellCLI> list iormplan detail
         name:                    qr01celadm01_IORMPLAN
         catPlan:
         dbPlan:
         objective:               balanced
         status:                  active

CellCLI>
```

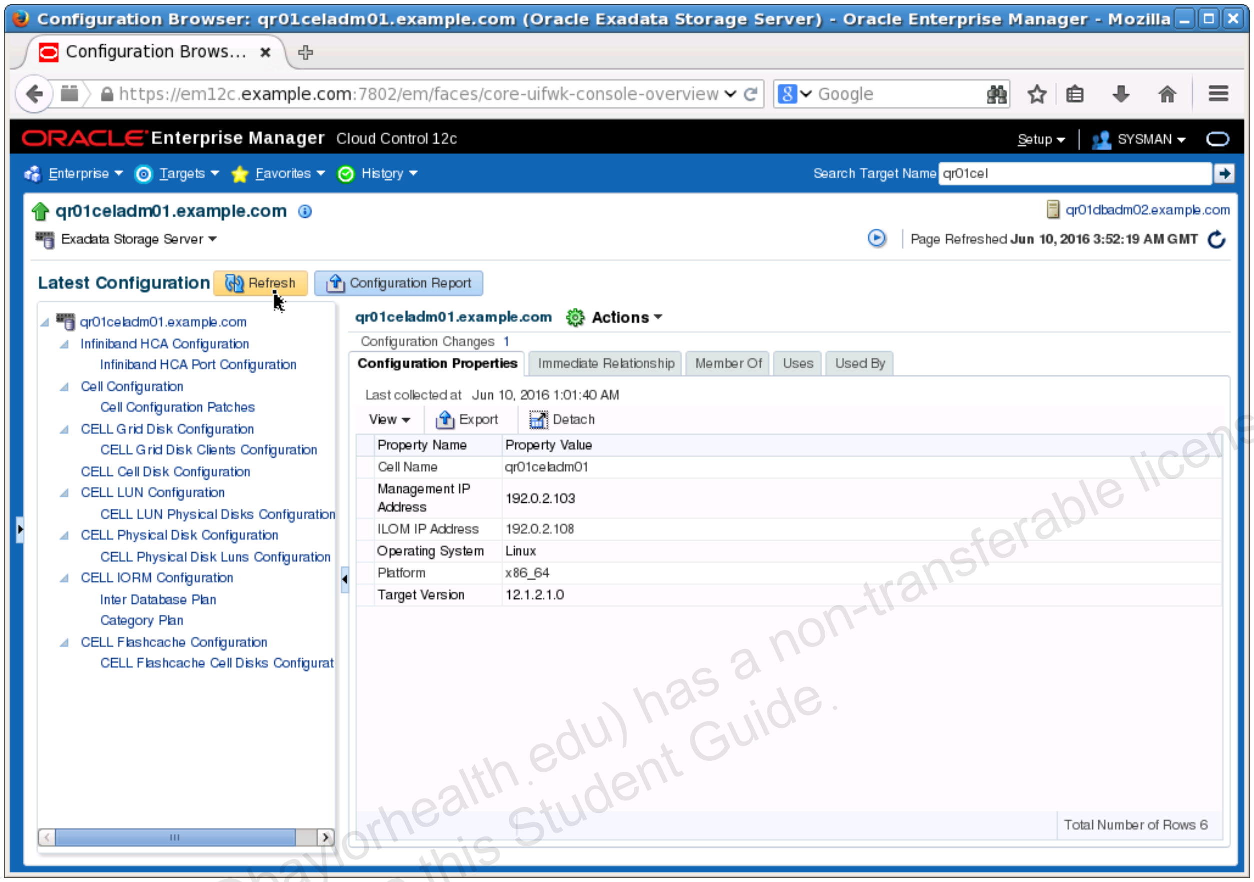
77. Back in your Enterprise Manager session, select the Exadata Storage Server > Configuration > Last Collected menu command.



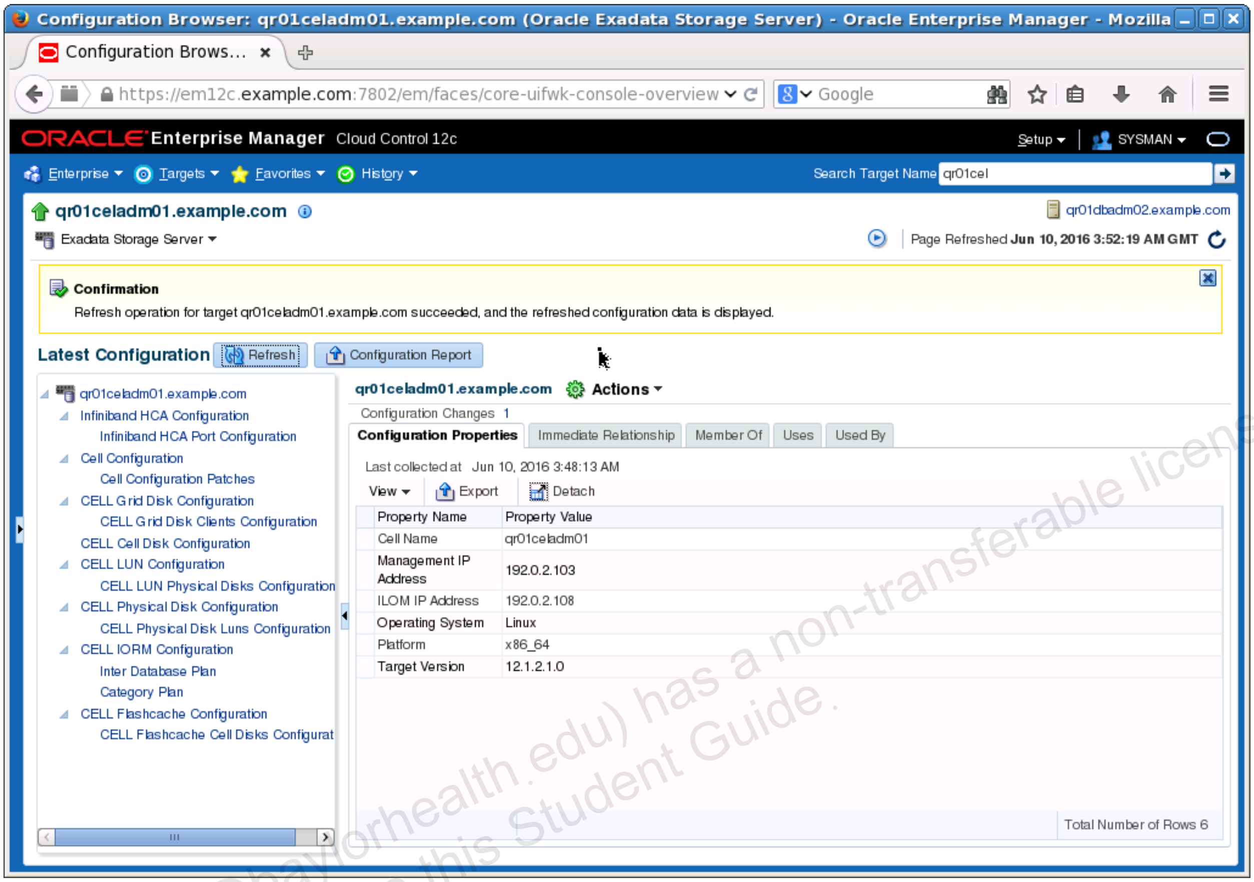
78. You should now see the configuration browser page for `qr01celadm01.example.com`. The configuration browser allows administrators to examine the configuration settings associated with a management target. It is similar to the All Metrics page that you examined earlier. The main difference between metrics and configuration settings is that metric values are expected to constantly vary over time while configuration settings should be more stable. You can use the hierarchical list on the left side of the page to examine different groups of configuration settings.



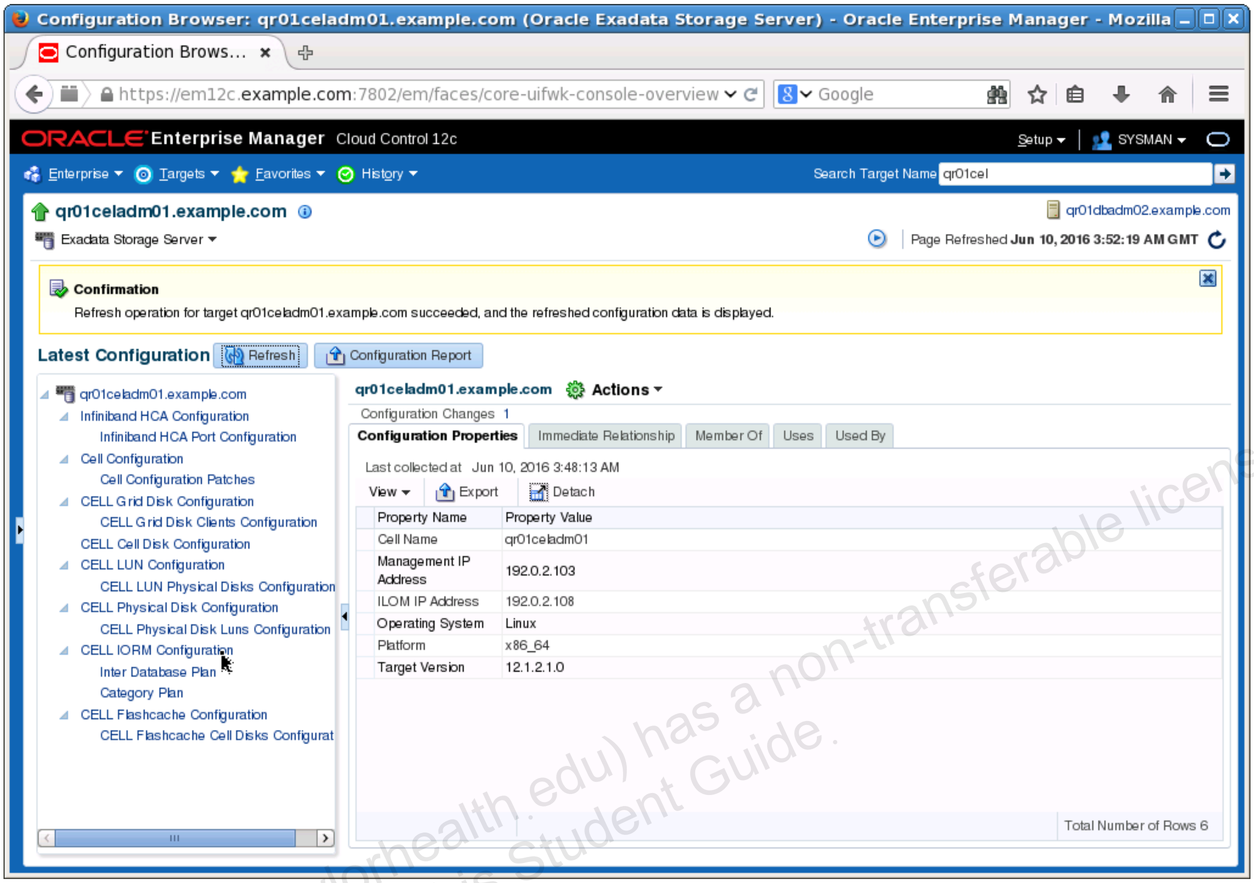
79. Click Refresh. This will cause Enterprise Manager to re-examine the storage server and refresh its configuration information.



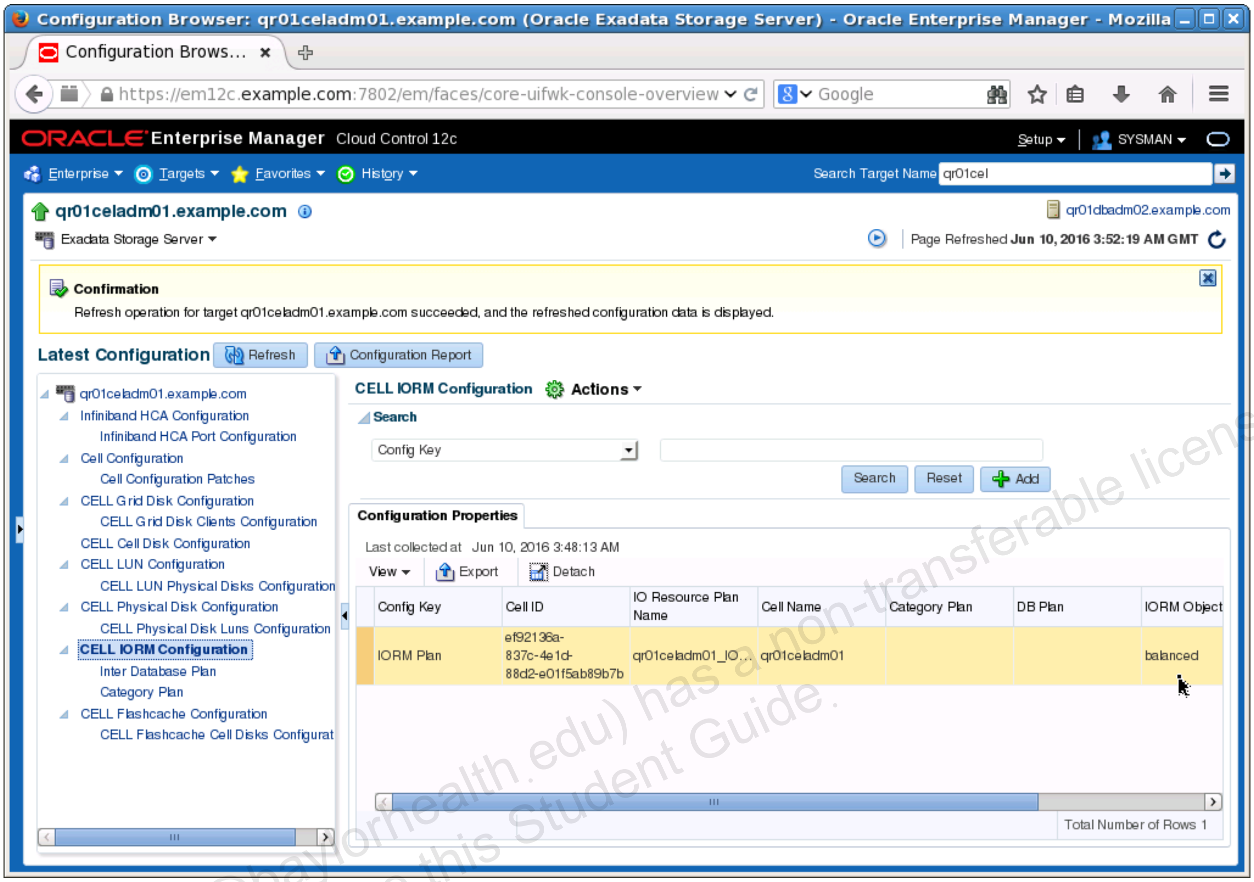
80. Wait until you see a message indicating that the configuration information has been refreshed.



81. Click CELL IORM Configuration in the hierarchical list on the left side of the page.

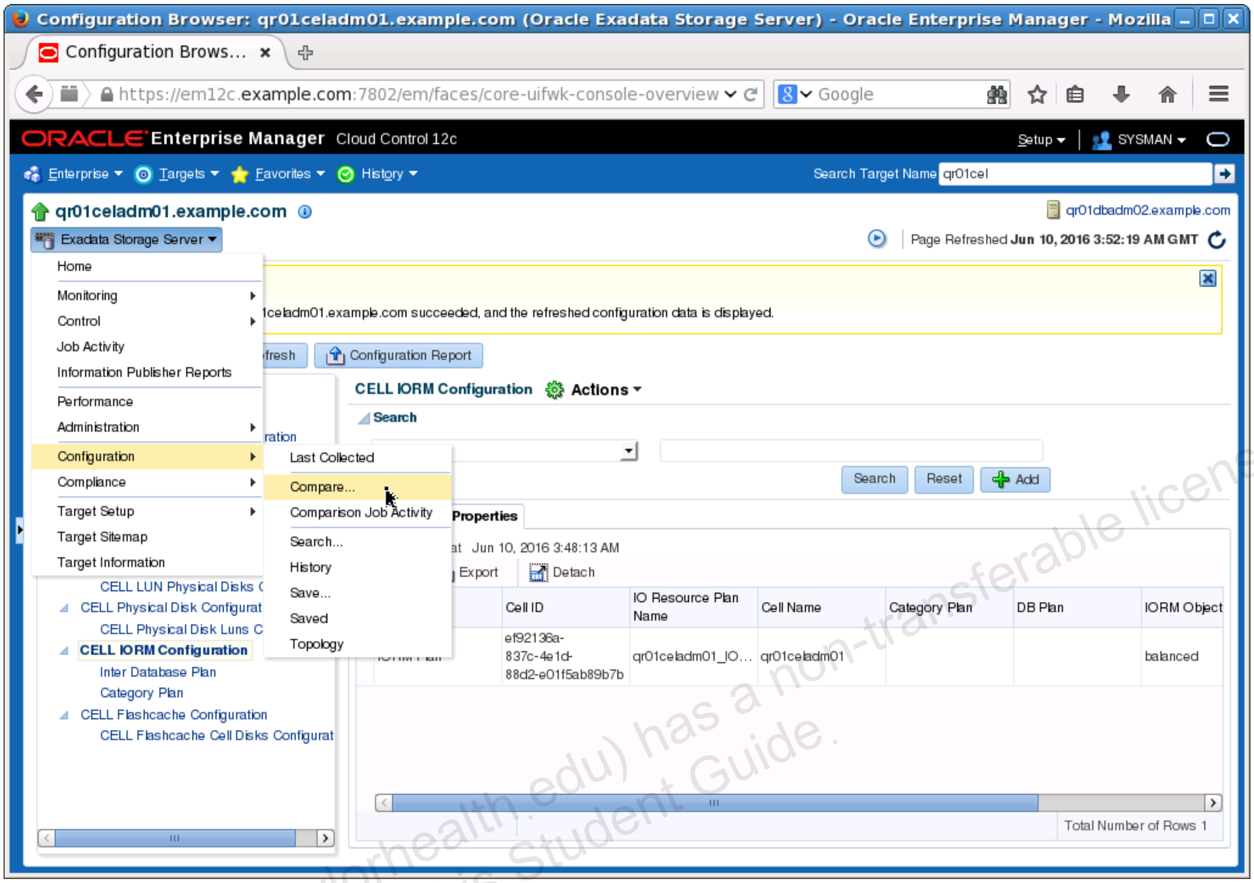


82. Examine the CELL IORM Configuration. Verify that the IORM Objective is listed as `balanced`, reflecting the configuration change you made in step 75.

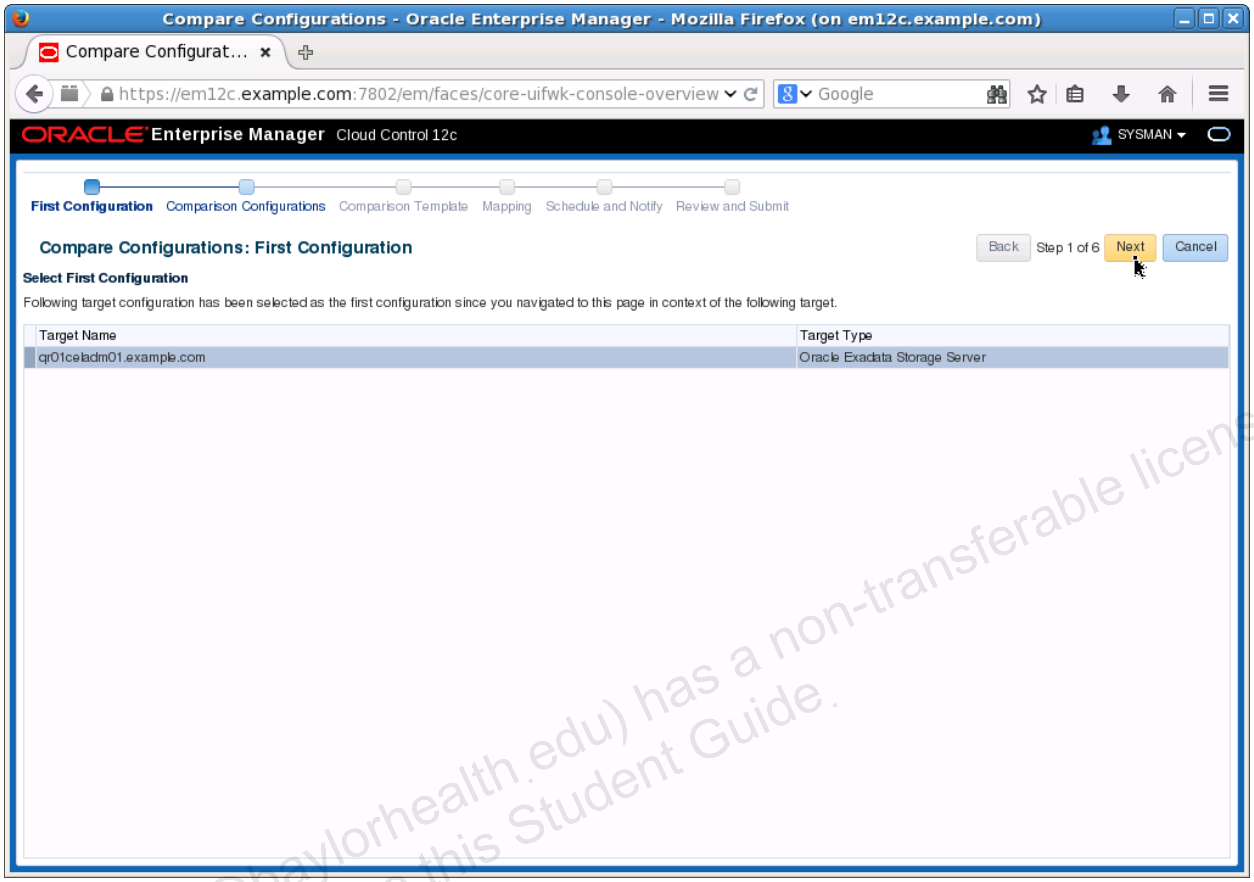


At this point, you have adjusted the IORM objective on one cell (`qr01celadm01`) and you have verified that the adjusted setting is visible in Enterprise Manager. Next, you will compare the configuration of `qr01celadm01` with your other cells to identify if there are any differences.

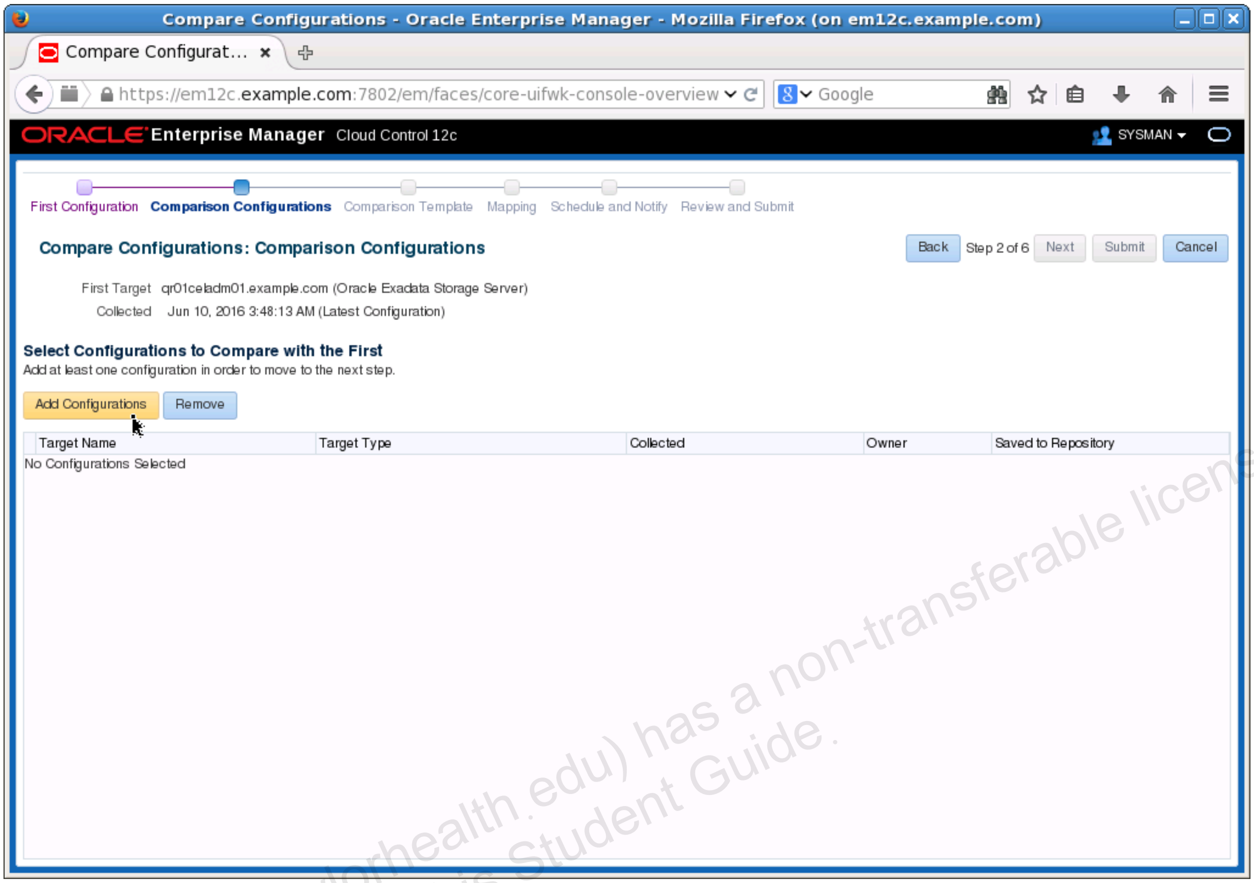
83. Select the Exadata Storage Server > Configuration > Compare... menu command.



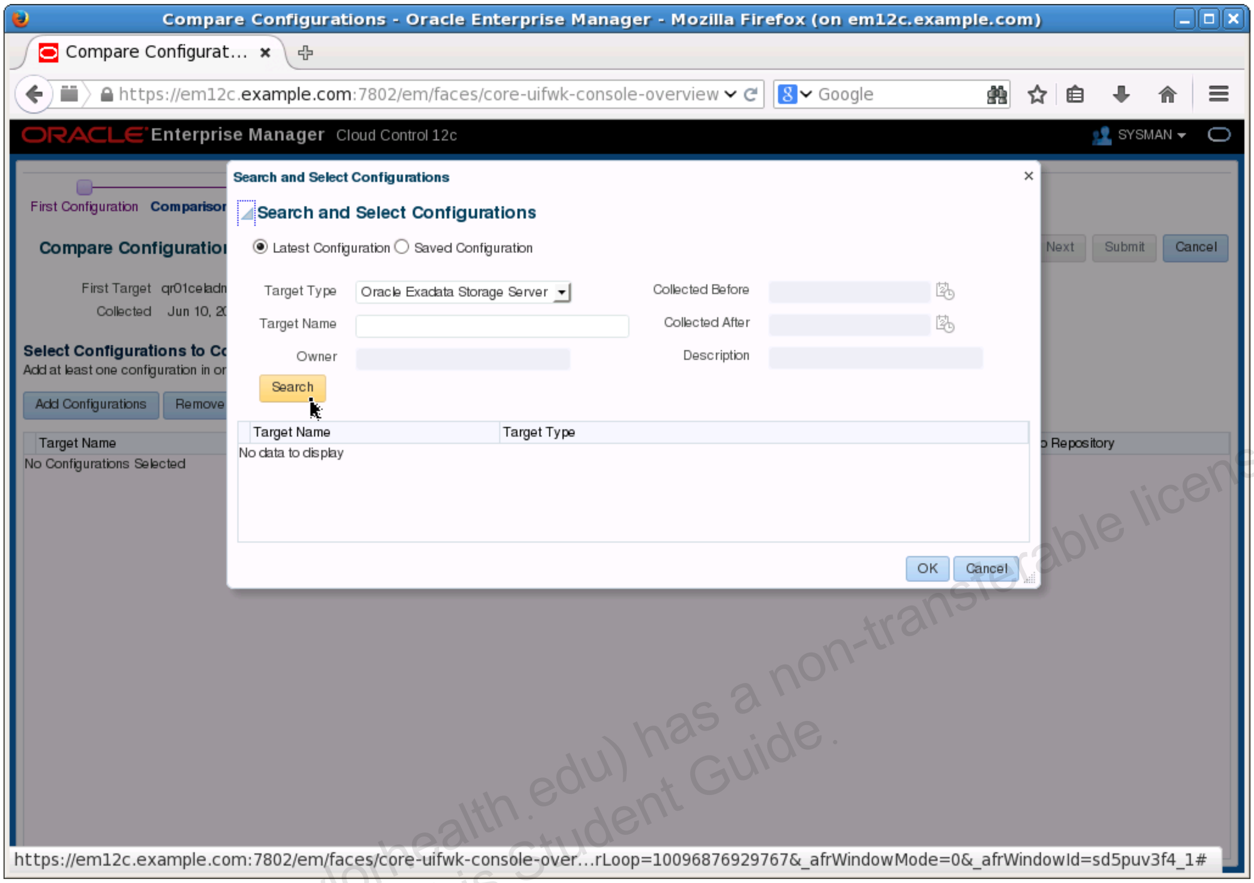
84. Confirm that `qr01celadm01.example.com` is selected as the comparison baseline (first configuration target). Then click Next to proceed.



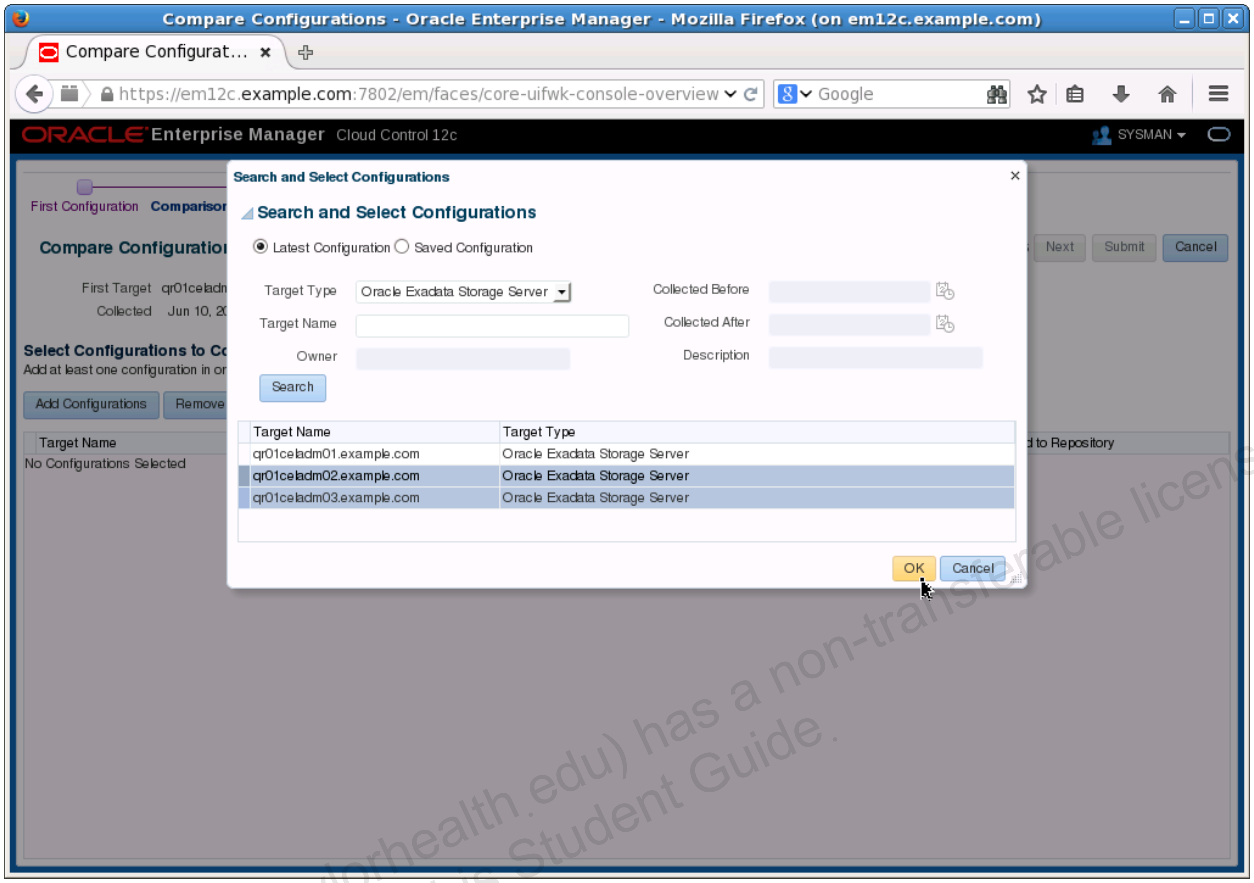
85. Click Add Configurations.



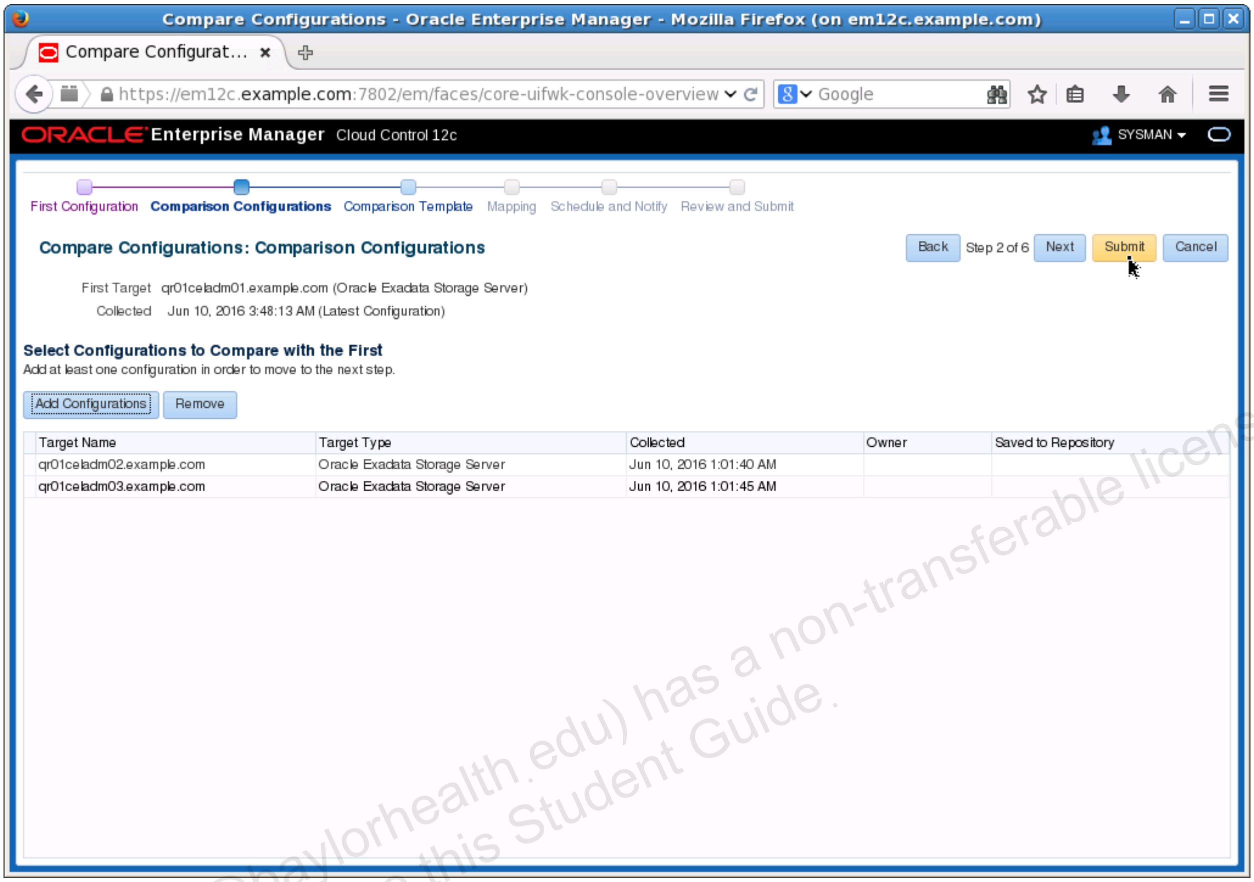
86. Click Search.



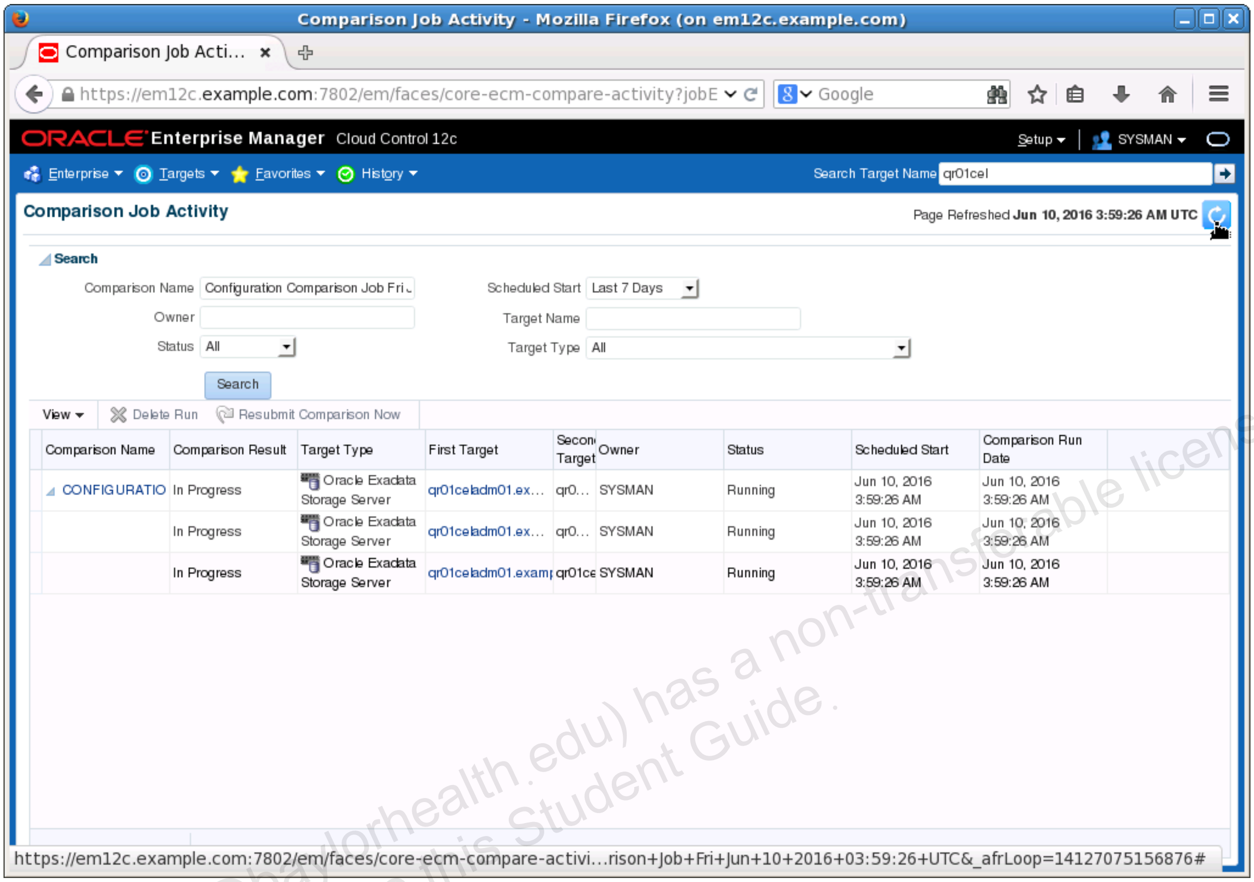
87. Select `qr01celadm02.example.com` and `qr01celadm03.example.com`, and click OK.



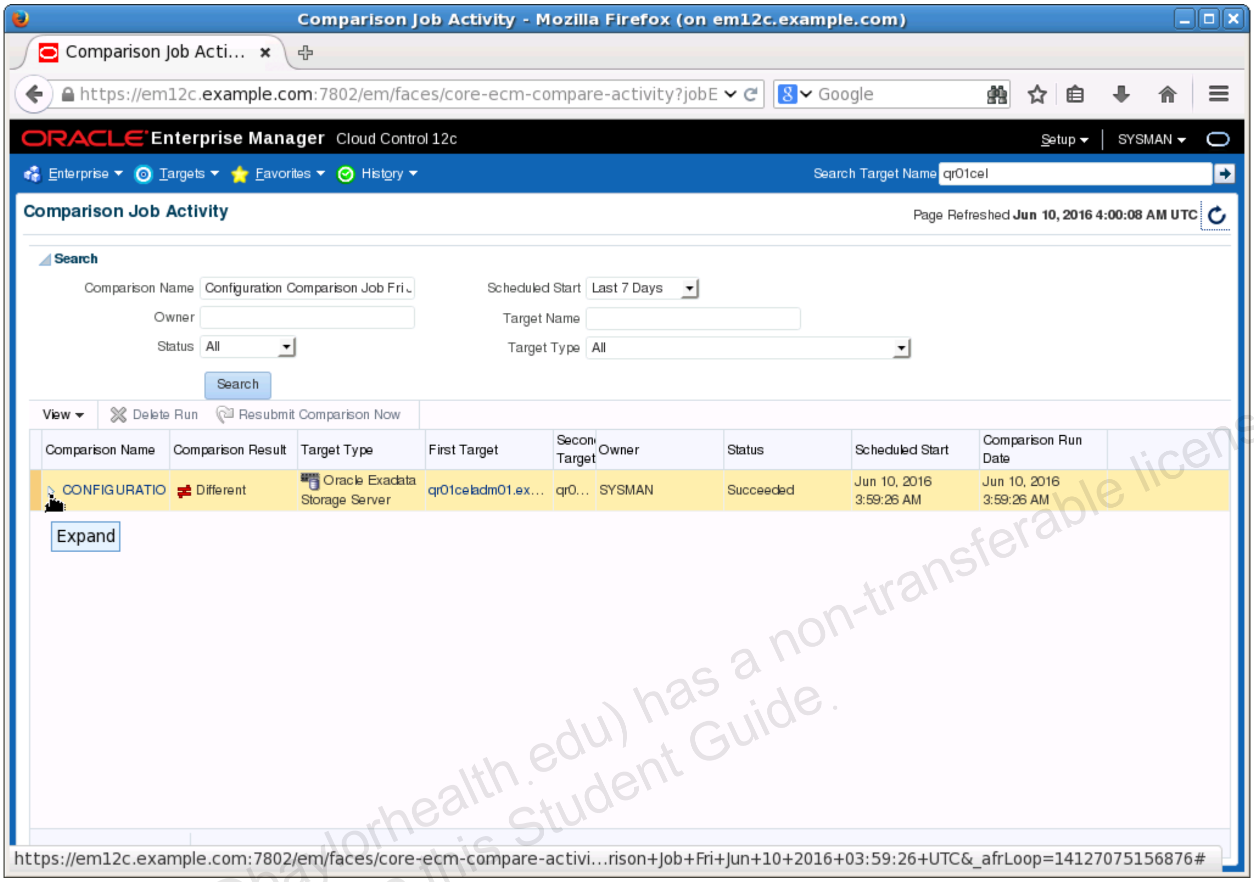
88. Confirm that `qr01celadm02.example.com` and `qr01celadm03.example.com` are selected as the comparison configurations. Then click Submit to proceed.



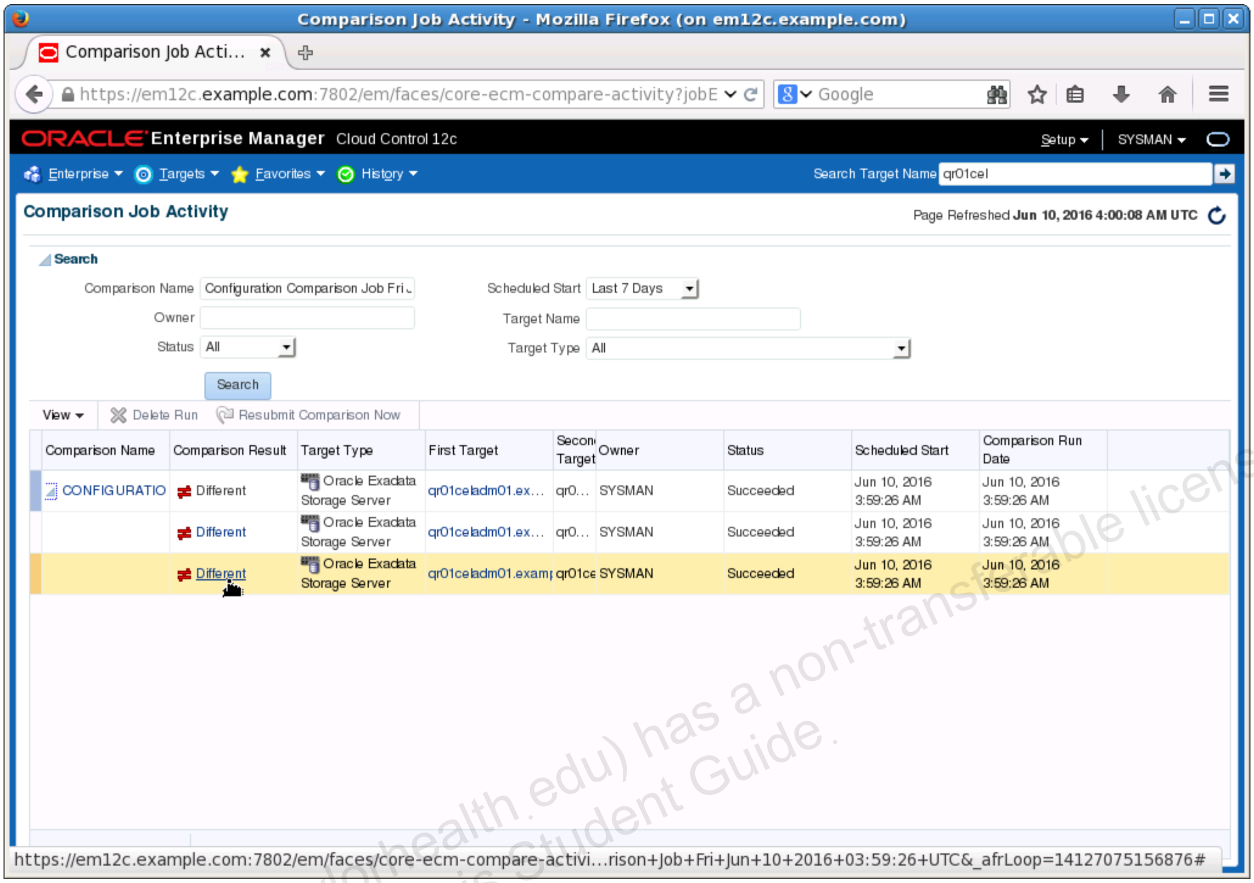
89. On the resulting page, examine the Comparison Result. If the comparison is Scheduled or In Progress, periodically refresh the page until it completes.



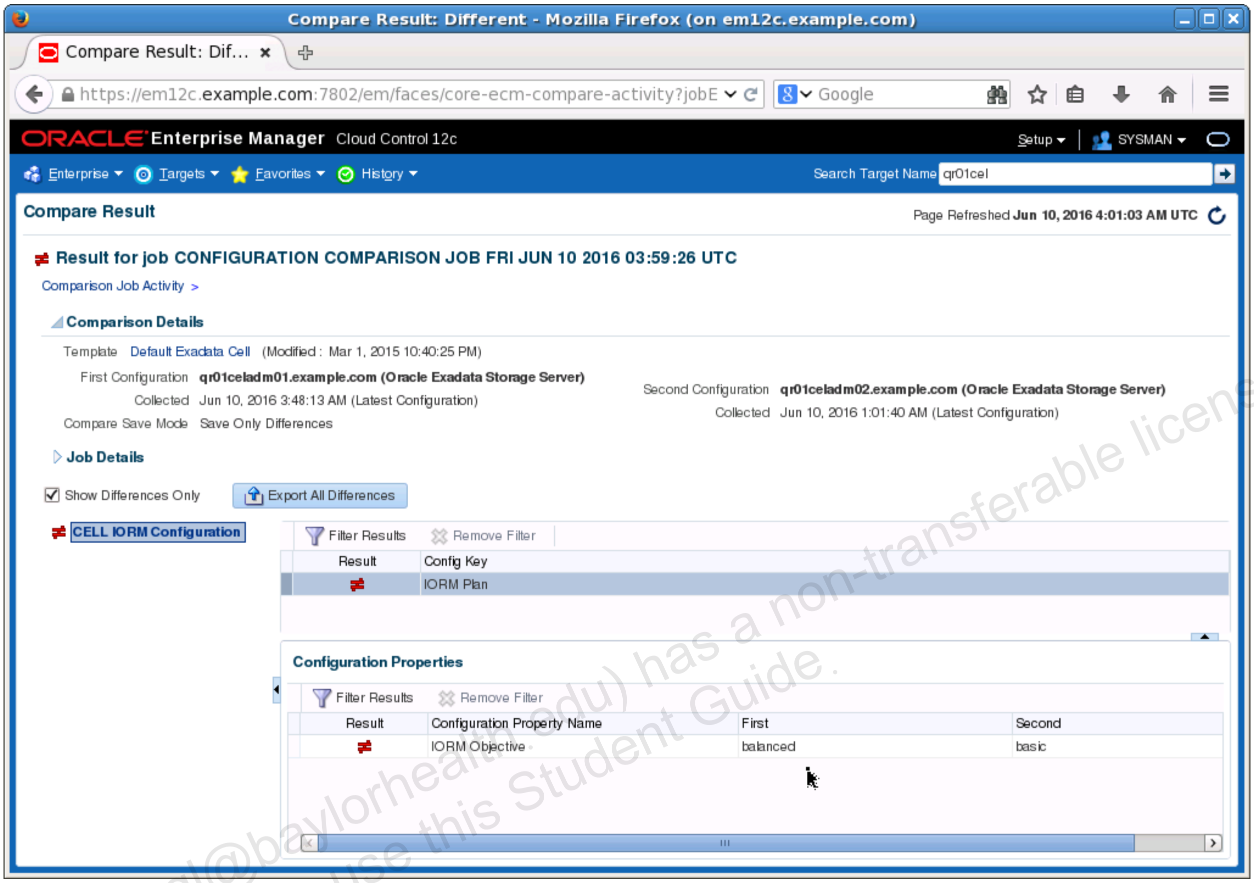
90. When the comparison completes, you should see that the Comparison Result is Different. Click the Expand icon beside the Comparison Name.



91. Click one of the available Different links.

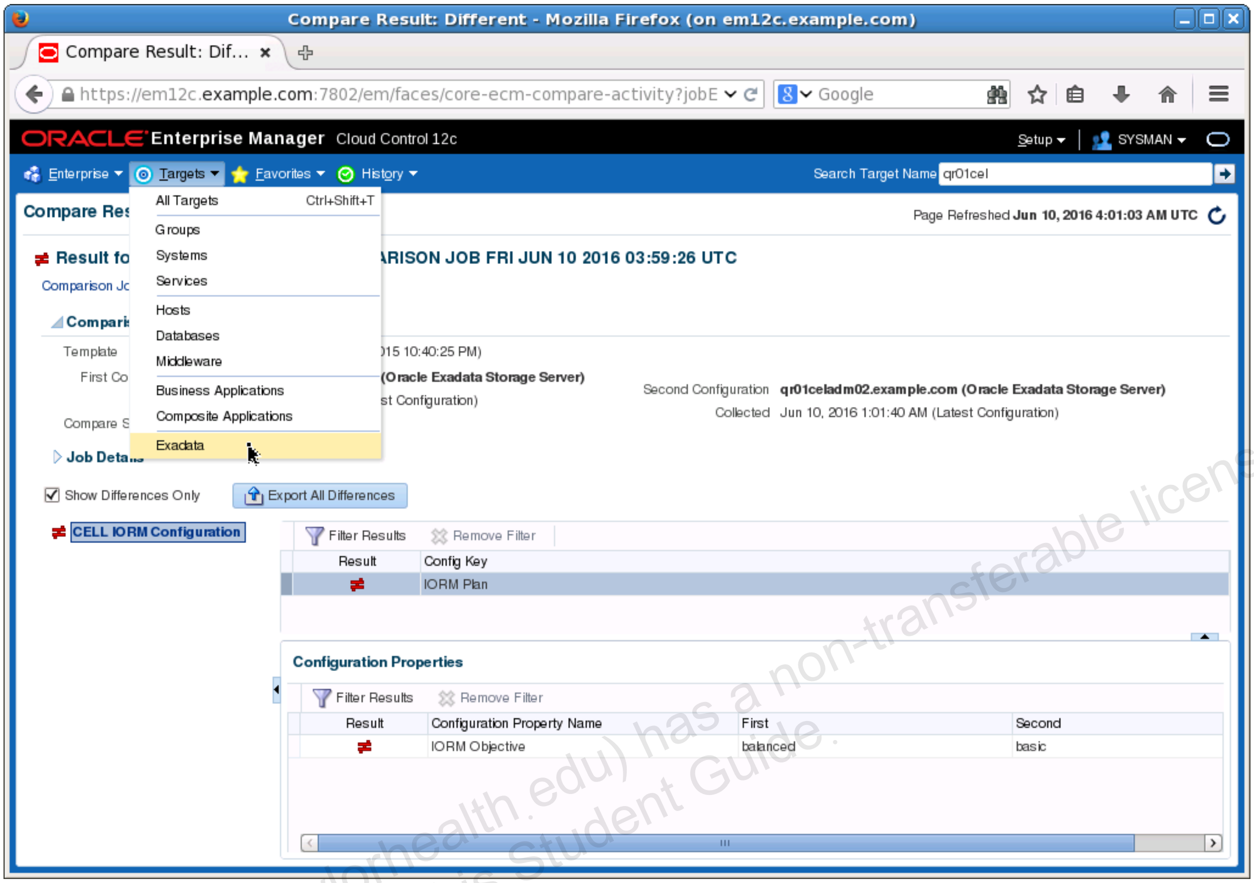


92. Examine the comparison result. You should see that the IORM Objective for the first configuration (`qr01celadm01`) is `balanced`, while it is listed as `basic` (the default setting) in the other cell.

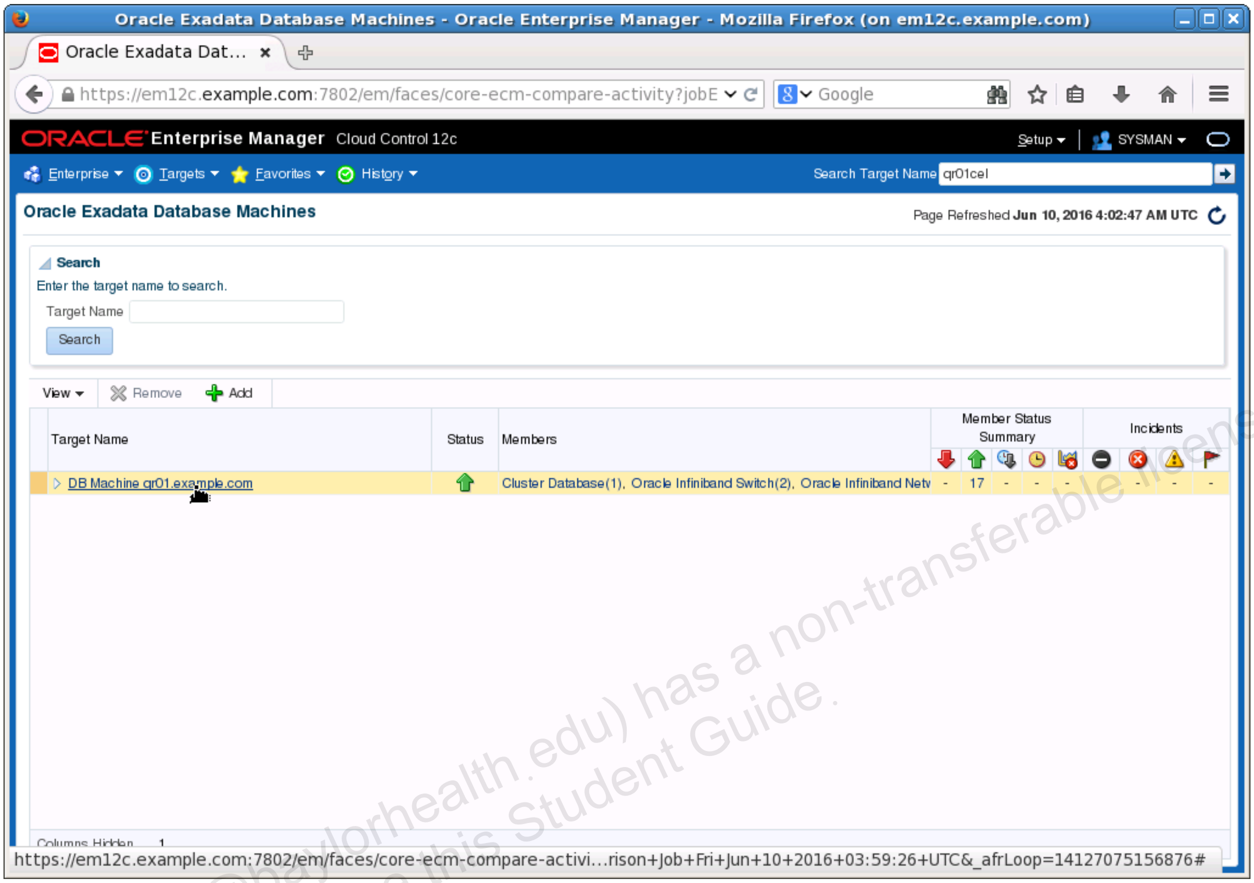


In the final part of this practice, you will use the IORM administration capabilities in Enterprise Manager to return your cells to a consistent configuration.

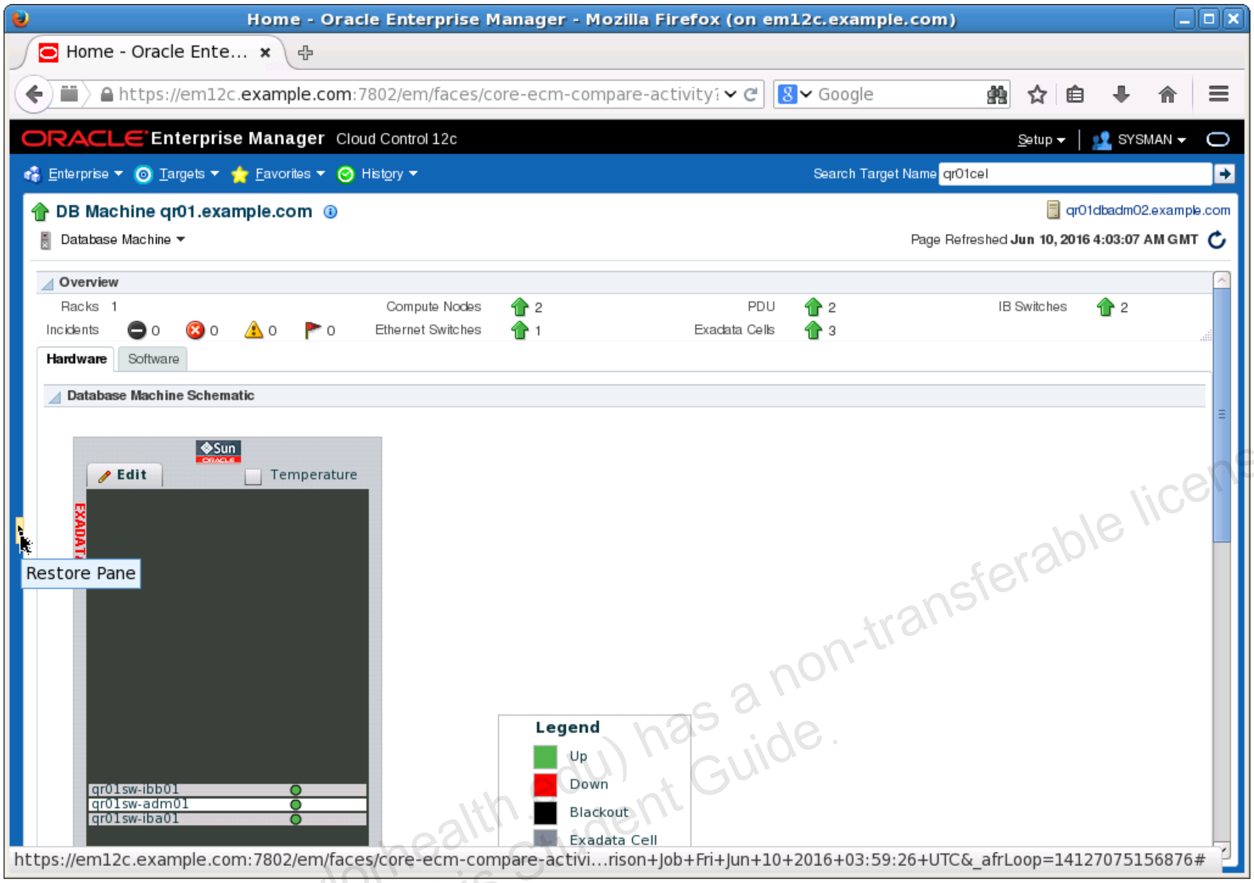
93. Select the Targets > Exadata menu command.



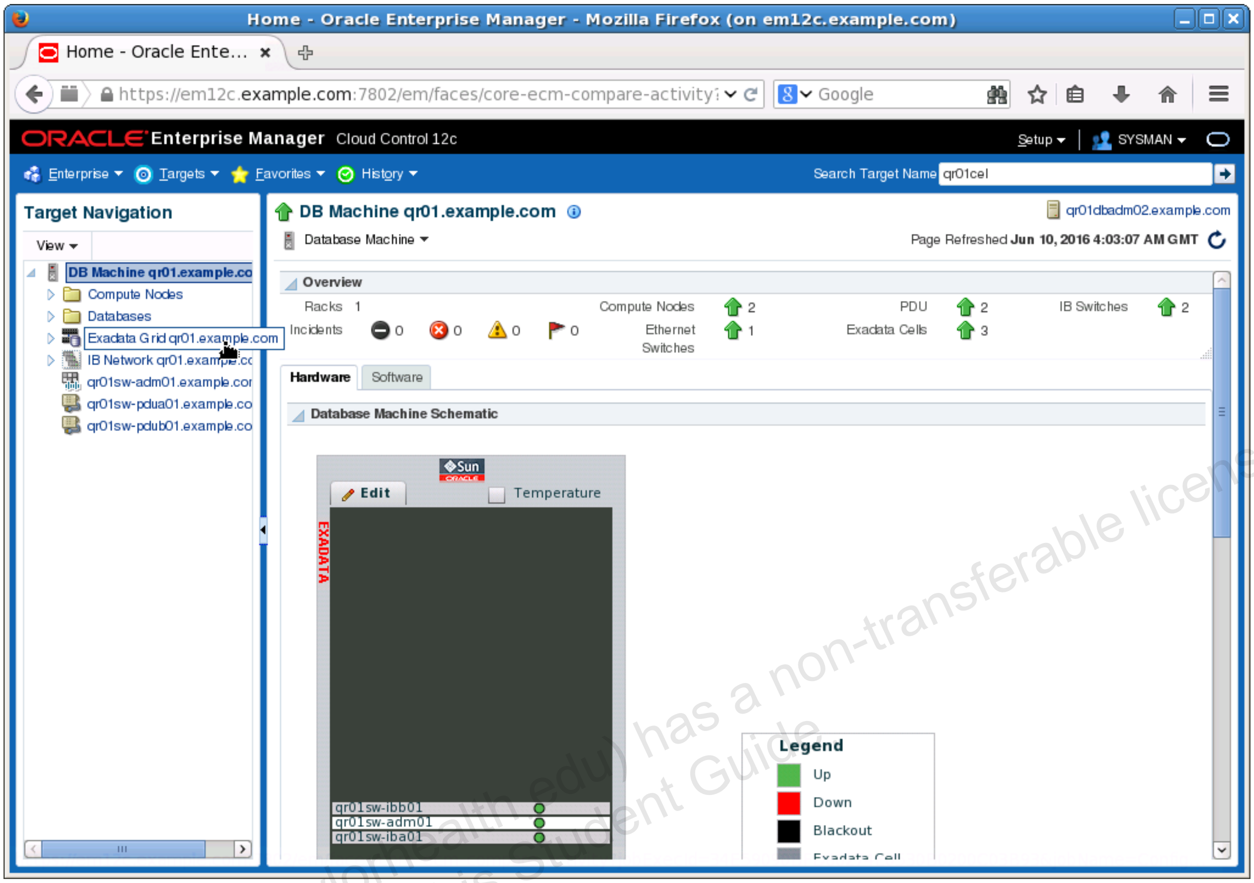
94. Click the link corresponding to your Exadata Database Machine (`DB Machine qr01.example.com`).



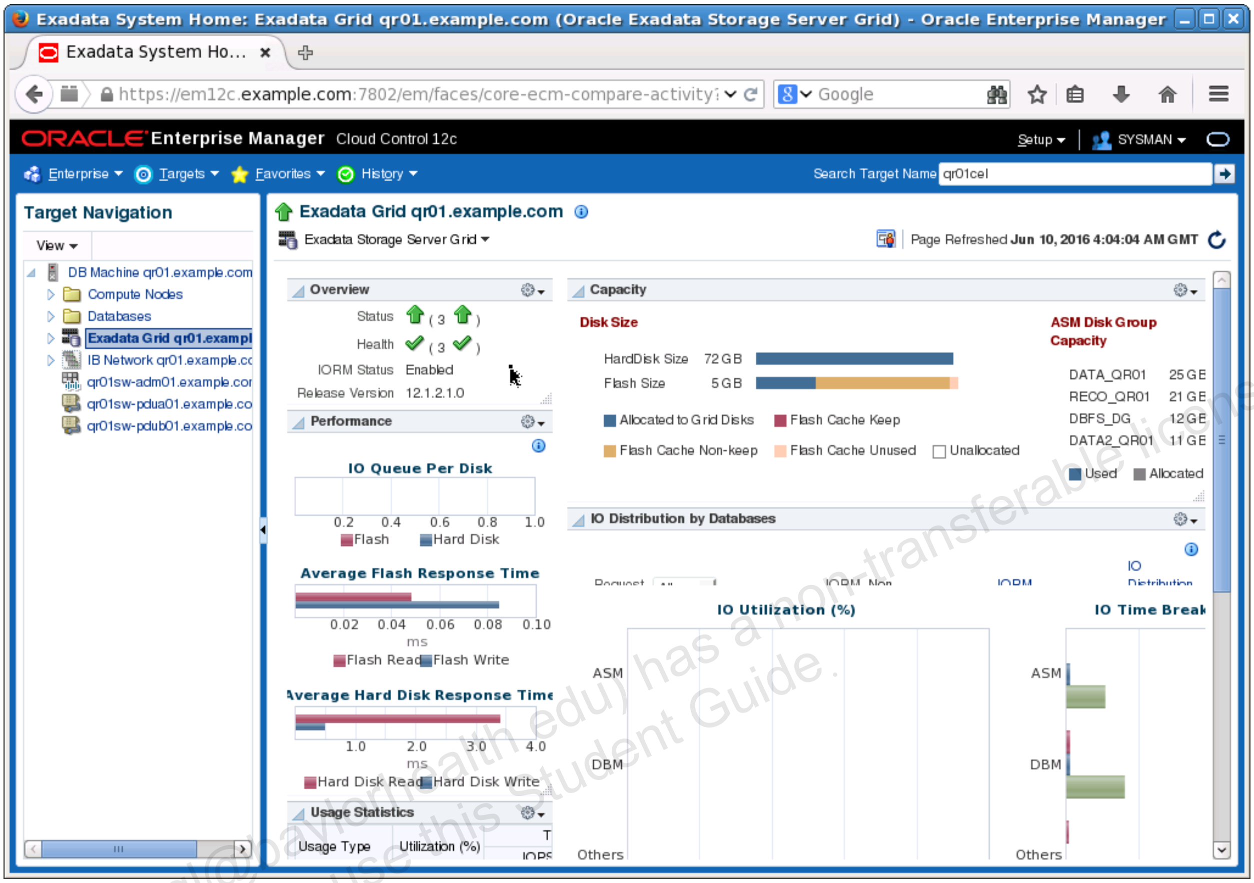
95. Restore the Target Navigation pane.



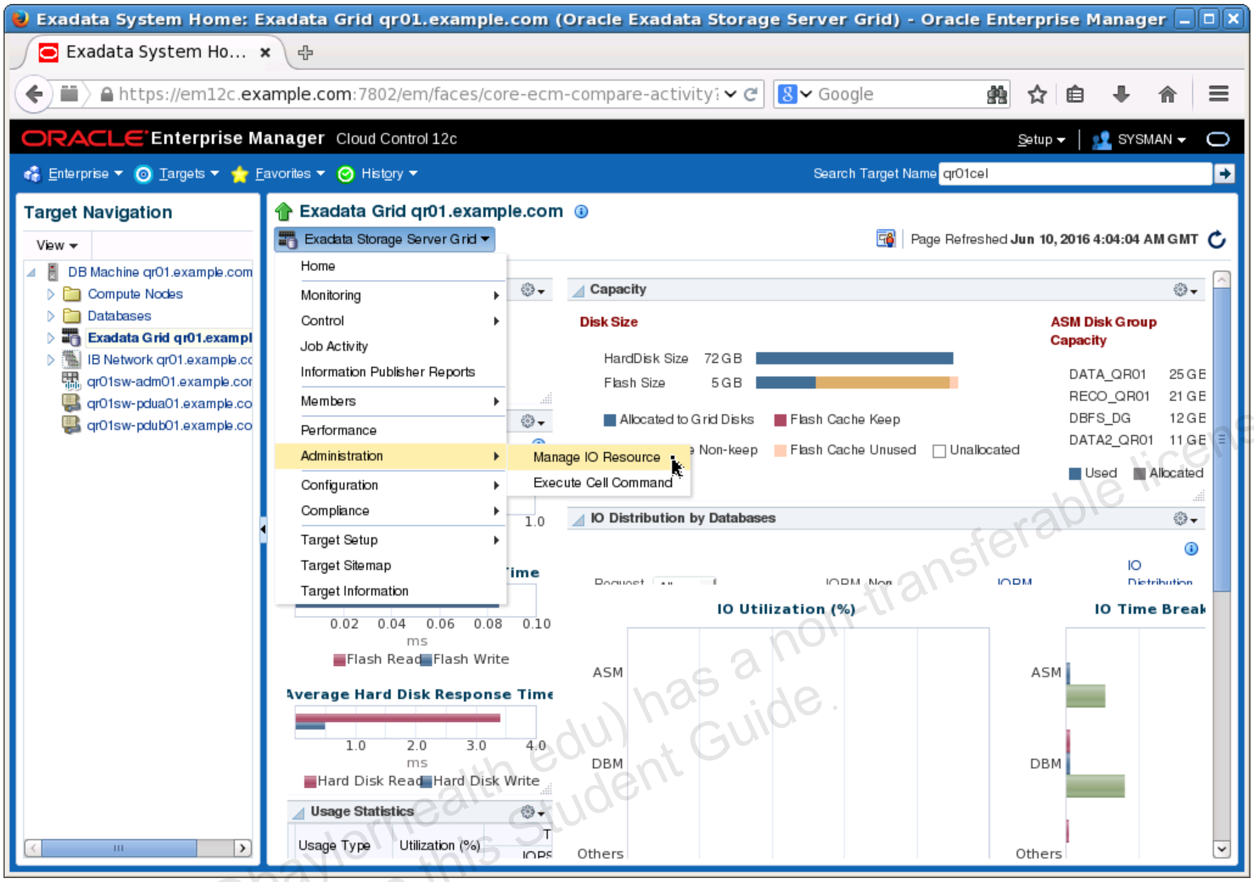
96. In the Target Navigation pane, click “Exadata Grid qr01.example.com”.



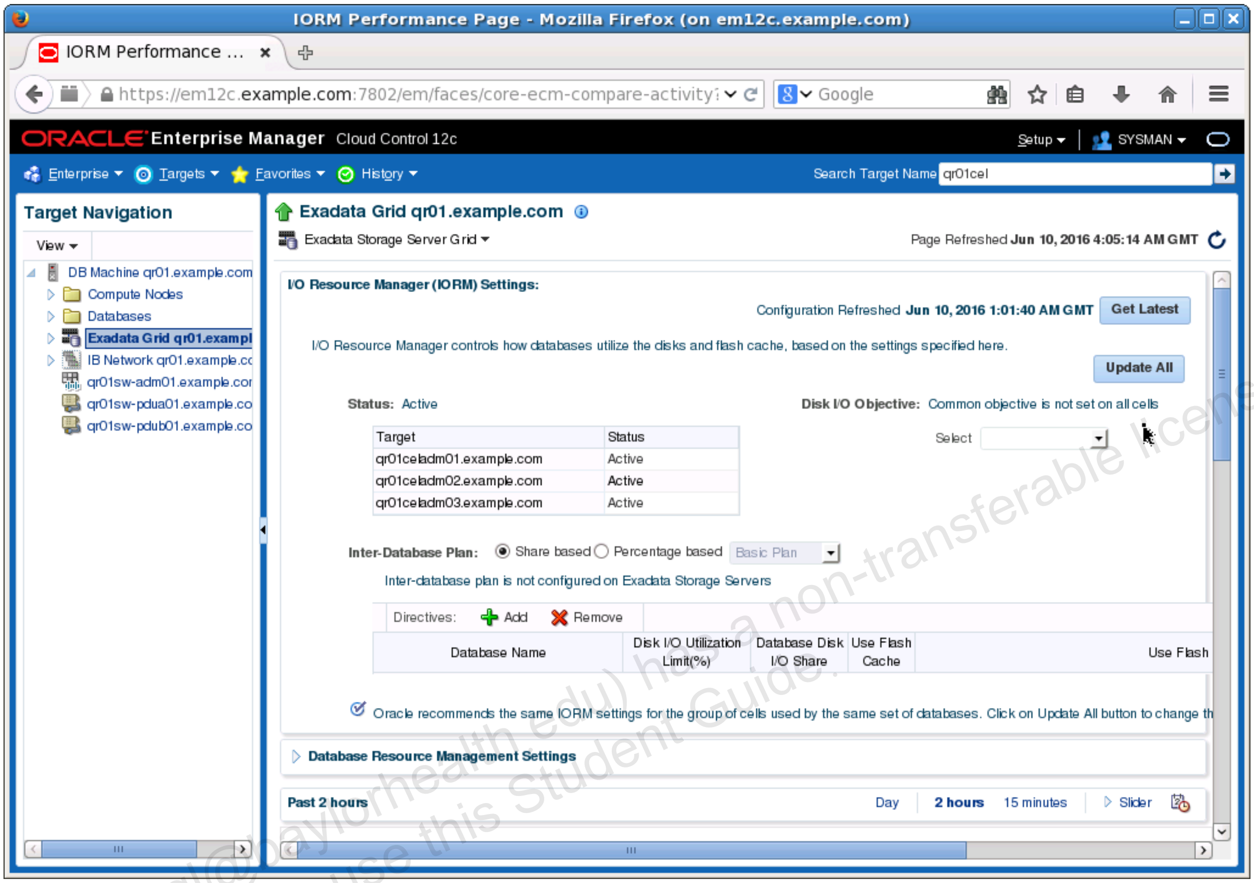
97. You now see the Exadata Storage Server Grid home page. This page is very similar to the home page for each individual Exadata Storage Server. However, the information presented on this page covers all the storage servers in the Exadata Database Machine.



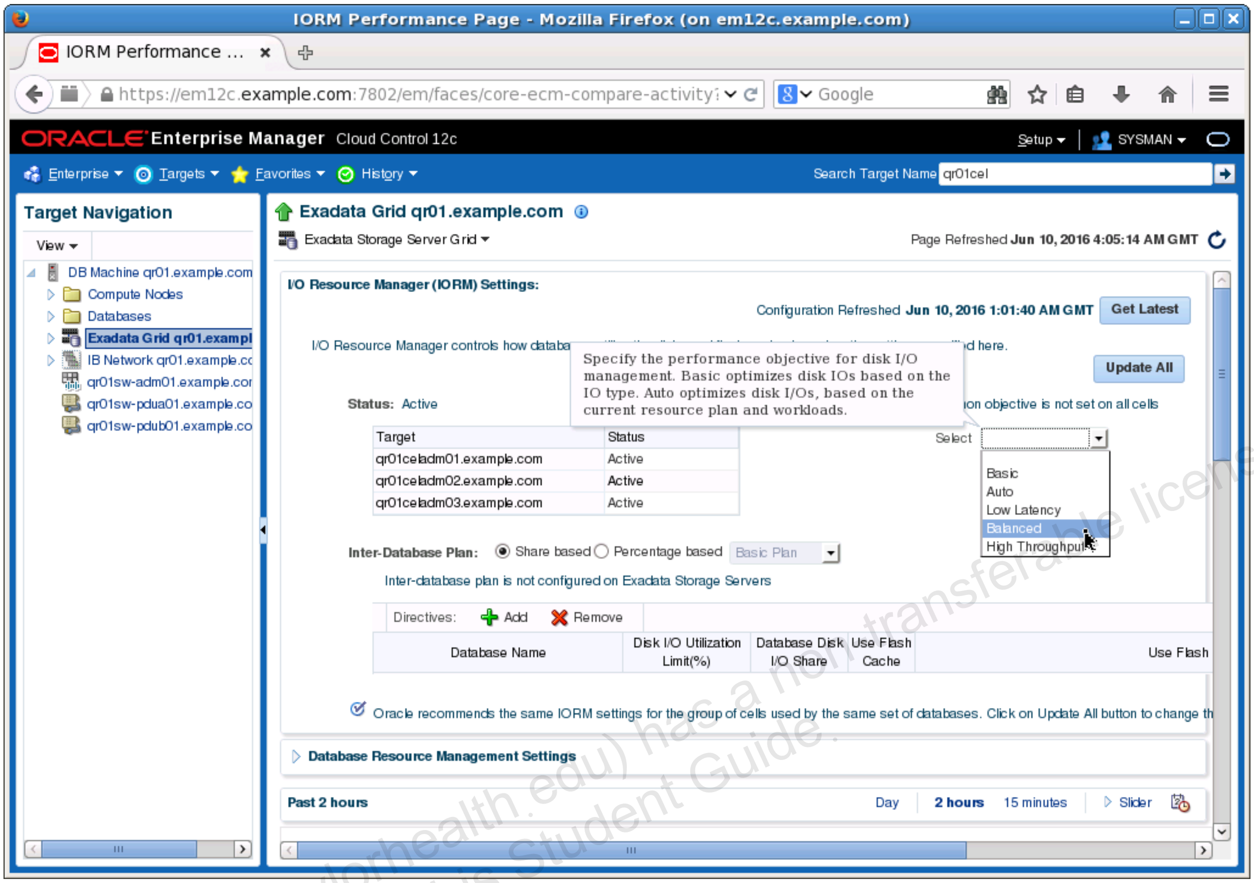
98. Select the Exadata Storage Server Grid > Administration > Manage IO Resource menu command.



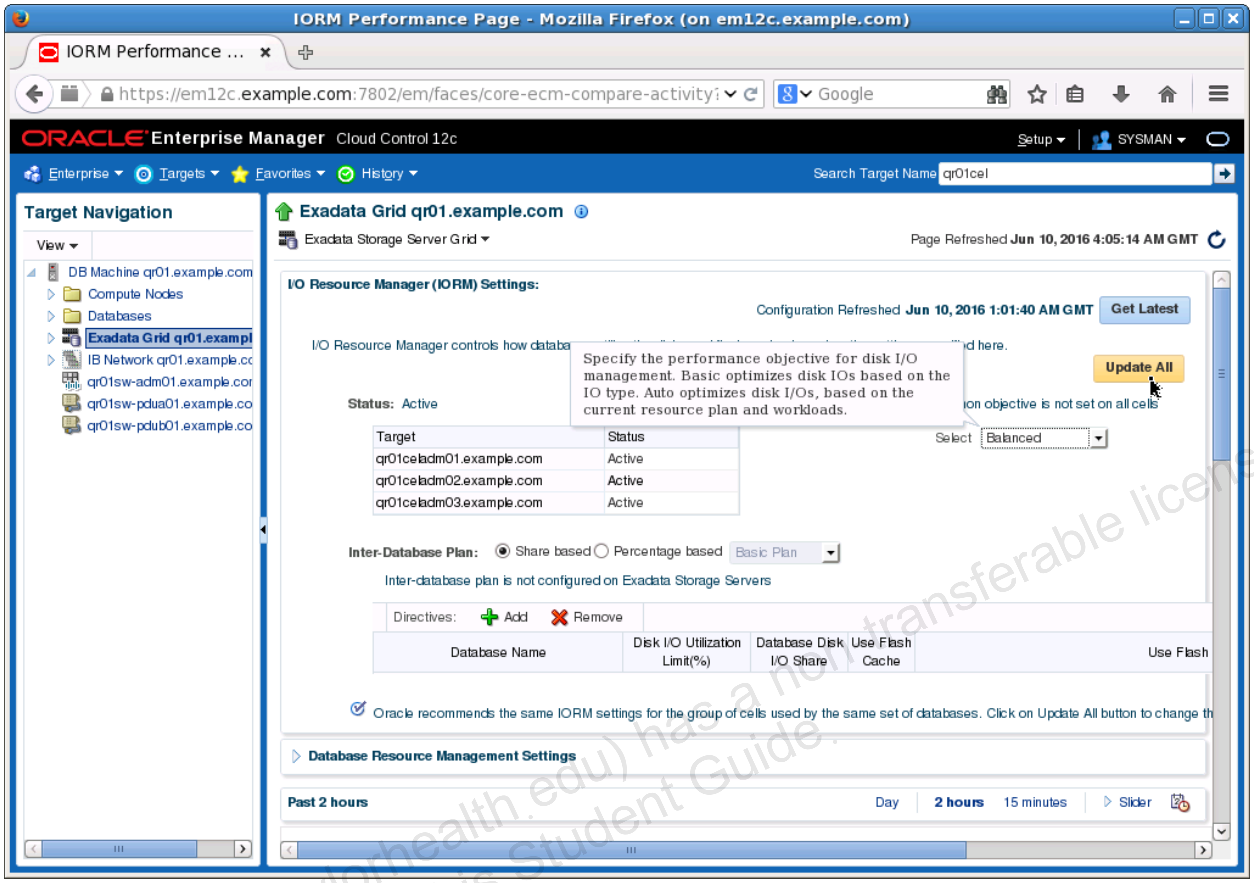
99. The resulting page provides administrators with an interface to manage IORM configuration settings across all the Exadata Storage Servers in the Exadata Database Machine. Notice that the page indicates a difference in the IORM objective across the cells.



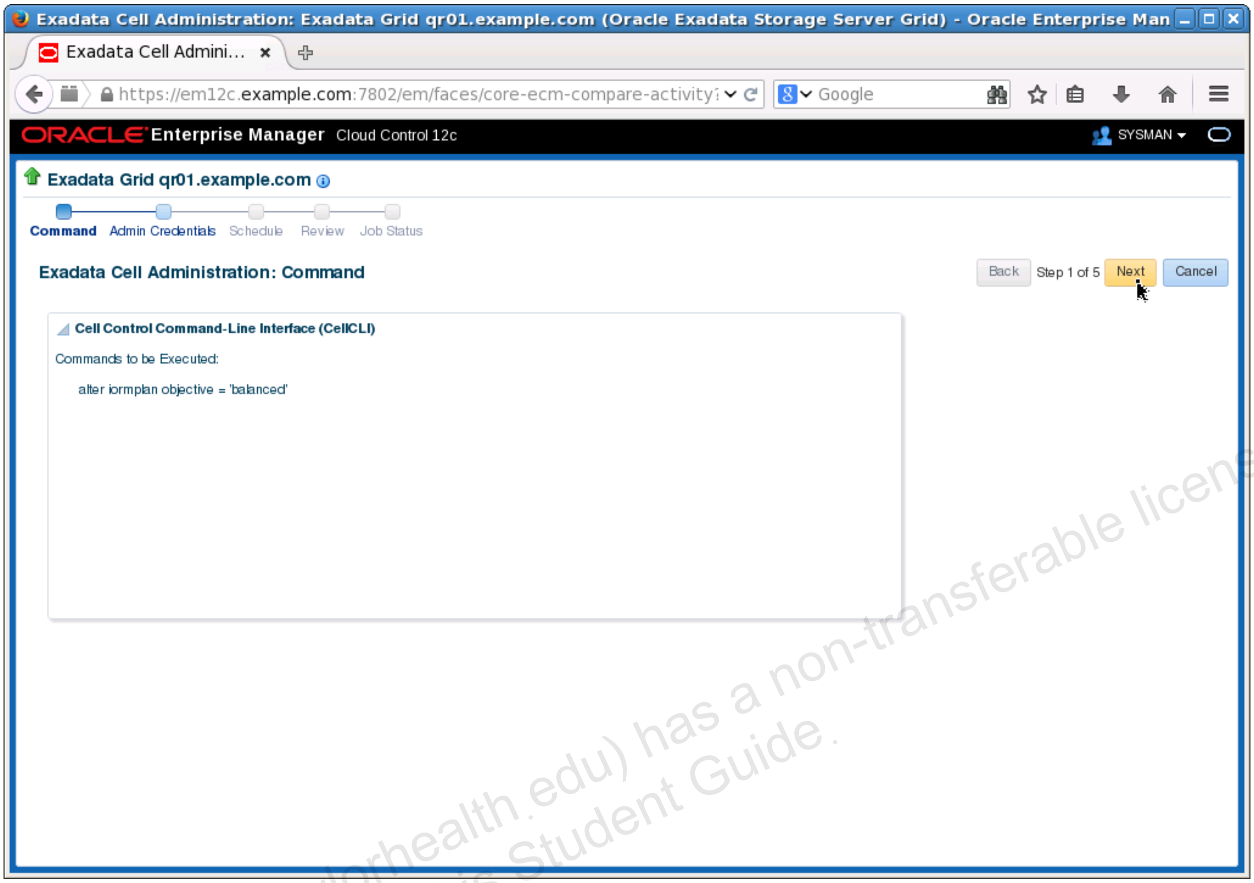
100. Select `Balanced` as the IORM objective for all your cells.



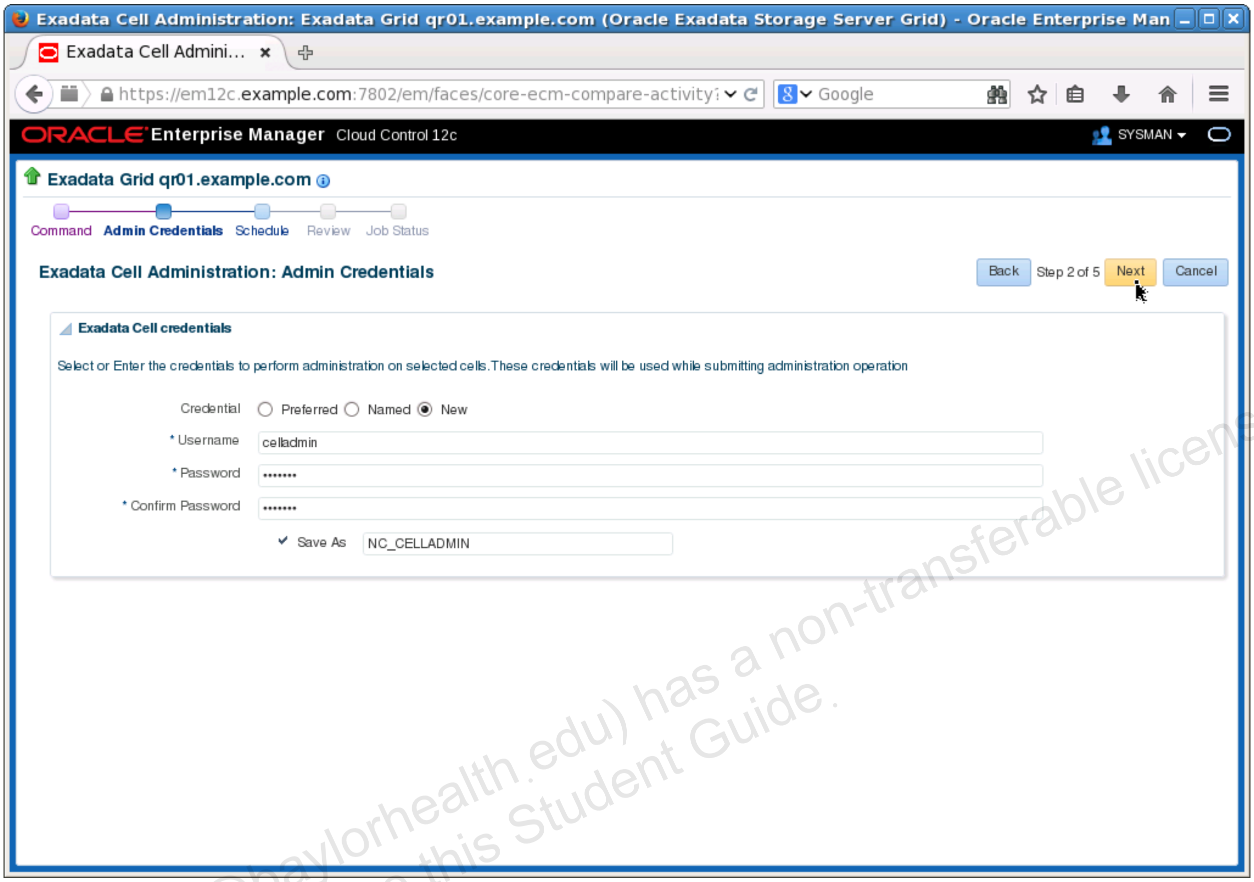
101.Click Update All to update the IORM objective across all your cells.



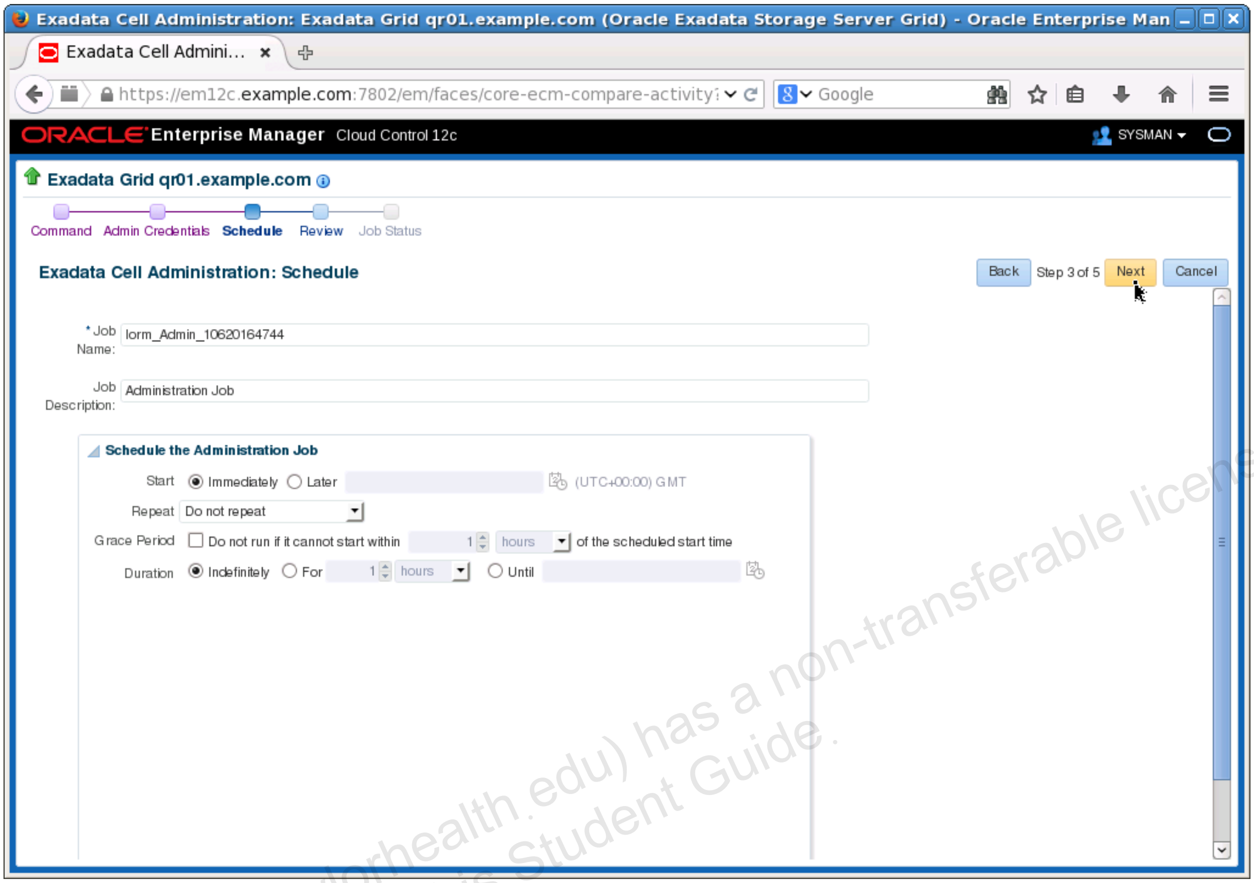
102. Verify that the command matches your expectation. Then click Next to proceed.



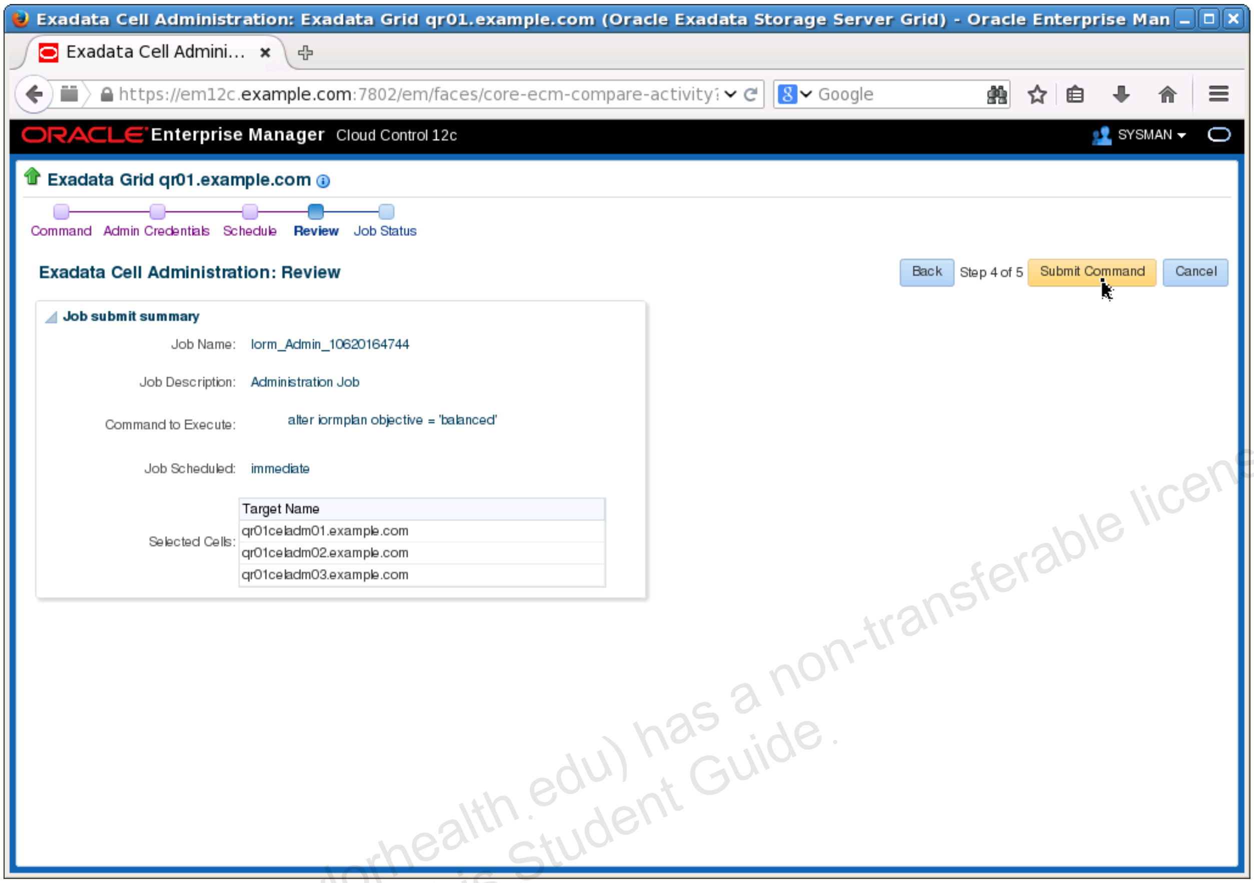
103. Specify new named credentials for the `celladmin` user.
Finally, click Next to proceed.



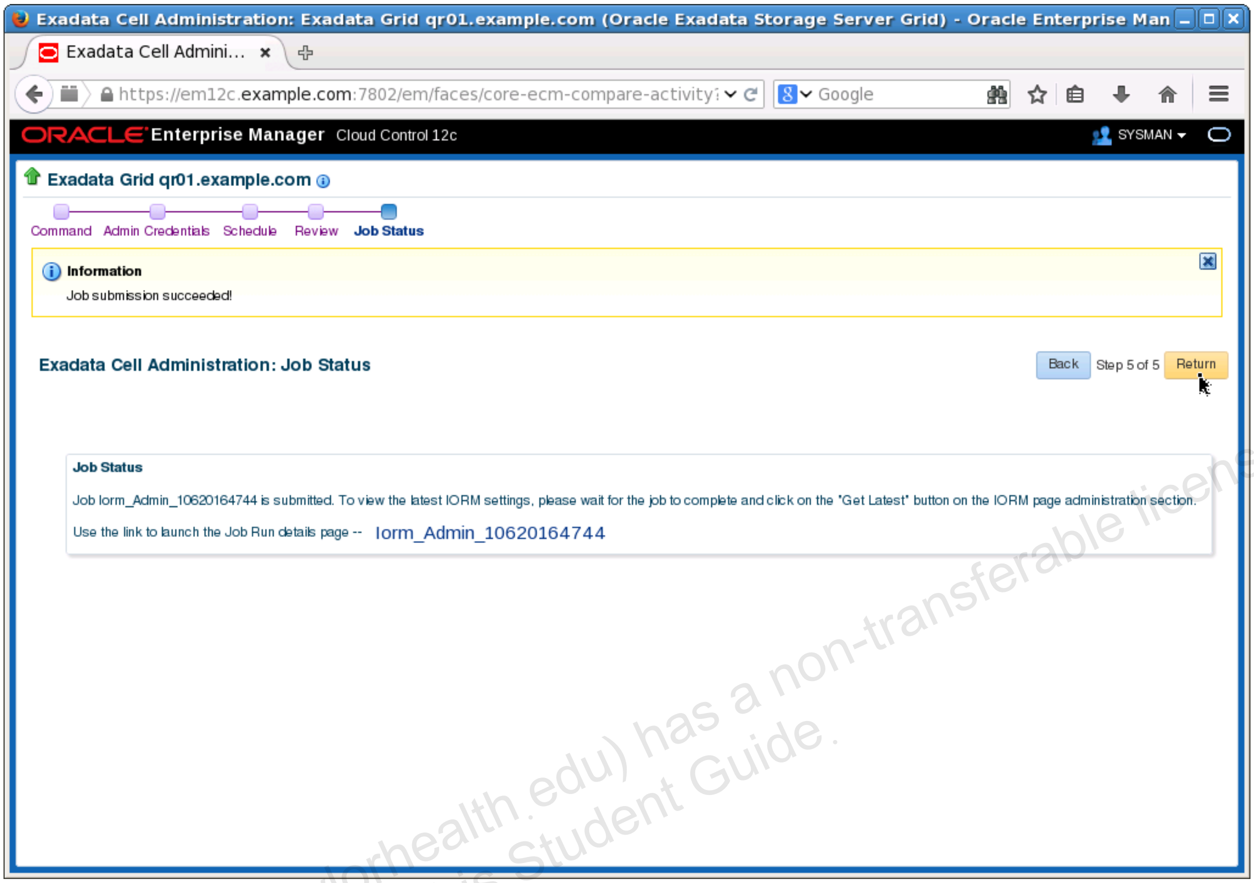
104.Accept the default schedule settings and click Next.



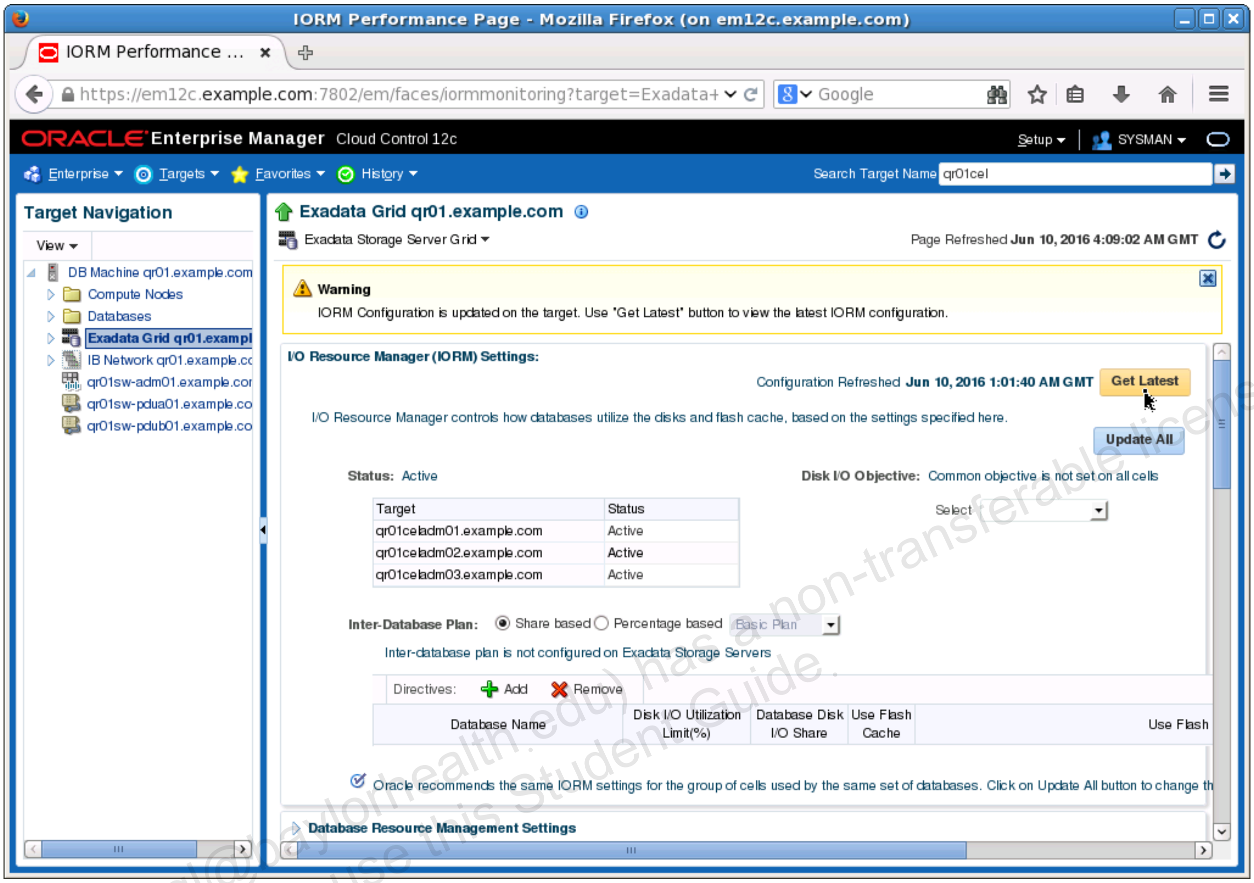
105.Review the “Job submit summary” and click Submit Command.



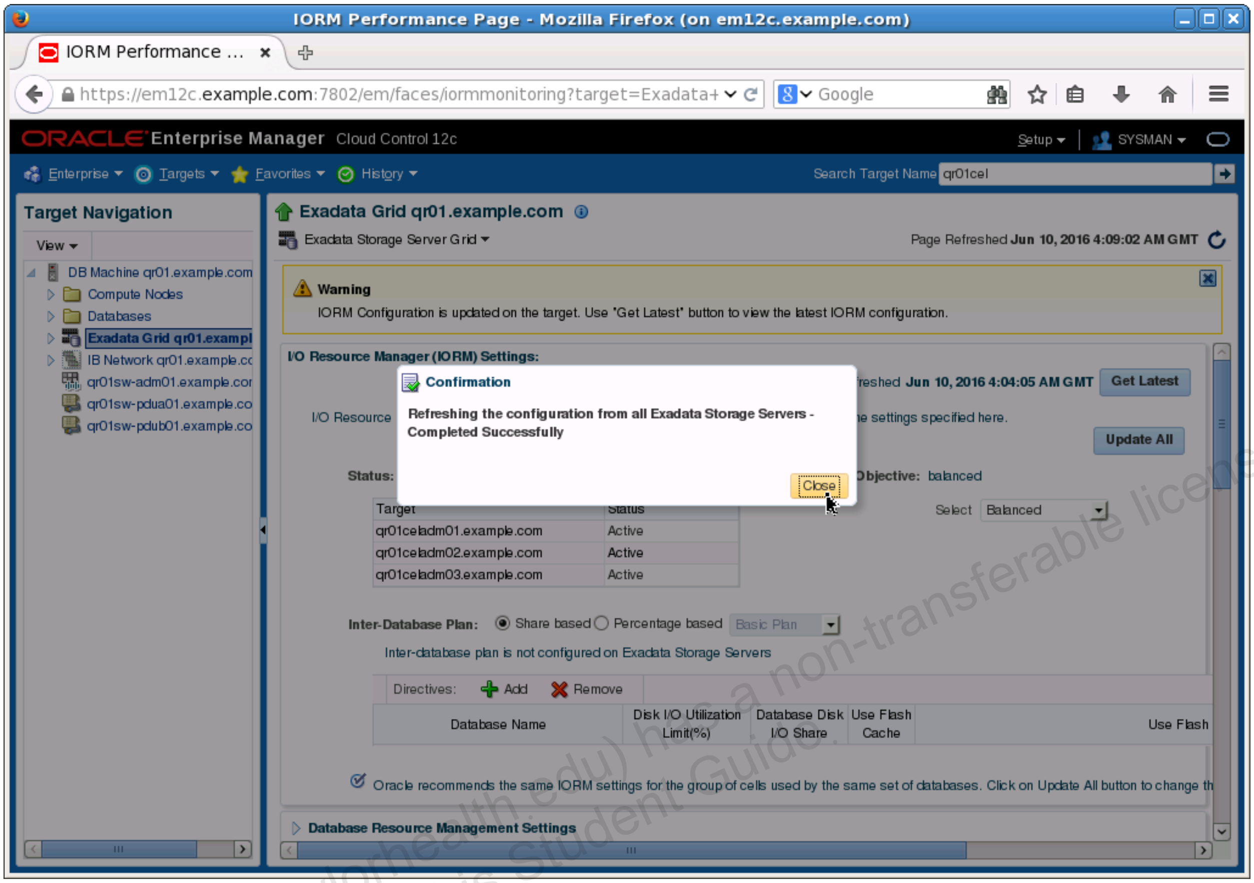
106.Confirm that job submission succeeded and click Return.



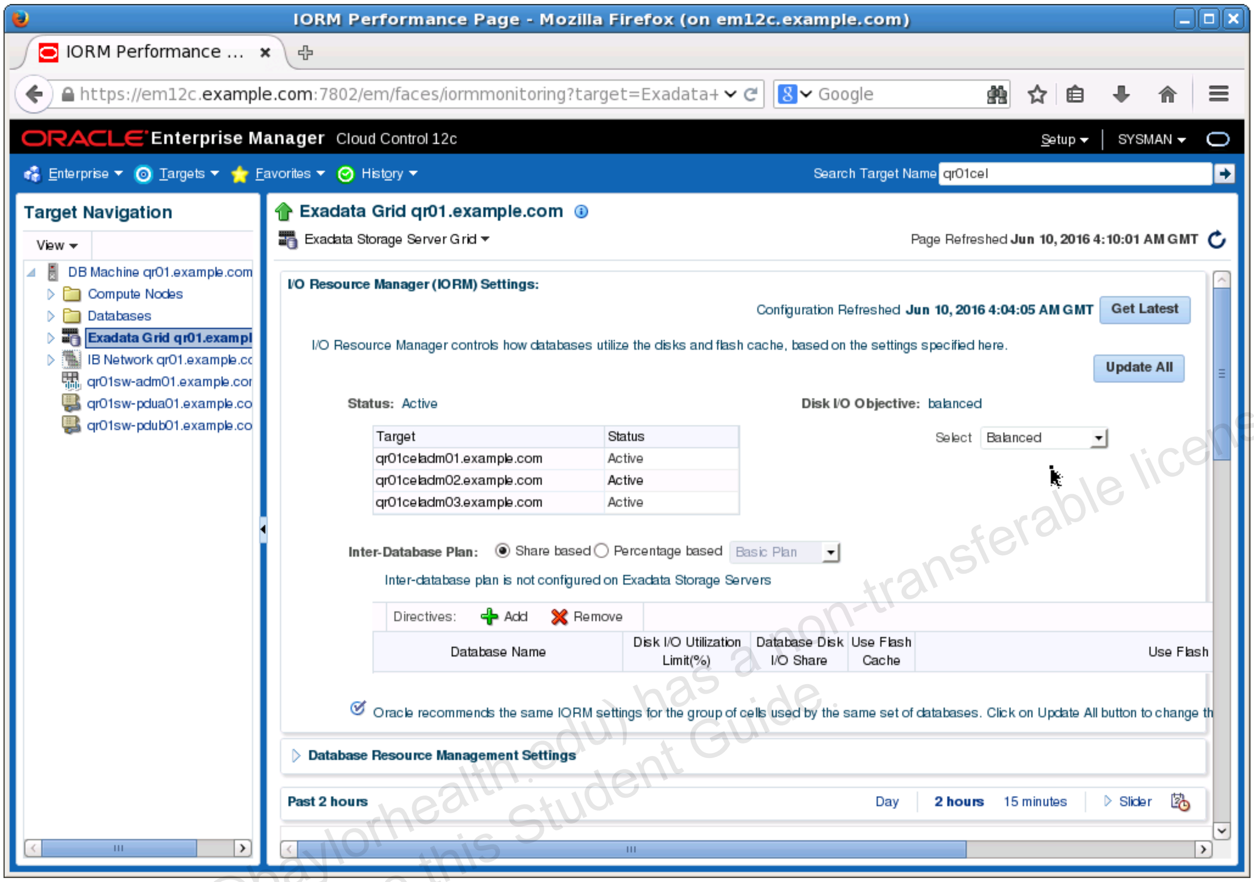
107. When you are returned to the IORM Settings page, you should notice that it still indicates that the Disk IO Objective is not common across all the cells. To update this page, click Get Latest.



108. Click Close to dismiss the confirmation dialog box.



109.Verify that the Disk I/O Objective is now listed as `balanced`. This confirms that the setting has been consistently applied across all your cells.



Congratulations! You have performed a variety of storage server monitoring and administration tasks by using Enterprise Manager Cloud Control 12*c*.

Practices for Lesson 16: Monitoring Exadata Database Machine Database Servers

Chapter 16

Practices for Lesson 16: Overview

Practices Overview

In this practice, you will be introduced to the Exadata-specific database monitoring capabilities provided by Enterprise Manager Cloud Control 12c.

Practice 16-1: Exadata Database Monitoring with Enterprise Manager

Overview

Using Enterprise Manager to monitor Oracle databases on Exadata Database Machine is essentially the same as using it to monitor databases outside of Exadata. However, for Oracle databases running on Exadata Database Machine, Enterprise Manager does include some additional Exadata-specific information. In this practice you will be introduced to the Exadata-specific database monitoring capabilities provided by Enterprise Manager Cloud Control 12c.

Assumptions

The practice relies on the configurations performed in Practice 14-1.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Change into the `labs/swingbench/bin` directory.

```
[oracle@qr01dbadm01 ~]$ cd labs/swingbench/bin
[oracle@qr01dbadm01 bin]$
```

3. Use the following command to execute a workload against your database. The workload consists of several long-running queries. Leave your terminal alone so that the workload can run to completion, which will take approximately 5 minutes.

```
[oracle@qr01dbadm01 bin]$ ./charbench -c sales.xml -rt 00:05
Author  :        Dominic Giles
Version :        2.5.0.971

Results will be written to results.xml.
Hit Return to Terminate Run...

Time            Users   TPM     TPS

4:54:55 AM      6       0       0
```

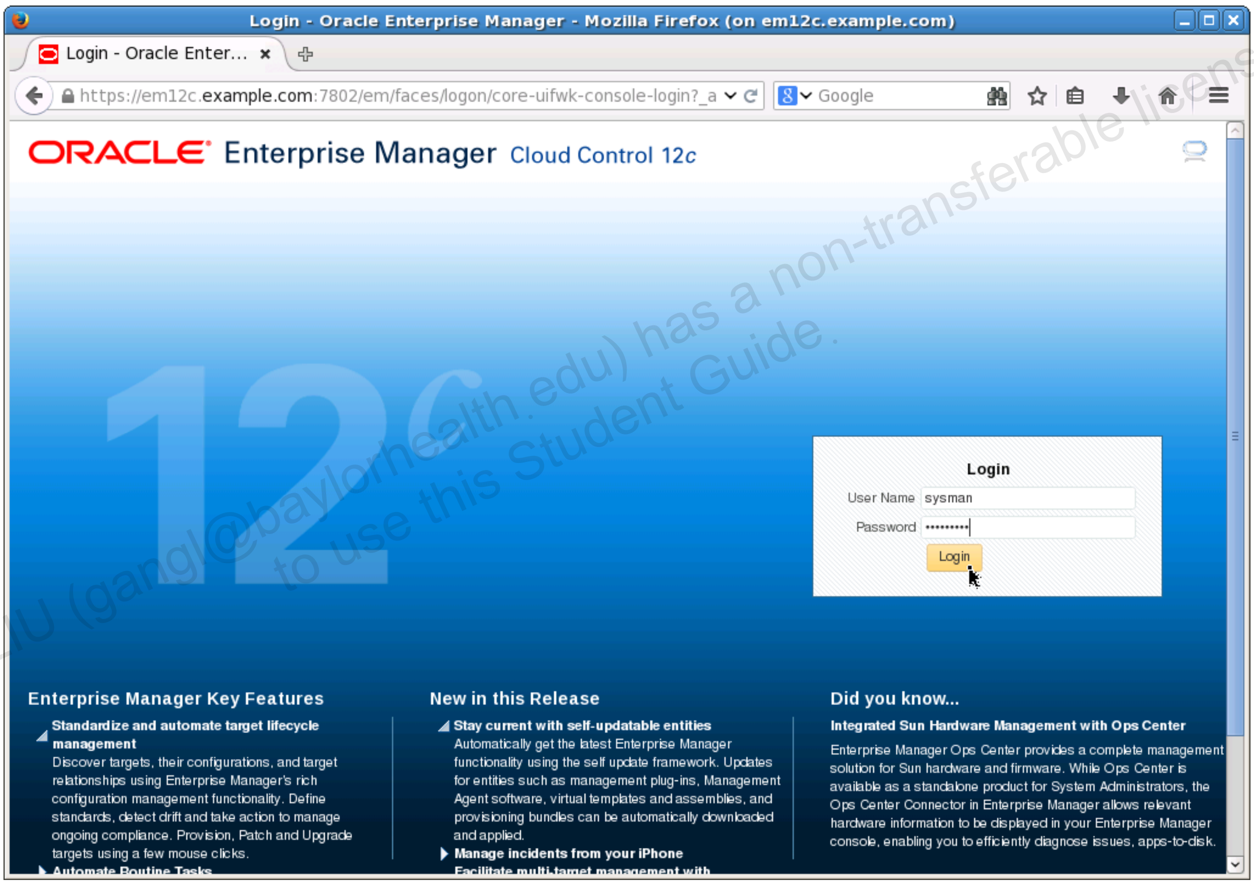
4. Establish another terminal session connected to `em12c` using the `oracle` OS user. Ensure that you specify the `-X` option for `ssh`.

```
$ ssh -X oracle@em12c
oracle@em12c password: ????????
[oracle@em12c ~]$
```

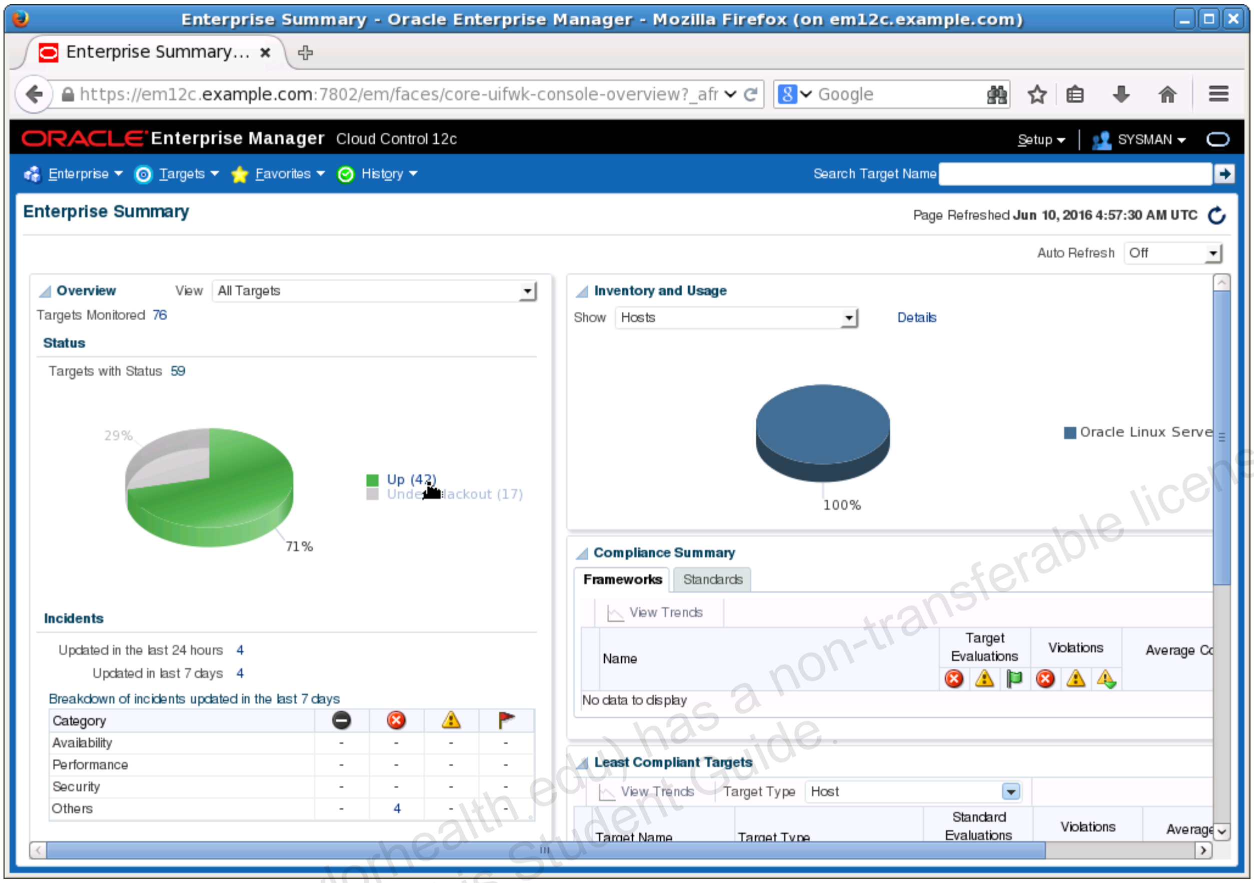
5. Start the Firefox web browser.

```
[oracle@em12c ~]$ firefox &
[1] 21592
```

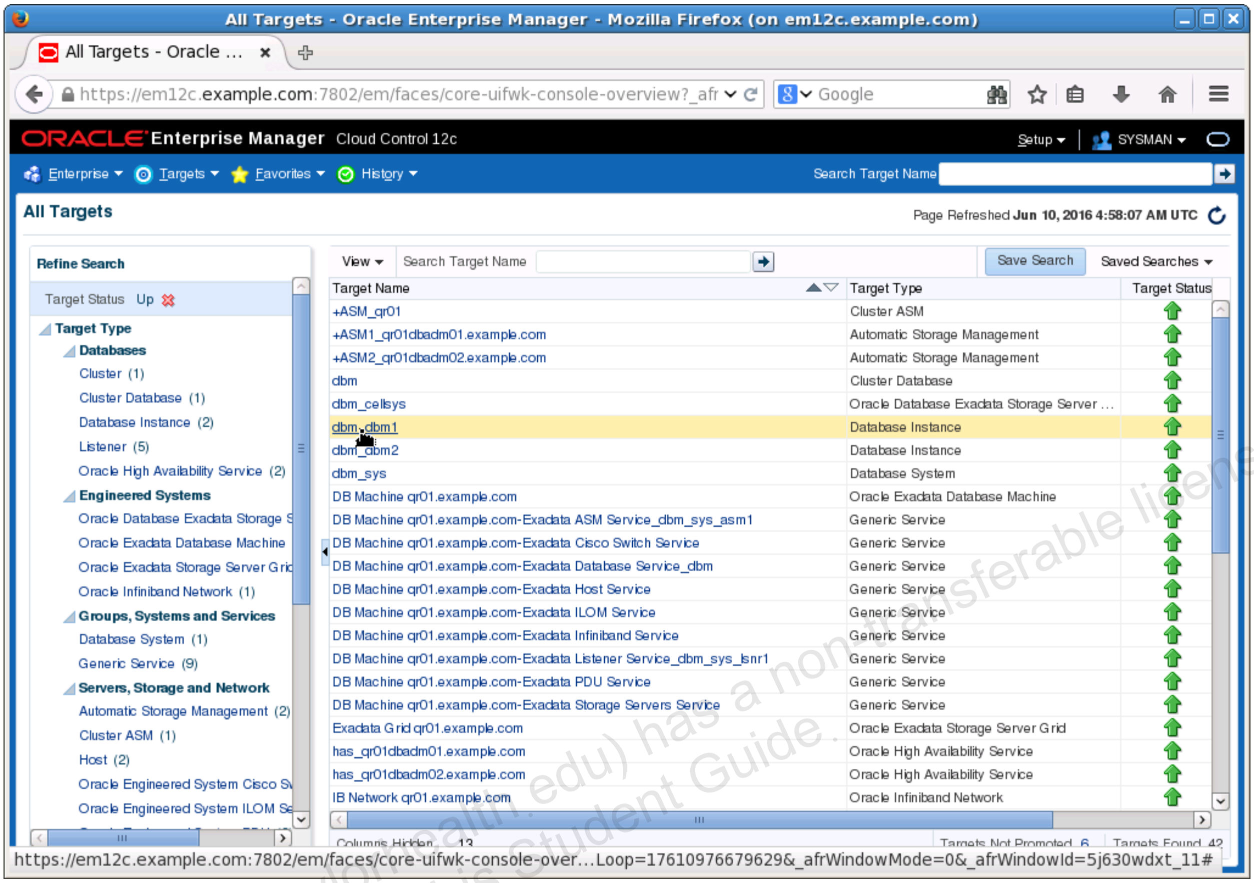
6. Log in to Enterprise Manager Cloud Control 12*c* as the `sysman` user.



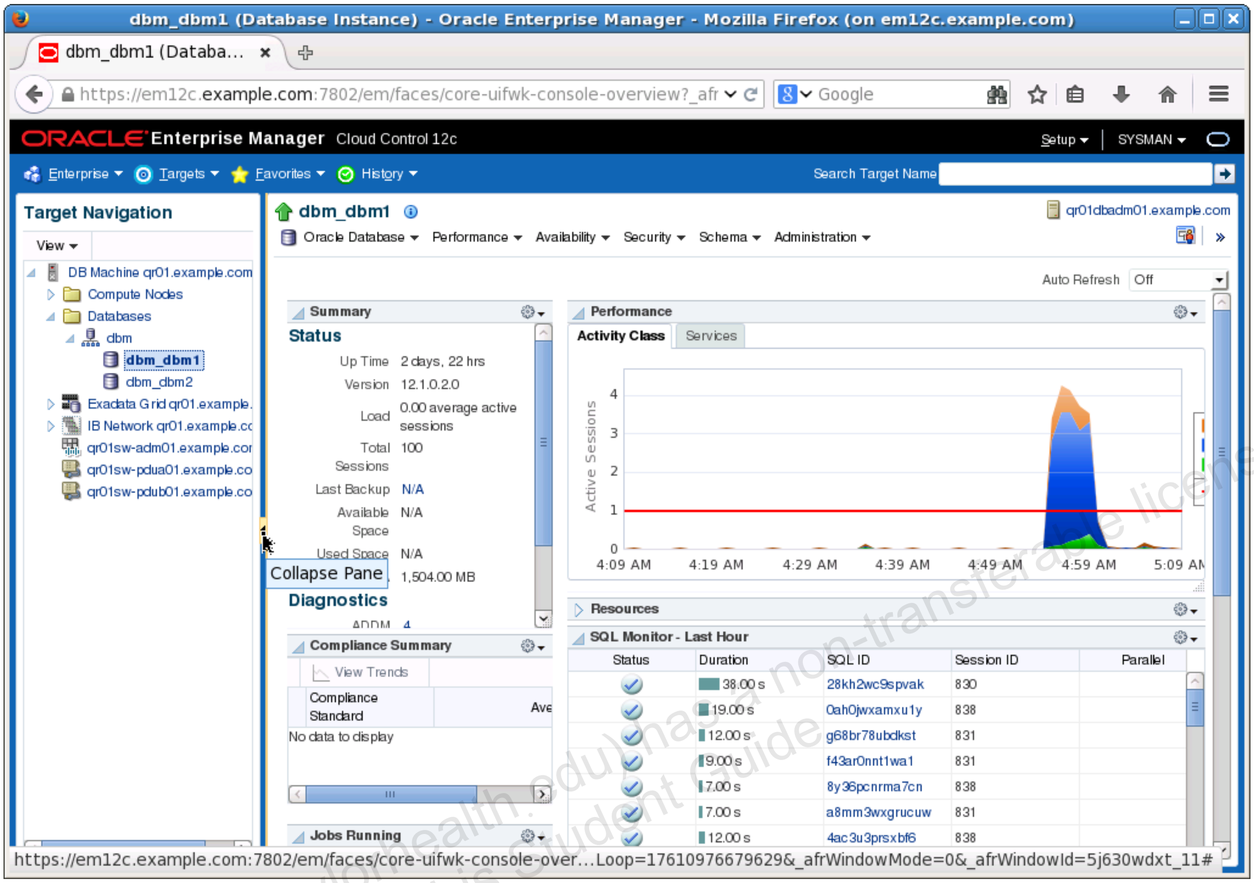
7. On the Enterprise Summary page, click the Up legend entry beside the target status chart.



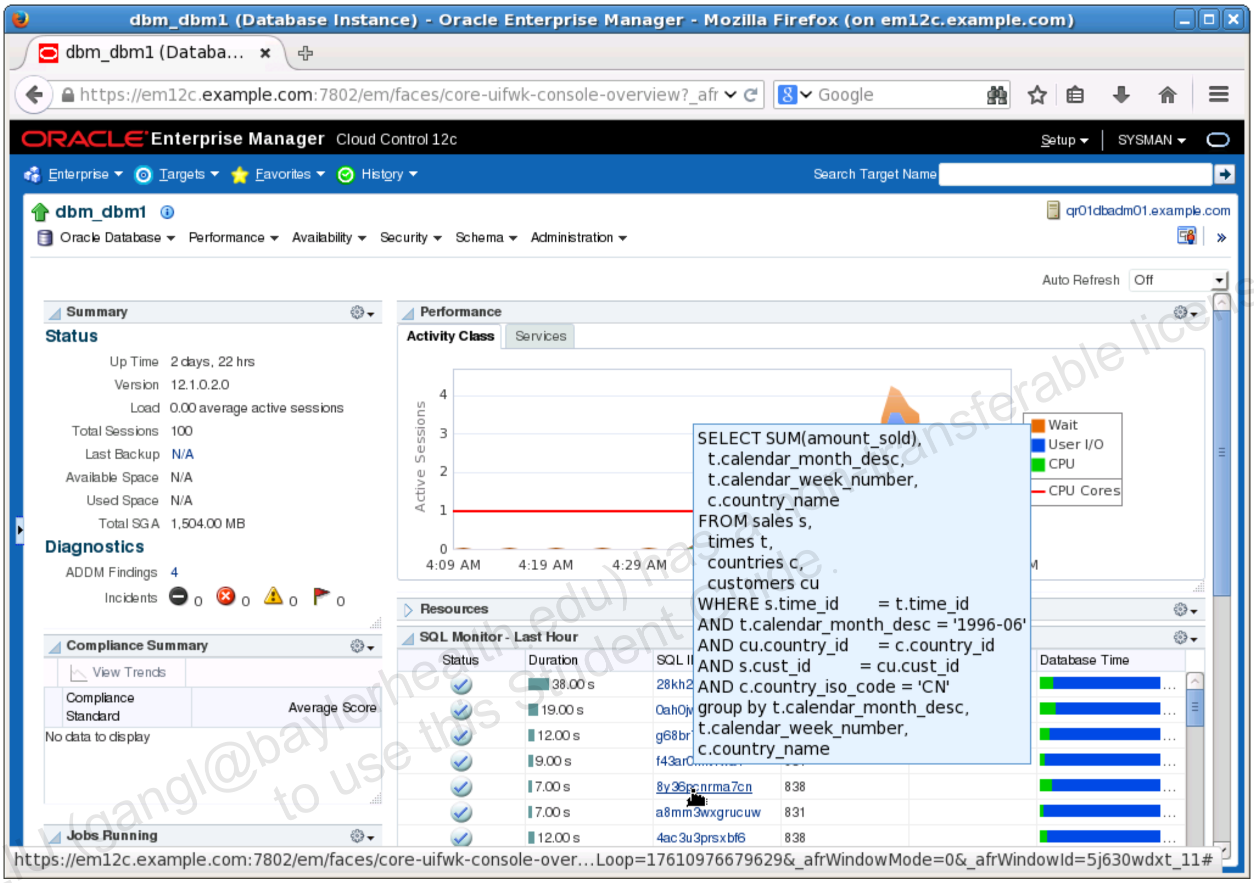
8. On the resulting target list, click the link associated with the database instance `dbm_dbm1`.



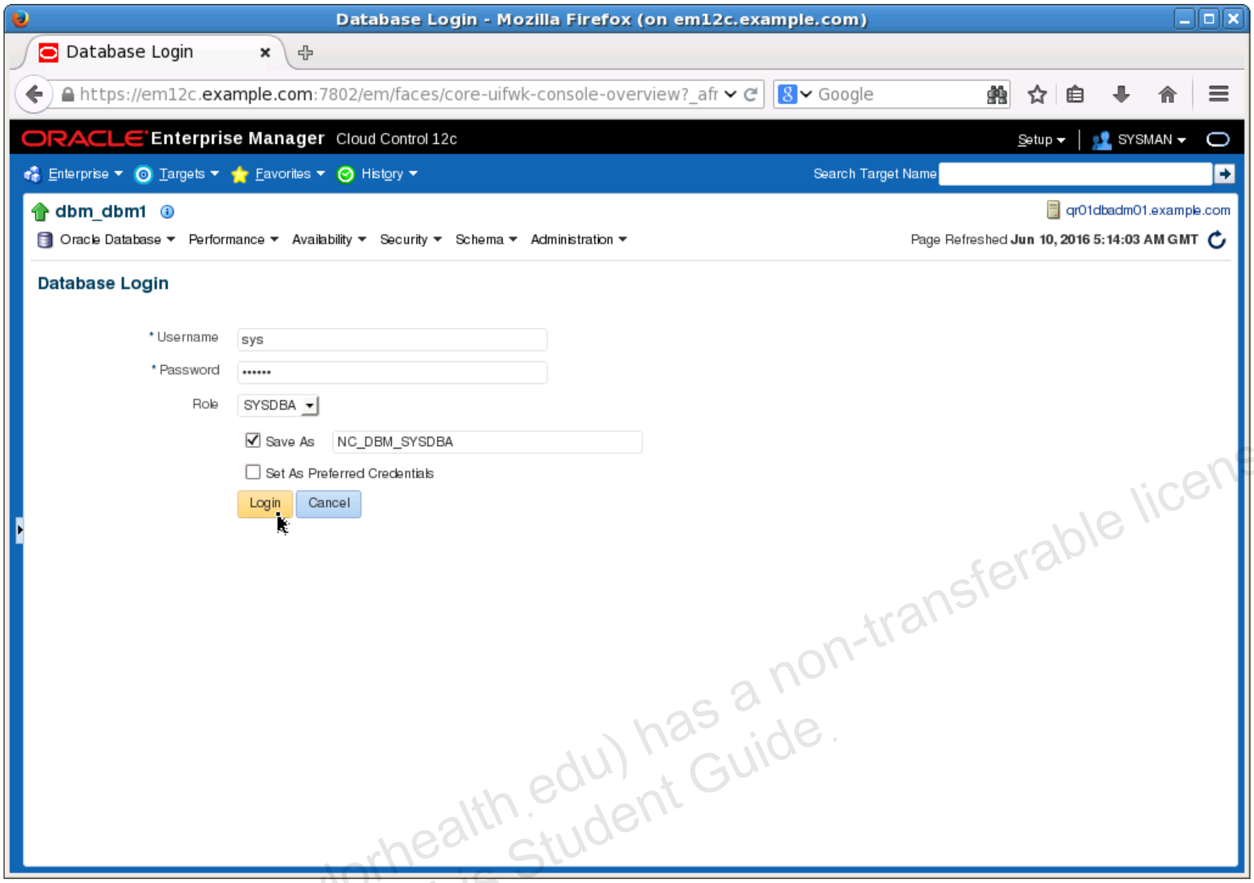
9. On the database instance home page for `dbm_dbm1`, collapse the Target Navigation pane.



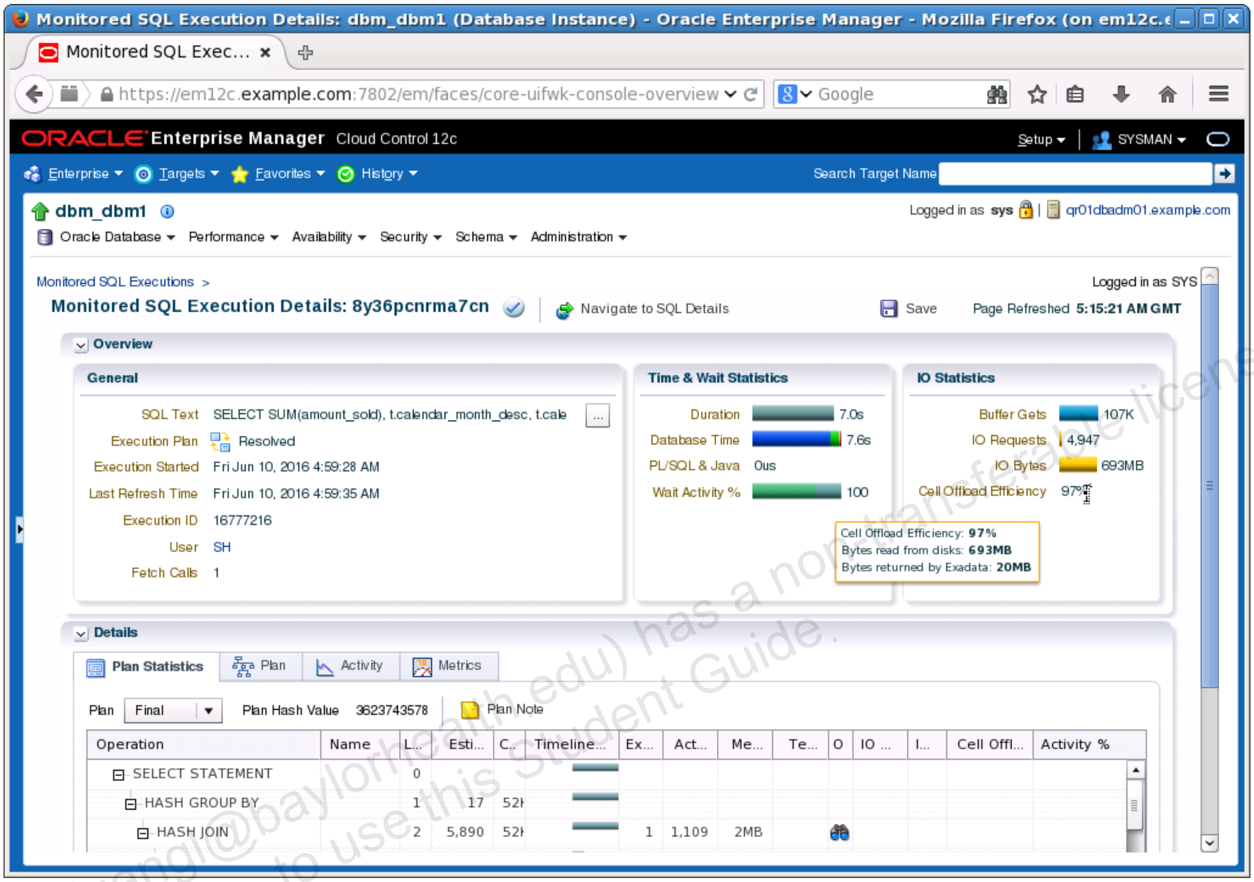
10. In the “SQL Monitor - Last Hour” area, you should see a list of SQL statements. Most of these statements should belong to the workload you executed at the beginning of this practice. Hover your mouse pointer over the SQL ID links in the list to reveal the SQL statement associated with each entry in the list. Find an interesting looking query that is based on the `sh` schema you have been using throughout all the practices in this course. Click the link associated with your selected query.



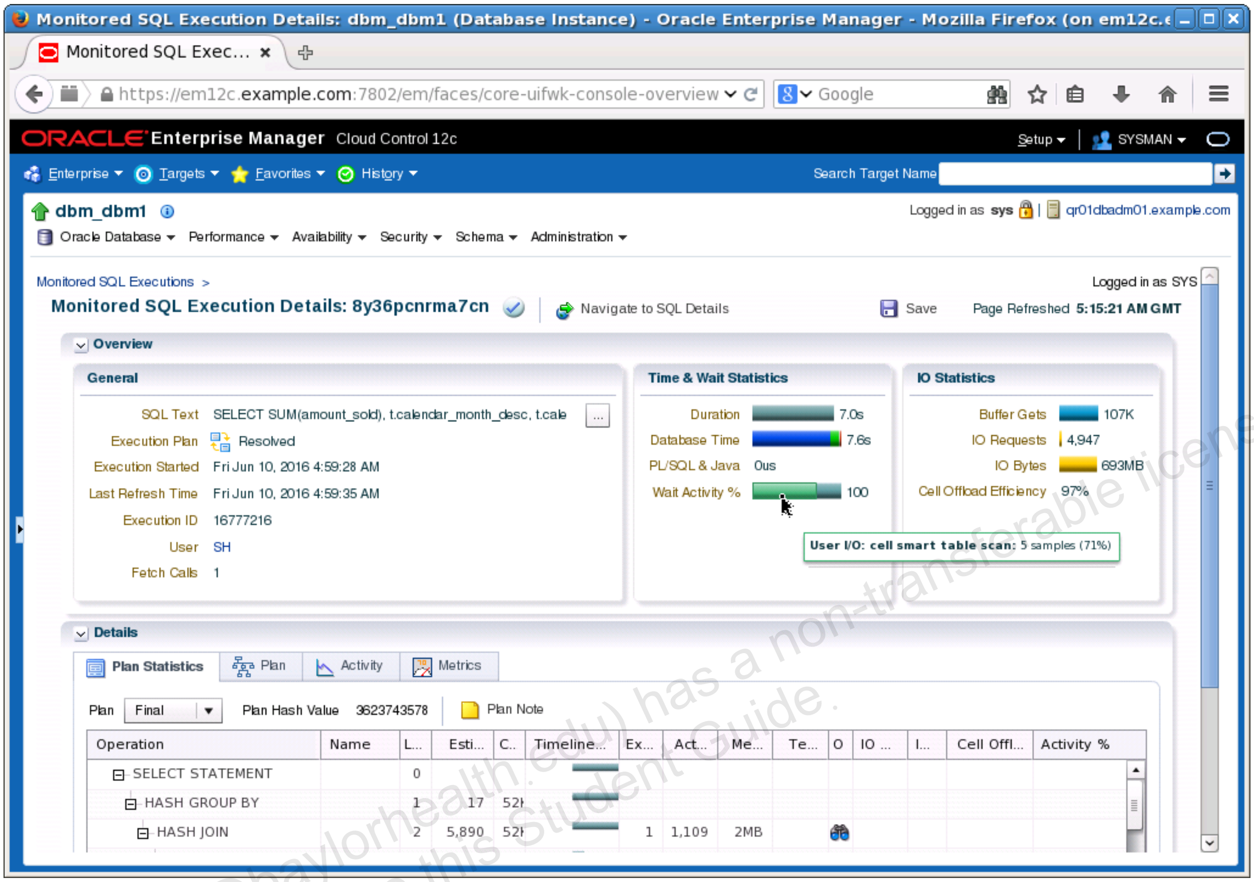
11. Login to the database as the `sys` user.



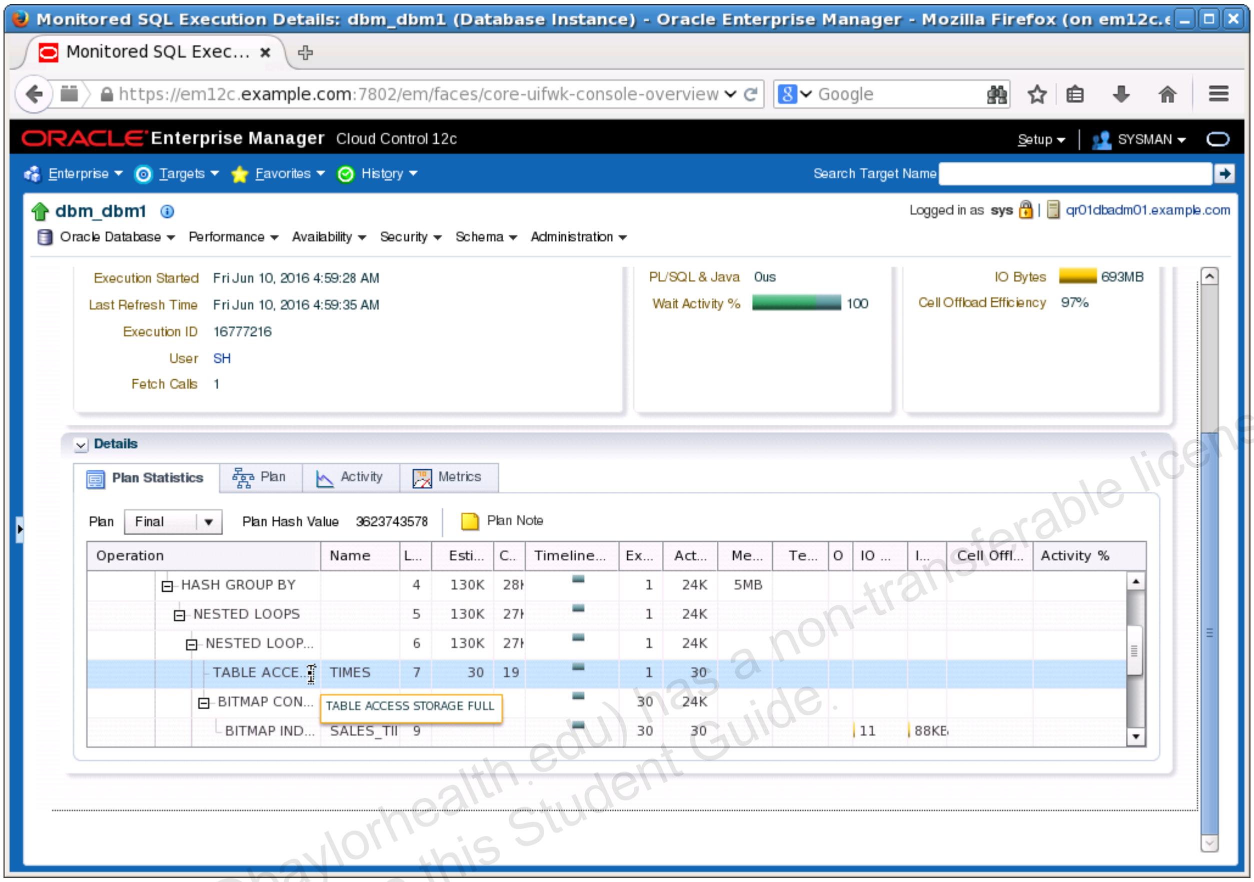
12. On the Monitored SQL Execution Details page, you will see detailed information relating to your selected query. When the database resides on Exadata, some additional Exadata-specific information is also presented. Notice the Cell Offload Efficiency statistic. Hover your mouse pointer over the Cell Offload Efficiency value to reveal more detailed information.



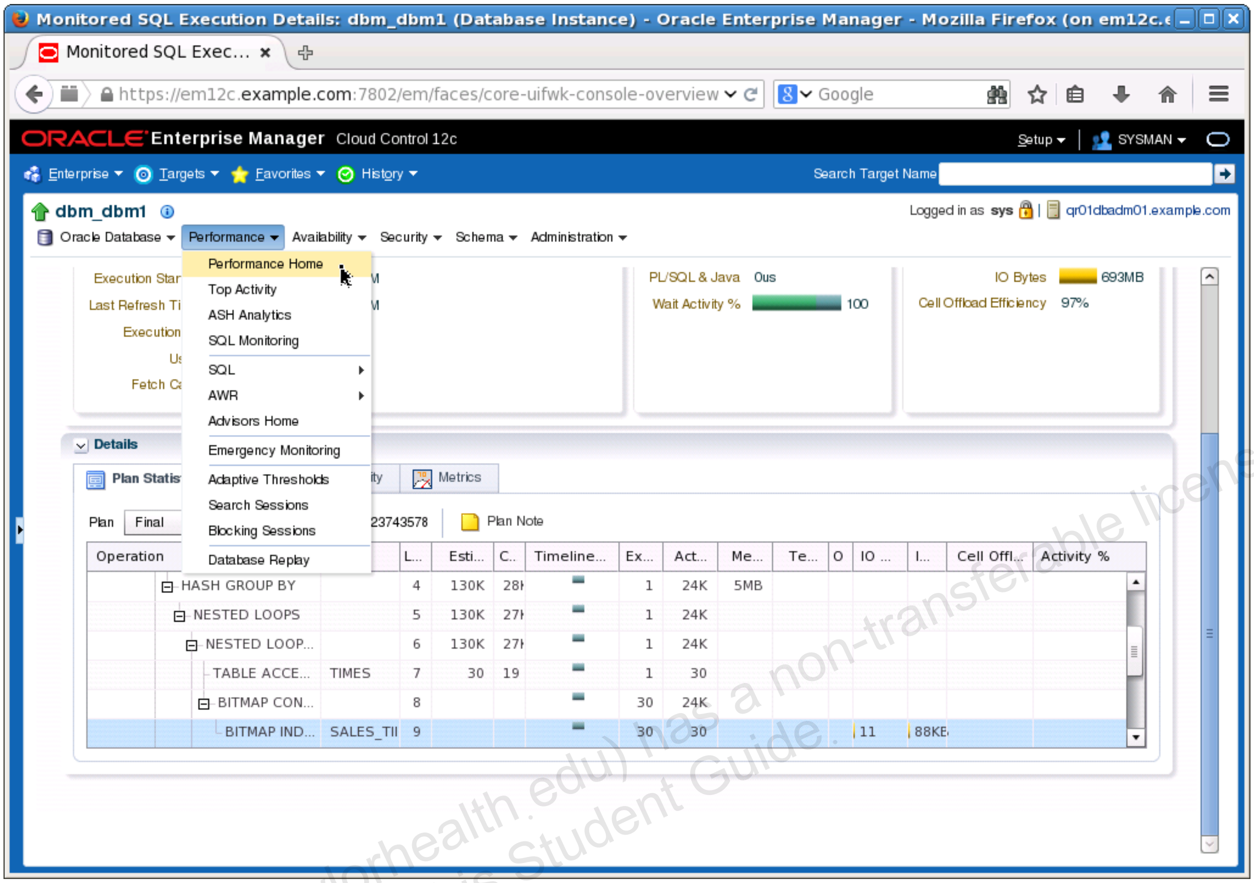
13. Hover your mouse pointer over the “Wait Activity %” bar and you should also see Exadata-specific wait event information.



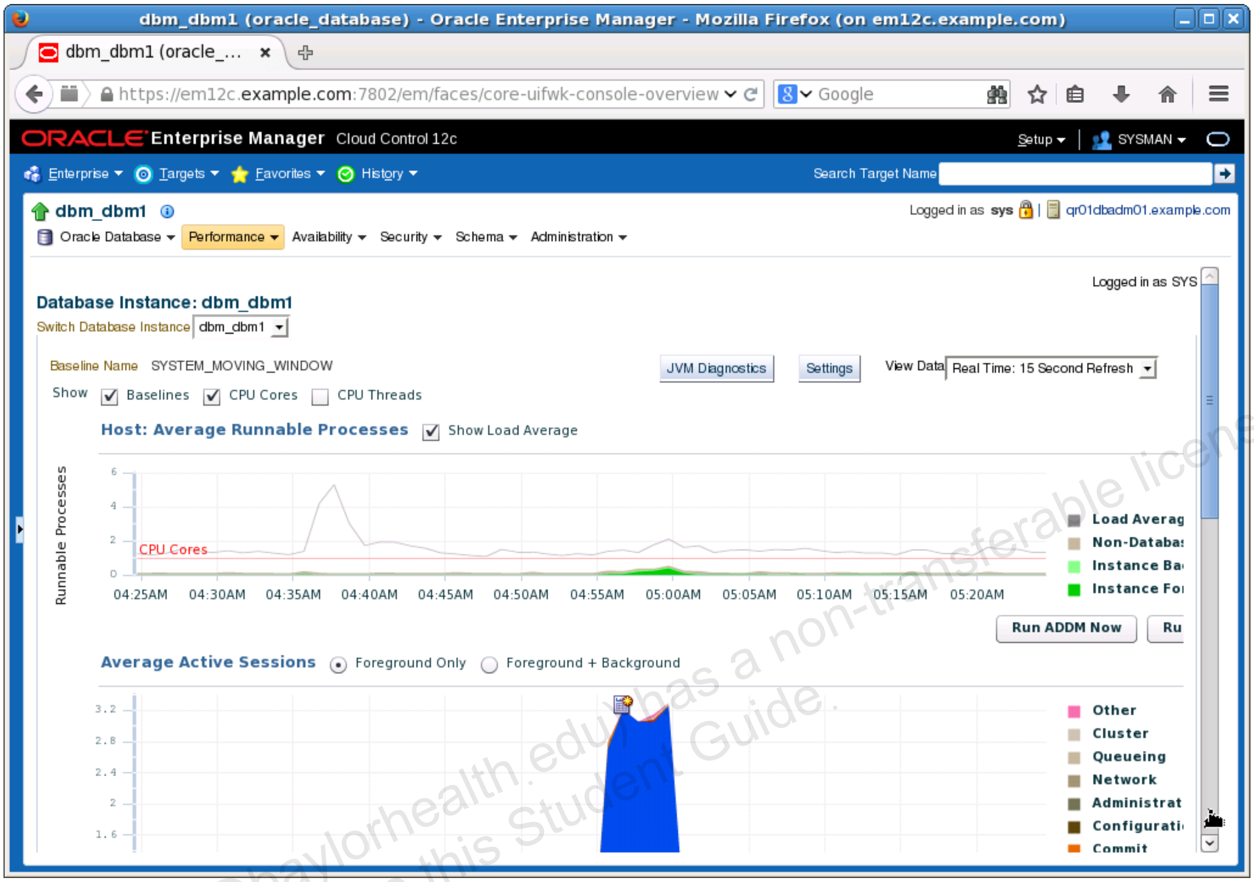
14. Examine the query execution plan and take note of the Smart Scan operations that are included within it.



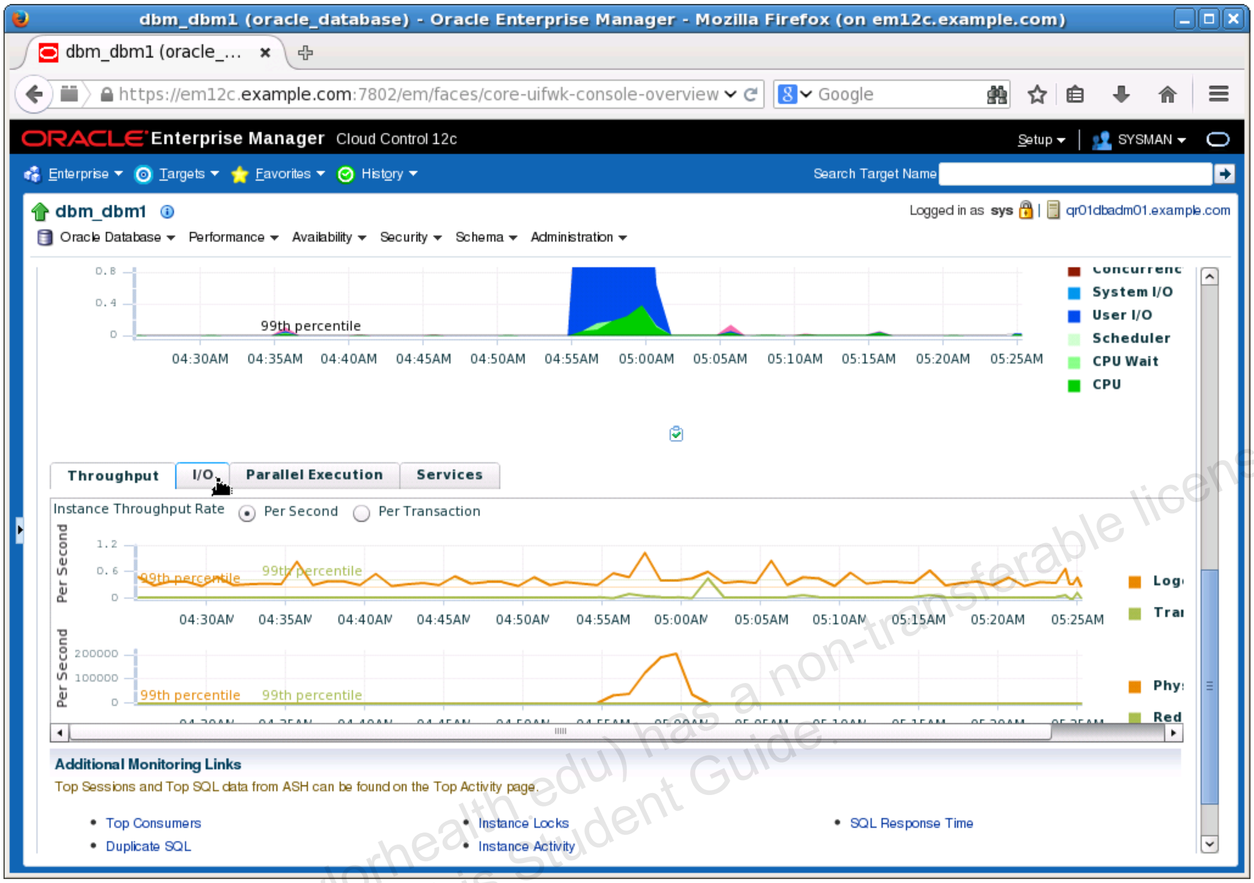
15. Select the Performance > Performance Home menu command.



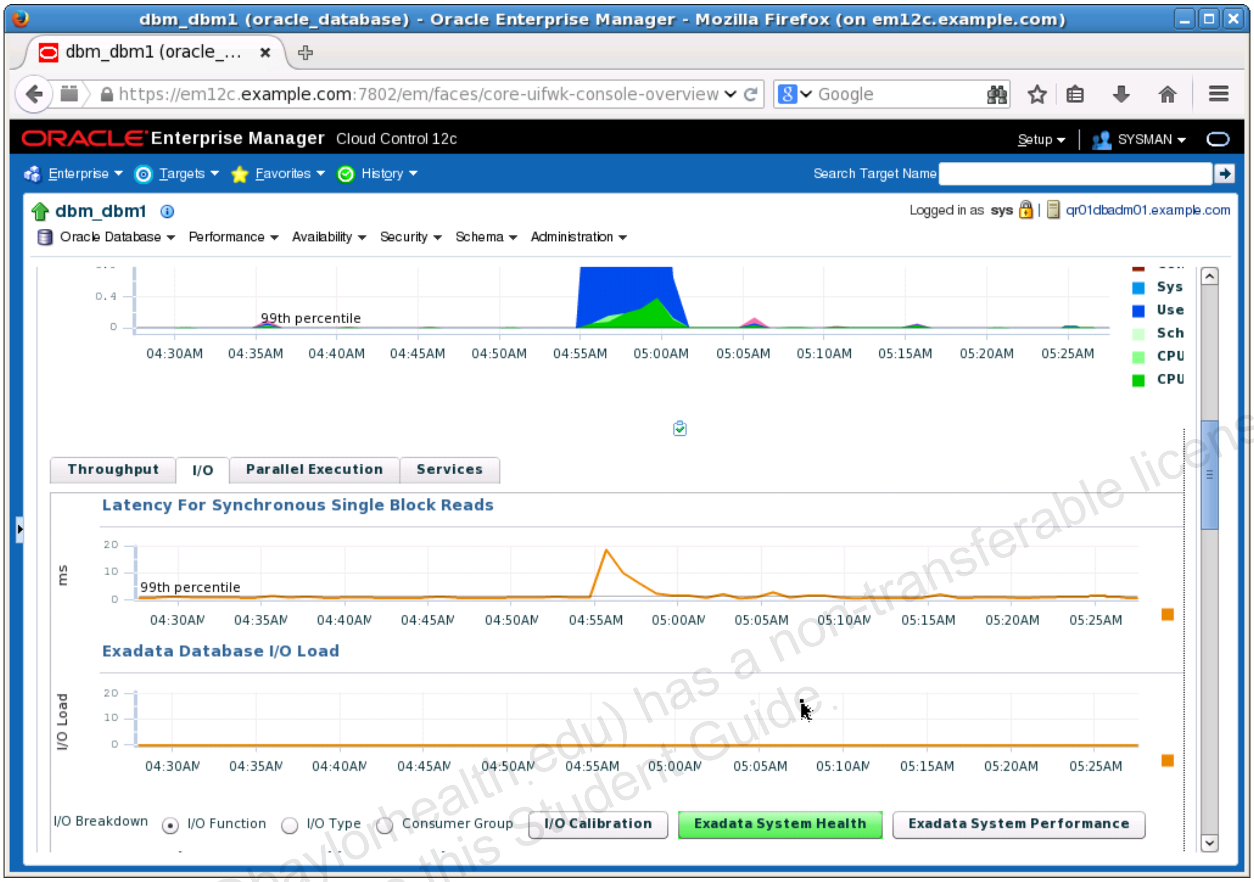
16. On the Database Instance performance home page, scroll down the page until you can see a series of tabs that include a tab labeled I/O.



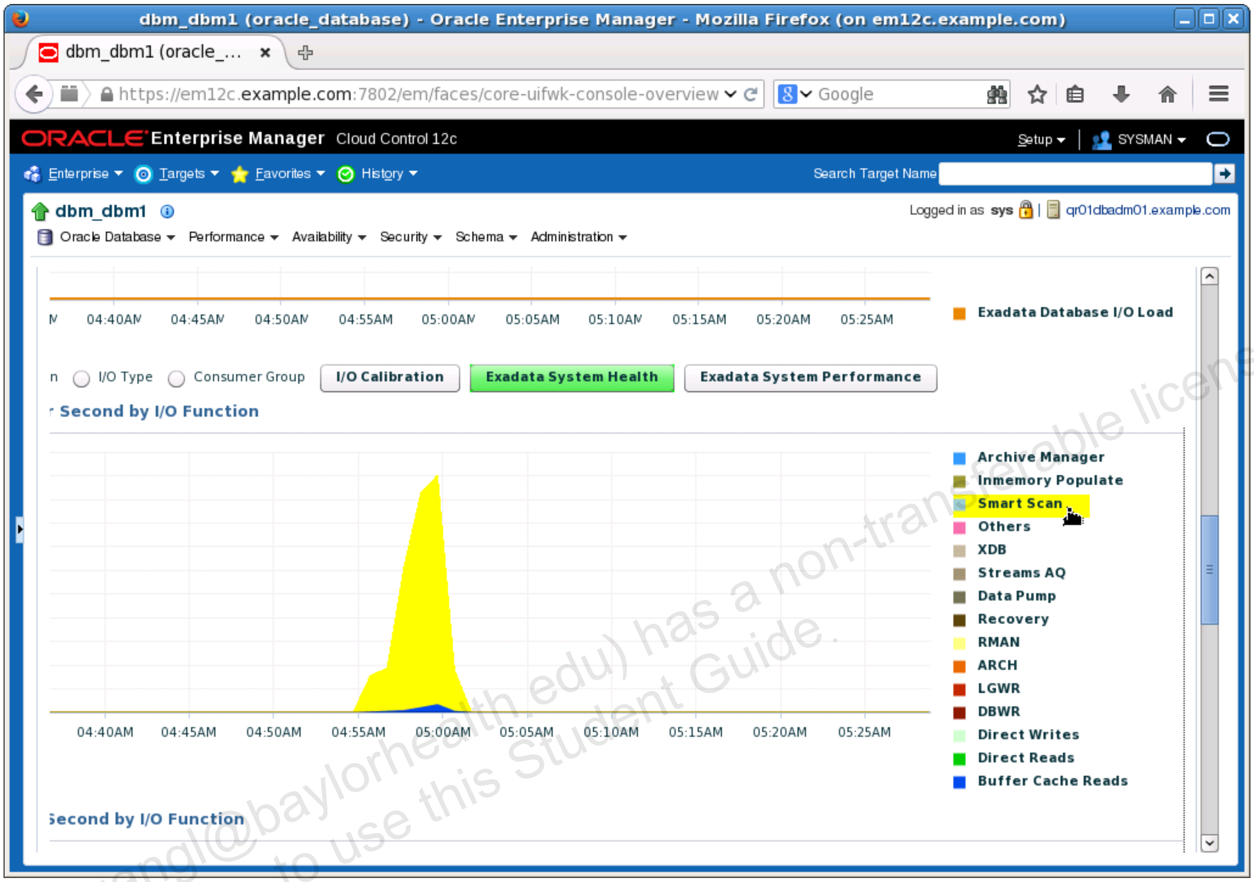
17. Click the I/O tab.



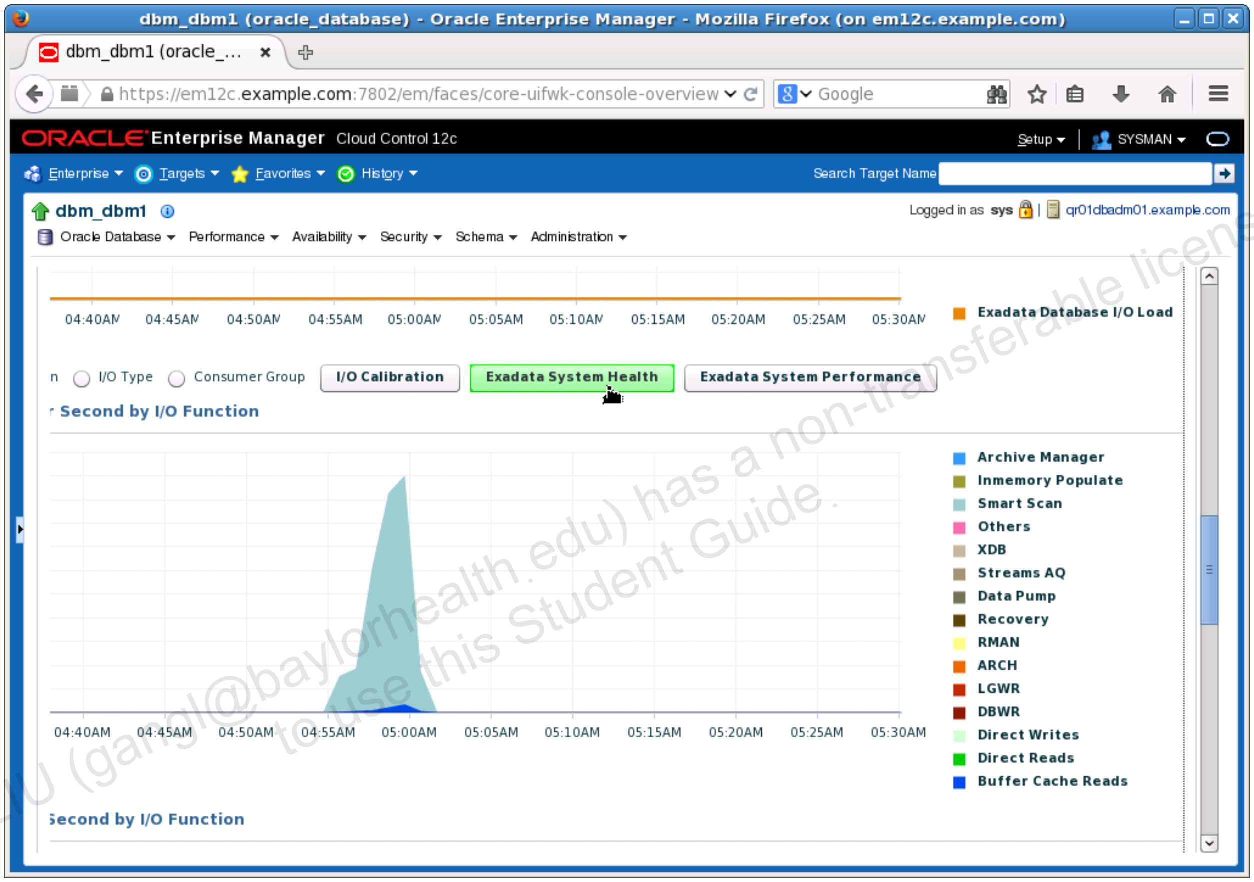
18. Notice that the IO performance information presented on the Database Instance performance page contains various Exadata-specific elements.



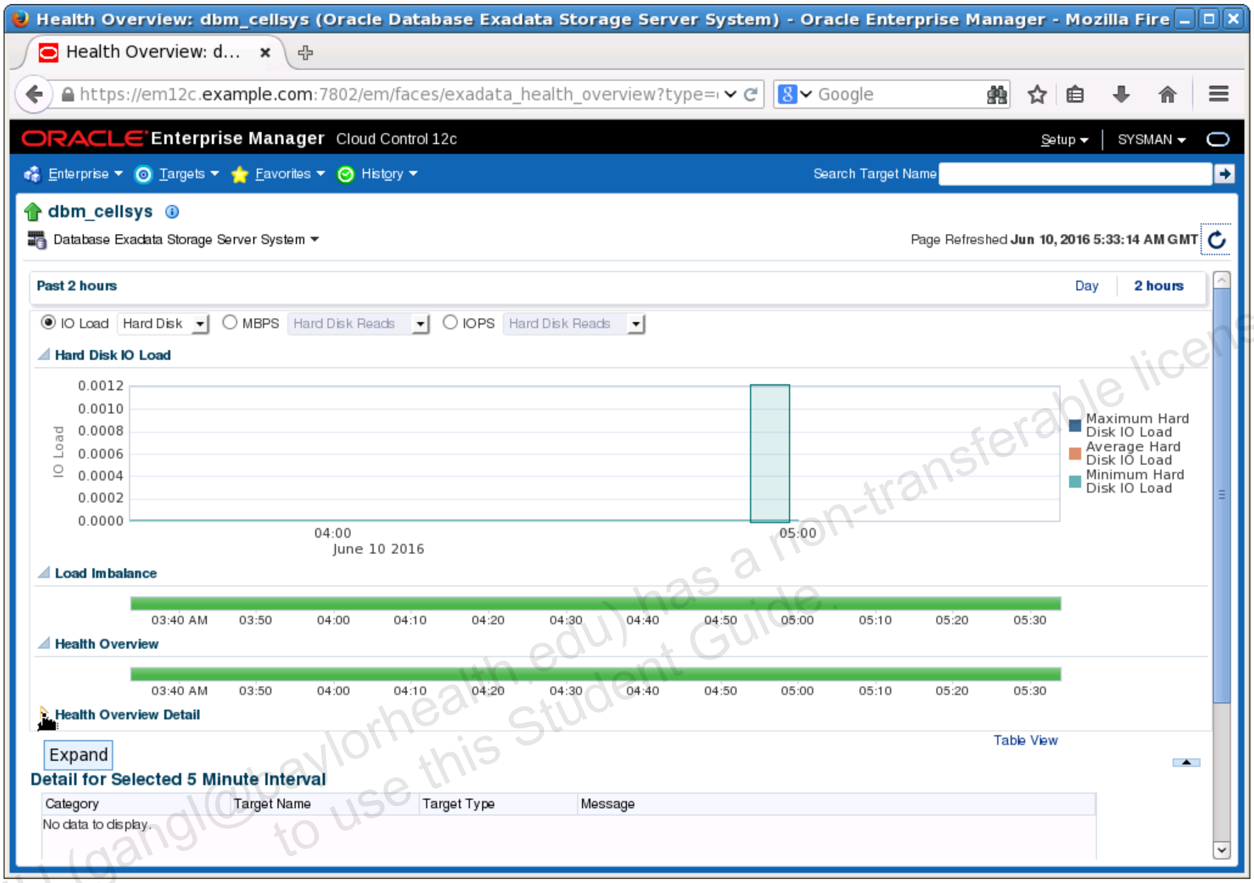
19. Hover your mouse pointer over the Smart Scan legend entry of the "I/O Megabytes per Second by I/O Function" chart. This will highlight the Smart Scan activity performed by the database instance and should clearly illustrate the amount of Smart Scan versus other IO operations on the instance.



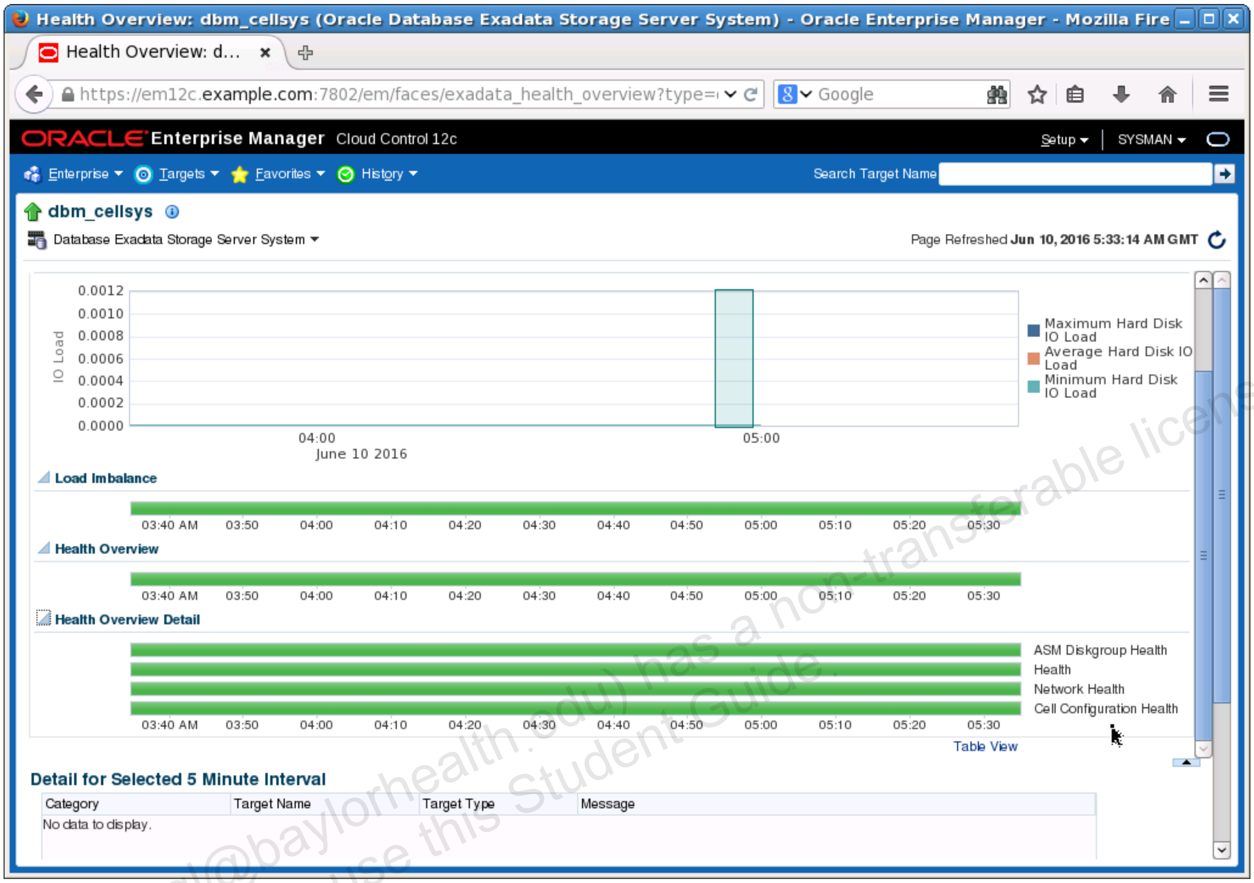
20. Notice also the two Exadata-specific buttons above the “I/O Megabytes per Second by I/O Function” chart.
The Exadata System Health button provides a visual indication of the health of the storage servers associated with the database. If the button appears green (as shown below) then the cells are in a healthy state.
The Exadata System Performance button can be used to navigate to a performance overview page for the storage servers associated with the database.
Click Exadata System Health.



21. Examine the Exadata System Health page. Because your laboratory environment isn't supporting a large user workload, you should expect to see mostly green bars in the timelines for Load Imbalance and Health Overview. These green bars indicate that the storage servers associated with your database are not reporting any problems. Expand the Health Overview Detail area.

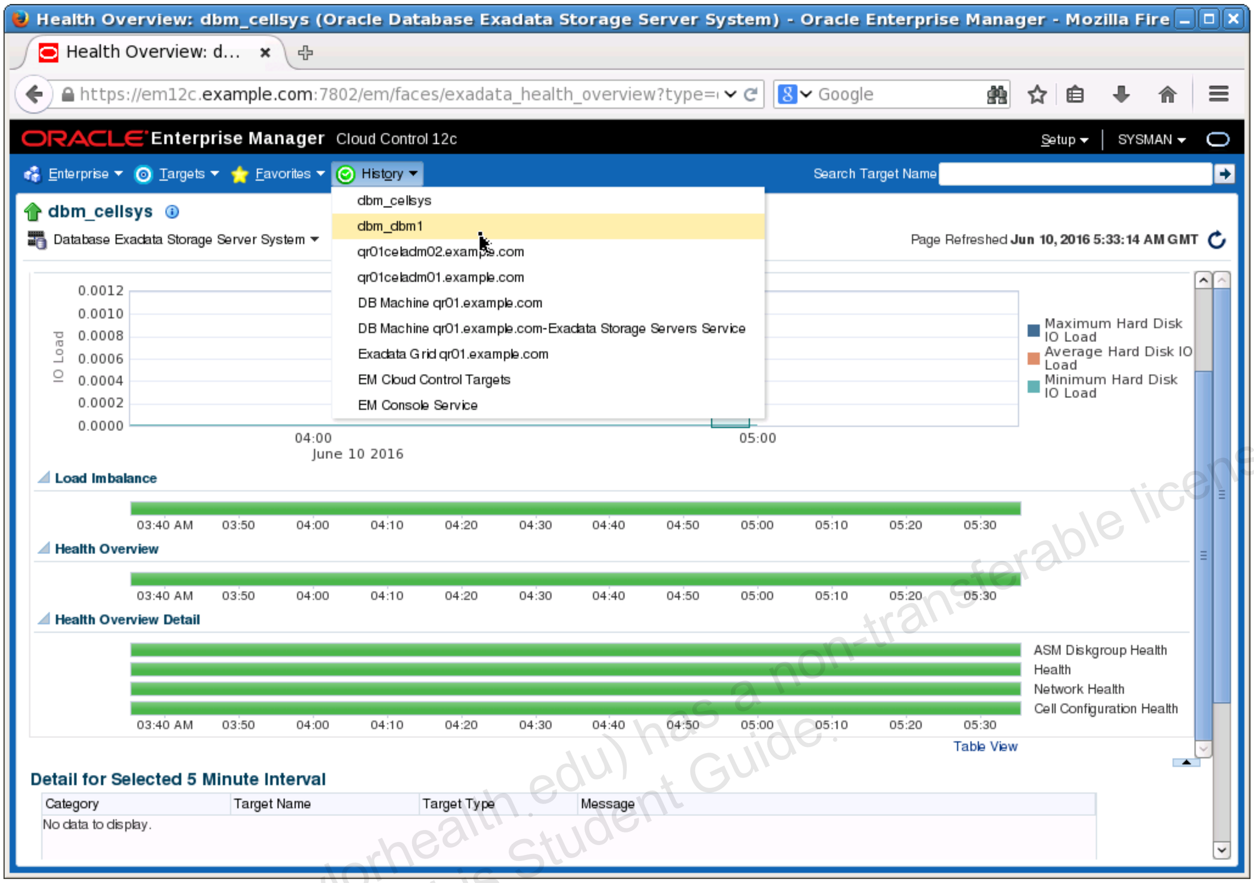


22. The Health Overview Detail area shows timelines and visual health indicators for key storage server components. Any area in these timelines that is not green indicates a warning or alert condition.

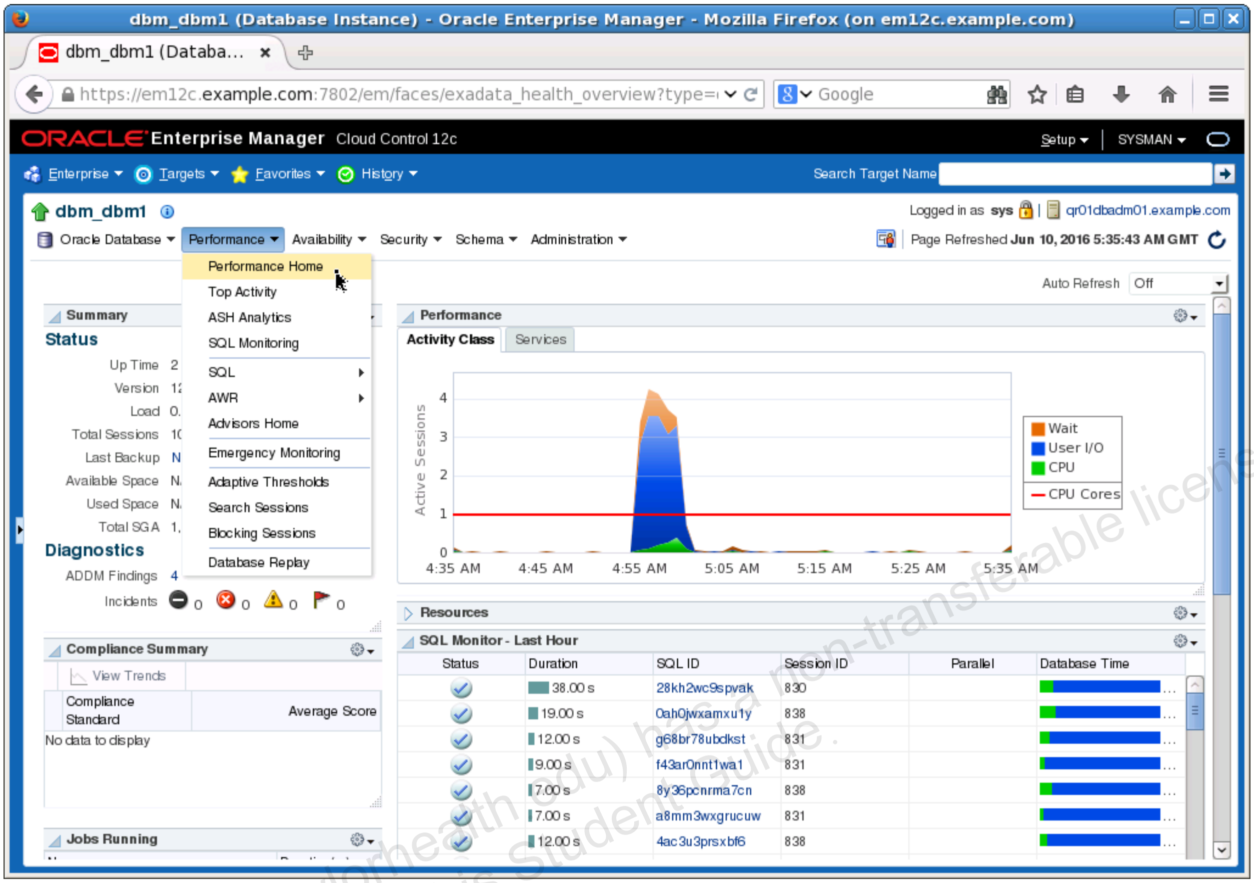


If you see any areas in the timelines that are not green, drag and drop the detail selector (the shaded box in the topmost chart, which is labeled Hard Disk IO Load by default) so that it covers the area of interest. Then look in the detail area (entitled Detail for Selected 5 Minute Interval by default) and examine the resulting messages to determine the reason for the condition.

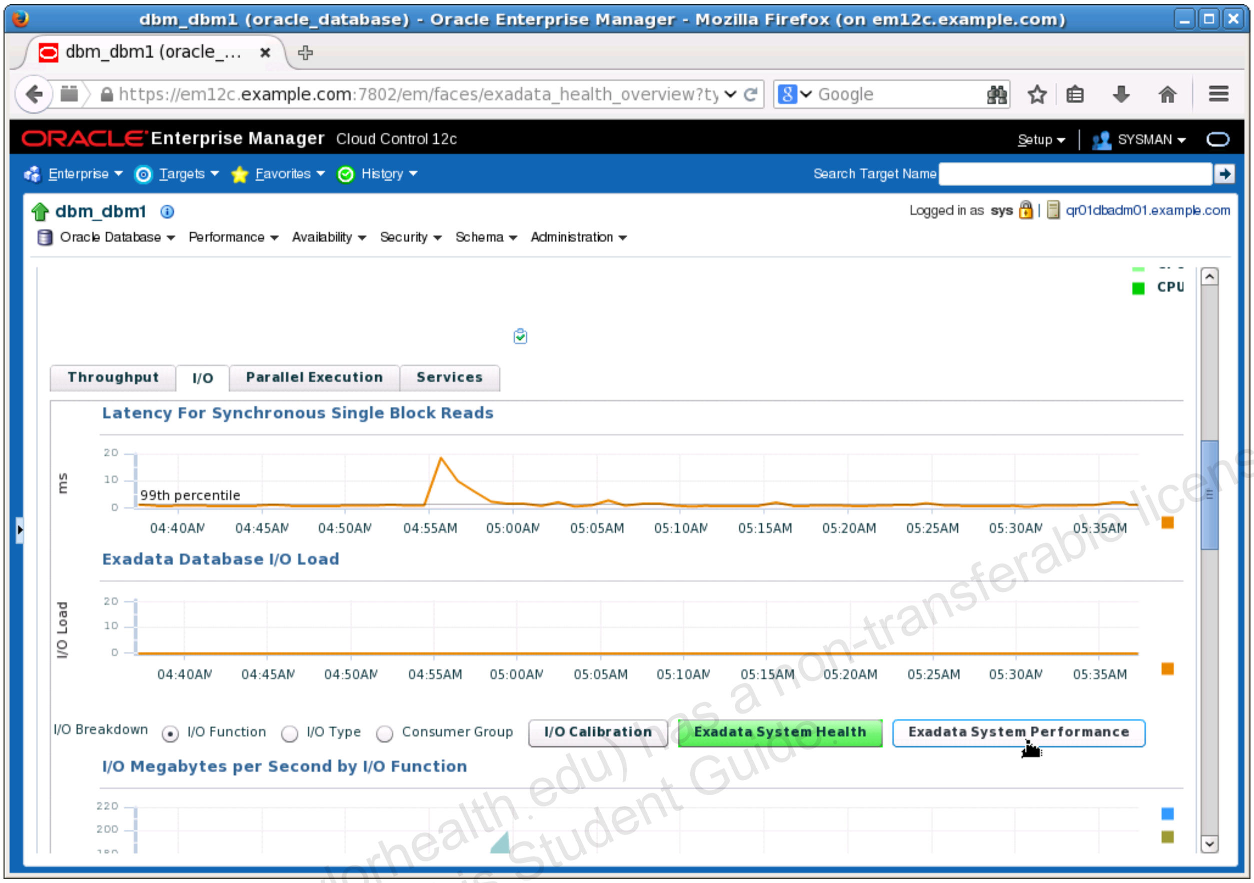
23. Use the History menu to navigate back to the database instance home page (`dbm_dbm1`).



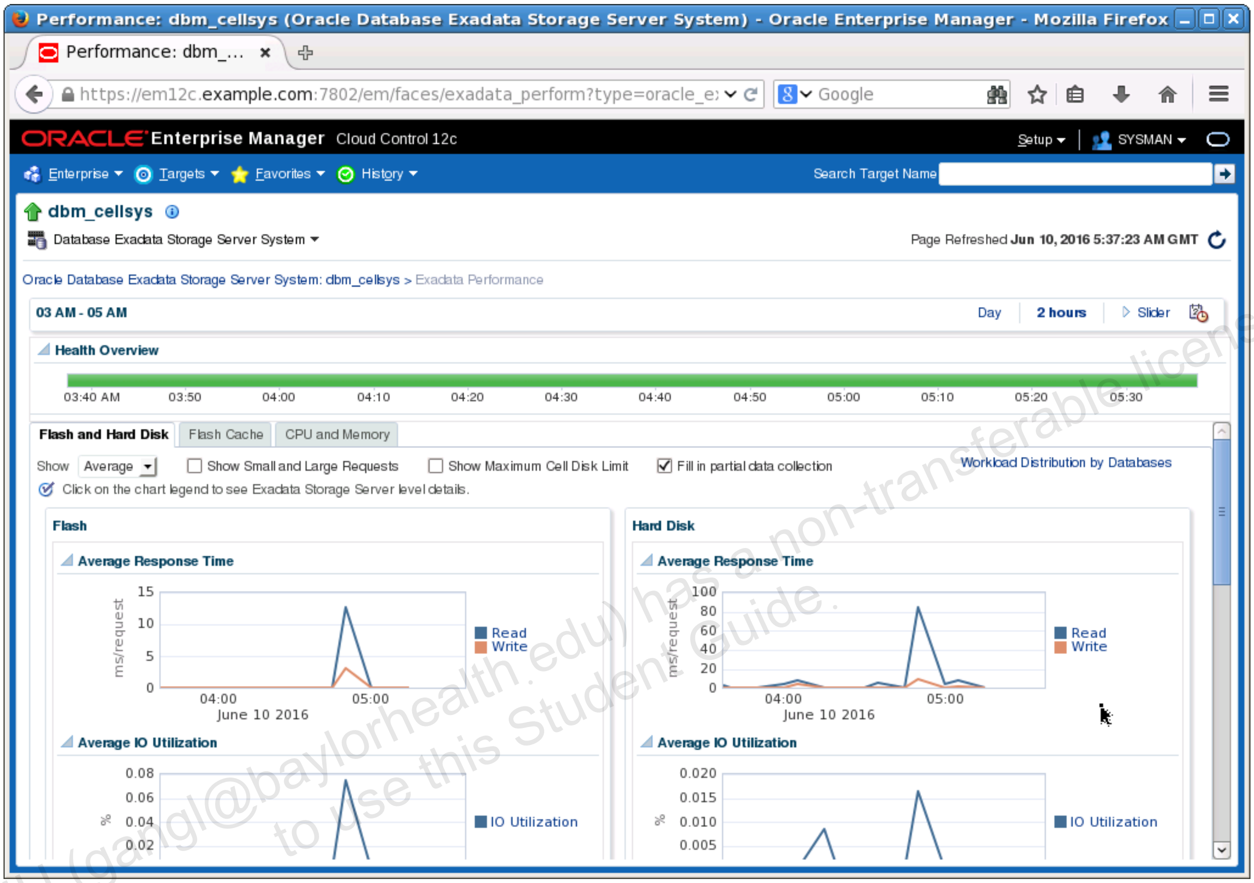
24. Select the Performance > Performance Home menu command.



25. Scroll down the database performance home page and click Exadata System Performance.



26. Here is the performance overview page for the storage servers associated with the database. It provides a series of charts that plot key Exadata Storage Server performance metrics over time for the storage servers associated with the database. It is very similar to the Exadata Storage Server Performance page that you examined in a previous practice. Take time to explore the page and the information it contains.



Congratulations! You have examined various Exadata-specific database monitoring capabilities provided by Enterprise Manager Cloud Control 12*c*.

Practices for Lesson 17: Monitoring the InfiniBand Network

Chapter 17

Practices for Lesson 17: Overview

Practices Overview

In this practice, you will examine the Exadata InfiniBand monitoring and administration capabilities provided by Enterprise Manager Cloud Control 12*c*.

Practice 17-1: Exadata InfiniBand Monitoring with Enterprise Manager

Overview

In this practice, you will examine the Exadata InfiniBand monitoring and administration capabilities provided by Enterprise Manager Cloud Control 12*c*.

Assumptions

The practice relies on the configurations performed in Practice 14-1.

Tasks

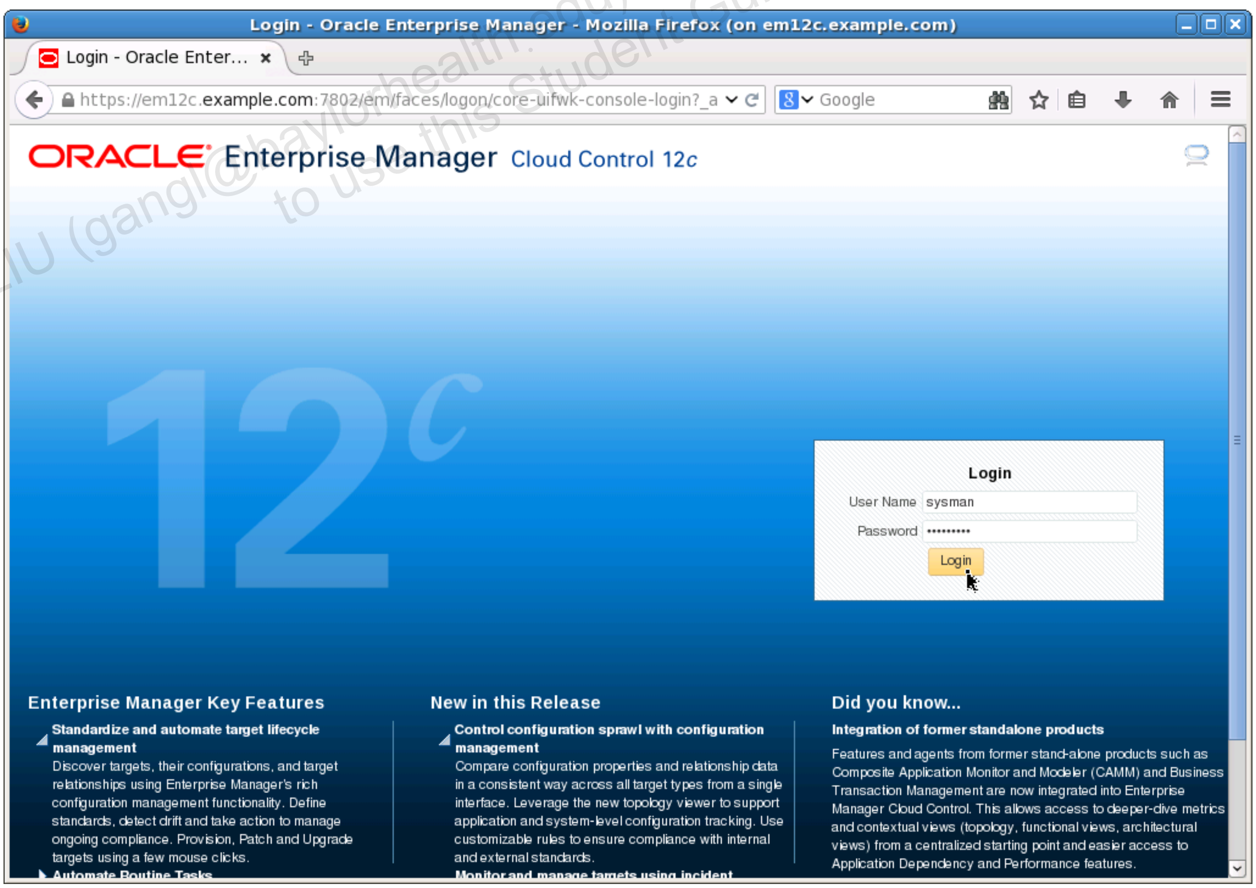
1. Establish a terminal session connected to `em12c` by using the `oracle` OS user. Ensure that you specify the `-X` option for `ssh`.

```
$ ssh -X oracle@em12c
oracle@em12c password: ????????
[oracle@em12c ~]$
```

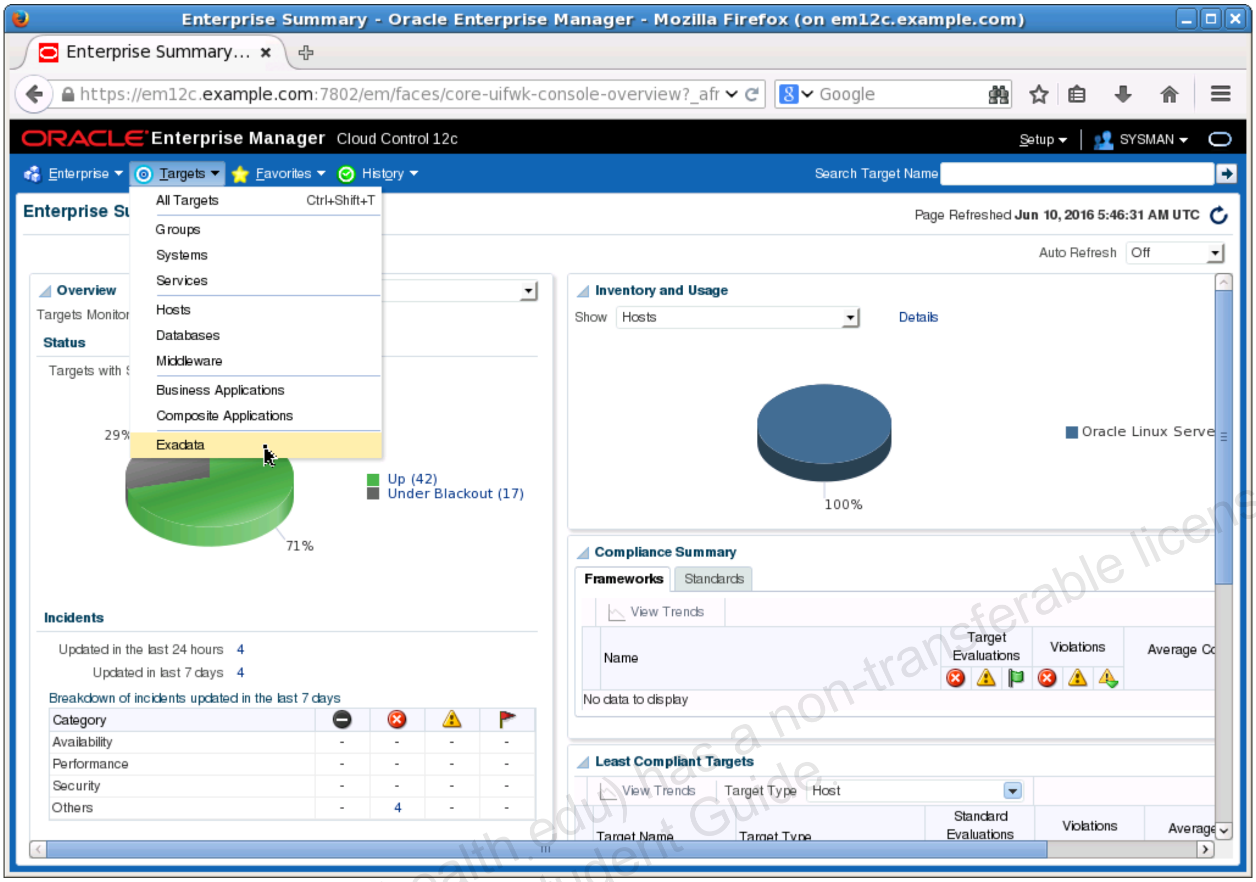
2. Start the Firefox web browser.

```
[oracle@em12c ~]$ firefox &
[1] 30345
```

3. Log in to Enterprise Manager Cloud Control 12*c* as the `sysman` user.

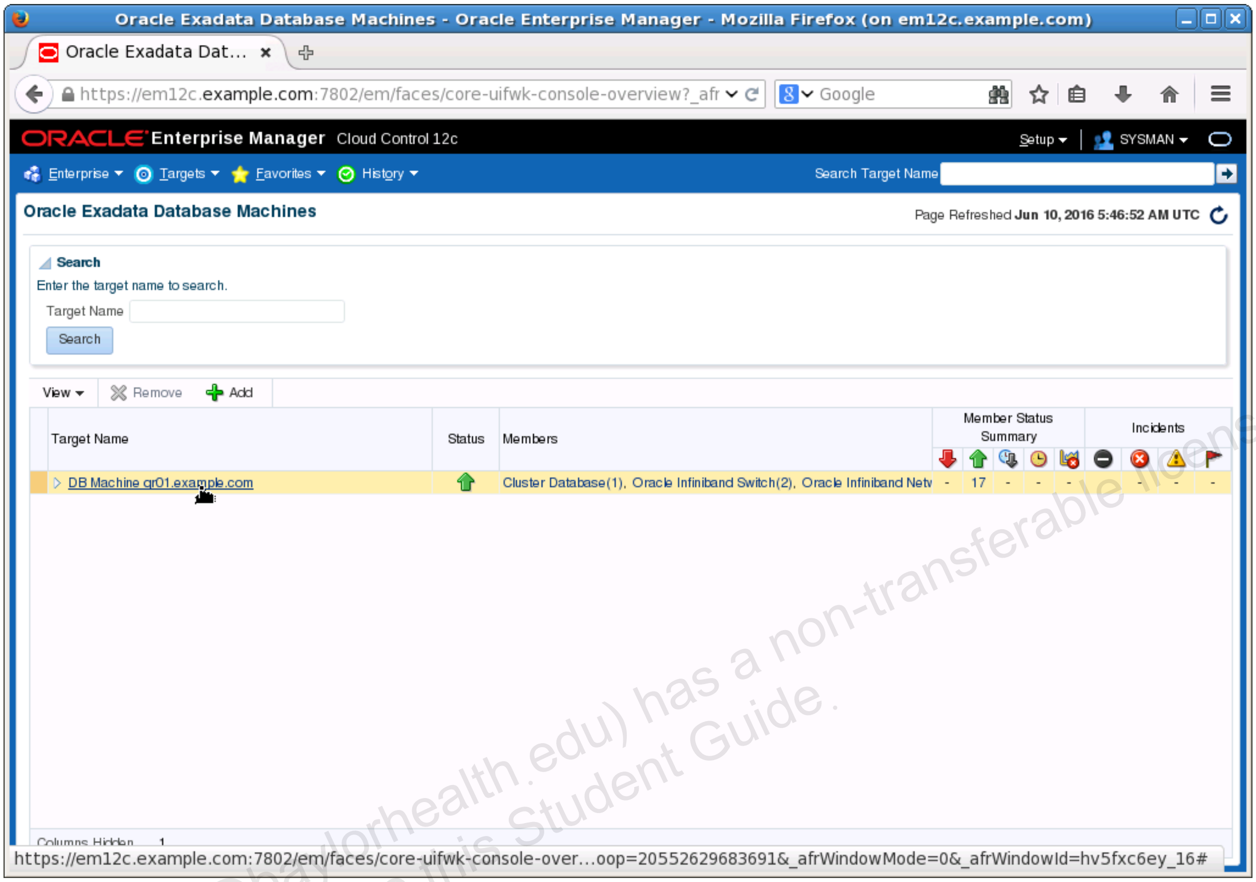


4. On the Enterprise Summary page, select the Targets > Exadata menu command.

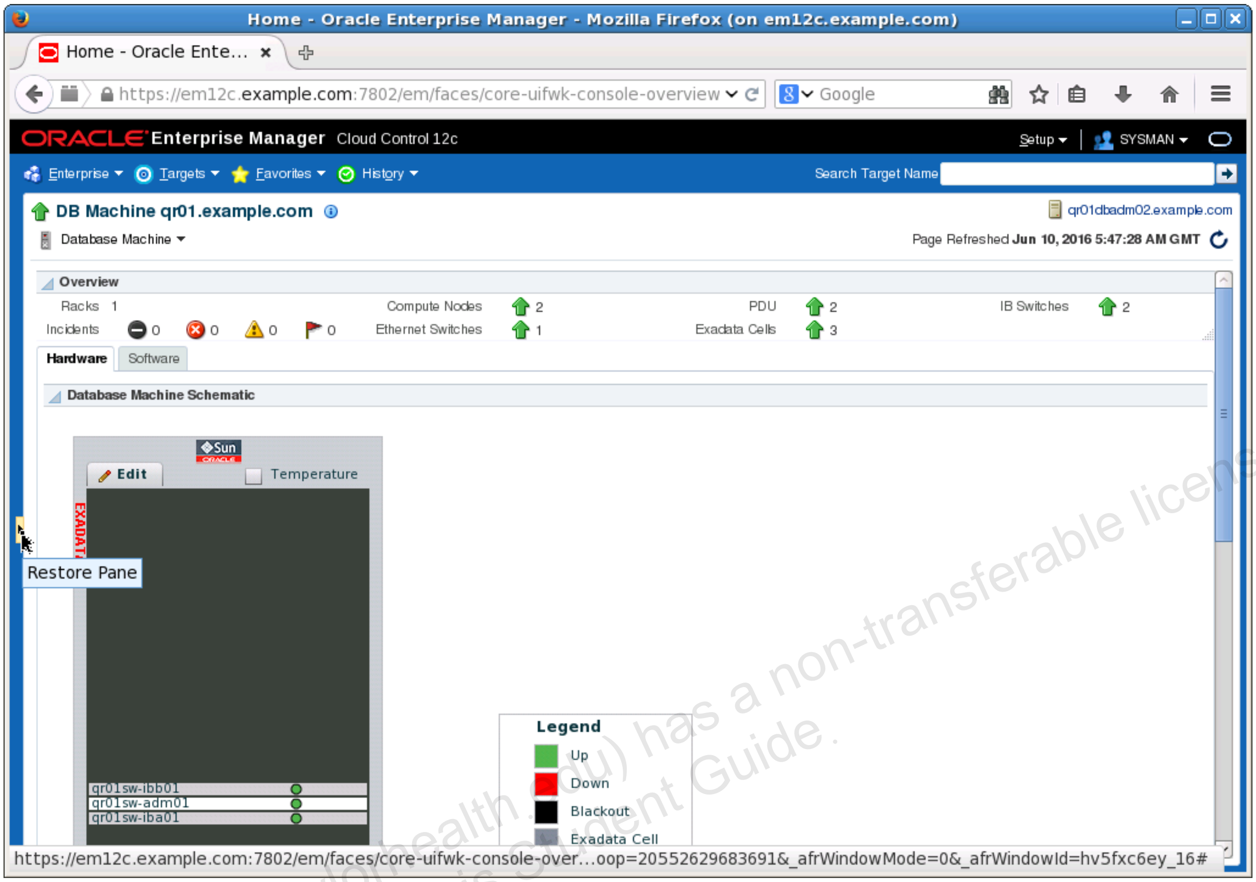


Enterprise Summary - Oracle Enterprise Manager - Mozilla Firefox (on em12c.example.com)
Enterprise Summary...
https://em12c.example.com:7802/em/faces/core-uifwk-console-overview?_afr
ORACLE Enterprise Manager Cloud Control 12c
Setup
SYSMAN
Enterprise
Targets
Favorites
History
Search Target Name
All Targets Ctrl+Shift+T
Groups
Systems
Services
Hosts
Databases
Middleware
Business Applications
Composite Applications
Exadata
Page Refreshed Jun 10, 2016 5:46:31 AM UTC
Auto Refresh Off
Overview
Targets Monitor
Status
Up (42)
Under Blackout (17)
71%
29%
Inventory and Usage
Show Hosts
Details
Oracle Linux Serve
100%
Compliance Summary
Frameworks
Standards
View Trends
Name
Target Evaluations
Violations
Average Co
No data to display
Incidents
Updated in the last 24 hours 4
Updated in last 7 days 4
Breakdown of incidents updated in the last 7 days
Category
Availability
Performance
Security
Others
Least Compliant Targets
View Trends
Target Type Host
Target Name
Target Type
Standard Evaluations
Violations
Average

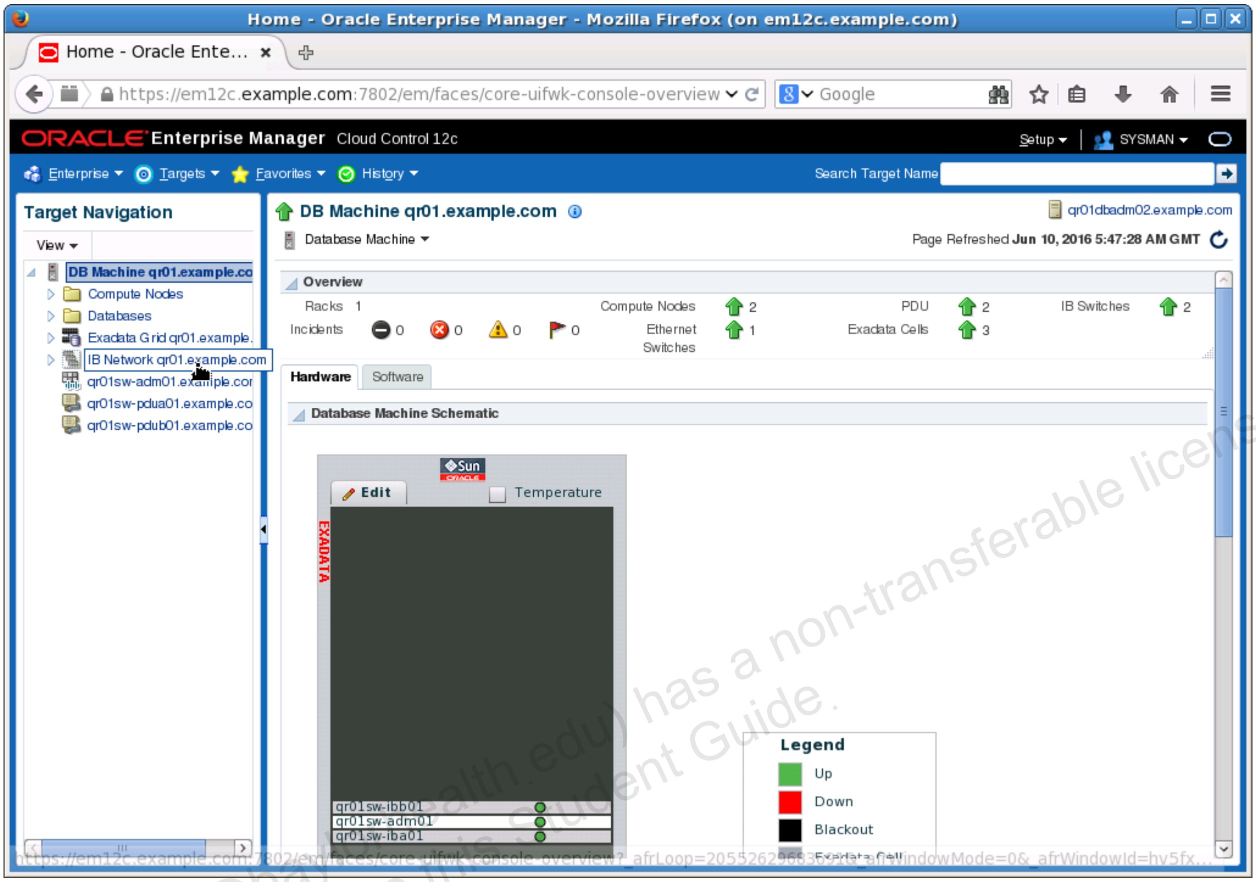
5. Click the link corresponding to your Exadata Database Machine (DB Machine qr01.example.com).



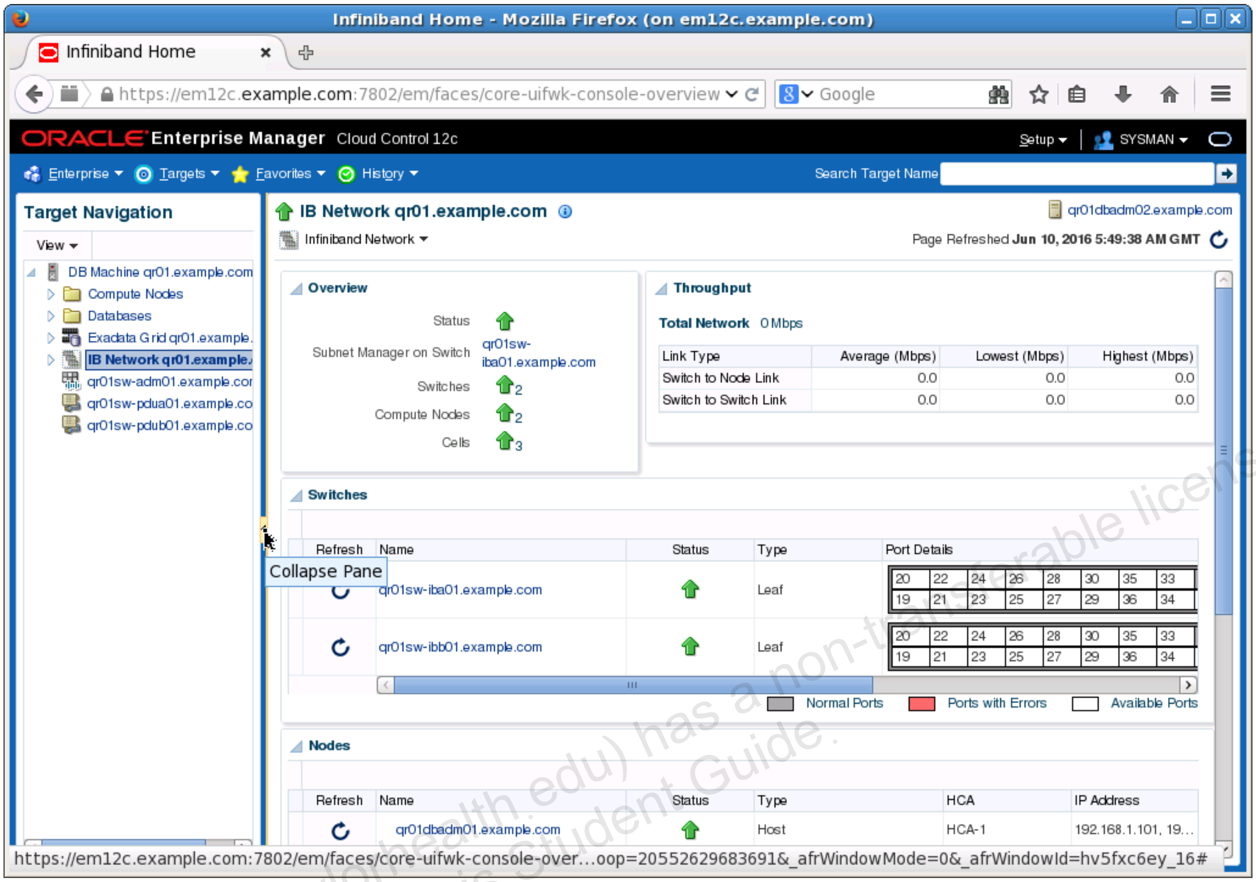
6. On the Database Machine home page, restore the Target Navigation pane.



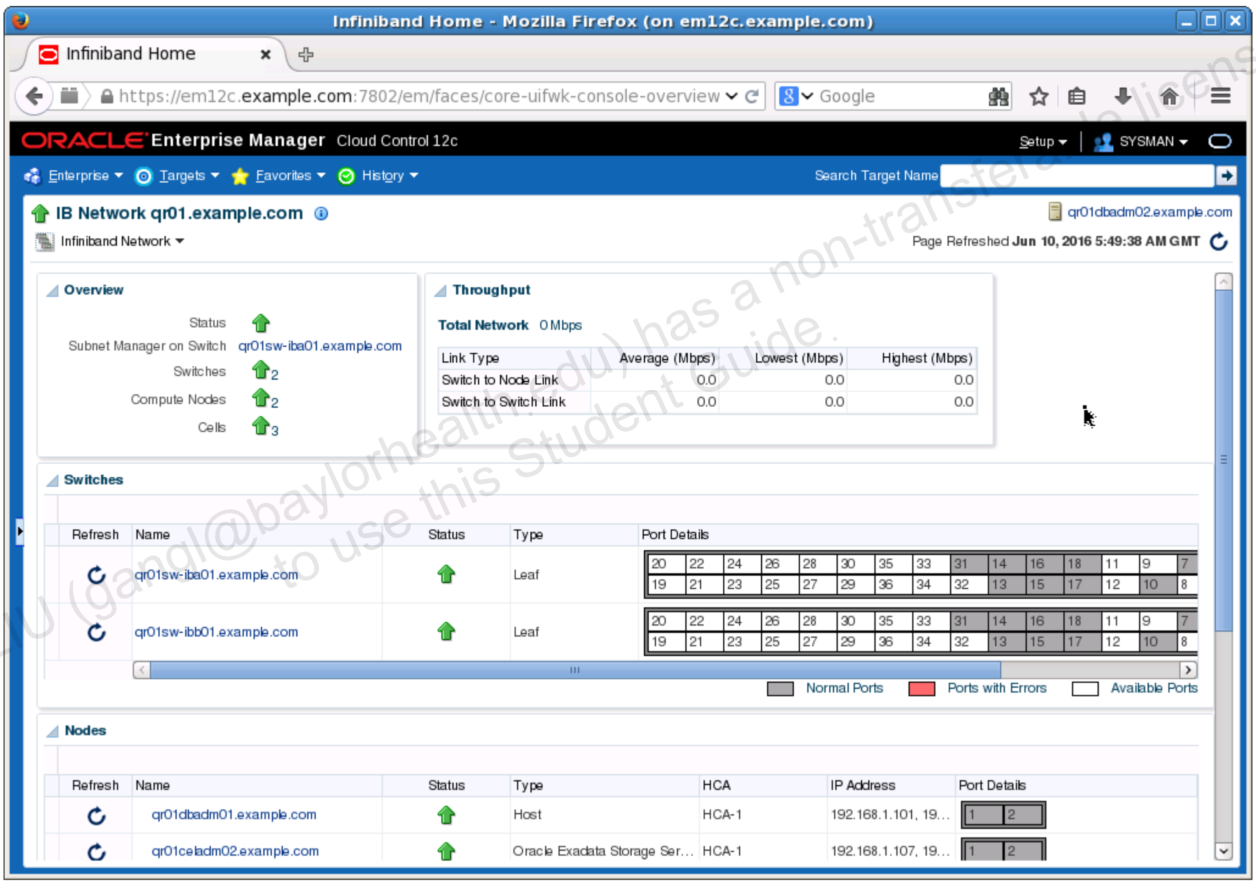
7. On the Database Machine home page, click “IB Network qr01.example.com” in the Target Navigation pane.



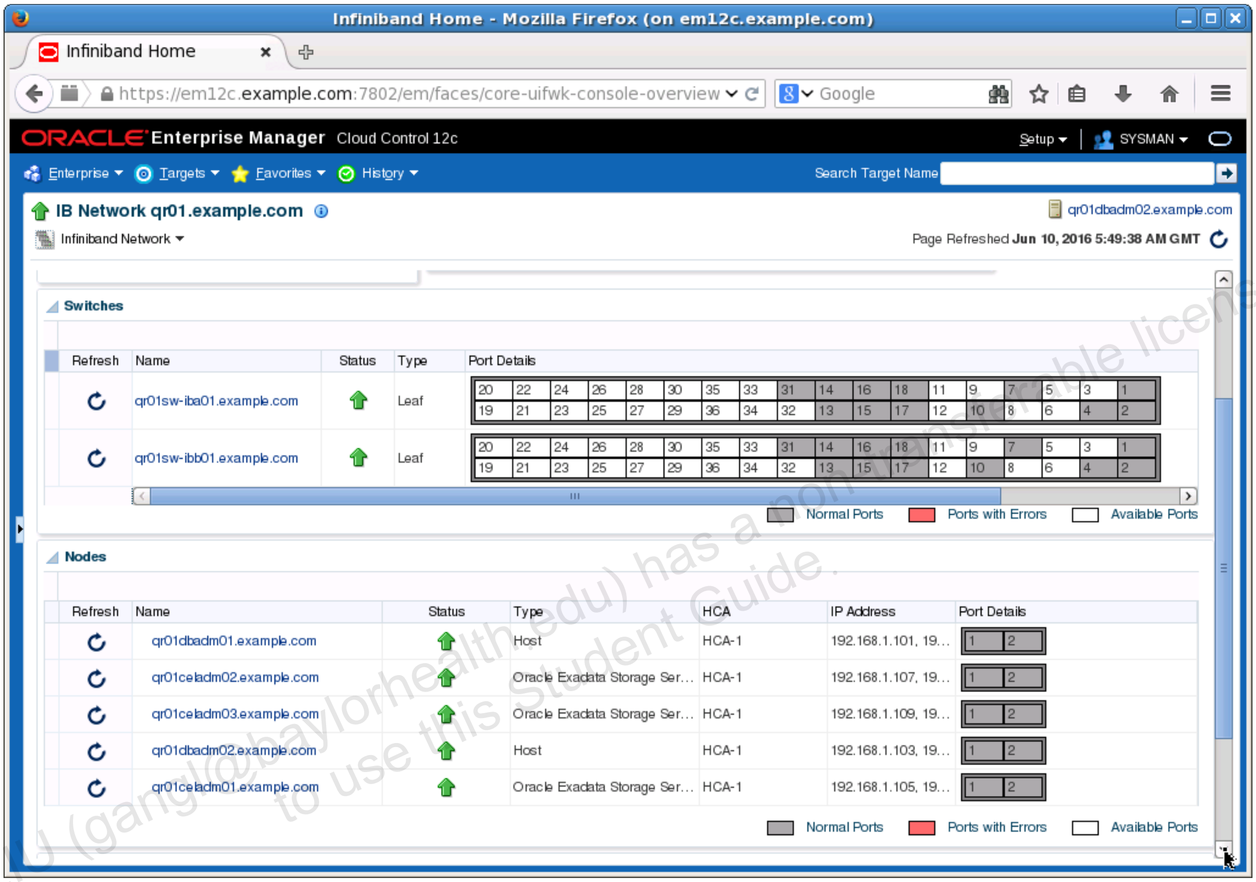
8. On the IB Network home page, collapse the Target Navigation pane.



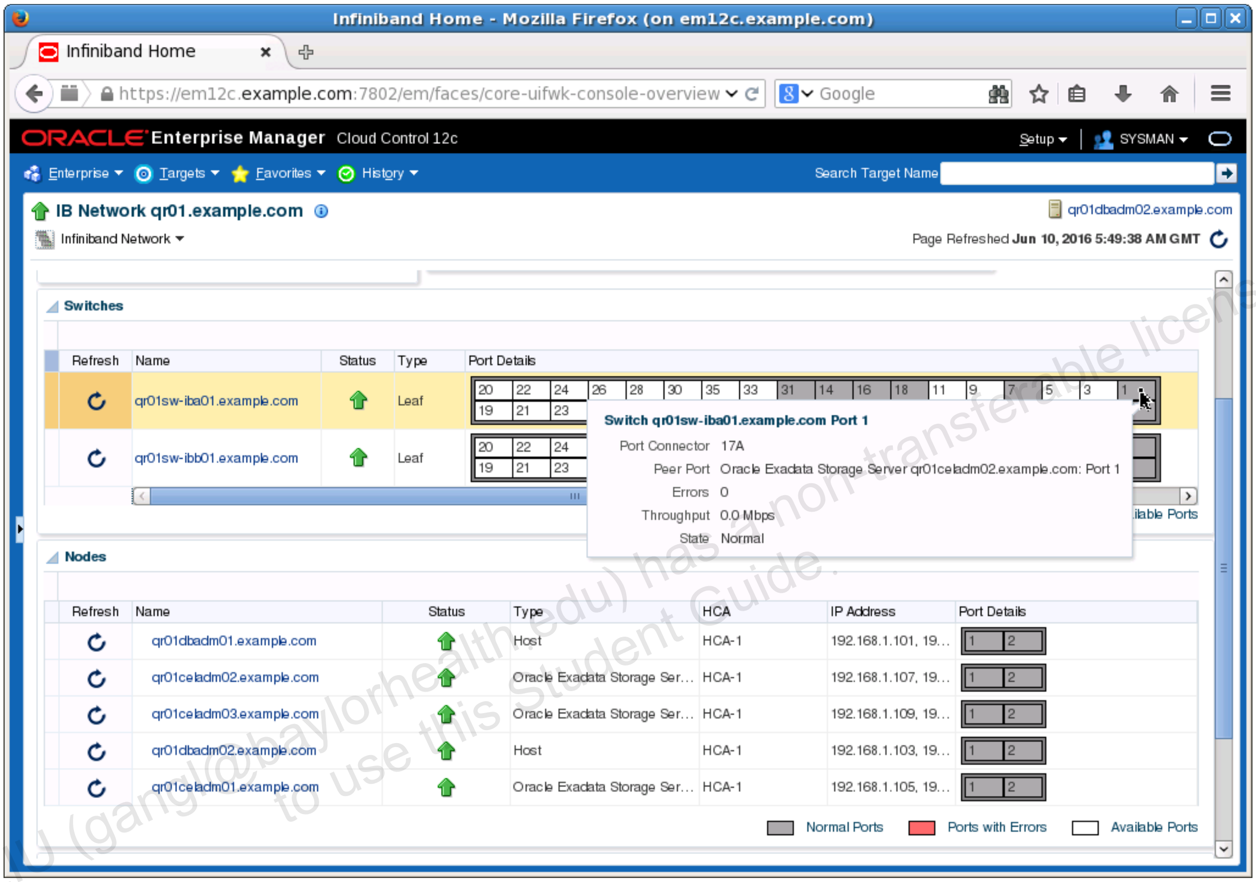
9. Take a moment to examine the IB Network home page. The Overview area provides a quick visual indication of the status and health of the InfiniBand network, including the switches, interfaces on the compute nodes (database servers), and cells (storage servers). The Throughput area summarizes the InfiniBand Network throughput. Below these lie areas that contain more detailed information about the InfiniBand switches and network interfaces. You will investigate these areas in greater detail later in the practice. At the bottom of the page, the Incidents area displays alerts related to the InfiniBand Network.
Note: Your laboratory environment does not contain any real InfiniBand switches or network interfaces. Therefore, to facilitate this practice, your laboratory environment has been seeded with information from a real quarter-rack Exadata Database Machine. As a result, some parts of the Enterprise Manager InfiniBand monitoring and administration interfaces will contain incomplete data. For example, the Throughput area on the IB Network home page will always contain zero values in your laboratory environment.



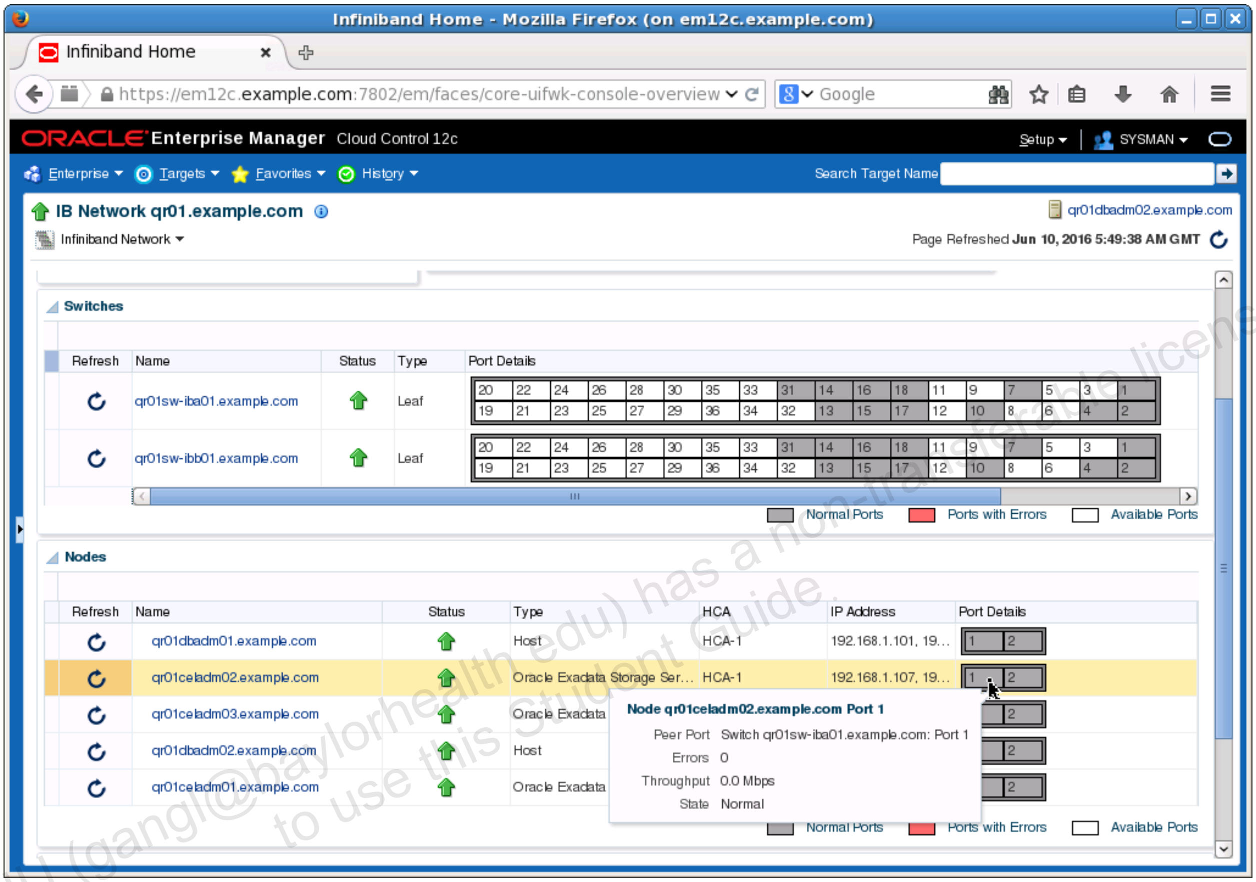
10. Scroll down the IB Network home page to display the Switches and Nodes areas. Note that these areas provide a visual representation of the InfiniBand network ports on the InfiniBand switches and Host Channel Adapters (HCAs). Grey colored ports indicate ports that are in use and functioning normally. White colored ports are available ports that are not connected to anything else. If a problem is detected, the corresponding port will be colored red.



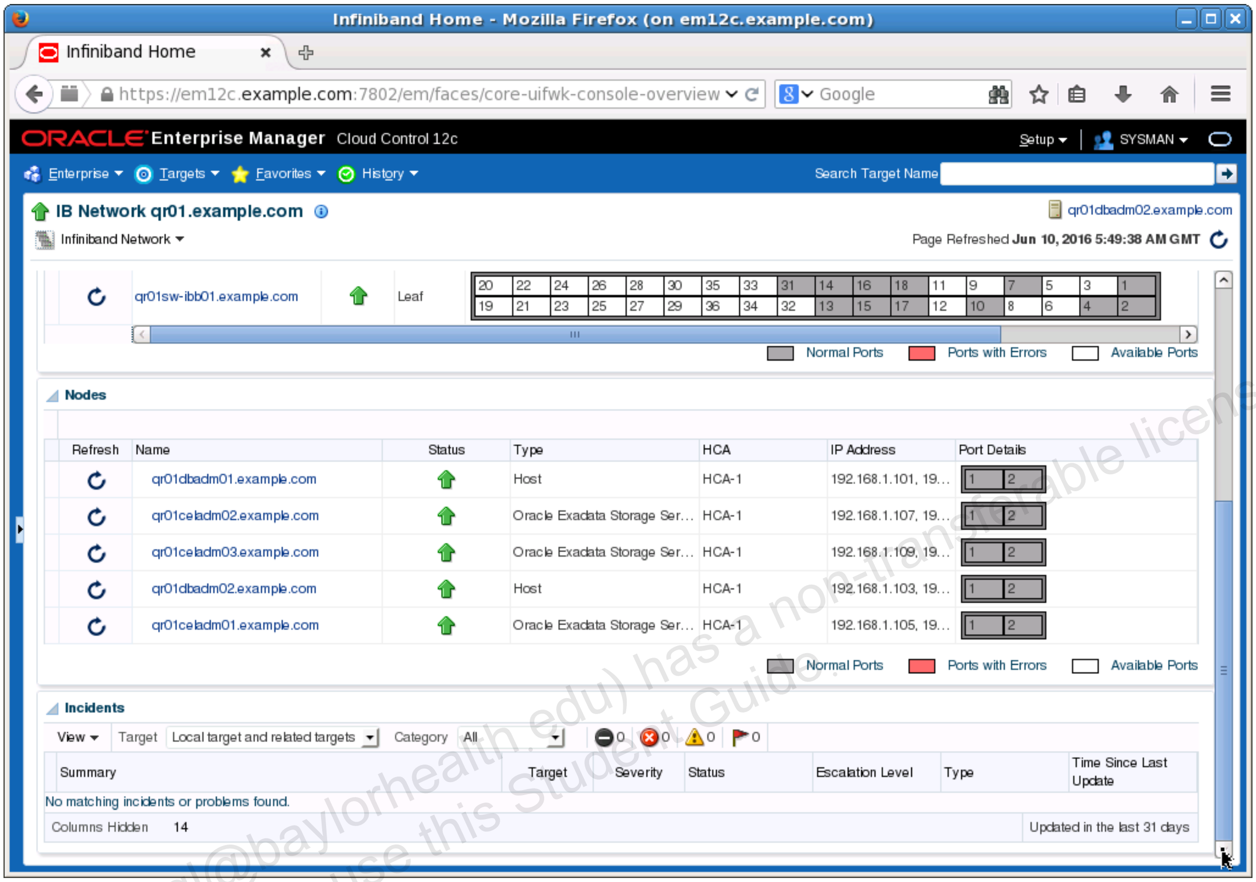
11. When you hover the cursor over the port diagram, a callout that contains detailed information about the port appears. If the port is in use, the callout also includes the details of the Peer Port; that is, the port at the other end of the connection. Using this information, you can determine the InfiniBand network topology. Hold the cursor over some of the switch ports. Can you determine which ports are connected to each of the database servers and Exadata Storage Servers? Can you determine which ports interconnect the switches?



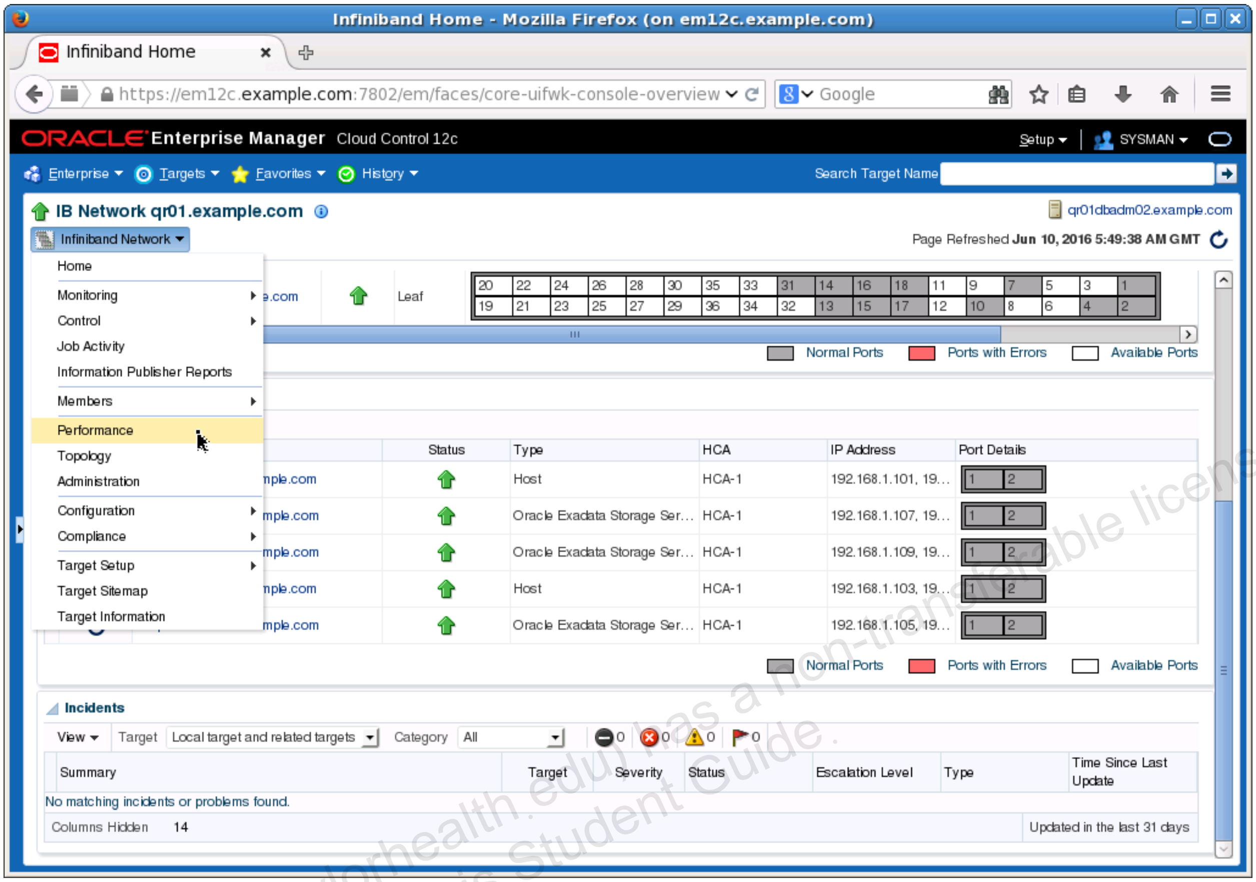
12. Hover the cursor over the database server and storage server HCA ports. Can you determine which switch each port is connected to? Verify that the information associated with each port matches its peer port. For example, if port 1 on `qr01sw-iba01` displays port 1 on `qr01celadm02` as its peer port (as shown above), verify that port 1 on `qr01celadm02` displays port 1 on `qr01sw-iba01` as its peer port (as shown below).



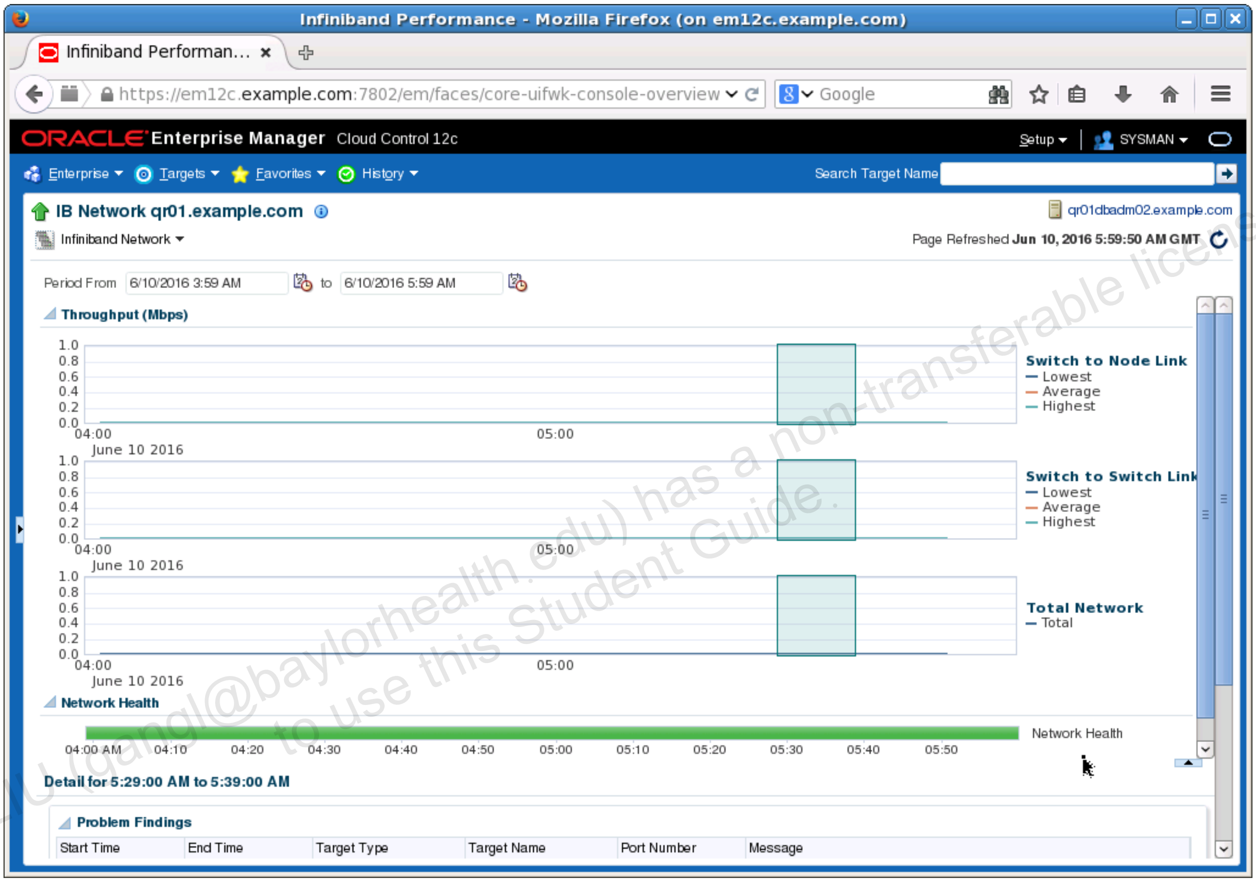
13. Scroll to the bottom of the IB Network home page to reveal the Incidents area. This is where you will see notification relating to problems and issues that are detected on the InfiniBand network.



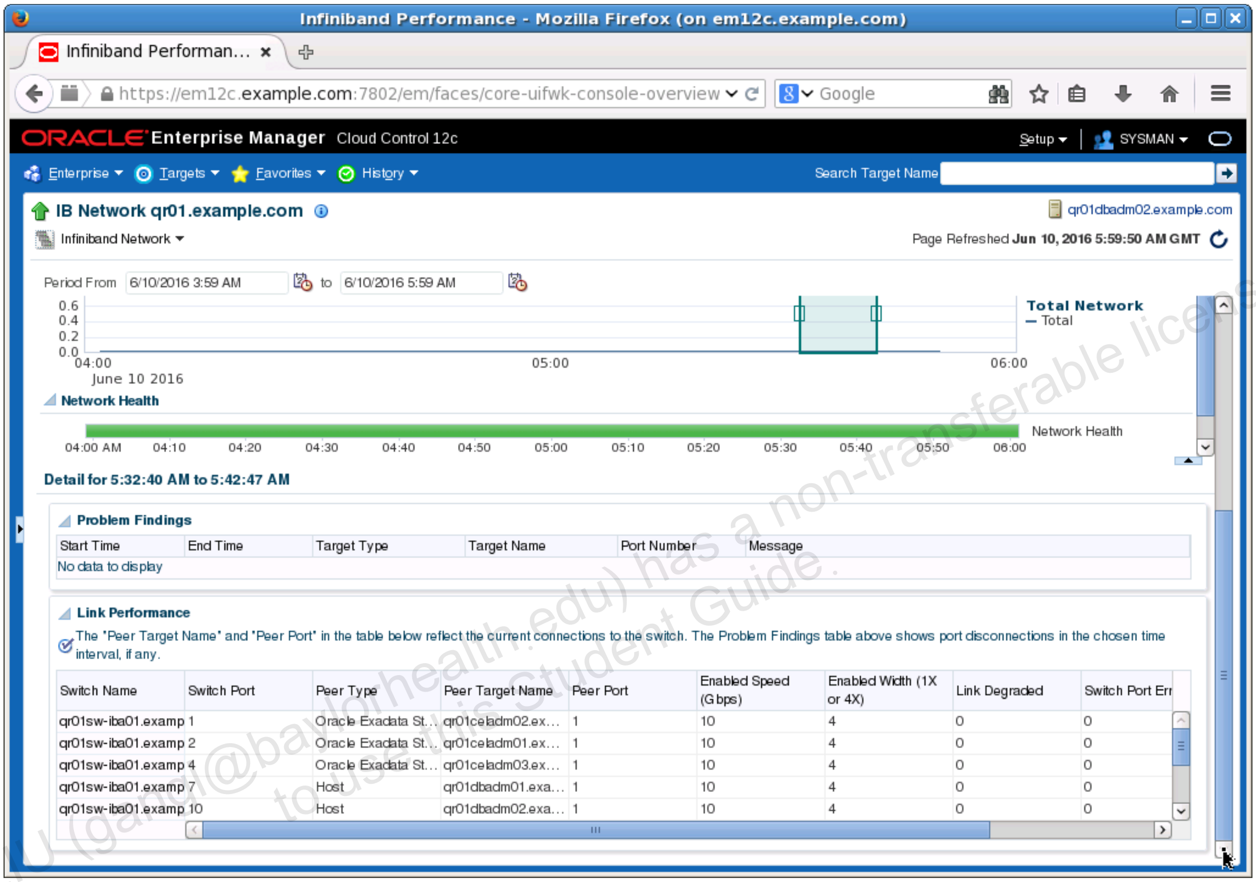
14. Select the Infiniband Network > Performance menu command.



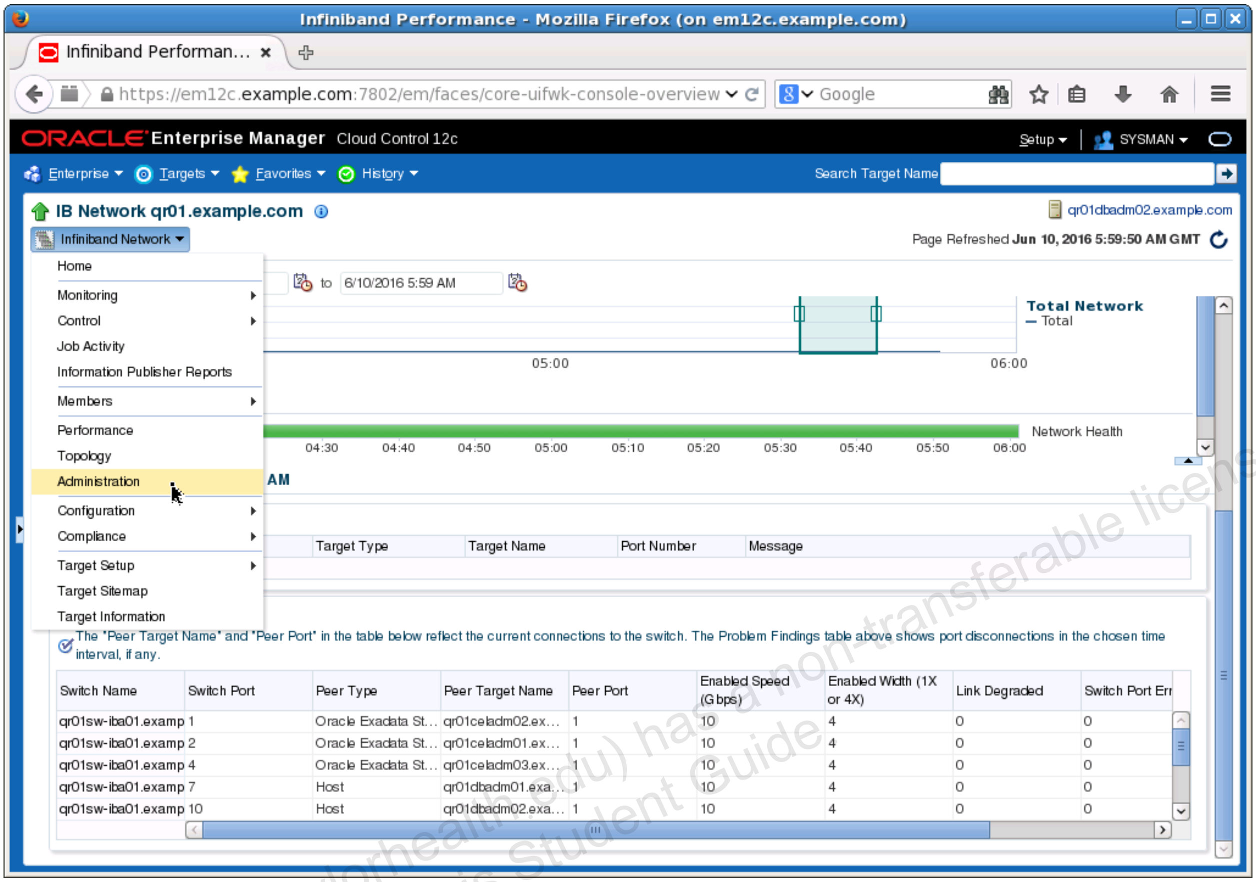
15. The InfiniBand network performance page displays a series of charts that plot InfiniBand network throughput over time. In your laboratory environment, these charts are essentially empty; however, in a production environment, administrators would use these charts to quickly identify if there is a significant variation in throughput on different parts of the network. Such a variation could indicate a problem with a network component or indicate an imbalance within the environment.
The InfiniBand network performance page also contains a visual indicator of the network health over time.



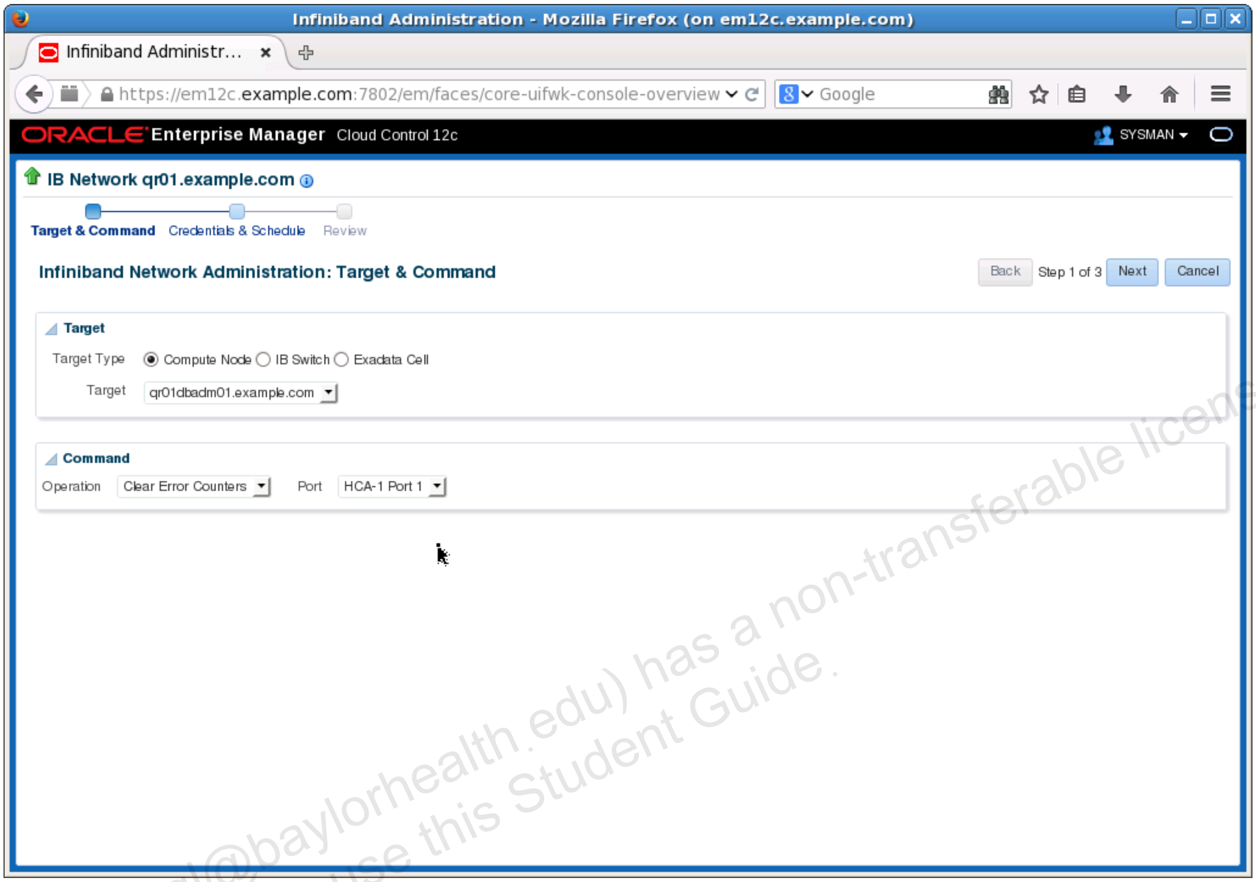
16. Scroll down the InfiniBand network performance page to reveal the details area. This area contains information related to the time interval highlighted in the throughput charts. If the throughput charts identify a significant variation, administrators can highlight a portion of the chart and investigate the details area to see if there are any associated problem findings, or if any of the links are degraded or showing errors. Note: If you see no data in the details area then try moving the highlighted area in the throughput charts.



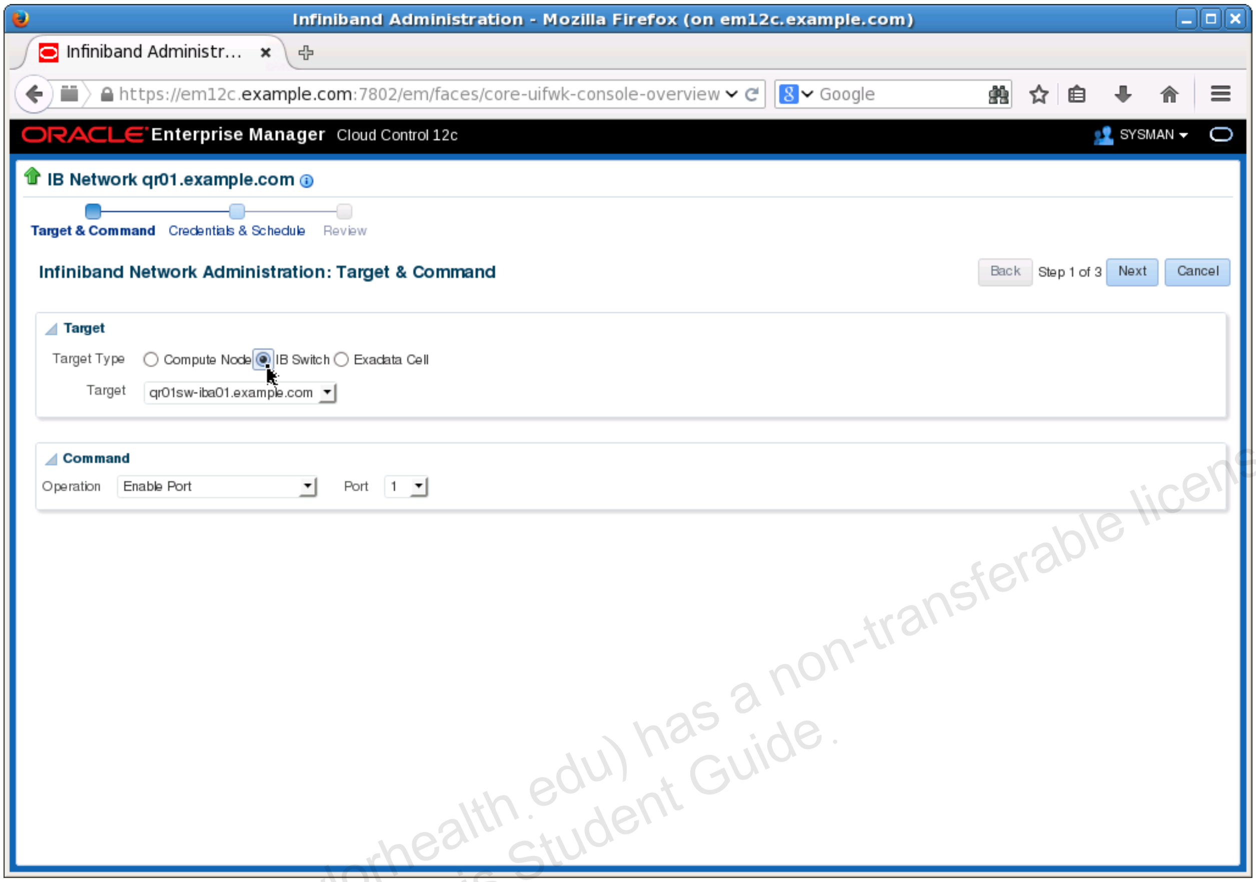
17. Select the Infiniband Network > Administration menu command.



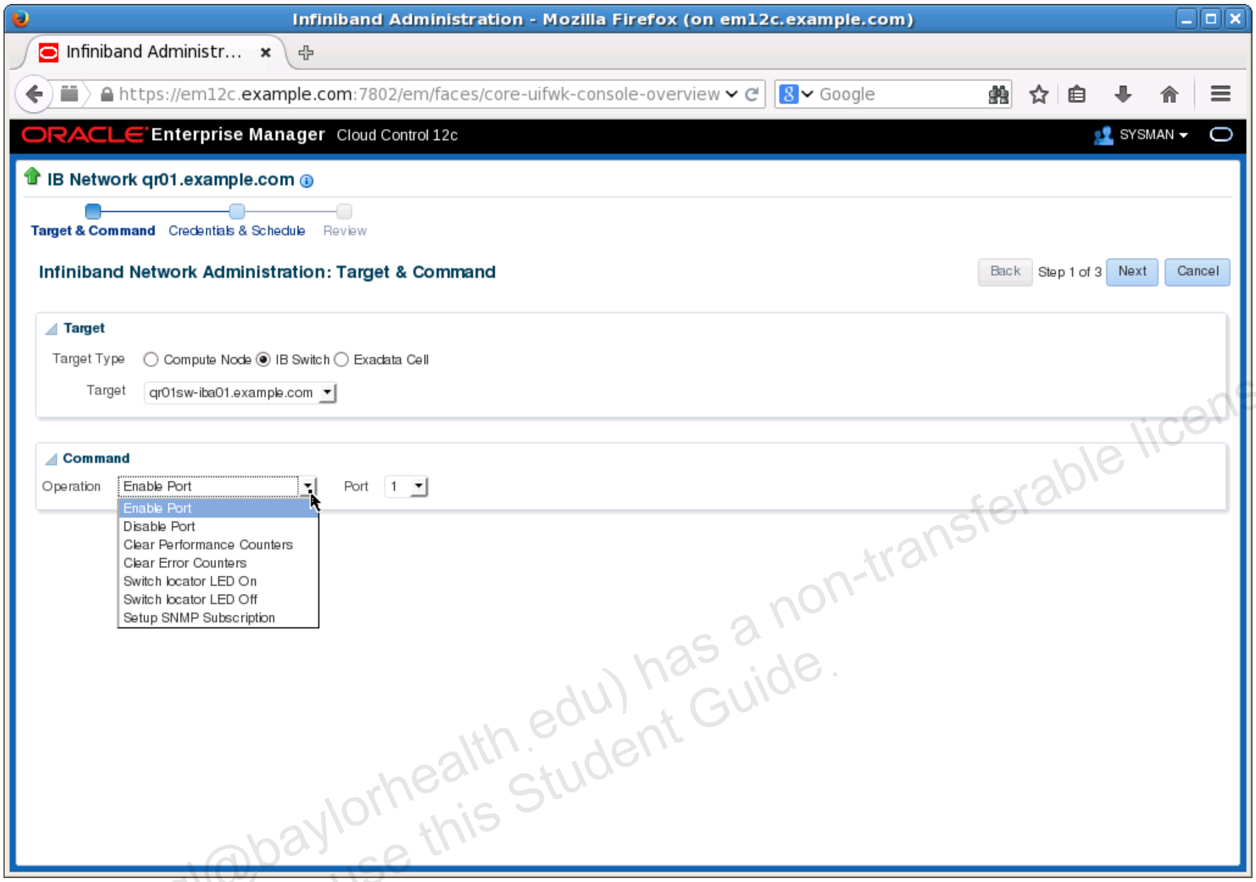
18. The first page of the Infiniband Network Administration Wizard appears. Using this wizard, you can perform an InfiniBand network administration task on an InfiniBand switch or server HCA.



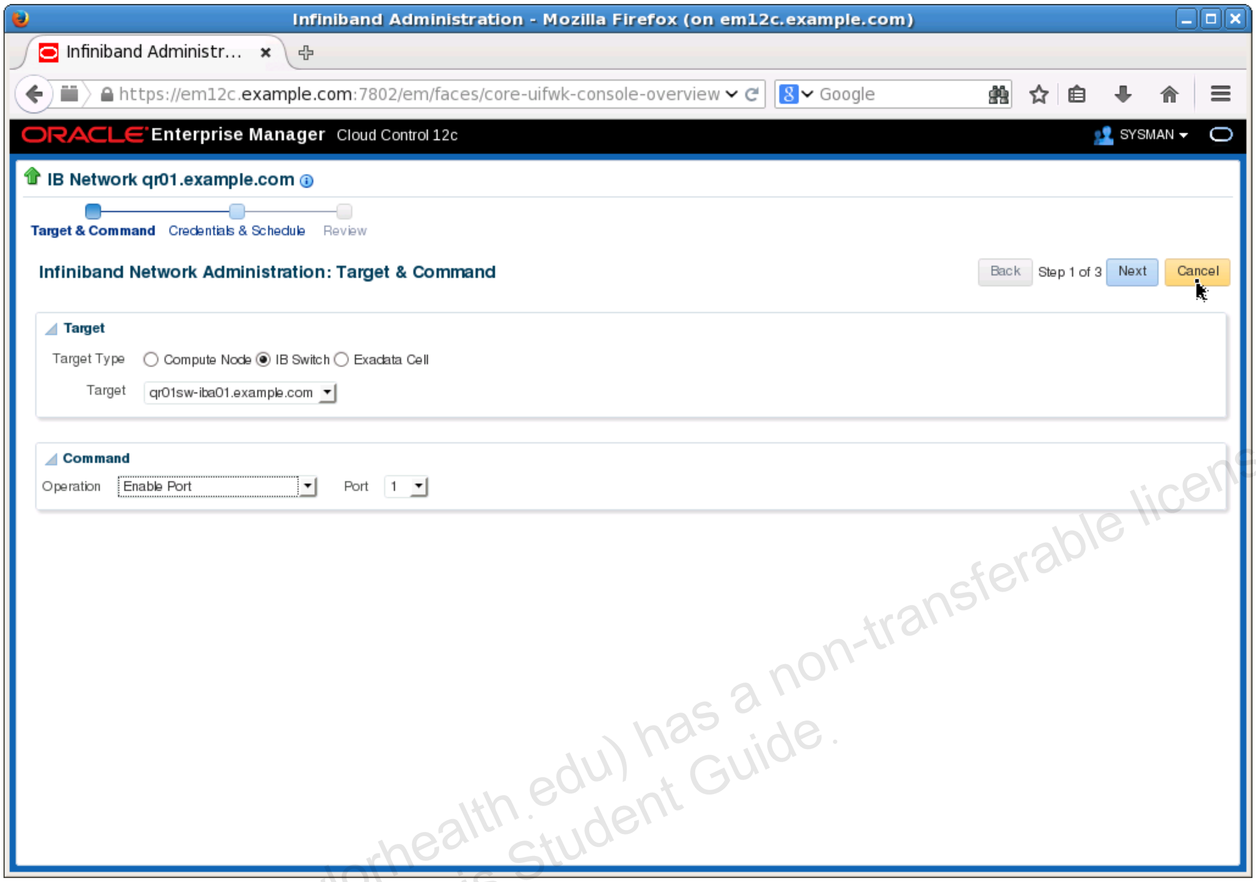
19. Select IB Switch as the Target Type.



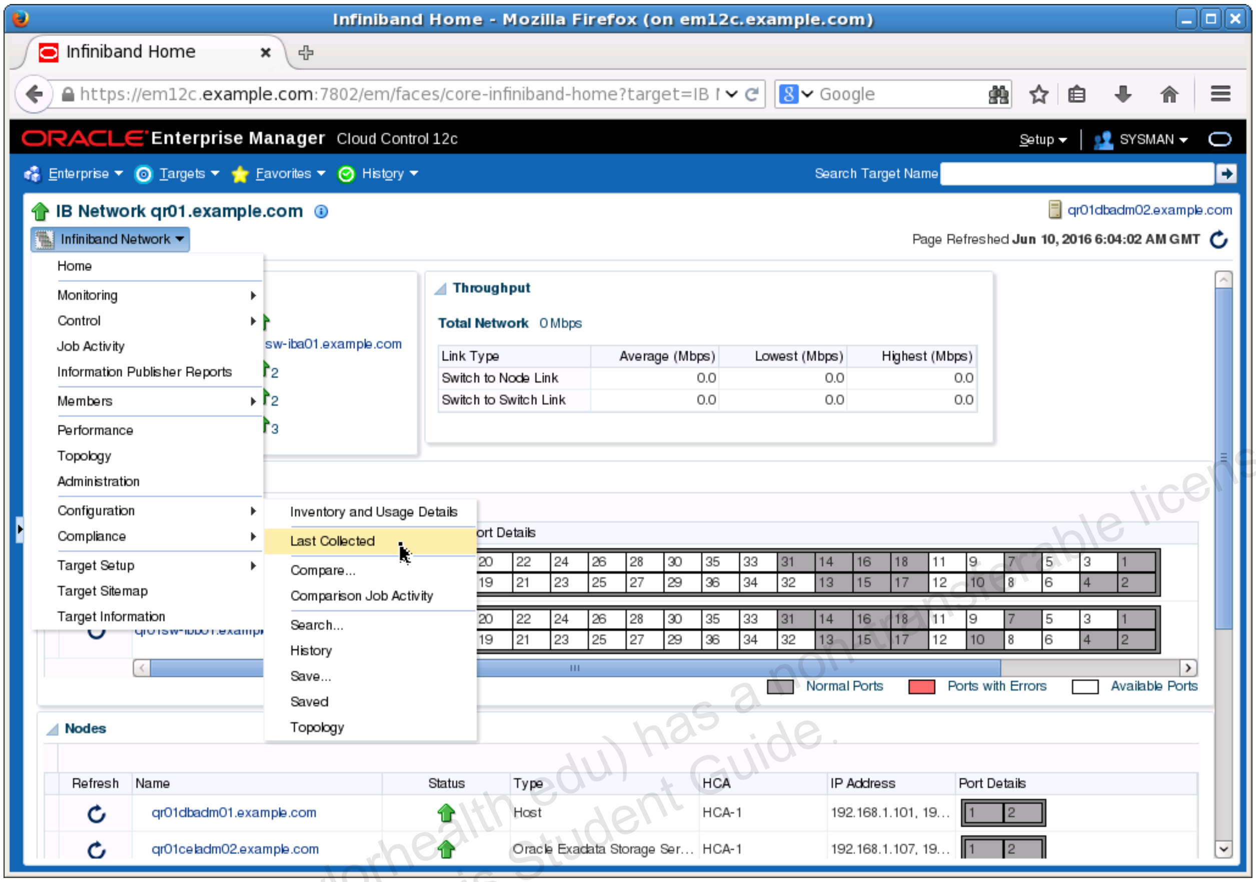
20. Click the Operation drop-down list and examine the list of available options. This gives you an idea of the administration tasks that you can perform on InfiniBand switch ports by using the InfiniBand Network Administration Wizard.



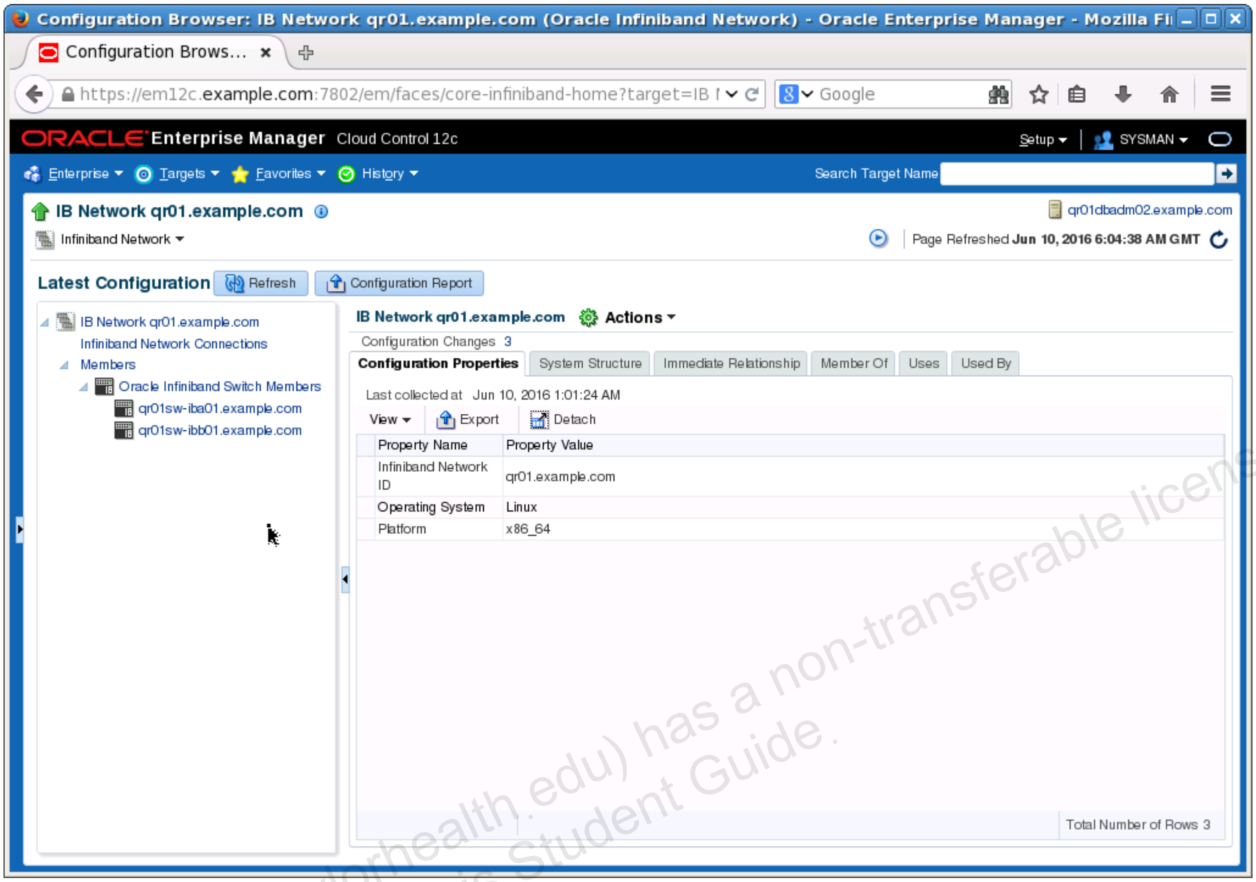
21. Click Cancel to exit the Infiniband Network Administration Wizard.



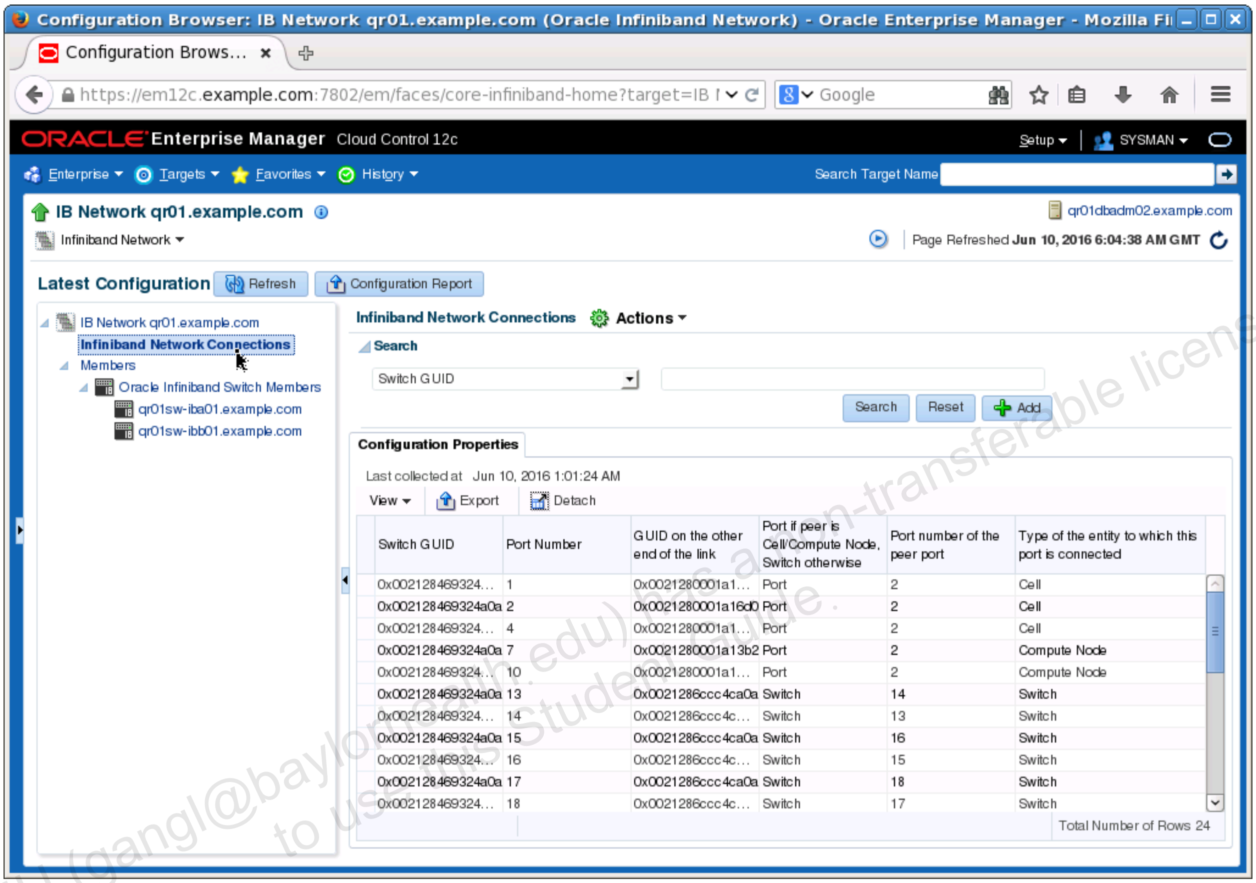
22. Select the Infiniband Network > Configuration > Last Collected menu command.



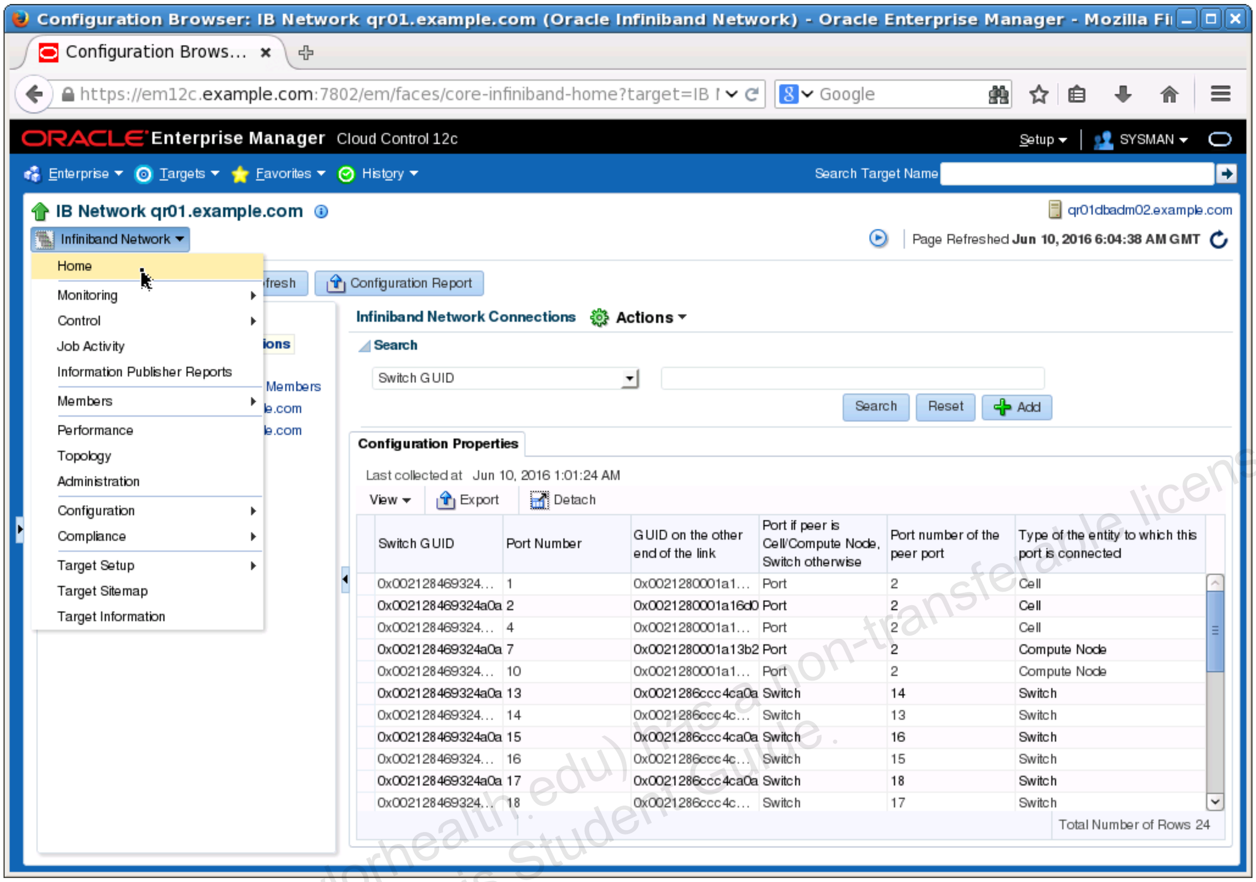
23. You should now see the configuration browser page for the InfiniBand network.



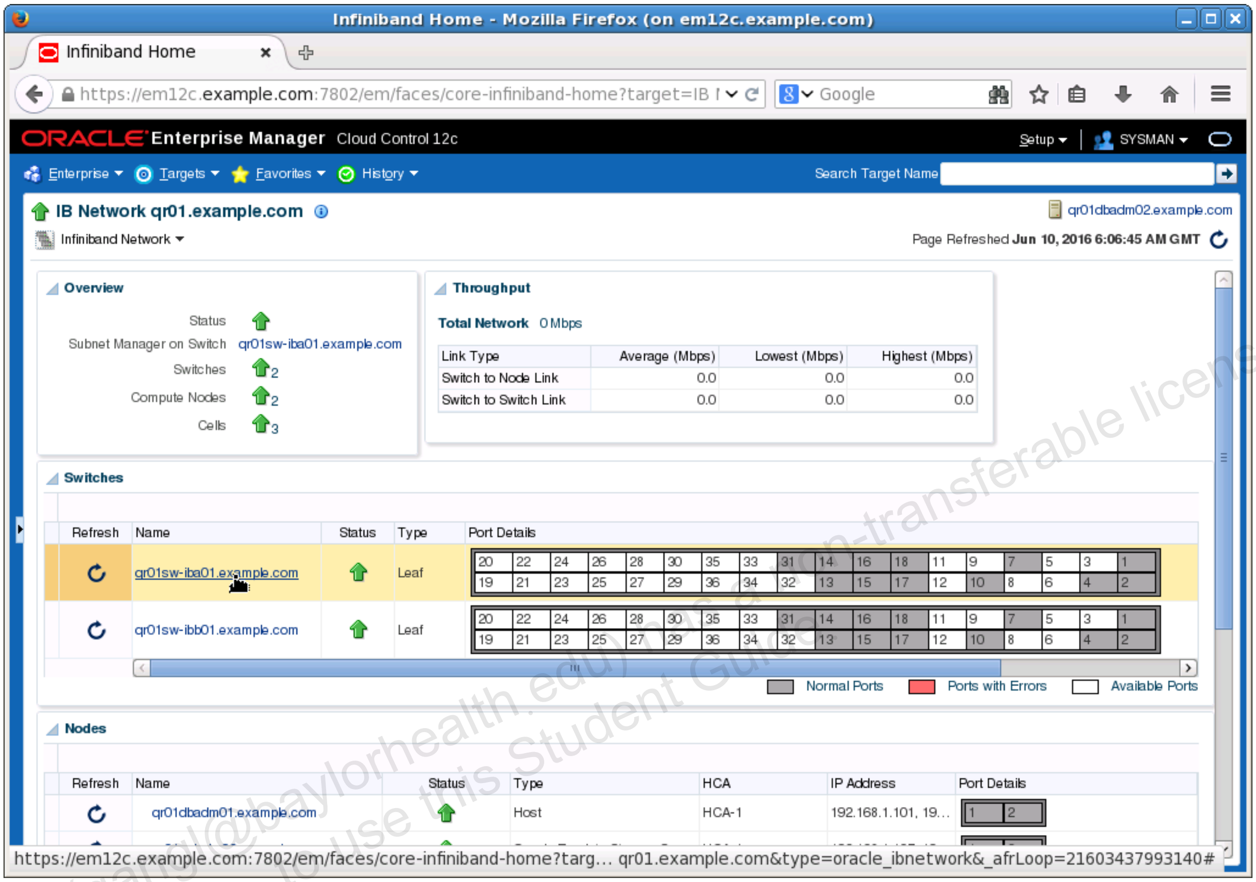
24. Click Infiniband Network Connections in the hierarchical list on the left of the page. This will display a table, which contains information about the InfiniBand network connections. This is the information that supports the port diagrams on the IB Network home page. Take a moment to examine the InfiniBand Network Connections along with any of the other available configuration information.



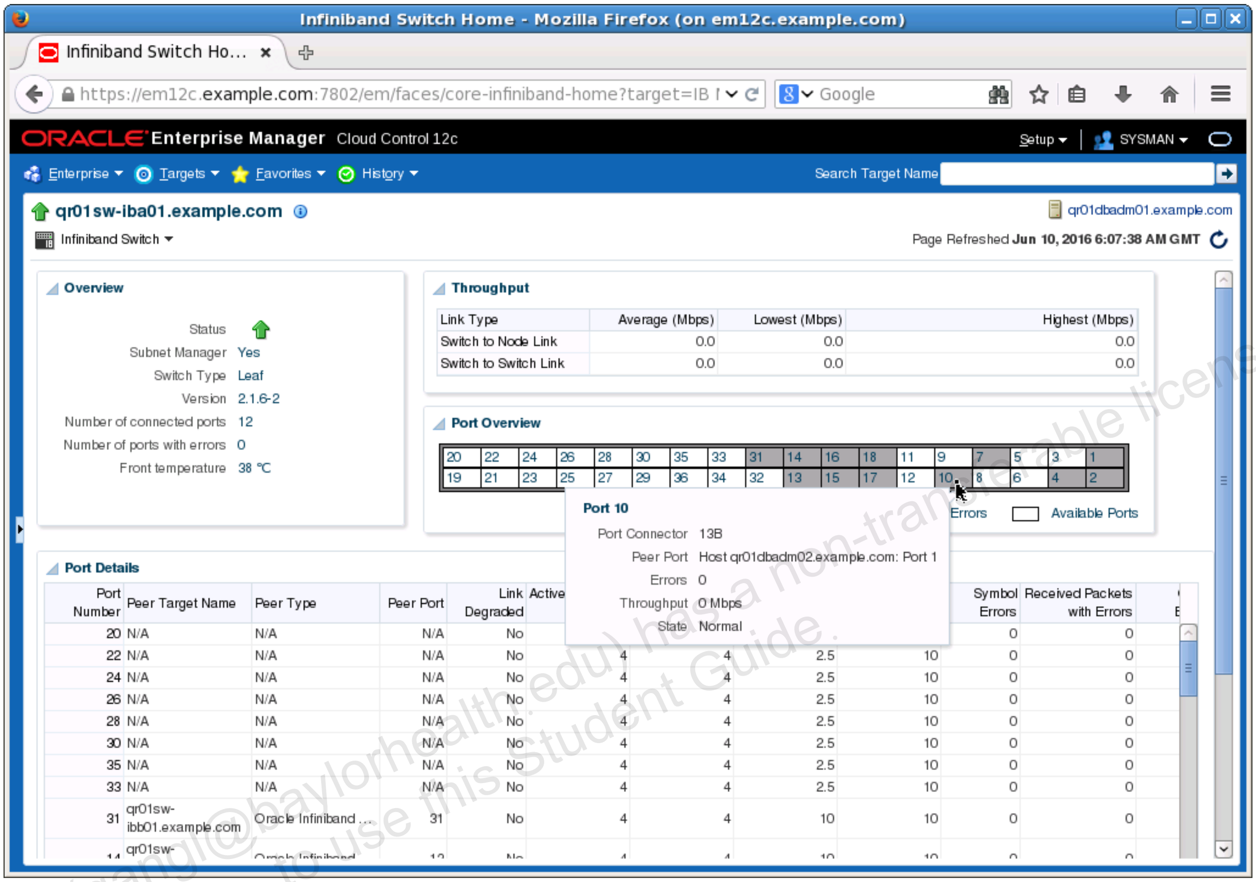
25. Select the Infiniband Network > Home menu command.



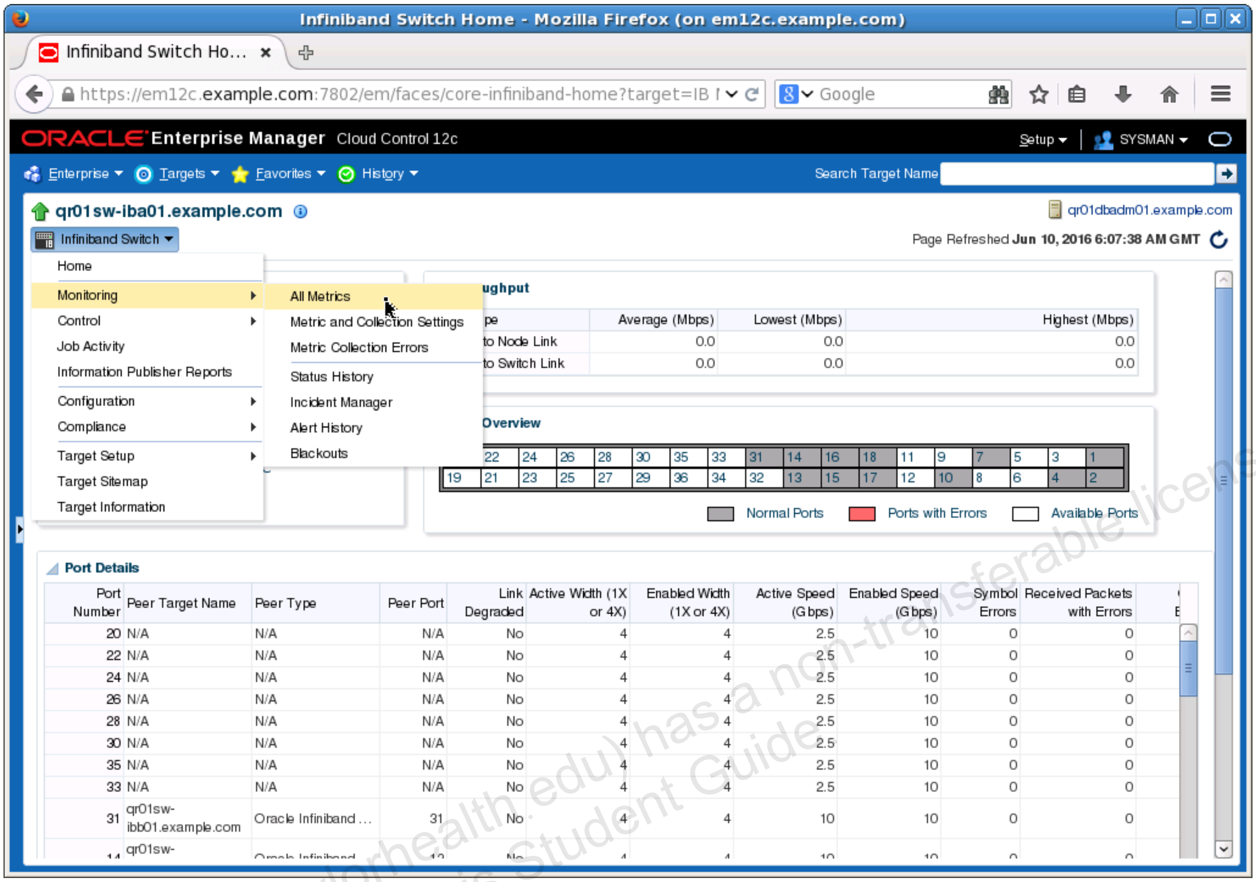
26. As you have already seen, the IB Network home provides an overview of the Database Machine InfiniBand network. In addition to this, each InfiniBand switch also has a home page associated with it. Click the link associated with the InfiniBand switch `qr01sw-iba01.example.com`.



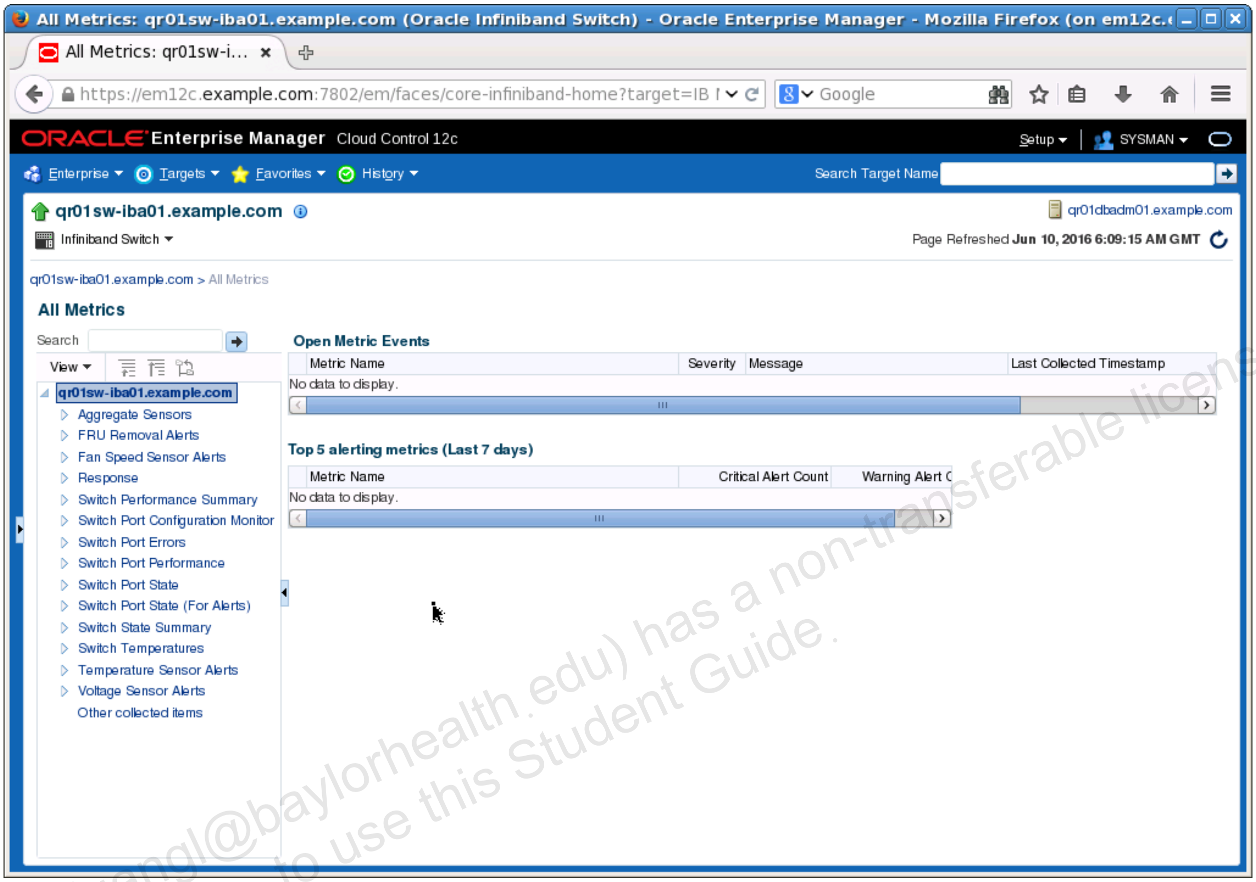
27. The InfiniBand switch home page for `qr01sw-iba01.example.com` appears. Like the IB Network home page, this page contains overview and throughput information, along with a port diagram for the switch. Below the port diagram, there is an area that lists port details, which administrators can use to quickly find information about a specific port.



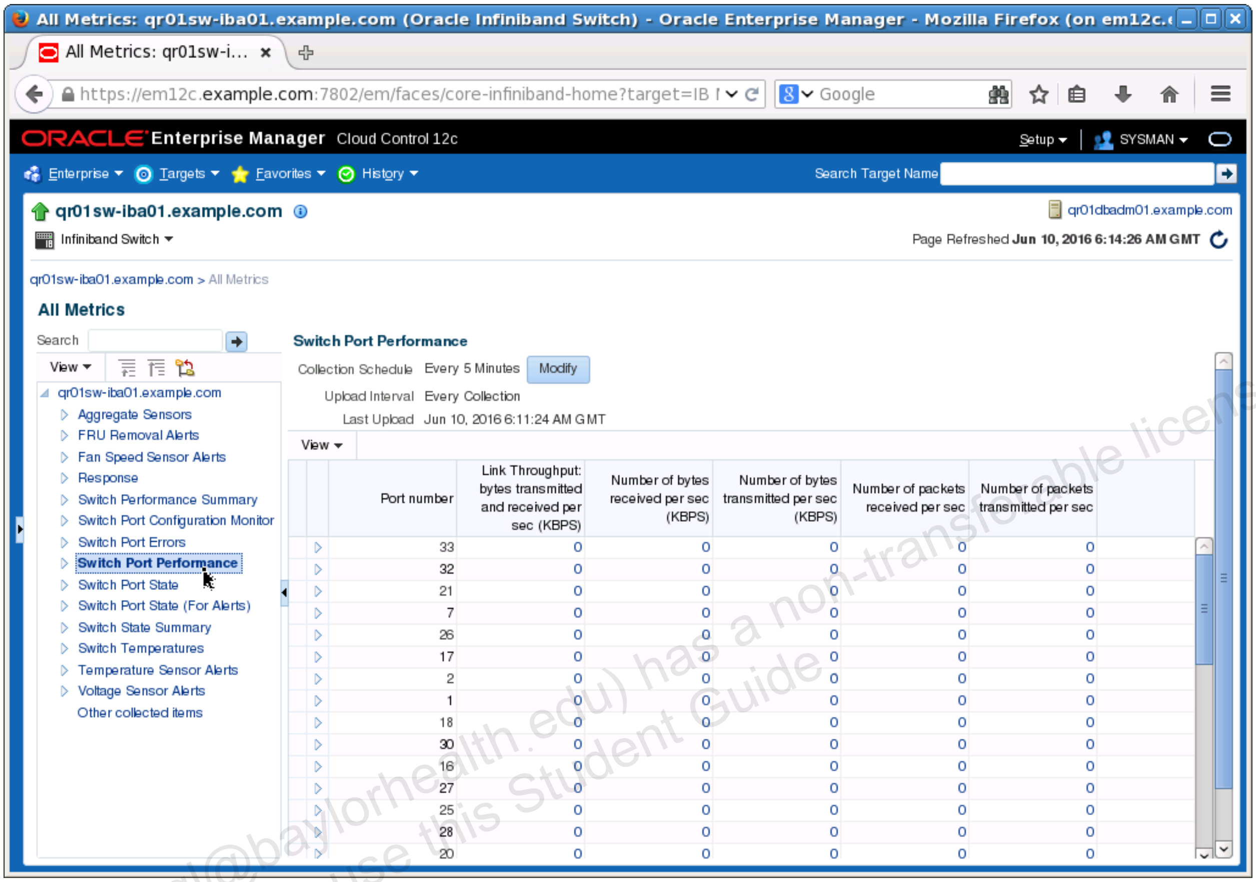
28. Select the Infiniband Switch > Monitoring > All Metrics menu command.



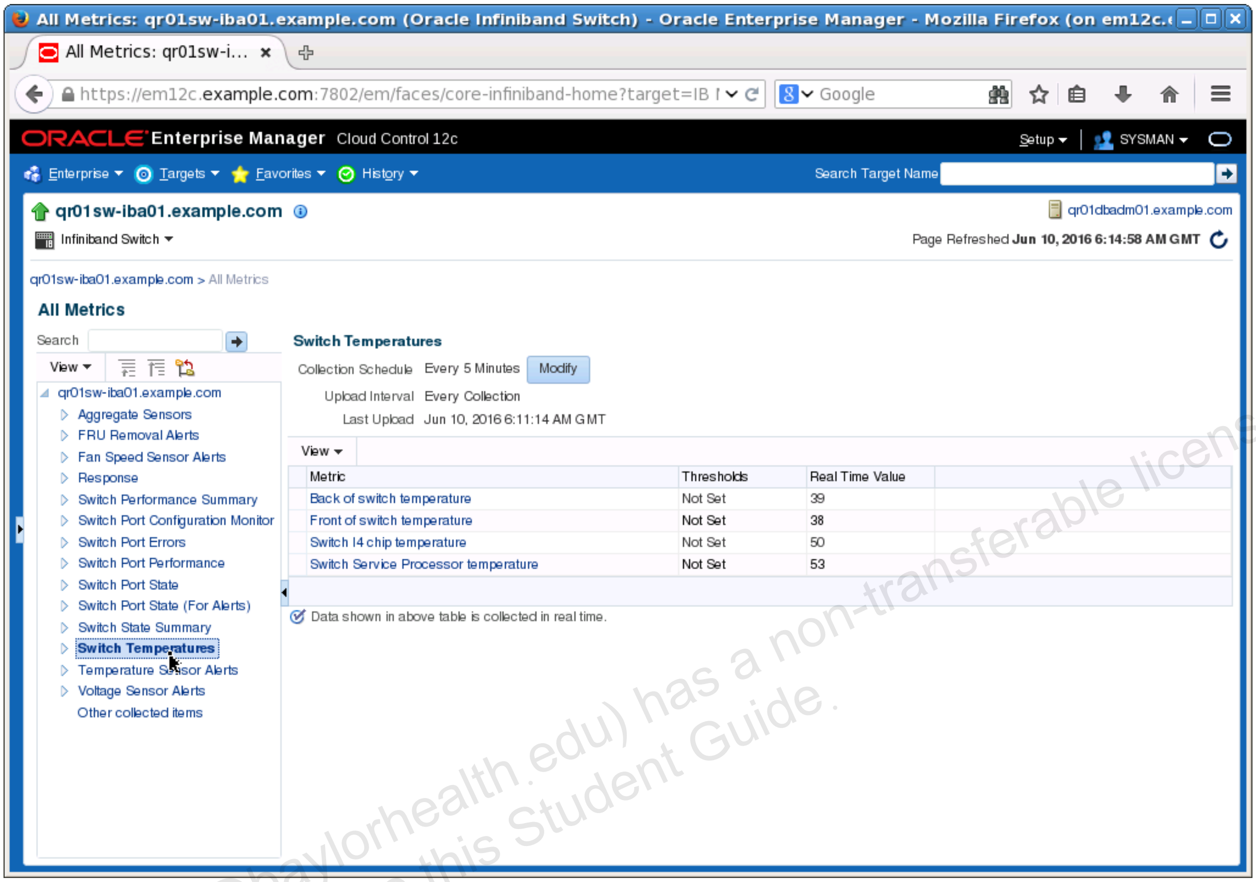
29. The All Metrics page provides an interface for administrators to examine the metrics associated with a monitoring target. For InfiniBand switches, this page provides access to numerous metrics and sensors associated with each switch. The metrics are organized into groups, which are displayed in a hierarchical list on the left side of the page.



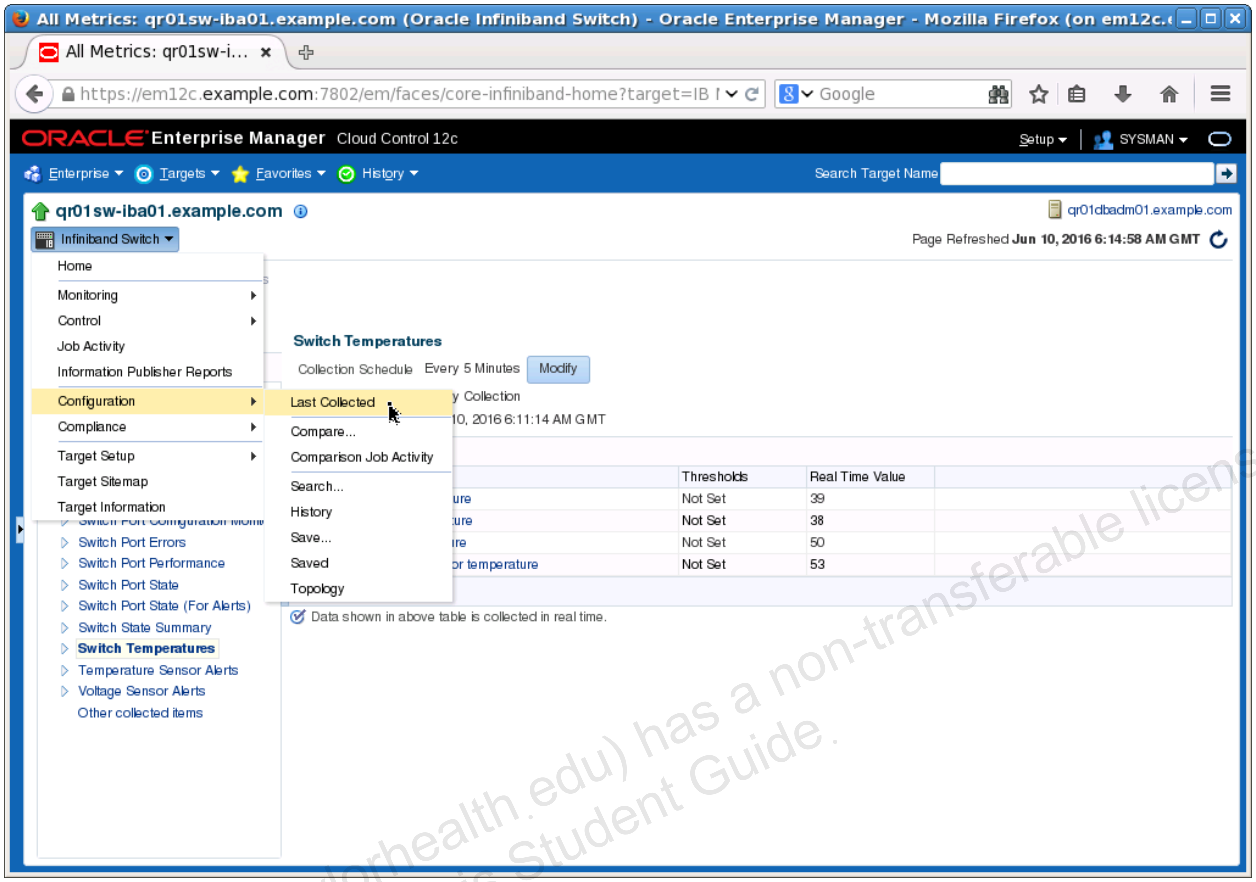
30. Click the Switch Port Performance metric group. Examine the available switch port performance metrics. Remember that administrators can set thresholds for these metrics, so that they will be alerted when certain conditions exist.



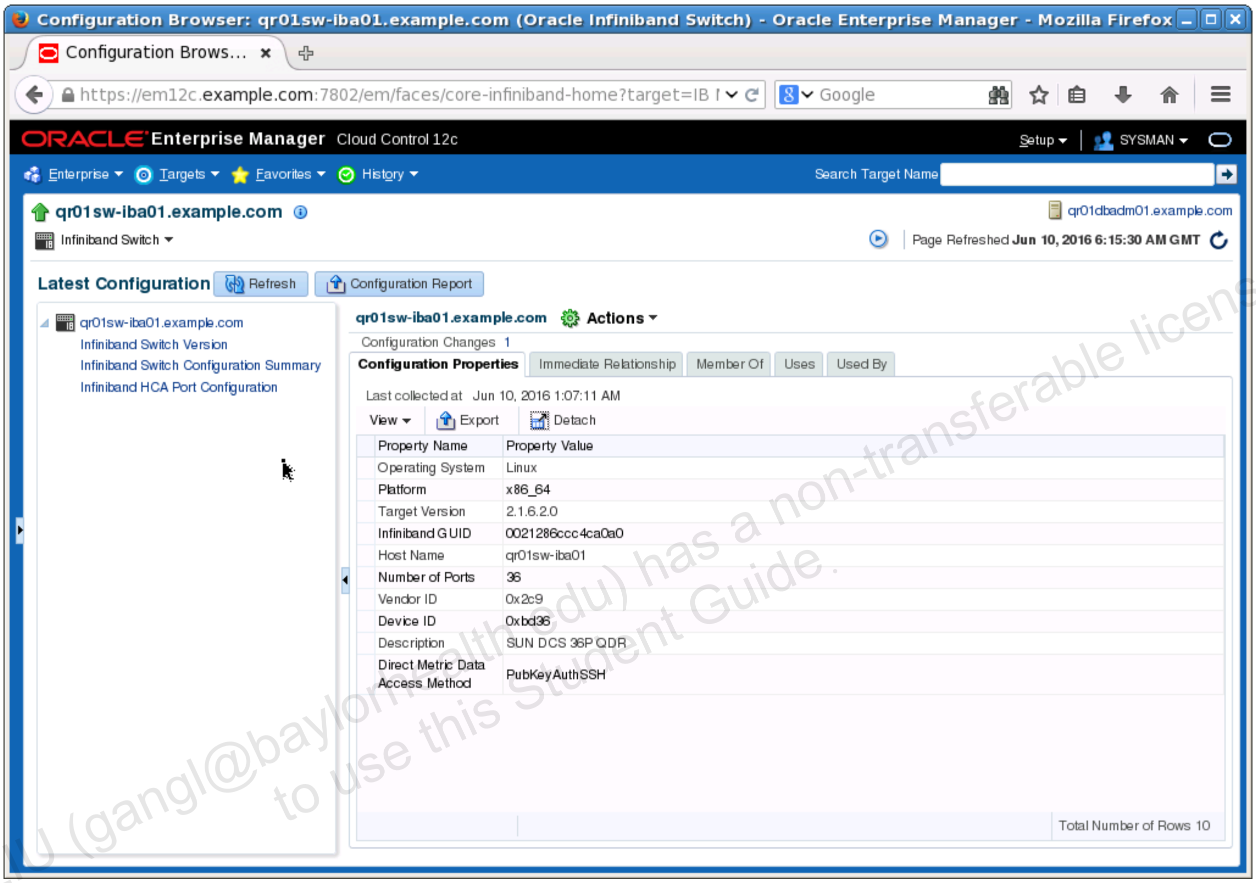
31. Click the Switch Temperatures metric group. The temperature readings are provided by various sensors contained inside the switch.



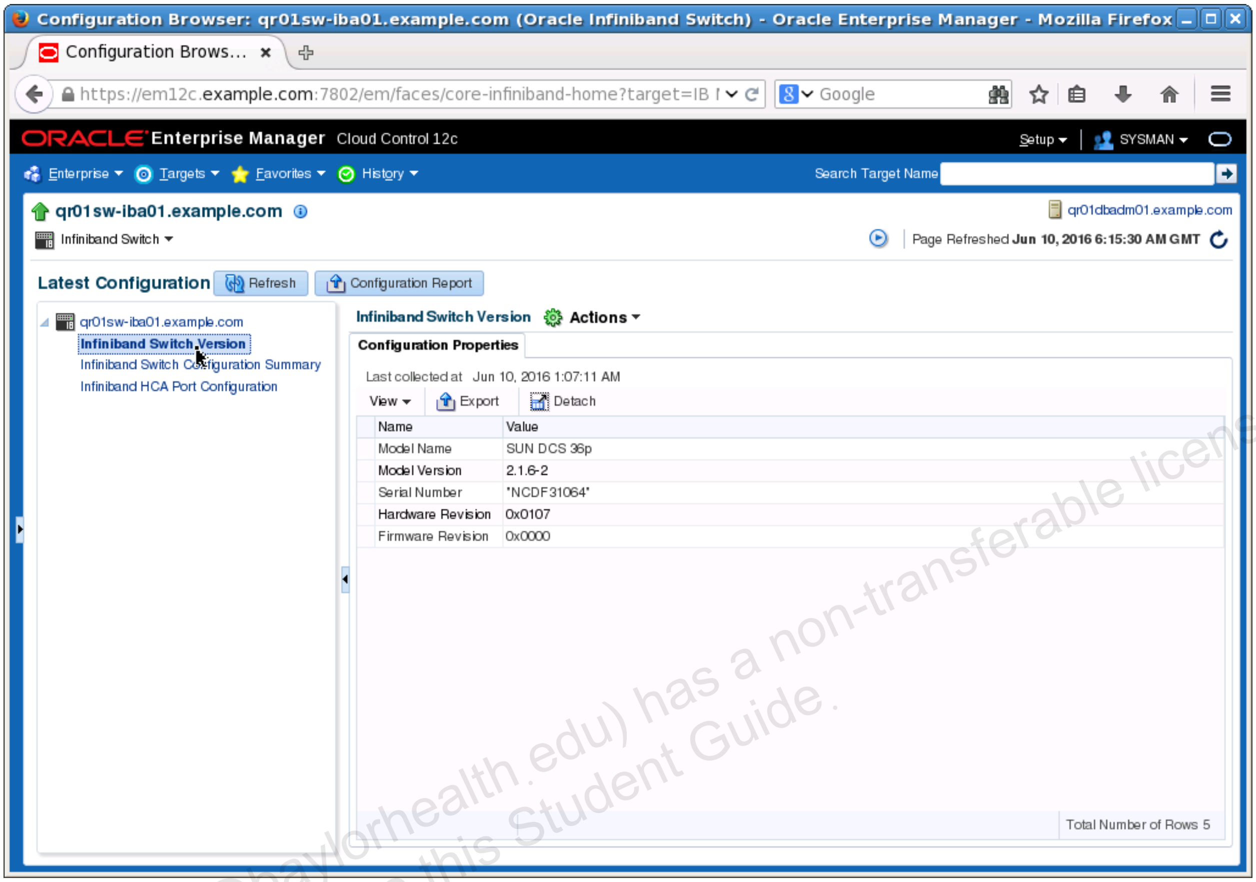
32. Select the Infiniband Switch > Configuration > Last Collected menu command.



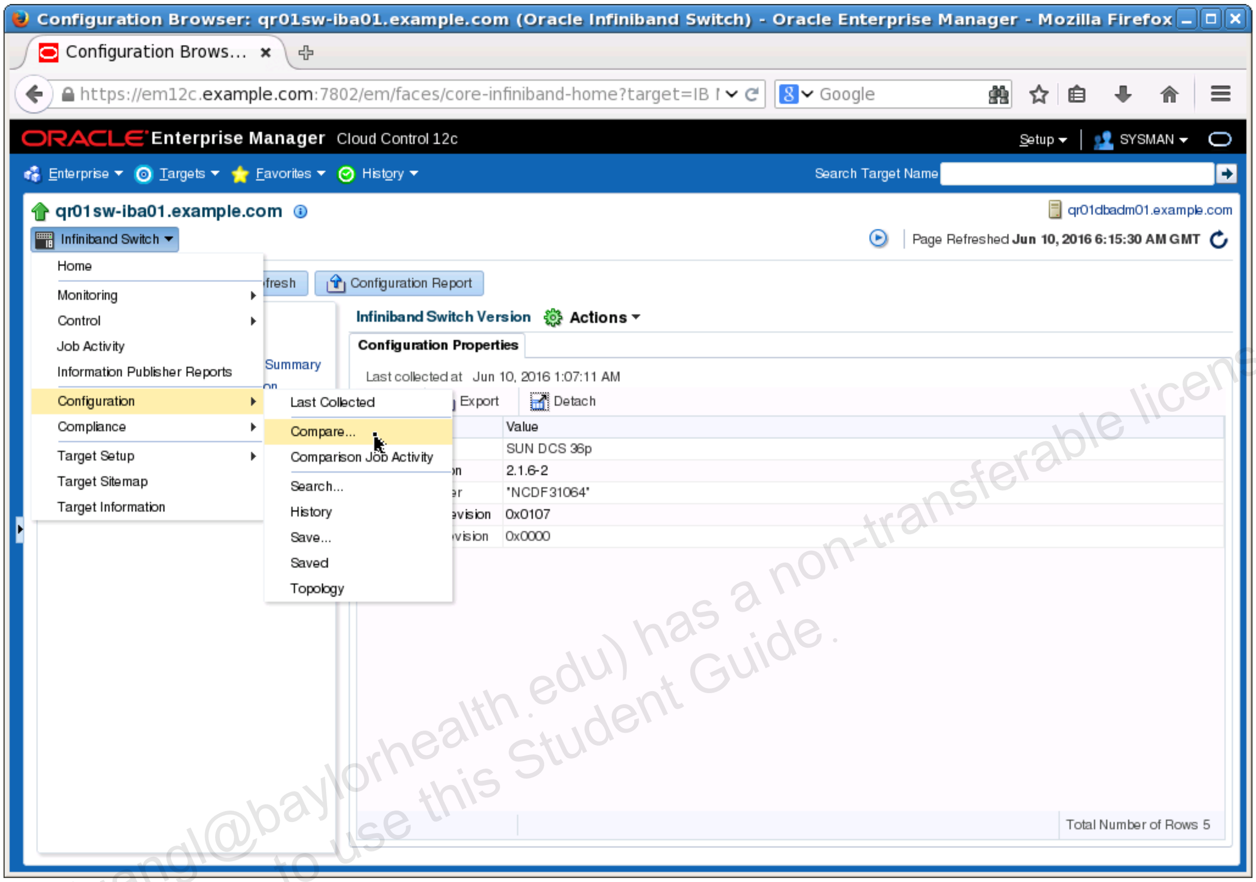
33. You should now see the configuration browser page for `qr01sw-iba01.example.com`. The configuration browser allows administrators to examine the configuration settings associated with a management target. It is similar to the All Metrics page that you just examined. The main difference between metrics and configuration settings is that metric values are expected to constantly vary over time while configuration settings should be more stable.



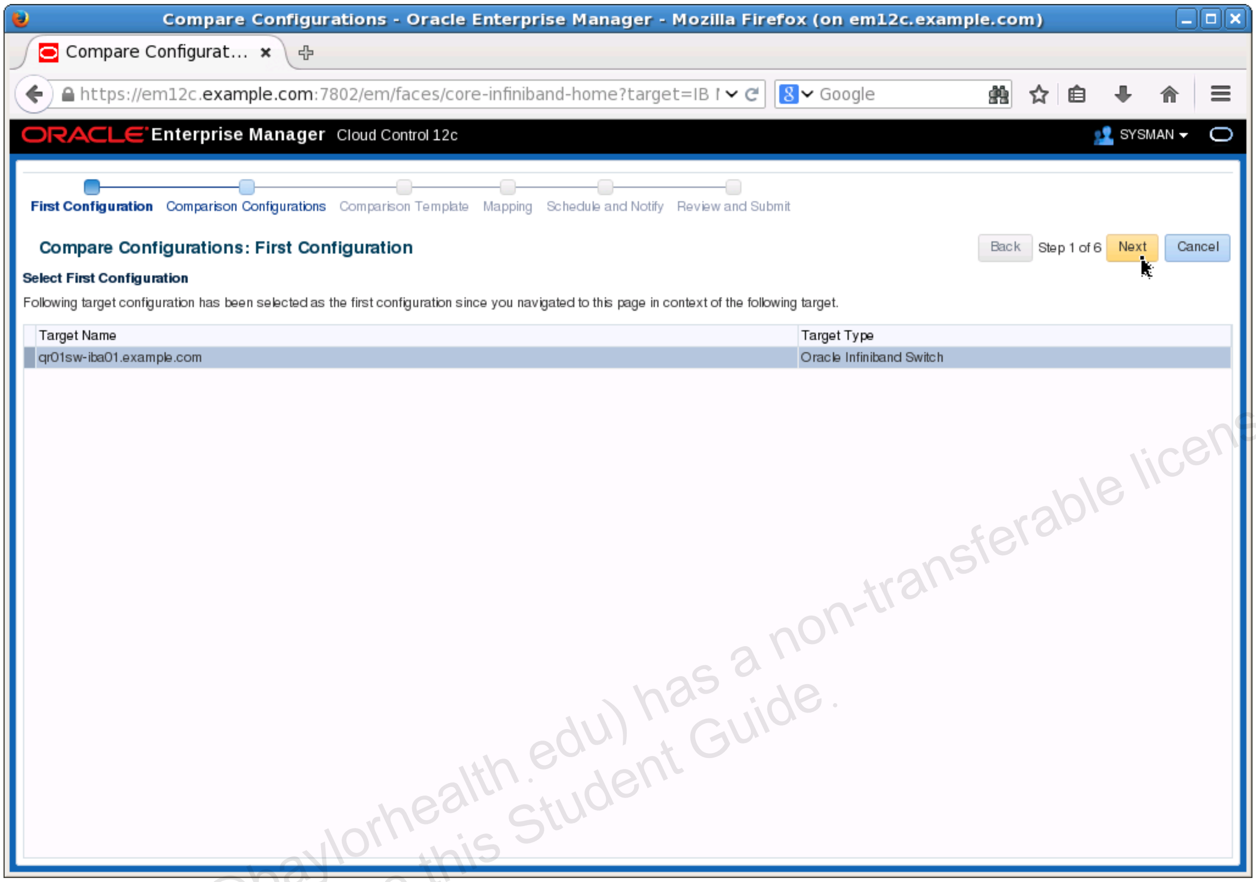
34. Click Infiniband Switch Version in the hierarchical list on the left side of the page and examine the available configuration attributes.



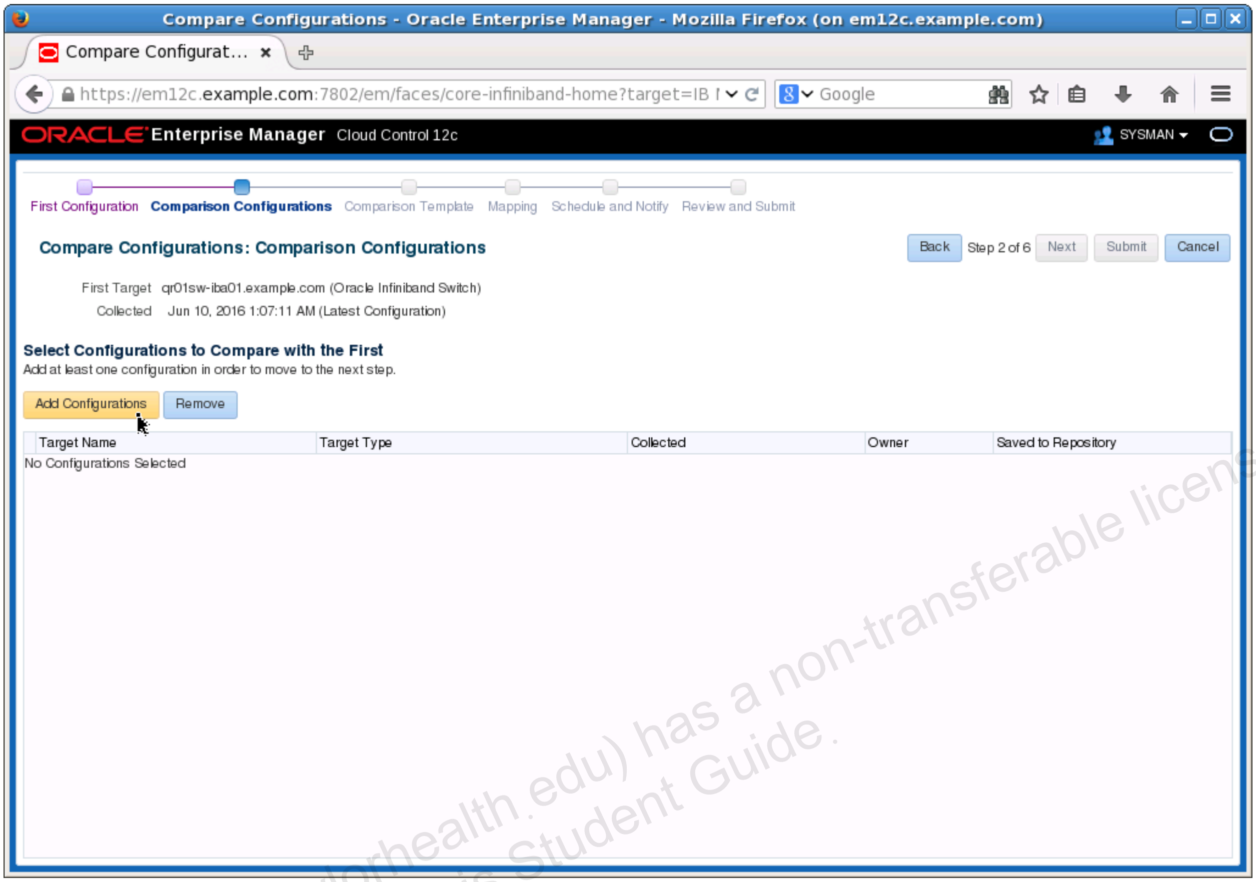
35. In a previous practice, you compared the configuration of your Exadata Storage Servers to ensure consistency across them. You can perform a similar check across the InfiniBand switches. To commence this process, select the Infiniband Switch > Configuration > Compare menu command.



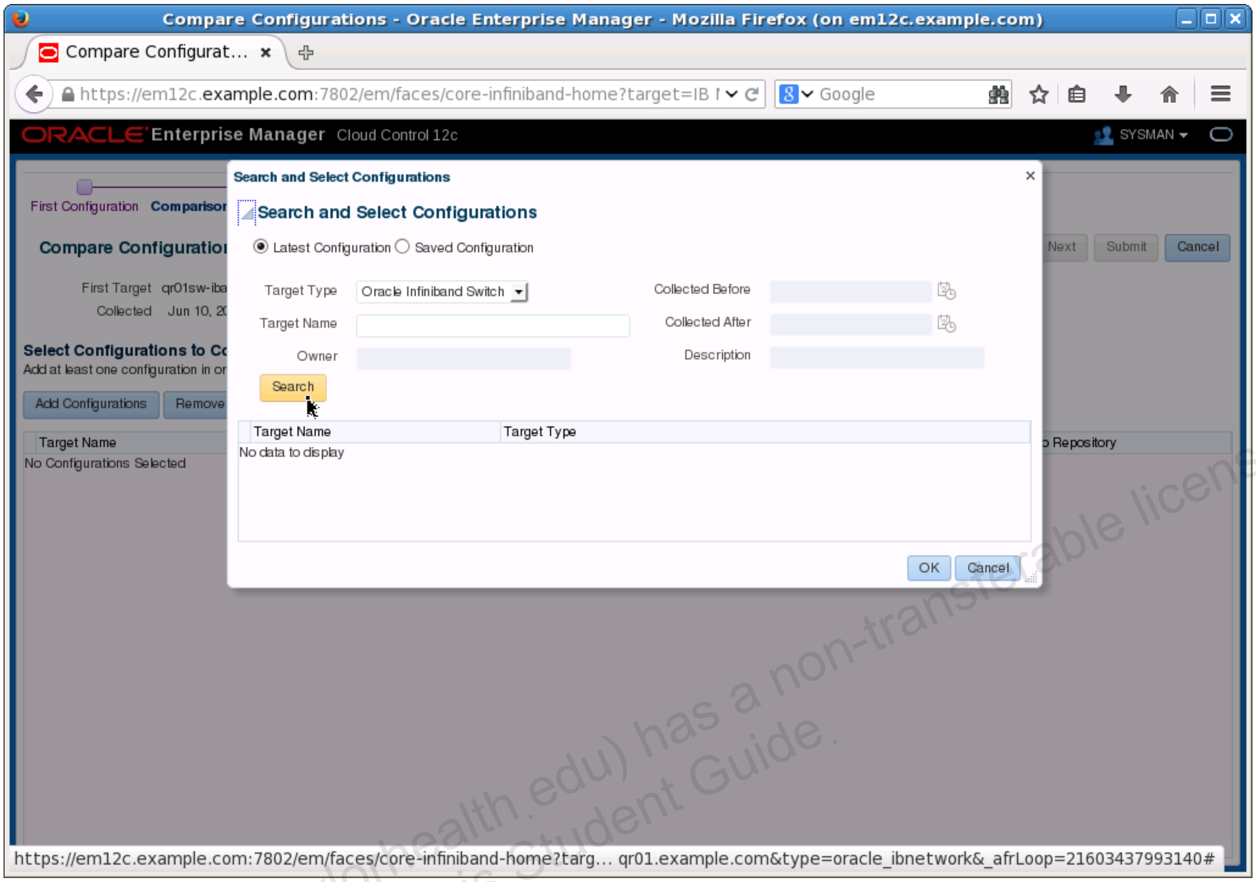
36. Confirm that `qr01sw-iba01.example.com` is selected as the comparison baseline (first configuration target). Then click Next to proceed.



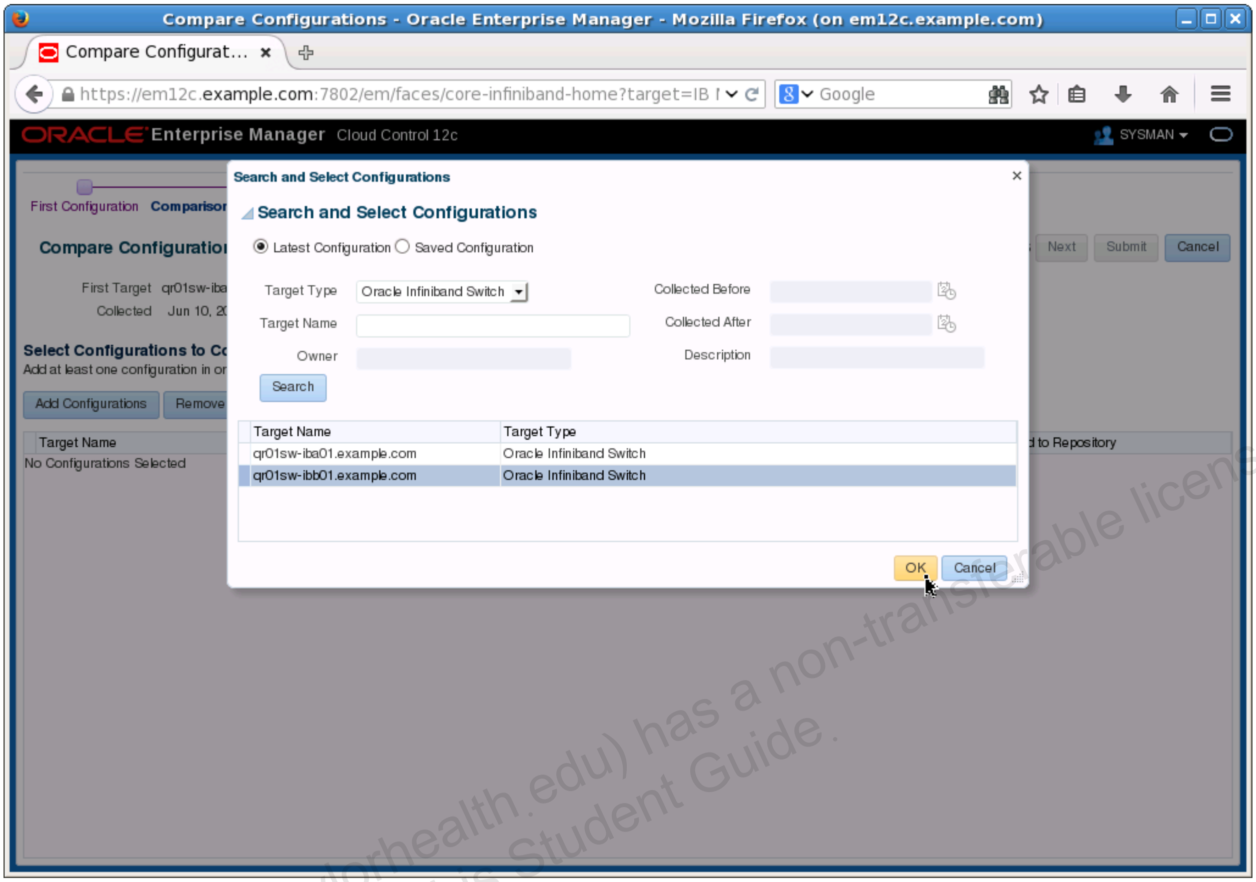
37. Click Add Configurations.



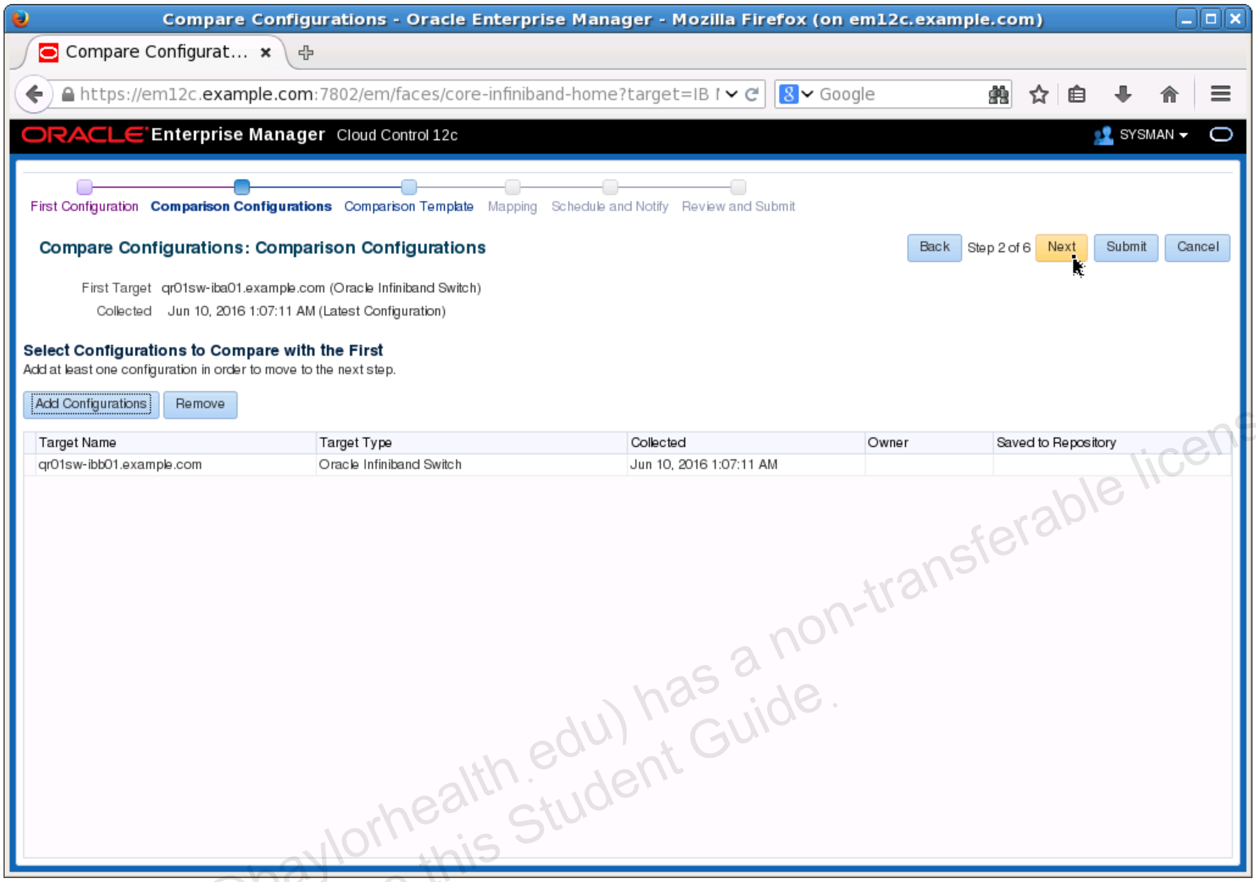
38. Click Search.



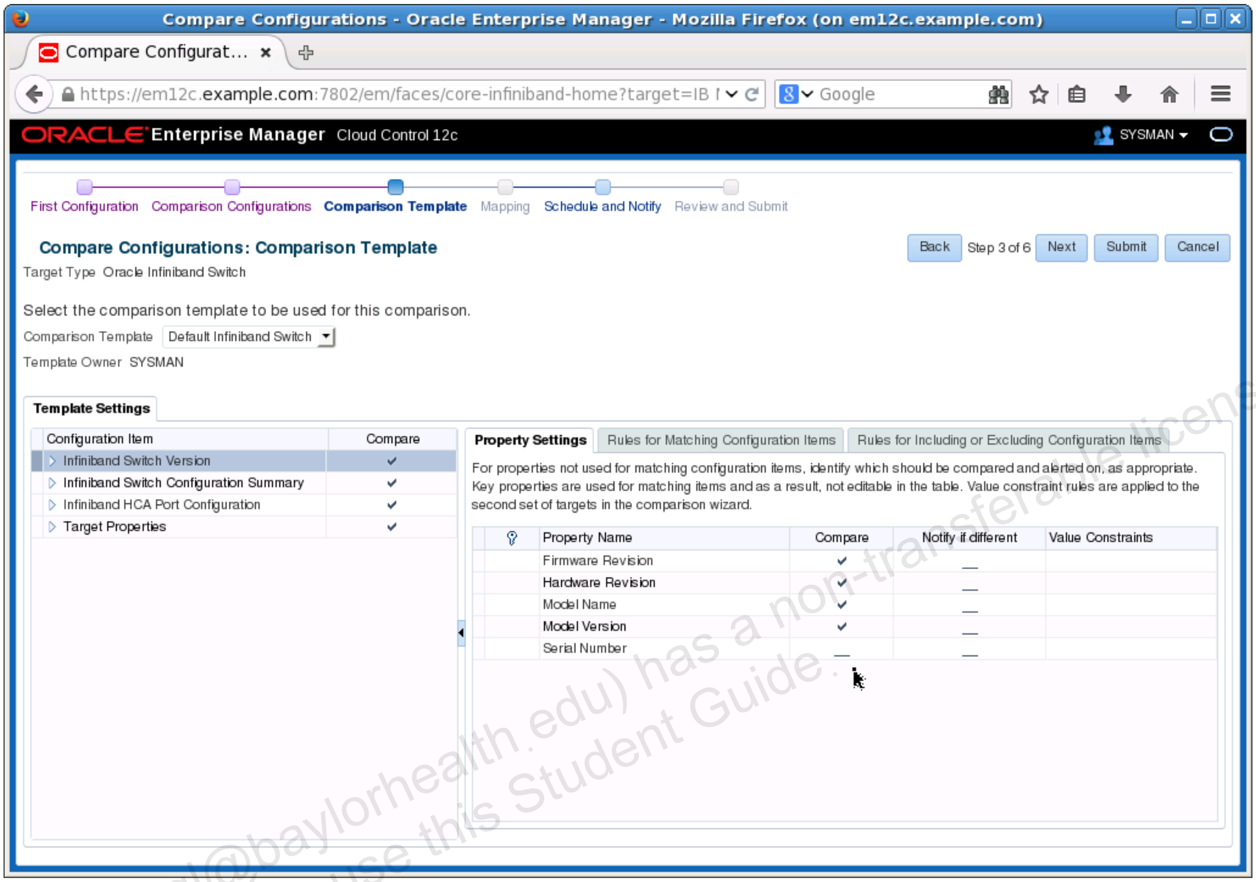
39. Select `qr01sw-ibb01.example.com` and click OK.



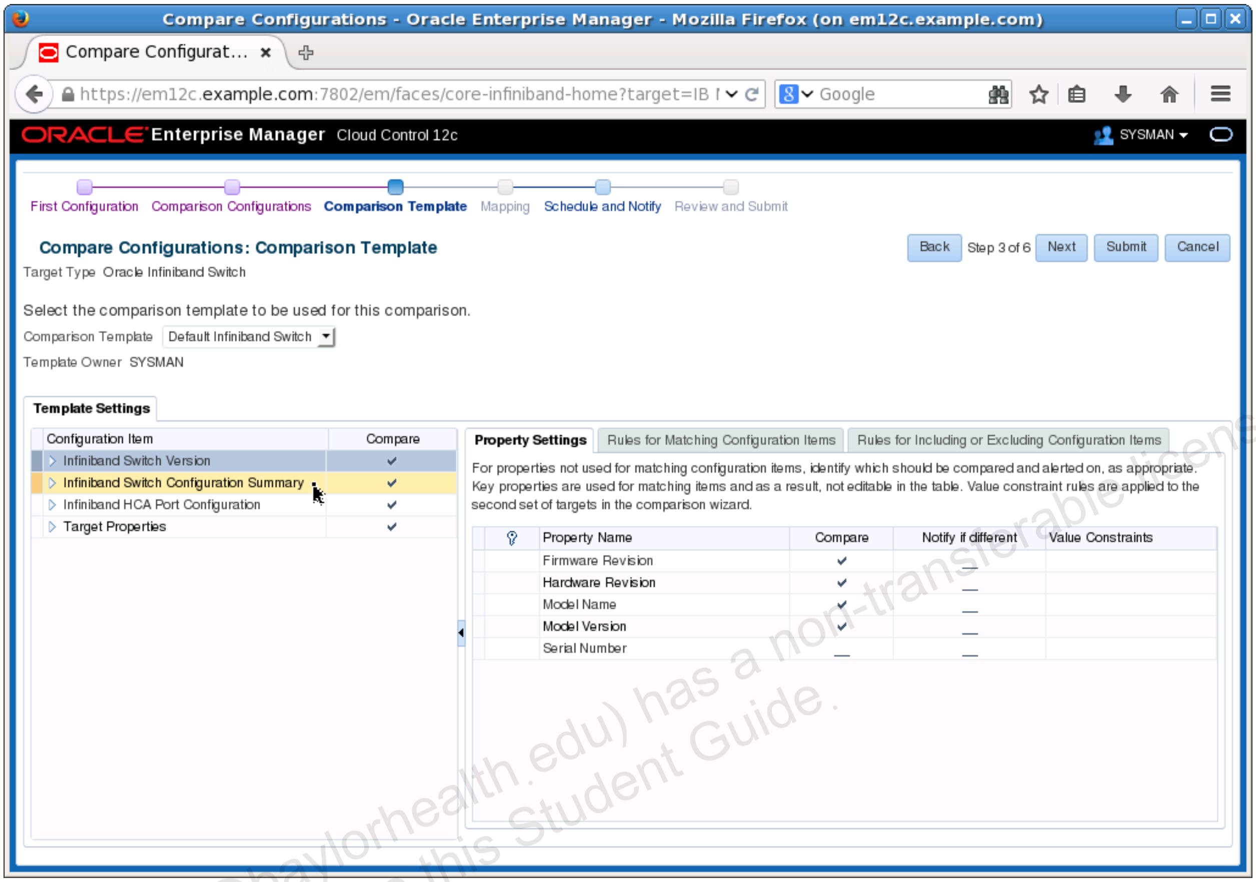
40. Confirm that `qr01sw-ibb01.example.com` is selected as the comparison configuration and click Next to proceed.



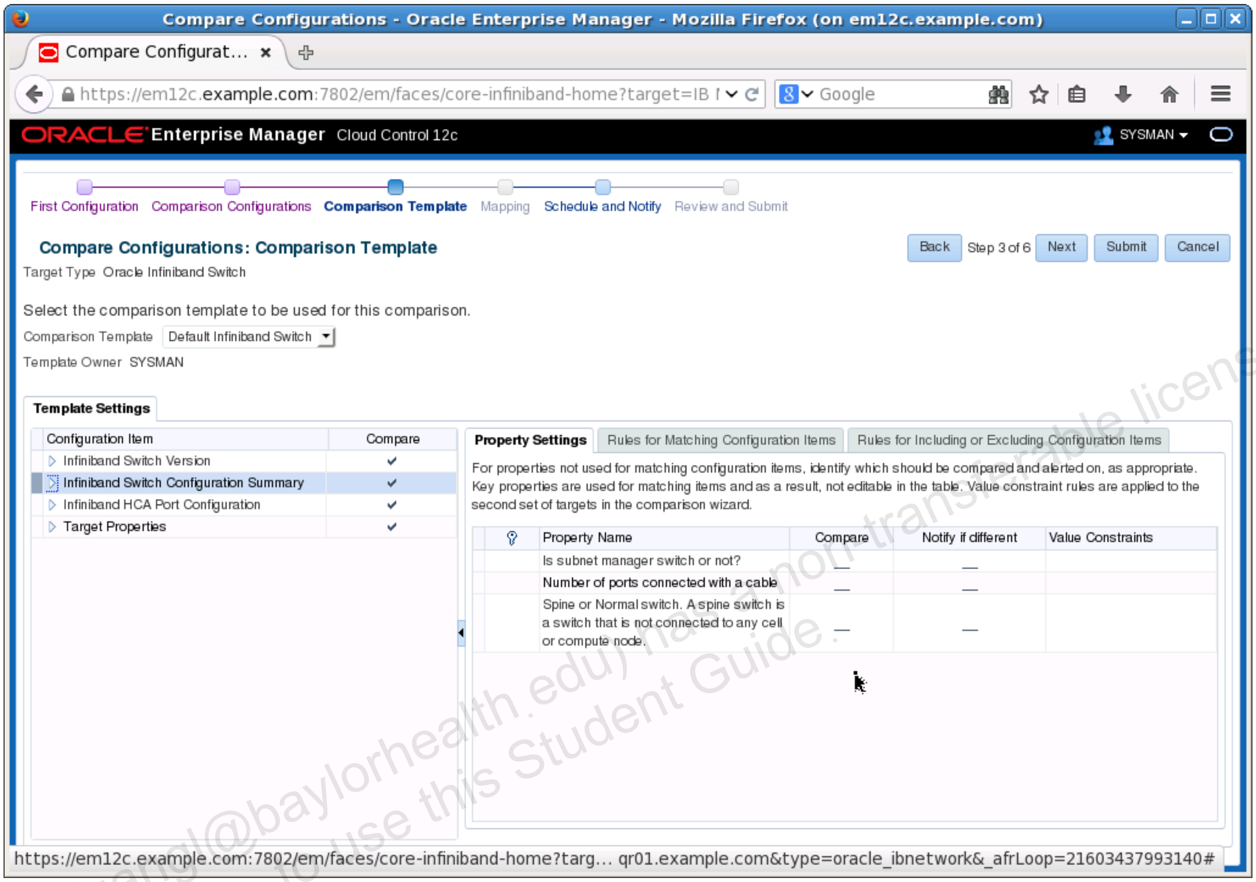
41. The comparison template defines how the comparison is performed. In the default comparison template for InfiniBand switches, serial number differences are ignored, which makes sense because the serial number should be unique for every switch.



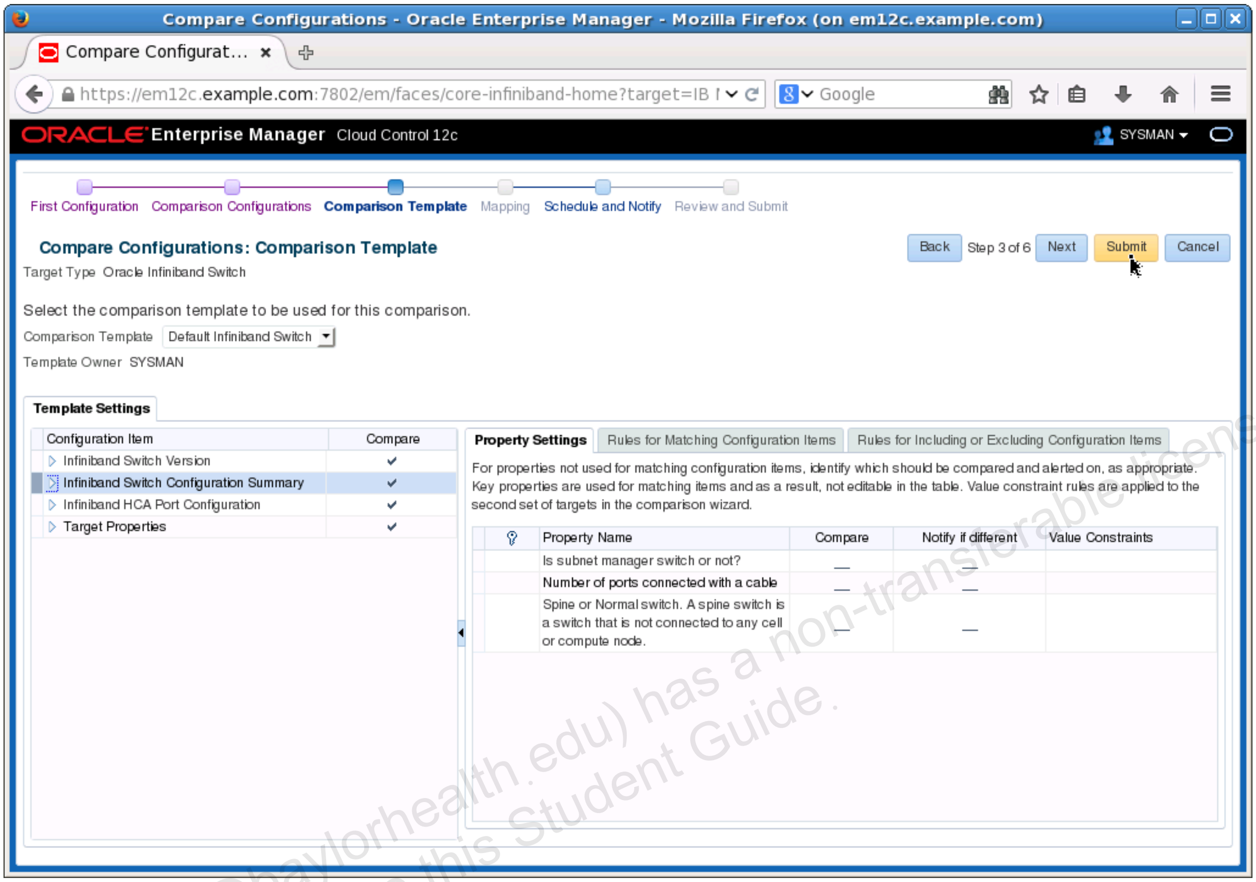
42. Click InfiniBand Switch Configuration Summary in the Template Settings area on the left side of the page.



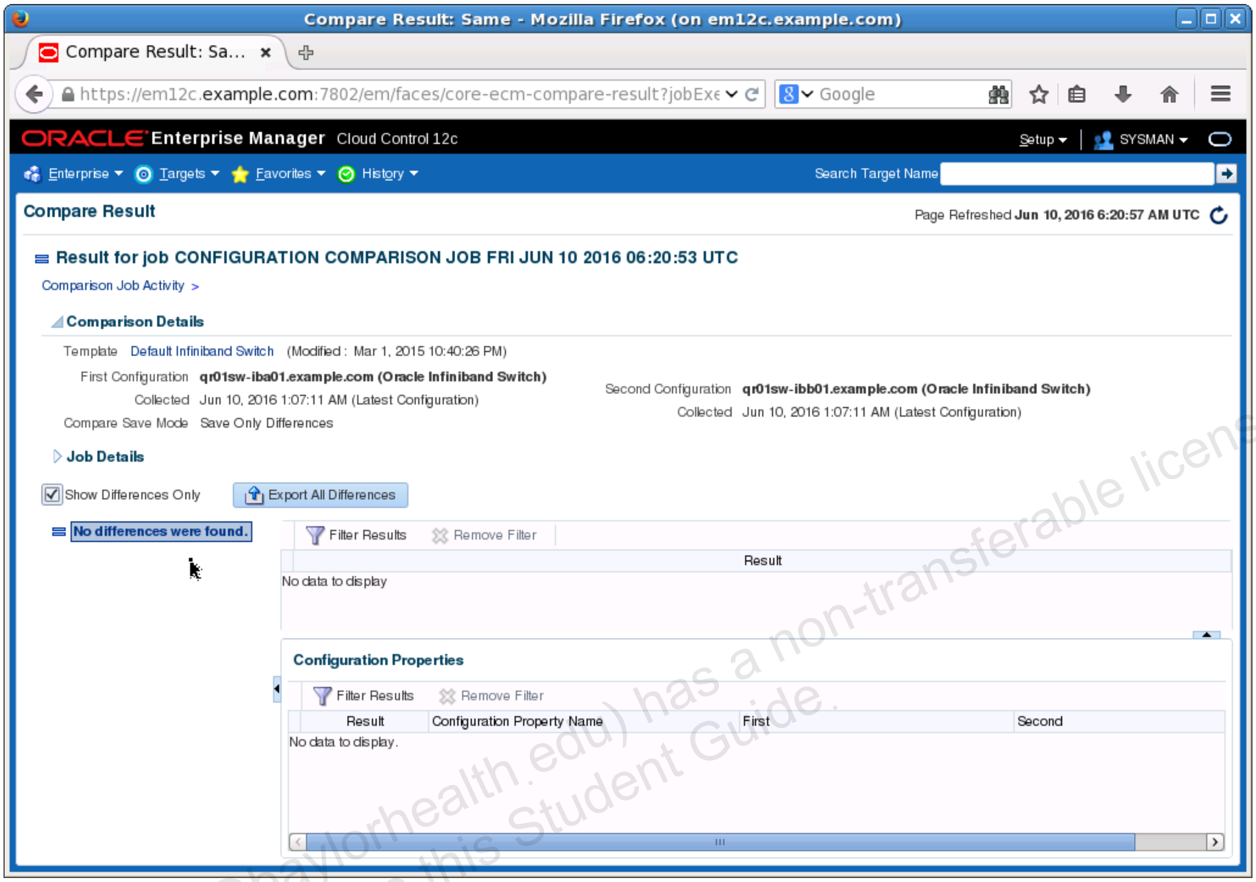
43. Here you can see that the default comparison template ignores a number of other differences. Depending on the goal of your comparison, this may make sense, or you may choose to modify the template and report the differences for a specific configuration property.



44. Take a moment to examine the rest of the comparison template. When you are ready, click Submit.



45. When the comparison completes, the Compare Result page appears. Examine the comparison result. Verify that the comparison found no differences.



Congratulations! You have examined the Exadata InfiniBand monitoring and administration capabilities provided by Enterprise Manager Cloud Control 12*c*.

Practices for Lesson 18: Monitoring Other Exadata Database Machine Components

Chapter 18

Practices for Lesson 18

There are no practices for this lesson.

Practices for Lesson 19: Other Useful Exadata Monitoring Tools

Chapter 19

Practices for Lesson 19

There are no practices for this lesson.

Practices for Lesson 20: Backup and Recovery

Chapter 20

Practices for Lesson 20: Overview

Practices Overview

In these practices, you will examine the backup and recovery optimizations that are enabled when Oracle Recovery Manager (RMAN) is used in conjunction with Exadata storage.

Practice 20-1: Backup Optimization

Overview

In this practice, you will see how Exadata optimizes the read I/O for an RMAN incremental backup.

Tasks

1. Establish a terminal connection to qr01dbadm01 as the oracle user, and configure your environment to access the dbm database (dbm1 instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Create a new tablespace named PRACTICE20, which you will use during this practice.

```
SQL> create tablespace practice20 datafile '+DATA_QR01' size
500M;

Tablespace created.

SQL>
```

4. Execute the following query (or execute the SQL script `/home/oracle/labs/lab20-01-04.sql`). The statistics show one of the ways in which Exadata can optimize file creation. In this case, the database server instructed the Exadata cells to format the data file on its behalf, rather than sending empty formatted data blocks over the storage interconnect. This can be seen by the value for `cell physical IO bytes saved during optimized file creation`, and also the very low value for `cell physical IO interconnect bytes` compared with `physical write total bytes`.

```
SQL> select a.name, b.value/1024/1024 MB
  2  from v$sysstat a, v$mystat b
  3  where a.statistic# = b.statistic# and
  4  (a.name in ('physical read total bytes',
  5              'physical write total bytes',
  6              'cell IO uncompressed bytes')
  7  or a.name like 'cell phy%');

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                          2.546875
physical write total bytes                                       1000.35938
cell physical IO interconnect bytes                                 3.59375
cell physical IO bytes saved during optimized file creation            1000
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes eligible for predicate offload                  1000
cell physical IO bytes saved by storage index                             0
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical IO interconnect bytes returned by smart scan                0
cell physical IO bytes saved by columnar cache                            0
cell physical write bytes saved by smart file initialization              0

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell IO uncompressed bytes                                                0
cell physical write IO bytes eligible for offload                         0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

SQL>
```

5. Move the `SH.CUST_INFO` table into the `PRACTICE20` tablespace;

```
SQL> alter table sh.cust_info move tablespace practice20;

Table altered.

SQL>
```

6. Enable block change tracking in your database and then query `V$BLOCK_CHANGE_TRACKING` to confirm the setting.

```
SQL> alter database enable block change tracking;

Database altered.

SQL> select status, filename from v$block_change_tracking;

STATUS
----------
FILENAME
--------------------------------------------------------------
ENABLED
+DATA_QR01/DBM/CHANGETRACKING/ctf.276.915085611


SQL>
```

7. Disconnect your SQL*Plus session, but leave SQL*Plus running.

```
SQL> disconnect
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0...
SQL>
```

8. Establish a second terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

9. Stop the `dbm` database. **Do not proceed until the database is stopped.**

```
[oracle@qr01dbadm01 ~]$ srvctl stop database -d dbm
[oracle@qr01dbadm01 ~]$ srvctl status database -d dbm
Instance dbm1 is not running on node qr01dbadm01
Instance dbm2 is not running on node qr01dbadm02
[oracle@qr01dbadm01 ~]$
```

10. Back in your SQL*Plus session, reconfigure your database to use archivelog mode by using the following sequence of commands:

```
SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1577058304 bytes
Fixed Size                  2924832 bytes
Variable Size             973082336 bytes
Database Buffers          587202560 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> alter database archivelog;

Database altered.

SQL> alter database open;

Database altered.

SQL>
```

11. Back in your second terminal window, restart the dbm database on all of your Exadata database servers.

```
[oracle@qr01dbadm01 ~]$ srvctl start database -d dbm
[oracle@qr01dbadm01 ~]$ srvctl status database -d dbm
Instance dbm1 is running on node qr01dbadm01
Instance dbm2 is running on node qr01dbadm02
[oracle@qr01dbadm01 ~]$
```

12. Launch Oracle Recovery Manager (RMAN) and connect to your database as shown below:

```
[oracle@qr01dbadm01 ~]$ rman target /

Recovery Manager: Release 12.1.0.2.0 - Production...

connected to target database: DBM (DBID=1355636031)

RMAN>
```

13. Configure RMAN to use two parallel execution channels for backup and recovery commands.

```
RMAN> configure device type disk parallelism 2;

using target database control file instead of recovery catalog
new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO
BACKUPSET;
new RMAN configuration parameters are successfully stored

RMAN>
```

14. Execute a level 0 incremental backup of the PRACTICE20 tablespace.

In this practice, you will perform backups on a single-file tablespace. It is not required to perform all backups this way on Database Machine. All the concepts in this practice apply to backups with different scopes, such as full database backups, and so on.

```
RMAN> backup as backupset incremental level 0 tablespace practice20;

Starting backup at 21-JUN-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=816 instance=dbm1 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=43 instance=dbm1 device type=DISK
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009
name=+DATA_QR01/DBM/DATAFILE/practice20.275.915085511
channel ORA_DISK_1: starting piece 1 at 21-JUN-16
channel ORA_DISK_1: finished piece 1 at 21-JUN-16
piece
handle=+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn0_tag20160621t063346_0.257.915
086029 tag=TAG20160621T063346 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:45
Finished backup at 21-JUN-16

RMAN>
```

15. Use the `LIST BACKUP` command to view details about the backup operation you just executed. Note the size of the backup and the amount of time required to perform the backup.

```
RMAN> list backup;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
1       Incr 0  251.19M    DISK        00:00:42     21-JUN-16
        BP Key: 1   Status: AVAILABLE  Compressed: NO  Tag: TAG20160621T063346
        Piece Name:
+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn0_tag20160621t063346_0.257.915086029
  List of Datafiles in backup set 1
  File LV Type Ckp SCN    Ckp Time  Name
  ---- -- ---- ---------- --------- ----
  9    0  Incr 13553891   21-JUN-16
+DATA_QR01/DBM/DATAFILE/practice20.275.915085511

RMAN>
```

16. Leave your RMAN session running and return to your SQL*Plus terminal. Execute the following query (or execute the SQL script `/home/oracle/labs/lab20-01-16.sql`) to show a selection of statistics relating to your RMAN session. You should see similar values for `physical read total bytes` and `physical write total bytes` because the level 0 incremental backup you executed in step 14 is essentially a full backup of the `PRACTICE20` tablespace.

```
SQL> select a.name, sum(b.value/1024/1024) MB
  2  from v$sysstat a, v$sesstat b, v$session c
  3  where a.statistic# = b.statistic# and
  4  b.sid = c.sid and
  5  upper(c.program) like 'RMAN%' and
  6  (a.name in ('physical read total bytes',
  7  'physical write total bytes',
  8  'cell IO uncompressed bytes')
  9  or a.name like 'cell phy%')
 10  group by a.name;

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell physical IO bytes eligible for predicate offload                     0
cell physical IO bytes saved by columnar cache                            0
cell physical IO interconnect bytes                              1114.23242
cell physical IO bytes saved during optimized file creation               0
cell physical IO interconnect bytes returned by smart scan                0
cell physical write IO bytes eligible for offload                         0
physical write total bytes                                       412.633301
cell physical write bytes saved by smart file initialization              0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes saved by storage index                             0
cell IO uncompressed bytes                                                0

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                         446.96582
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

SQL>
```

17. Execute the following `UPDATE` command (or execute the SQL script `/home/oracle/labs/lab20-01-17.sql`) to modify a small subset of the customer records scattered throughout the table. Commit the changes once the `UPDATE` statement completes.

```
SQL> update sh.cust_info set
  2  affinity_card = 0
  3  where remainder(cust_id,1000)=0;

2986 rows updated.

SQL> commit;

Commit complete.

SQL>
```

18. Back in your RMAN terminal, exit your current RMAN session and launch a fresh one. This will enable you to examine a fresh set of session statistics later in the practice.

```
RMAN> exit

Recovery Manager complete.
[oracle@qr01dbadm01 ~]$ rman target /

Recovery Manager: Release 12.1.0.2.0...

connected to target database: DBM (DBID=1355636031)

RMAN>
```

19. Execute a level 1 incremental backup of the `PRACTICE20` tablespace.

```
RMAN> backup as backupset incremental level 1 tablespace practice20;

Starting backup at 21-JUN-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=816 instance=dbm1 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=19 instance=dbm1 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009
name=+DATA_QR01/DBM/DATAFILE/practice20.275.915085511
channel ORA_DISK_1: starting piece 1 at 21-JUN-16
channel ORA_DISK_1: finished piece 1 at 21-JUN-16
piece
handle=+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn1_tag20160621t063717_0.258.915
086245 tag=TAG20160621T063717 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 21-JUN-16

RMAN>
```

20. As before, use the LIST BACKUP command to view details about the backup operation you just executed. Note the size of the backup and the amount of time required to perform the backup.

```
RMAN> list backup;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
1       Incr 0  251.19M    DISK        00:00:42     21-JUN-16
        BP Key: 1   Status: AVAILABLE  Compressed: NO  Tag: TAG20160621T063346
        Piece Name:
+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn0_tag20160621t063346_0.257.915086029
  List of Datafiles in backup set 1
  File LV Type Ckp SCN    Ckp Time  Name
  ---- -- ---- ---------- --------- ----
  9    0  Incr 13553891   21-JUN-16
+DATA_QR01/DBM/DATAFILE/practice20.275.915085511

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
2       Incr 1  23.23M     DISK        00:00:10     21-JUN-16
        BP Key: 2   Status: AVAILABLE  Compressed: NO  Tag: TAG20160621T063717
        Piece Name:
+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn1_tag20160621t063717_0.258.915086245
  List of Datafiles in backup set 2
  File LV Type Ckp SCN    Ckp Time  Name
  ---- -- ---- ---------- --------- ----
  9    1  Incr 13554098   21-JUN-16
+DATA_QR01/DBM/DATAFILE/practice20.275.915085511

RMAN>
```

21. Leave your RMAN session running and return to your SQL*Plus terminal. As you did in step 16, execute the following query (or execute the SQL script `/home/oracle/labs/lab20-01-16.sql`) to show a selection of statistics relating to your current RMAN session.

    This time the value for `physical read total bytes` will be smaller than the value you observed in step 16. This because block change tracking enables RMAN to concentrate its read requests on areas that have changed since the last backup. In addition, you should see that Exadata Smart Scan further optimized the RMAN reads. Compare `cell physical IO bytes eligible for predicate offload` with `cell physical IO interconnect bytes returned by smart scan` to determine the effect of the optimization.

```
SQL> select a.name, sum(b.value/1024/1024) MB
  2  from v$sysstat a, v$sesstat b, v$session c
  3  where a.statistic# = b.statistic# and
  4  b.sid = c.sid and
  5  upper(c.program) like 'RMAN%' and
  6  (a.name in ('physical read total bytes',
  7  'physical write total bytes',
  8  'cell IO uncompressed bytes')
  9  or a.name like 'cell phy%')
 10  group by a.name;

NAME                                                              MB
---------------------------------------------------------------- ----------
cell physical IO bytes eligible for predicate offload            253.015625
cell physical IO bytes saved by columnar cache                            0
cell physical IO interconnect bytes                               418.904297
cell physical IO bytes saved during optimized file creation               0
cell physical IO interconnect bytes returned by smart scan       23.1953125
cell physical write IO bytes eligible for offload                         0
physical write total bytes                                        183.984863
cell physical write bytes saved by smart file initialization              0
cell physical IO bytes saved during optimized RMAN file restore           0
cell physical IO bytes saved by storage index                             0
cell IO uncompressed bytes                                                0

NAME                                                              MB
---------------------------------------------------------------- ----------
physical read total bytes                                         439.442383
cell physical IO bytes sent directly to DB node to balance CPU            0
cell physical write IO host network bytes written during offloa           0

14 rows selected.

SQL>
```

22. Examine `V$BACKUP_DATAFILE` by using the following query (or execute the SQL script `/home/oracle/labs/lab20-01-22.sql`). The `BLOCKS_SKIPPED_IN_CELL` value associated with the level 1 incremental backup shows the amount of I/O that was saved by the optimization.

```
SQL> select file#,incremental_level,datafile_blocks,
  2  blocks,blocks_read,blocks_skipped_in_cell
  3  from v$backup_datafile;

     FILE# INCREMENTAL_LEVEL DATAFILE_BLOCKS     BLOCKS BLOCKS_READ
---------- ----------------- --------------- ---------- -----------
BLOCKS_SKIPPED_IN_CELL
----------------------
         9                 0           64000      32149       32896
                     0

         9                 1           64000       2970        2970
                 29417

SQL>
```

23. Exit your SQL*Plus and RMAN sessions.

Practice 20-2: Recovery Optimization

Overview

In this practice, you will exercise the recovery optimization that is provided by Exadata in conjunction with RMAN.

Assumptions

Before beginning this practice, you must complete Practice 20-1. Your ability to complete this practice depends on the existence of the backups that are created in practice 20-1.

Tasks

1. Establish a terminal connection to `qr01dbadm01` as the `oracle` user, and configure your environment to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

2. Connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

3. Offline the `PRACTICE20` tablespace and exit your SQL*Plus session.

```
SQL> alter tablespace practice20 offline;

Tablespace altered.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0...
[oracle@qr01dbadm01 ~]$
```

4. Use the following `su` command to launch the ASM command tool (ASMCMD) using the privileges of the `grid` OS user.

```
[oracle@qr01dbadm01 ~]$ su - grid -c ". oraenv;asmcmd -p"
Password: ????????
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid
ASMCMD [+] >
```

5. Navigate to the directory inside ASM that houses your database files.

```
ASMCMD [+] > cd +DATA_QR01/dbm/datafile
ASMCMD [+DATA_QR01/dbm/datafile] >
```

6. List your database files using the `ls` command. Note the name of the data file associated with the `PRACTICE20` tablespace.

```
ASMCMD [+DATA_QR01/dbm/datafile] > ls
PRACTICE20.275.915085511
SH.273.874117569
SOE.274.915083869
SYSAUX.261.874115745
SYSTEM.260.874115729
UNDOTBS1.262.874115757
UNDOTBS2.264.874115775
USERS.265.874115777
ASMCMD [+DATA_QR01/dbm/datafile] >
```

7. Delete the data file associated with the `PRACTICE20` tablespace, then verify the deletion and exit ASMCMD.

```
ASMCMD [+DATA_QR01/dbm/datafile] > rm PRACTICE20.275.915085511
ASMCMD [+DATA_QR01/dbm/datafile] > ls
SH.273.874117569
SOE.274.914043297
SYSAUX.261.874115745
SYSTEM.260.874115729
UNDOTBS1.262.874115757
UNDOTBS2.264.874115775
USERS.265.874115777
ASMCMD [+DATA_QR01/dbm/datafile] > exit
[oracle@qr01dbadm01 ~]$
```

8. Launch Oracle Recovery Manager (RMAN) and connect to your database as shown below:

```
[oracle@qr01dbadm01 ~]$ rman target /

Recovery Manager: Release 12.1.0.2.0 - Production...

connected to target database: DBM (DBID=1355636031)

RMAN>
```

9. Restore the `PRACTICE20` tablespace.

```
RMAN> restore tablespace practice20;

Starting restore at 21-JUN-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=41 instance=dbm1 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=816 instance=dbm1 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00009 to
+DATA_QR01/DBM/DATAFILE/practice20.275.915085511
channel ORA_DISK_1: reading from backup piece
+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn0_tag20160621t063346_0.257.915086029
channel ORA_DISK_1: piece
handle=+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn0_tag20160621t063346_0.257.915
086029 tag=TAG20160621T063346
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:55
Finished restore at 21-JUN-16

RMAN>
```

10. Recover the `PRACTICE20` tablespace.

```
RMAN> recover tablespace practice20;

Starting recover at 21-JUN-16
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
destination for restore of datafile 00009:
+DATA_QR01/DBM/DATAFILE/practice20.275.915086505
channel ORA_DISK_1: reading from backup piece
+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn1_tag20160621t063717_0.258.915086245
channel ORA_DISK_1: piece
handle=+RECO_QR01/DBM/BACKUPSET/2016_06_21/nnndn1_tag20160621t063717_0.258.915
086245 tag=TAG20160621T063717
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:07

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

Finished recover at 21-JUN-16

RMAN>
```

11. Leave your RMAN session running and establish a second terminal connection to your database server. Configure the terminal session to access the `dbm` database (`dbm1` instance).

```
$ ssh oracle@qr01dbadm01
oracle@qr01dbadm01's password: ????????
[oracle@qr01dbadm01 ~]$ . oraenv
ORACLE_SID = [oracle] ? dbm
The Oracle base has been set to /u01/app/oracle
[oracle@qr01dbadm01 ~]$ export ORACLE_SID=dbm1
[oracle@qr01dbadm01 ~]$
```

12. In your second terminal session, connect to your database with SQL*Plus. Log in as the database administrator.

```
[oracle@qr01dbadm01 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production...

SQL>
```

13. Execute the following query (or execute the SQL script `/home/oracle/labs/lab20-02-13.sql`) to show a selection of statistics relating to your RMAN recovery session. You should see a value for `cell physical IO bytes saved during optimized RMAN file restore`. When RMAN restores a file, any blocks in the file that have not been altered since the file was first formatted can be re-created by Exadata. This optimization removes the need to transport empty formatted blocks across the storage network. Rather, RMAN is able to instruct Exadata to conduct the I/O on its behalf, which is similar to the way that optimized file creation is performed.

```
SQL> select a.name, sum(b.value/1024/1024) MB
  2  from v$sysstat a, v$sesstat b, v$session c
  3  where a.statistic# = b.statistic# and
  4  b.sid = c.sid and
  5  upper(c.program) like 'RMAN%' and
  6  (a.name in ('physical read total bytes',
  7  'physical write total bytes',
  8  'cell IO uncompressed bytes')
  9  or a.name like 'cell phy%')
 10  group by a.name;

NAME                                                                     MB
---------------------------------------------------------------- ----------
cell physical IO bytes eligible for predicate offload              11.71875
cell physical IO bytes saved by columnar cache                             0
cell physical IO interconnect bytes                              1343.23389
cell physical IO bytes saved during optimized file creation                0
cell physical IO interconnect bytes returned by smart scan                 0
cell physical write IO bytes eligible for offload                          0
physical write total bytes                                       530.328125
cell physical write bytes saved by smart file initialization               0
cell physical IO bytes saved during optimized RMAN file restore    11.71875
cell physical IO bytes saved by storage index                              0
cell IO uncompressed bytes                                                 0

NAME                                                                     MB
---------------------------------------------------------------- ----------
physical read total bytes                                        304.796387
cell physical IO bytes sent directly to DB node to balance CPU             0
cell physical write IO host network bytes written during offloa            0

14 rows selected.

SQL>
```

14. Bring the `PRACTICE20` tablespace back online.

```
SQL> alter tablespace practice20 online;

Tablespace altered.
```

15. Exit your SQL*Plus and RMAN sessions.

Practices for Lesson 21: Exadata Database Machine Maintenance Tasks

Chapter 21

Practices for Lesson 21

There are no practices for this lesson.

Practices for Lesson 22: Patching Exadata Database Machine

Chapter 22

Practices for Lesson 22

There are no practices for this lesson.

Practices for Lesson 23: Exadata Database Machine Automated Support Ecosystem

Chapter 23

Practices for Lesson 23

There are no practices for this lesson.