**PY and DW Friday pre steps**

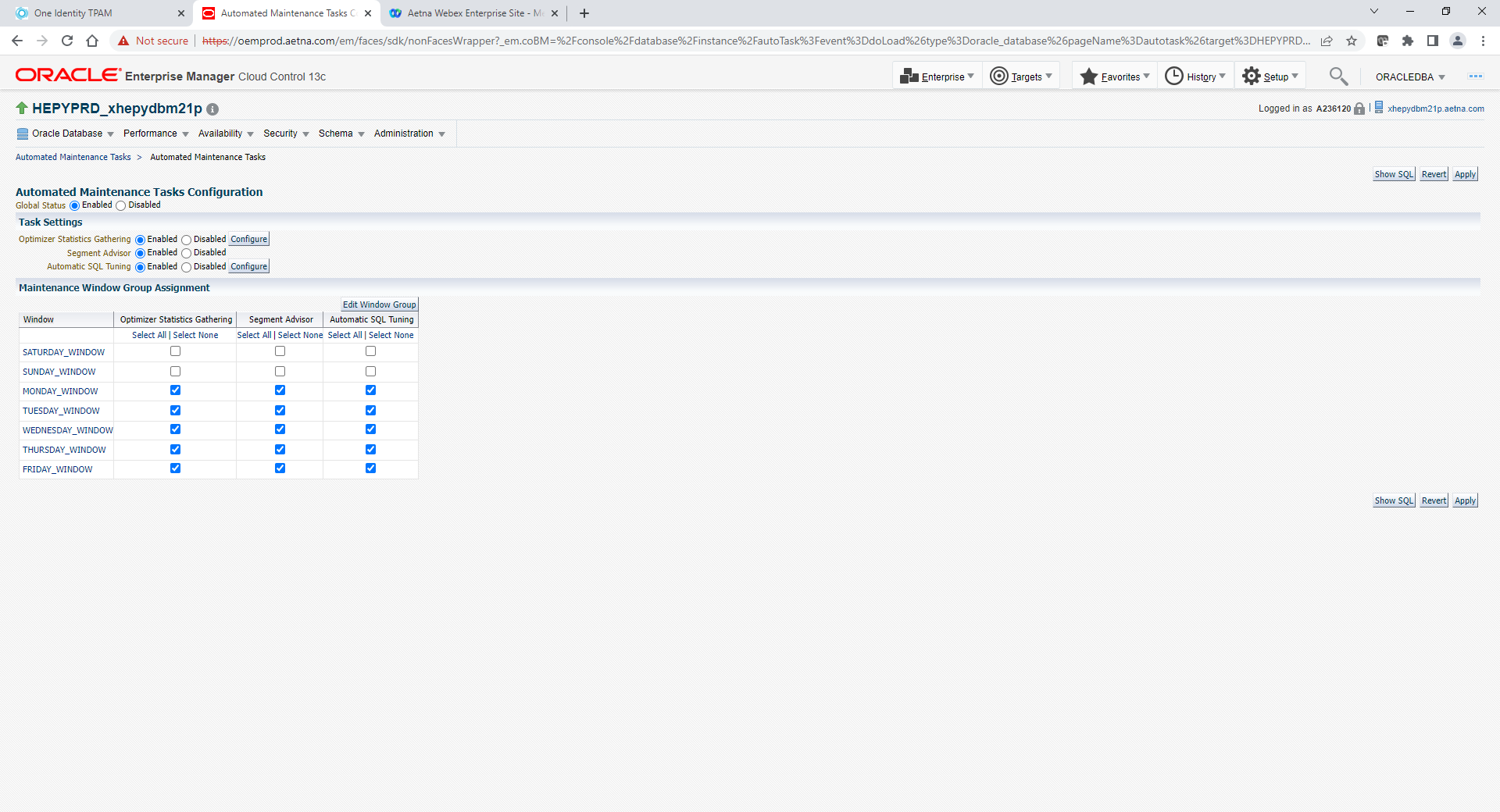
**Adjust schedule for both HEPYPRD and HEDWPRD Level 0 jobs to start earlier on Saturday**. Based on latest average completion time figure out best time to start with goal to finish at least 1 hour prior to start of HRP upgrade.

**Adjust schedule for both HEPYPRD and HEDWPRD Archivelog jobs schedule** so they can finish right before HRP Upgrade start time. This will ensure FLASH\_01 Disk space freed up.

**Adjust schedule for both HEPYRD (StandBy) and HEDWRPD (StandBy) archive purge jobs** **schedule** so they can finish right before HRP Upgrade start time. This will ensure FLASH\_01 Disk space freed up.

**Disable OEM Saturday and Sunday and Automated Maintenance Tasks for both HEPYPRD and HEDWPRD.** See example below

These jobs occasionally run longer into HRP upgrade start time and we want to keep them down during HRP upgrade.



**PY and DW Saturday pre upgrade steps (you can do this while HRP bringing apps down or earlier during the day)**

-- Comment out PY crontab stats jobs. **If you see stats job running prior to Upgrade start time kill it.**

-- Check make sure PY Level 0 and DW Level 0 completed (should start earlier on Saturday and should be completed before HRP upgrade start). **If backup still running prior to Upgrade start time kill it**

-- Disable PY OEM Kill Job (comment out crontab job below)

#KILL JOB

\*/5 \* \* \* \* /home/oracle/eb/kill\_job/kill\_jobs.sh HEPYPRD > /dev/null 2>/dev/null

-- Check make sure FLASH\_01 space freed up on both PY Primary, PY Standby and DW Primary, DW Standby servers.

--Example below. Look for FLASH\_01 DISK space.

+ASM> asmcmd lsdg

\*\*\*\* HEPYPRD \*\*\*\*\*\*

PLACE HOLDER FOR EXTRA DBA TASKS THAT NEEDS TO BE DONE PRIOR TO UPGRADE. Like changing DB parameters, bounce DB, index related work A lot of times DBA’s assigned extra tasks during HRP upgrade window to make changes not related to HRP upgrade…

**~~Deploy PY Upgrade Automation (only add crontab entries all other completed)~~**

**Process parameter changes Primary and Standby, bounce database. Follow srvctl\_serverControl\_bounce\_DB.txt**

In Both Primary and Standby

alter system set db\_recovery\_file\_dest\_size = 1000G scope=both;

\*\*\*\* HEDWPRD \*\*\*\*\*\*

PLACE HOLDER FOR EXTRA DBA TASKS THAT NEEDS TO BE DONE PRIOR TO UPGRADE. Like changing DB parameters, bounce DB, index related work A lot of times DBA’s assigned extra tasks during HRP upgrade window to make changes not related to HRP upgrade…

**~~Deploy DW Upgrade Automation (only add crontab entries all other completed)~~**

In Both Primary and Standby

alter system set db\_recovery\_file\_dest\_size = 800G scope=both;

**PY Saturday upgrade steps (after apps down)**

Primary Server : xhepydbm1p (HEPYPRD)

StandBy Server: xhepydbw21p (HEPYPRD)

--Comment heartbeat in cron on **Primary** server

#\*/5 \* \* \* \* /home/oracle/tls/rman/heartbeat.ksh HEPYPRD AEDBA > /dev/null 2>&1

--Manually update RMAN HeartBeat Table in **Primary** Database. Document outputs for reference.

/home/oracle/tls/rman/heartbeat.ksh HEPYPRD AEDBA

DBNAME Start Time

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

HEPYPRD 10-08-2022 06:48:53 PM

Previous TimeStamp

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

10-08-2022 06:45:01 PM

Current TimeStamp

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

10-08-2022 06:48:53 PM

End Time

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

10-08-2022 06:48:53 PM

--Get Flashback Database status in **Primary** database (Should return NO)

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

--Get Flashback Database status in **Standby** database (Should return NO)

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

From **Primary** database

alter system set db\_recovery\_file\_dest\_size = 1000G scope=both;

SQL> show parameter db\_recovery\_file\_dest

From **Standby** database

alter system set db\_recovery\_file\_dest\_size = 1000G scope=both;

SQL> show parameter db\_recovery\_file\_dest

**--Create the Guaranteed Restore Point**

**--To be run on the Primary database server - Script will also configure the standby database**

cd $SCRIPTS

create\_guaranteed\_restore\_point.sh HEPYPRD b4\_app\_upgrade

NAME INCARNATION SCN TIME GUAR

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

B4\_APP\_UPGRADE 2 13624228151450 10-JUL-22 12.05.14.000000000 AM YES

Uncomment the heartbeat script in cron on primary server

\*/5 \* \* \* \* /home/oracle/tls/rman/heartbeat.ksh HEPYPRD AEDBA > /dev/null 2>&1

**Confirm flashback\_database configured in standby database**

Run below select in standby database

select flashback\_on from v$database;

**DW Saturday upgrade steps**

Primary Server: xhedwdbm21p (HEDWPRD)

Standby Server: xhedwdbw21p (HEDWPRD)

--Run **AEDBA.LOCKUSER** procedure in HEDWPRD.

sqlplus / as sysdba

EXECUTE aedba.LOCKUSER;

--Manually kill N and A ID connections if you see any. Should not be any active at this point.

--Comment heartbeat in cron on **Primary** server

#\* /home/oracle/tls/rman/heartbeat.ksh HEDWPRD AEDBA

--Manually update RMAN HeartBeat Table in **Primary** Database. Document for reference.

/home/oracle/tls/rman/heartbeat.ksh HEDWPRD AEDBA

DBNAME Start Time

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

HEDWPRD 10-08-2022 06:59:13 PM

Previous TimeStamp

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

10-08-2022 06:55:01 PM

Current TimeStamp

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

10-08-2022 06:59:13 PM

End Time

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

10-08-2022 06:59:13 PM

-- Get Flashback Database status in **Primary** database (Should return NO)

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

--Get Flashback Database status in **Standby** database (Should return NO)

sql => select flashback\_on from v$database;

FLASHBACK\_ON

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

NO

From **Primary** database

alter system set db\_recovery\_file\_dest\_size = 800G scope=both;

SQL> show parameter db\_recovery\_file\_dest

From **Standby** database

alter system set db\_recovery\_file\_dest\_size = 800G scope=both;

SQL> show parameter db\_recovery\_file\_dest

**--Create the Guaranteed Restore Point**

**--To be run on the Primary database server - Script will also configure the standby database**

cd $SCRIPTS

create\_guaranteed\_restore\_point.sh HEDWPRD b4\_app\_upgrade

NAME INCARNATION SCN TIME GUAR

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

B4\_APP\_UPGRADE 2 13633812145040 10-JUL-22 12.10.11.000000000 AM YES

Uncomment the heartbeat script in cron on primary server

\*/5 \* \* \* \* /home/oracle/tls/rman/heartbeat.ksh HEDWPRD AEDBA > /dev/null 2>&1

**Confirm flashback\_database configured in standby database**

Run below select in Standby database

select flashback\_on from v$database;

**-----------------------------> Sunday morning steps (whenever time you get green light after checkout).**

**PY Sunday steps**

--To Check flashback logs size that being generated during upgrade

. oraenv

HEPYPRD

sqlplus / as sysdba

SELECT

SUM(bytes/1024/1024/1024) as "Size(GB)"

FROM

v$flashback\_database\_logfile;

**Get total from Size(GB) column and note size here for reference: 264 GB**

**Drop restore points**

**Drop Guaranteed Restore Point and turn off flashback\_database in the standby database**

**To be run on the primary database server - Script will also configure the standby database**

**Primary**:

cd $SCRIPTS

drop\_guaranteed\_restore\_point.sh HEPYPRD b4\_app\_upgrade

Confirm restore point dropped

sqlplus / as sysdba

select name from v$restore\_point where guarantee\_flashback\_database='YES';

**Confirm flashback\_database not configured in standby database**

Run below select in standby database

select flashback\_on from v$database;

Resize the FRA in the primary database and the standby database

Primary DataBase:

sqlplus / as sysdba

alter system set db\_recovery\_file\_dest\_size = 7G scope=both;

show parameter db\_recovery\_file\_dest\_size;

Standby DataBase:

sqlplus / as sysdba

alter system set db\_recovery\_file\_dest\_size = 7G scope=both;

show parameter db\_recovery\_file\_dest\_size;

**Uncomment stats jobs (reschedule to start Sunday jobs that supposed to start earlier. Remember to put old schedule back after that.)**

PLACE HOLDER FOR EXTRA DBA TASKS. Like changing DB parameters, bounce DB, index related work, BLOB table migration to Big File Tablespaces etc. A lot of times DBA’s assigned extra tasks during HRP upgrade window to make changes not related to HRP upgrade…

\*\*\*\*\* HEPYPRD\*\*\*\*\*

select INDEX\_NAME

from dba\_indexes

where owner = 'PROD'

and VISIBILITY = 'INVISIBLE';

ALTER INDEX PROD.SPPLR\_LATEST VISIBLE;

ALTER INDEX PROD.FK\_SUPPLIER\_PRVDRIDCD\_DATES VISIBLE;

ALTER INDEX PROD.VIDDATES\_PRACT\_LATEST VISIBLE;

ALTER INDEX PROD.PKDATES\_PRACT\_LATEST VISIBLE;

ALTER INDEX PROD.ALIAS\_PRACT\_LATEST VISIBLE;

ALTER INDEX PROD.V\_PRACT\_HCCID\_INDX VISIBLE;

ALTER INDEX PROD.PRACT\_LATEST\_IDENT\_NBR VISIBLE;

ALTER INDEX PROD.FK\_PRACT\_L\_DENT\_NBR\_PRACT\_EST VISIBLE;

ALTER INDEX PROD.FK\_PRACT\_LAT\_IDENT\_NBR\_CD\_E\_RY VISIBLE;

ALTER INDEX PROD.PRACT\_LATEST\_LICENSE\_NBR VISIBLE;

ALTER INDEX PROD.FK\_PRACT\_LA\_ENSE\_NBR\_PRAC\_EST VISIBLE;

ALTER INDEX PROD.V\_PRACT\_NPI\_INDX VISIBLE;

ALTER INDEX PROD.V\_PRACT\_NPI\_UPPER\_IX VISIBLE;

ALTER INDEX PROD.PRACT\_LATEST\_IDENT\_NBR\_TXT VISIBLE;

ALTER INDEX PROD.PRACT\_LATEST\_LICENSE\_NBR\_TXT VISIBLE;

ALTER INDEX PROD.VIDDATES\_SPPLR\_LATEST VISIBLE;

ALTER INDEX PROD.PKDATES\_SPPLR\_LATEST VISIBLE;

ALTER INDEX PROD.ALIAS\_SPPLR\_LATEST VISIBLE;

ALTER INDEX PROD.SPPLR\_LATEST\_IDENT\_NBR VISIBLE;

ALTER INDEX PROD.FK\_SPPLR\_L\_DENT\_NBR\_SPPLR\_EST VISIBLE;

ALTER INDEX PROD.FK\_SPPLR\_LAT\_IDENT\_NBR\_CD\_E\_RY VISIBLE;

ALTER INDEX PROD.UP\_PRIMARY\_NAME\_SPPLR\_LATEST VISIBLE;

ALTER INDEX PROD.FK\_SUPPLIER\_LATEST\_TAX\_ENTITY VISIBLE;

ALTER INDEX PROD.SPPLR\_LATEST\_IDENT\_NBR\_TXT VISIBLE;

ALTER INDEX PROD.PRACT\_LATEST VISIBLE;

ALTER INDEX PROD.FK\_SUPPLIER\_HCCID\_ENDORDT VISIBLE;

ALTER INDEX PROD.V\_SUPPLIER\_LATEST\_PCP VISIBLE;

CREATE INDEX PROD.PAYABLE\_AEDBA\_3

ON PROD.PAYABLE("TENANT\_ID", "SPPLR\_ID", "PAYABLE\_ID")

TABLESPACE INDX4 ONLINE PARALLEL 6 ;

ALTER INDEX PROD.PAYABLE\_AEDBA\_3 NOPARALLEL;

~~ALTER INDEX PROD.FK\_PAYABLE\_SUPPLIER INVISIBLE~~

~~/~~

DROP INDEX PROD.FK\_PAYABLE\_SUPPLIER

/

CREATE INDEX PROD.PAYABLE\_AEDBA\_2

ON PROD.PAYABLE("SPPLR\_ID", "SUBTYPE\_NM", "CYCLE\_ID", "BANK\_ACCT\_ID", "TENANT\_ID", "RELEASE\_DT")

TABLESPACE INDX4 ONLINE PARALLEL 6;

ALTER INDEX PROD.PAYABLE\_AEDBA\_2 NOPARALLEL;

CREATE INDEX PROD.ADJUSTMENT\_PAYABLE\_AEDBA\_1

ON PROD.ADJUSTMENT\_PAYABLE("TENANT\_ID", "PAYABLE\_ID")

TABLESPACE INDX4 ONLINE PARALLEL 6;

ALTER INDEX PROD.ADJUSTMENT\_PAYABLE\_AEDBA\_1 NOPARALLEL;

!!!

**Deploy latest kill script changes**

**DW Sunday steps**

-- To Check flashback logs size that being generated during upgrade

. oraenv

HEDWPRD

sqlplus / as sysdba

SELECT

SUM(bytes/1024/1024/1024) as "Size(GB)"

FROM

v$flashback\_database\_logfile;

**Get total from Size(GB) column and note size here for reference: 24 GB**

**Drop restore points**

cd $SCRIPTS

drop\_guaranteed\_restore\_point.sh HEDWPRD b4\_app\_upgrade

**Confirm restore point dropped**

sqlplus / as sysdba

select name from v$restore\_point where guarantee\_flashback\_database='YES';

**Confirm flashback\_database not configured in standby database**

Run below select in standby database

select flashback\_on from v$database;

Resize the FRA in the primary database and the standby database

Primary DataBase:

sqlplus / as sysdba

alter system set db\_recovery\_file\_dest\_size = 7G scope=both;

show parameter db\_recovery\_file\_dest\_size;

Standby DataBase:

sqlplus / as sysdba

alter system set db\_recovery\_file\_dest\_size = 7G scope=both;

show parameter db\_recovery\_file\_dest\_size;

--Following script will add appropriate grants to newly created tables during upgrade. It will ask for **database name** and **number of days or hours** to check for new objects back. Just put 12 hours like this 12/24 or query database for last PROD\_DW table create date to get an idea how many hours new tables were added.

cd /home/oracle/tls/security

grant\_permissions.ksh

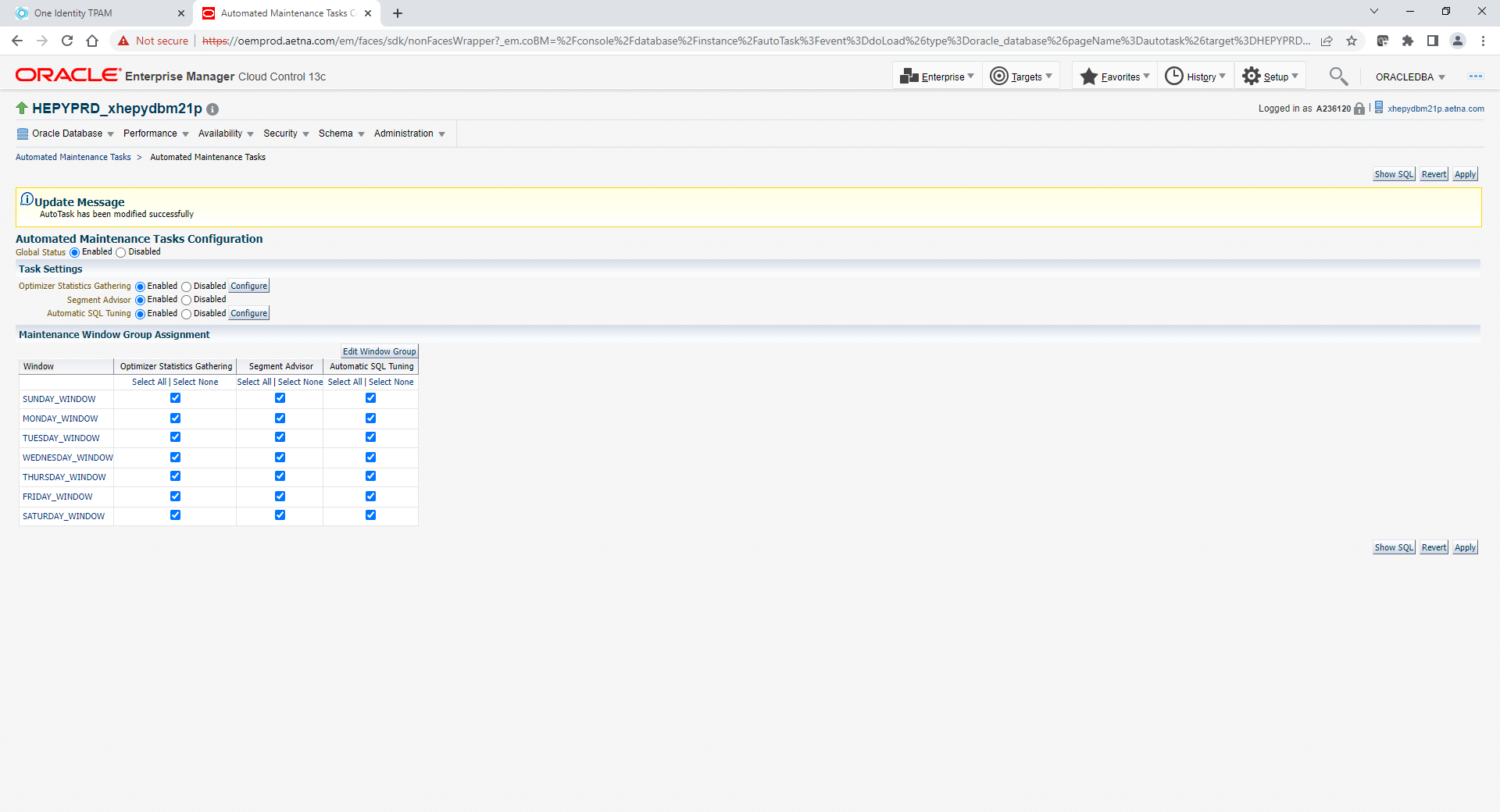
-- Run **AEDBA.UNLOCKUSER** procedure in HEDWPRD

sqlplus / as sysdba

EXECUTE aedba.UNLOCKUSER;

--Deploy DSR to update views. Pam should submit DSR with bunch of scripts mostly views to redeploy.

--Enable OEM Saturday and Sunday Automated Maintenance Tasks for both PY and DW



--Uncomment PY crontab stats job that were commented out night before. If it already not completed. If it past Sunday’s job start time just adjust start time based on current time.

-- Enable PY OEM Kill Job (Uncomment job below)

#KILL JOB

\*/5 \* \* \* \* /home/oracle/eb/kill\_job/kill\_jobs.sh HEPYPRD > /dev/null 2>/dev/null