Table of Contents

[Oracle databases migration approaches from AIX to Red Hat Linux 3](#_Toc435619115)

[1. Datapump only 4](#_Toc435619116)

[1.1. RHEL: Check mountpints, disk space 4](#_Toc435619117)

[1.2. RHEL: Create an empty database 4](#_Toc435619118)

[1.3. RHEL: Set up directory DATA\_PUMP\_DIR. 4](#_Toc435619119)

[1.4. RHEL: Pre-build user profiles 4](#_Toc435619120)

[1.5. AIX: Build tablespaces creation scripts: 5](#_Toc435619121)

[1.6. RHEL: Pre-create tablespaces. 5](#_Toc435619122)

[1.7. RHEL: Pre-create certain users: 5](#_Toc435619123)

[1.8. RHEL: Consider dropping SQLT if it exists. 5](#_Toc435619124)

[1.9. AIX: Export source database: 5](#_Toc435619125)

[1.10. RHEL: Import database 6](#_Toc435619126)

[1.11. Create sqlfile for indexes, constraints and materialized views 6](#_Toc435619127)

[1.12. Edit sqlfile 6](#_Toc435619128)

[1.13. Run sqlfile 6](#_Toc435619129)

[1.14. Compile invalid objects 6](#_Toc435619130)

[1.15. Review all logfiles and fix and error 6](#_Toc435619131)

[1.16. Gather statistics 6](#_Toc435619132)

[2. Datapump plus rman transportable tablespaces 7](#_Toc435619133)

[2.1. AIX: Perform TTS check 7](#_Toc435619134)

[2.2. RHEL: Check mountpints, disk space 7](#_Toc435619135)

[2.3. AIX: Check mountpints, disk space 7](#_Toc435619136)

[2.4. RHEL: Create an empty database 7](#_Toc435619137)

[2.5. RHEL: Set up directory DATA\_PUMP\_DIR. 7](#_Toc435619138)

[2.6. RHEL: Pre-build user profiles 7](#_Toc435619139)

[2.7. RHEL: Pre-create all tablespace 8](#_Toc435619140)

[2.8. AIX: Export metadata for users, profiles, roles, grants 8](#_Toc435619141)

[2.9. RHEL: Create users 8](#_Toc435619142)

[2.10. AIX: Full metadata export users, profiles, roles, grants 8](#_Toc435619143)

[2.11. RHEL: Drop tablespaces 8](#_Toc435619144)

[2.12. AIX: Put tablespaces in read-only mode. 8](#_Toc435619145)

[2.13. AIX: Run rman to convert endian format of datafiles. 8](#_Toc435619146)

[2.14. RHEL: FTP all datafiles from AIX. 9](#_Toc435619147)

[2.15. AIX: Export tablespaces metadata. 9](#_Toc435619148)

[2.16. RHEL: Import tablespaces metadata. 9](#_Toc435619149)

[2.17. AIX: Full metadata import 9](#_Toc435619150)

[2.18. Review all logfiles and fix and error 9](#_Toc435619151)

[2.19. Gather statistics 9](#_Toc435619152)

[3. Datapump plus rman transportable tablespaces plus GoldenGate……………………………………………..10](#_Toc435619153)

3.1. AIX: Create GoldenGate Extract process…………..……………………………………….…………..……….10

3.2. AIX: Create duplicate database using RMAN (Clonedb)…………………………………….…….………10

3.3. RHEL: Create skeleton database……………………………………………………………………………….….…10

3.4. AIX: Take full export of duplicate database (Clonedb) without any table data……….….…..11

3.5. RHEL: Import dumpfile into target Linux database………………………………………………………...11

3.6. AIX: Export tablespace metadata from duplicate database (Clonedb)…………………………...11

3.7. RHEL: Copy datafiles from Clonedb database and convert datafiles using RMAN…………..11

3.8. RHEL: Import tablespace metadata into target Linux database…………………………………...…11

3.9. RHEL: Create and start GoldenGate Replicat process…………………………………………………..…11

3.10. RHEL: Point users and application to target Linux db with no or minimal downtime…..…11

[4. Delphix 11](#_Toc435619154)

[4.1. AIX: Enabling oracle dsource(dsource/VDB) for Cross-Platform provisioning 11](#_Toc435619155)

[4.2. Transform to linux 12](#_Toc435619156)

[4.3. RHEL: check linux server and convert VDB to physical DB 13](#_Toc435619157)

[4.4. Review all logfiles and fix and error 14](#_Toc435619158)

[4.5. Gather statistics 14](#_Toc435619159)

# Oracle databases migration approaches from AIX to Red Hat Linux

Moving Oracle databases from AIX to Red Hat Linux environments will require database migration. Straight rman backup and restore cannot work because endian format is different in these environments - AIX uses big endian format whereas Linux uses little endian format. Though, Oracle rman supports database conversion (all tablespaces) across different platforms, it works only if both source and target environments have same endian format. If source and target environments have different endian format, tablespace or datafile level endian conversion is the option.

Following approaches for database migration have been considered:

1. Datapump only
2. Datapump plus rman transportable tablespaces
3. Datapump plus rman transportable tablespaces plus GoldenGate
4. Delphix

Each approach has some advantages and disadvantages. The main consideration to select an approach will be answer to question, “***How much downtime can database afford***”, which will also depend upon size of database.

* Option “1” is simpler than other approaches. It is a good option if database can afford downtime. Downtime is for the duration of export, FTP and import which is proportional to size of database.
* Option “2” is similar to option “1” with following steps.
  + Requires datapump export/import for metadata
  + Required conversion of endian format of tablespaces/datafiles. ***If any user schema object references objects in SYSTEM or SYSAUX tablespace, it will fail transportable tablespace checks.***
  + This option is faster than datapump only option because it does not involve DML activities during import for table data, indexes, constraints creation.
  + Downtime is for the whole duration because transporting tablespace requires tablespace in read only mode.
* Option “3” has minimum downtime. This option requires:
  + Starting GoldenGate capture process at source
  + Clone source database in AIX environment
  + Use option “1” or “2” to create a copy of clone database in the Linux server
  + Run GoldenGate apply process to apply changes
  + Switch database when both source (AIX) and target (RHEL) are in sync
  + This option has least downtime but it requires goldengate license at both source and target. It also requires additional disk space at the source for cloned copy of database.
* Option “4” was discussed with Delphix representatives
  + It can automate many steps in option “1” and “2” but it cannot capture and apply changes like goldengate.
  + While it can simplify process, it does not reduce downtime required for migration. As a matter of fact, there is an additional step to convert virtual database to physical database.
  + Also, unlike other options, it cannot be used to migrate database & upgrade.

Given below are details of each step and some sample scripts:

## Datapump only

### RHEL: Check mountpoints, disk space

Make sure that Linux host has necessary mountpoints for datafiles, redo logs and archived redo logs with required disk space.

### RHEL: Create an empty database

Database version should not be lower than source database.

### RHEL: Set up directory DATA\_PUMP\_DIR.

Set up directory DATA\_PUMP\_DIR in the database to appropriate location

### RHEL: Pre-build user profiles

***Why is this needed***? During my tests, I found that certain profiles had verify\_function which referenced object owned by SYS in SYSTEM schema. SYS objects do not get exported by datapump even with FULL=Y option. As a result, creation of such profiles fails during import. Users’ creation during import fails for all users with such profiles. This is particularly true for heritage Coventry databases (I think true for heritage Aetna databases as well because I remember similar issue during CXTPROD import)

Following link has details of process to create profiles using verify\_function

<http://teams2013.cvty.com/sites/dbinfrastructure/docs/_layouts/15/start.aspx#/Published%20Documents/Forms/AllItems.aspx>

Compare profiles created with profiles in the source database and manually create any missing profiles.

For any missing profile:

Extract DDL script for a profile

set pages 0 lines 132

set long 50000

SELECT DBMS\_METADATA.GET\_DDL('PROFILE', 'TRUSTED\_ID\_NO\_EXPIRE') from dual;

### AIX: Build tablespaces creation scripts:

Build tablespaces creation scripts from source database. It is not required if preference is to let datapump import create tablespaces which will require identical mountpoints at source (AIX) and target (RHEL)

run script similar to one below at the source database for each tablespace.

set pages 0 lines 132

set long 50000

SELECT DBMS\_METADATA.GET\_DDL('TABLESPACE', 'COMMON\_MED\_DATA02') from dual;

### RHEL: Pre-create tablespaces.

* Edit scripts created above for mountpoint and/or file name changes.
* Note: There were issues creating datafiles larger than 36 GB in RHEL environment. Edit scripts created above for datafiles larger than 36 GB.
* Run script in RHEL

### RHEL: Pre-create certain users:

Some users in heritage Coventry databases have been intentionally created with password hash value as 'G00dLuckGuessingThis' e.g.

CREATE USER "TUXCCMUU" IDENTIFIED BY VALUES 'G00dLuckGuessingThis'

Oracle 12c checks for correctness of password hash value. Creation of users with hash value 'G00dLuckGuessingThis' fails during datapump import. Such users must be pre-created

### RHEL: Consider dropping SQLT if it exists.

SQLT references certain SYS objects it is good idea to drop SQLT and recreate after import is done (Doc ID 1454160.1). Alternatively, SQLTXPLAIN & SQLTADMIN can be excluded during expdp run.

### AIX: Export source database:

expdp aprakash@claimdev dumpfile=claimdev\_full.dmp logfile=exp\_claimdev\_full.log full=y exclude=user:\"=\'SQLTXPLAIN\'\" exclude=user:\"=\'SQLTXADMIN'\"

### RHEL: Import database

Run datapump import at target location excluding tablespaces, indexes, constraints and materialized view logs.

impdp aprakash dumpfile=claimdev\_empty.dmp exclude=index exclude=constraint exclude=tablespace exclude=MATERIALIZED\_VIEW\_LOG

***Some observations during datapump import:***

* *During import procedures are imported before views, if there are procedures referencing views, these can be invalid after import. Such objects require recompilation.*
* *Even after excluding SQLTXPLAIN and SQLTXADMIN during export, I found import tried to create synonym & grants for these users. I ignored errors.*
* *I saw some errors such as below (I am not sure about the reason for this error. Failing statement is invalid in both 11g and 12c database):*

*ORA-39083: Object type AUDIT:"PRAKASAN"."PRAKASAN" failed to create with error:*

*ORA-32595: DDL statement cannot be audited with BY SESSION specified*

*Failing sql is:*

*AUDIT CREATE SESSION BY "PRAKASAN" BY SESSION*

### Create sqlfile for indexes, constraints and materialized views

impdp aprakash dumpfile=claimdev\_empty.dmp include=index include=constraint include=MATERIALIZED\_VIEW\_LOG sqlfile=indexes.sql logfile=impdp2.log

### Edit sqlfile

Consider editing sqlfile to increase degree of parallelism for indexes – it speeds up process

### Run sqlfile

Create indexes, constraints, and materialized views Using sqlfile created above

### Compile invalid objects

Check for invalid objects and compile these.

### Review all logfiles and fix and error

### Gather statistics

Though, statistics are imported during datapump import, I prefer gather stats after import.

## Datapump plus rman transportable tablespaces

### AIX: Perform TTS check

execute sys.dbms\_tts.transport\_set\_check('anand\_ts',true); **-- substitute anand\_ts with list of tablespaces.**

select \* from sys.transport\_set\_violations;

If there are violation, tablespaces cannot be transported. Either resolve violation issues or use different method.

### RHEL: Check mountpoints, disk space

Make sure that Linux host has necessary mountpoints for datafiles, redo logs and archived redo logs with required disk space.

### AIX: Check mountpoints, disk space

On AIX server, a temporary mountpoint is required for datafiles converted into little endian format. Amount of disk space required will be little more than size of all datafiles.

### RHEL: Create an empty database

Database version should not be lower than source database.

### RHEL: Set up directory DATA\_PUMP\_DIR.

Set up directory DATA\_PUMP\_DIR in the database to appropriate location

### RHEL: Pre-build user profiles

***Why is this needed***? During my tests, I found that certain profiles had verify\_function which referenced object owned by SYS in SYSTEM schema. SYS objects do not get exported by datapump even with FULL=Y option. As a result, creation of such profiles fails during import. Users’ creation during import fails for all users with such profiles. This is particularly true for heritage Coventry databases (I think true for heritage Aetna databases as well because I remember similar issue during CXTPROD import)

Following link has details of process to create profiles using verify\_function

<http://teams2013.cvty.com/sites/dbinfrastructure/docs/_layouts/15/start.aspx#/Published%20Documents/Forms/AllItems.aspx>

Compare profiles created with profiles in the source database and manually create any missing profiles.

For any missing profile:

Extract DDL script for a profile

set pages 0 lines 132

set long 50000

SELECT DBMS\_METADATA.GET\_DDL('PROFILE', 'TRUSTED\_ID\_NO\_EXPIRE') from dual;

### RHEL: Pre-create all tablespace

***Why is this needed*** ***if we are going to transport tablespaces***? Transportable tablespaces requires users owning objects in the tablespace(s) must exist. Next steps will export and import users from AIX to RHEL. These users creation will fail if default tablespace for the user does not exist. These tablespaces will have to be dropped after creation of users.

### AIX: Export metadata for users, profiles, roles, grants

expdp system DIRECTORY=data\_pump\_dir dumpfile=users\_exp.dmp LOGFILE=user\_exp.log FULL=y INCLUDE=PROFILE,USER,ROLE,ROLE\_GRANT

FTP export dumpfile from AIX to RHEL

### RHEL: Create users

Import dumpfile created above to create users profiles, roles, grants

impdp system DIRECTORY=data\_pump\_dir dumpfile=users\_exp.dmp LOGFILE=user\_imp.log

### AIX: Full metadata export users, profiles, roles, grants

expdp system DIRECTORY=data\_pump\_dir dumpfile=metadata\_exp.dmp LOGFILE=metadata\_exp.log FULL=y EXCLUDE=PROFILE,USER,ROLE,ROLE\_GRANT content=METADATA\_ONLY

FTP export dumpfile from AIX to RHEL

### RHEL: Drop tablespaces

Drop tablespace created above because tablespaces will be transported from source (AIX)

### AIX: Put tablespaces in read-only mode.

### AIX: Run rman to convert endian format of datafiles.

This should be run for each tablespace ( as below)

convert tablespace anand\_ts to platform 'Linux x86 64-bit'

DB\_FILE\_NAME\_CONVERT '/oradata/exports/TESTDB/TESTDB','/oradata/exports/TESTDB/TESTDB/backup';

In the example above, I had datafiles at /oradata/exports/TESTDB/TESTDB. I used DB\_FILE\_NAME\_CONVERT to create converted files at /oradata/exports/TESTDB/TESTDB/backup

In reality, it is likely to be more complex.

### RHEL: FTP all datafiles from AIX.

FTP all datafiles from AIX converted to little endian format to RHEL at location for datafiles .

### AIX: Export tablespaces metadata.

expdp system DIRECTORY=data\_pump\_dir dumpfile=exp\_ts.dmp logfile=exp\_ts.log transport\_full\_check=no transport\_tablespaces= **put\_list\_of\_*tablespaces here***.

### RHEL: Import tablespaces metadata.

impdp aprakash  dumpfile=anand\_ts.dmp transport\_datafiles='/oradata/u01/TESTFS/anand\_ts01.dbf' logfile=tts6.log

This is only a small example, list of datafiles is likely to be very long in reality.

### AIX: Full metadata import

impdp system DIRECTORY=data\_pump\_dir dumpfile=metadata\_exp.dmp LOGFILE=metadata\_imp.log FULL=y

Ignore “already exists” error for objects.

### Review all logfiles and fix and error

### Gather statistics

Though, statistics are imported during datapump import, I prefer gather stats after import.

## Datapump plus rman transportable tablespaces plus GoldenGate

* 1. **AIX: Create GoldenGate Extract process**

Create GoldenGate Extract process on source AIX database and start the same. This extract process will capture changes as they occur on the source AIX database in the remote trail files located on the Linux target system. Since the Replicat process is not running on the target system at this time, the source database changes will accumulate in the extract trail files. If source AIX database is 11g then install GoldenGate 12c for 11g database. If source AIX database is 12c then install GoldenGate 12c for 12c database.

**3.2. AIX: Create duplicate database using RMAN (Clonedb)**

Create duplicate database using RMAN in the source AIX environment (Clonedb) – this database will be used as the source for the export of database structure (no rows export) and tablespace metadata. If source AIX database is 11g, create 11g duplicate database. If source AIX database is 12c, create 12c duplicate database.

**3.3. RHEL: Create skeleton database**

Create a skeleton 12c database on the Linux platform – target Linux database. Also set up GoldenGate user in this database and grant it the required privileges.

**3.4. AIX: Take full export of duplicate database (Clonedb) without any table data**

Take full export of the database without any table data to get just the structure of the database – this is taken from the Clonedb duplicate database created in step 3.2.

**3.5. RHEL: Import dumpfile into target Linux database**

Import dumpfile into the target Linux database which has the database structure without the table data – this will create all the users, roles, synonyms etc.

**3.6. AIX: Export tablespace metadata from duplicate database (Clonedb)**

On the Clonedb database, export the tablespace metadata – make the required tablespaces read only. Note that the original source AIX database is in read write mode and is being accessed by the users with no downtime as yet.

**3.7. RHEL: Copy datafiles from Clonedb database and convert datafiles using RMAN**

Copy datafiles from read only tablespaces (from Clonedb) to the target Linux system and using RMAN convert the datafiles from the AIX platform to the Linux platform.

**3.8. RHEL: Import tablespace metadata into target Linux database**

Import tablespace metadata into the target Linux database and plug in the tablespaces - make the tablespaces read write.

**3.9. RHEL: Create and start GoldenGate Replicat process**

On target Linux environment, create and start the GoldenGate Replicat process. This process will start reading from the Extract trail files created in Step 3.1 and will start applying them to the target Linux database.

**3.10. RHEL: Point users and application to target Linux db with no or minimal downtime**

Once all the changes in the trail files have been applied by the Replicat process and it is confirmed that both source and target databases are in sync, point the users and application to the target Linux database with no or minimal downtime which will depend on the infrastructure.

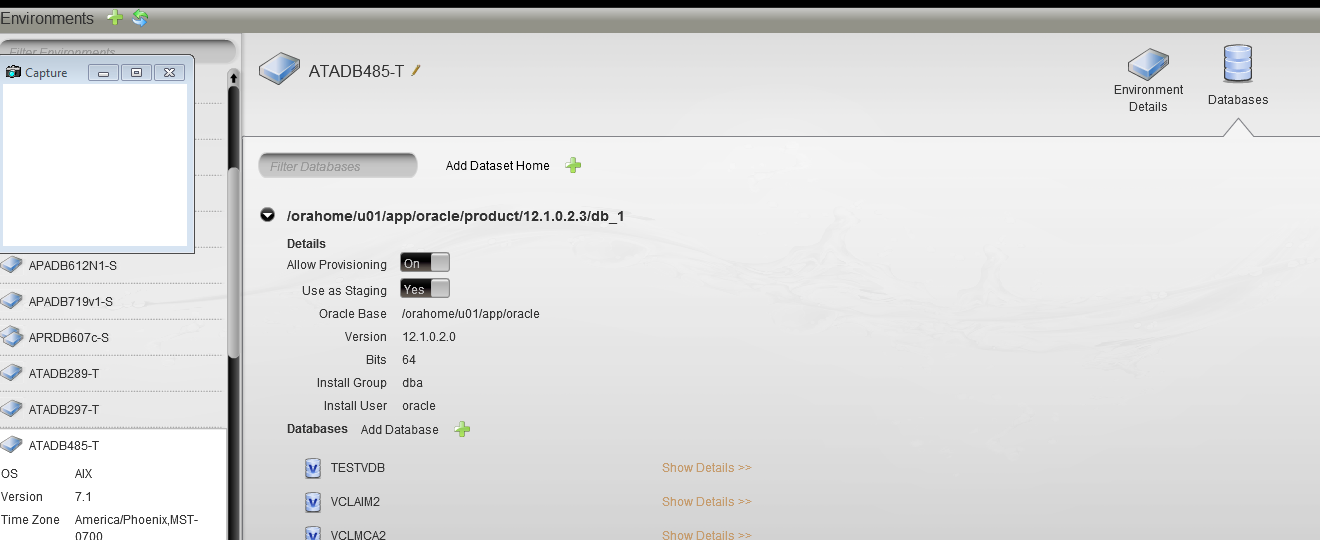
## Delphix

The underlying Oracle technology used to transform to Linux imposes several requirements, including:

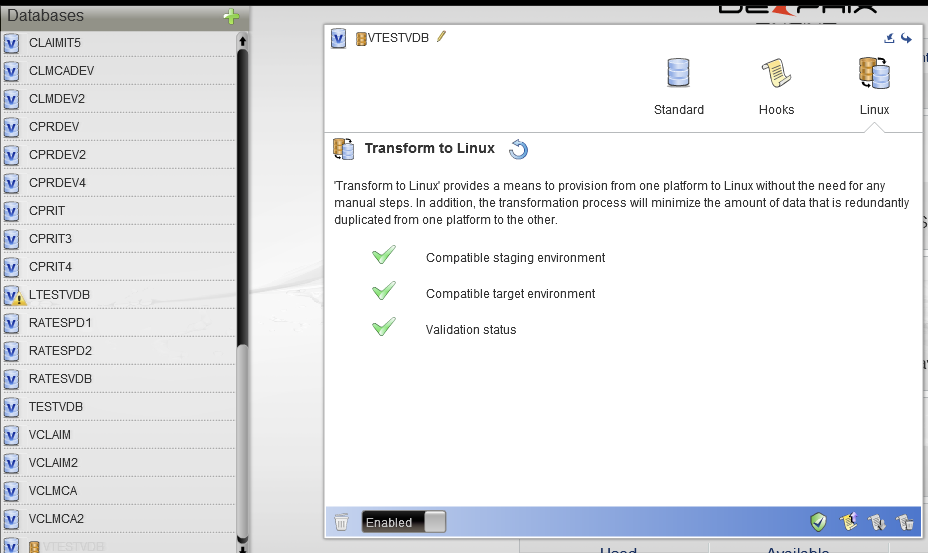
* Encryption cannot be used
* Tablespace Transport Set must be self-contained
* Tablespaces with XML types cannot be used before Oracle version 11.2
* Advanced queues versions 8 or later
* Spatial indexes cannot be used before Oracle version 11.2

### AIX: Enabling oracle dsource(dsource/VDB) for Cross-Platform provisioning

* Log into the Delphix Admin application using delphix\_admin credentials
* In the **manage** menu, select **Environments**>select server name for source>click **database** tab>select ORACLE\_HOME and enable **User as Staging**



* In the **Manage** menu, select **Databases > My Databases > click source db expand icon to open source db card. Then click the Flip icon on the car>Click Linux tab>click Validate Transformation button>** If there are violation, Either resolve violation issues or use different method

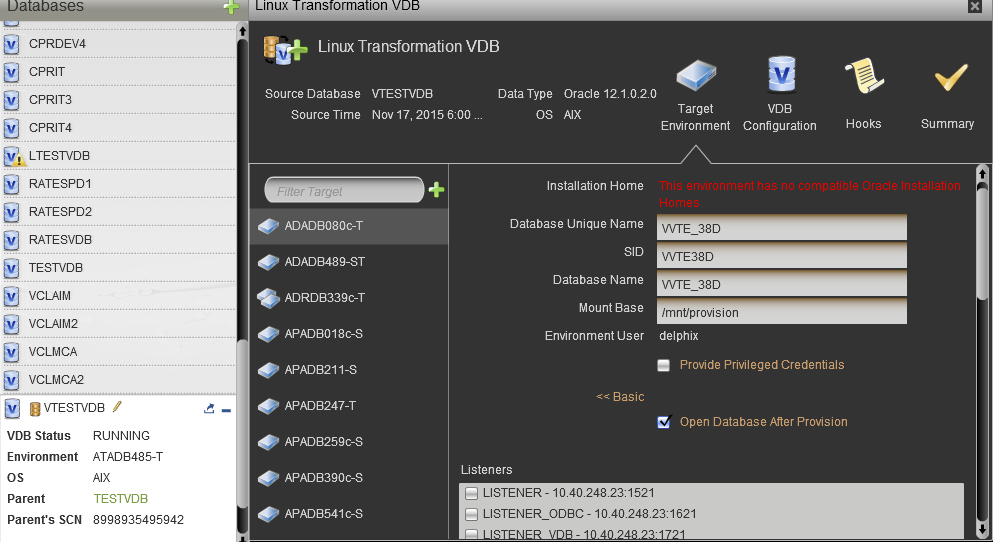


### Transform to linux

* If the validation process is successful, green check marks will appear next to each validation requirement. Click **Transform to Linux button**

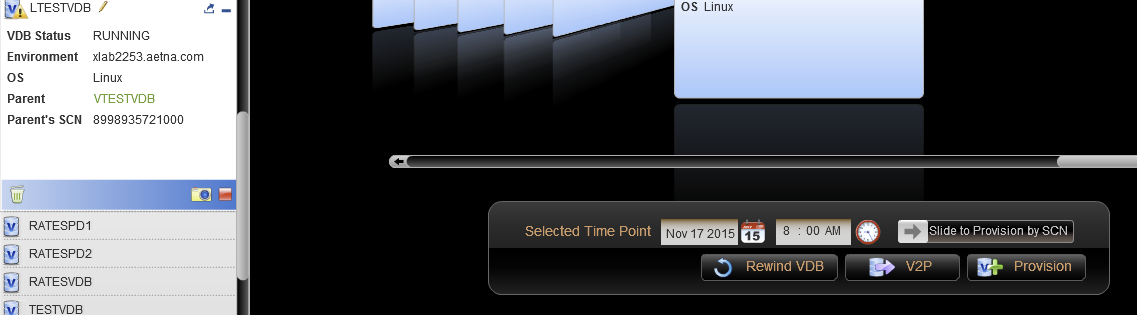


* Follow the wizard : select server>enter vdb name>choose listener>choose VDB configuration Template



### RHEL: check linux server and convert VDB to physical DB

* Check mountpoints , disk space and Make sure that Linux host has necessary mountpoints for datafiles, redo logs and archived redo logs with required disk space.
* Delphix host check on linux server
  1. create generic host account delphix on the server . Ulimit for the account processes(per user)  unlimited
  2. create mountpoint /mnt/provision directory (owner - delphix, group - dba)
  3. run hostcheck
* Convert VDB to physical DB: Click linux VDB>click **V2P>**Follow the wizard



### Review all logfiles and fix and error

### Gather statistics

Though, statistics are imported during datapump import, I prefer gather stats after import.