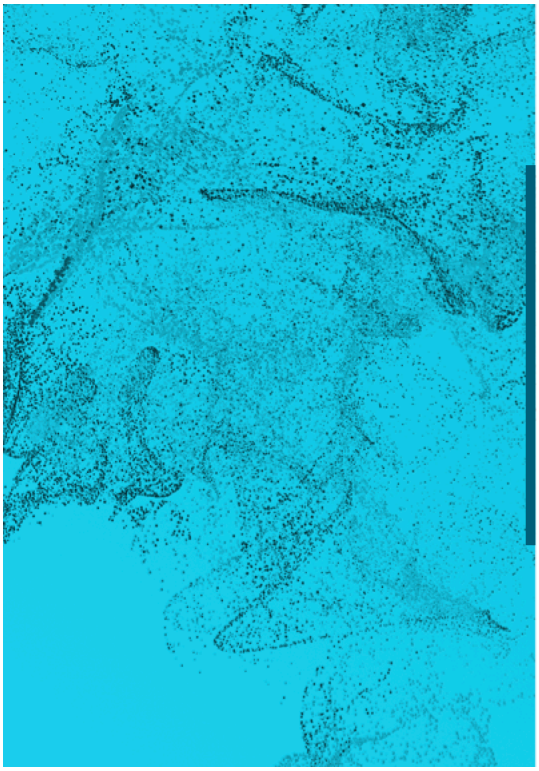
DELPHIX



Upgrading Oracle and MS Sql

Field Services, April 2018

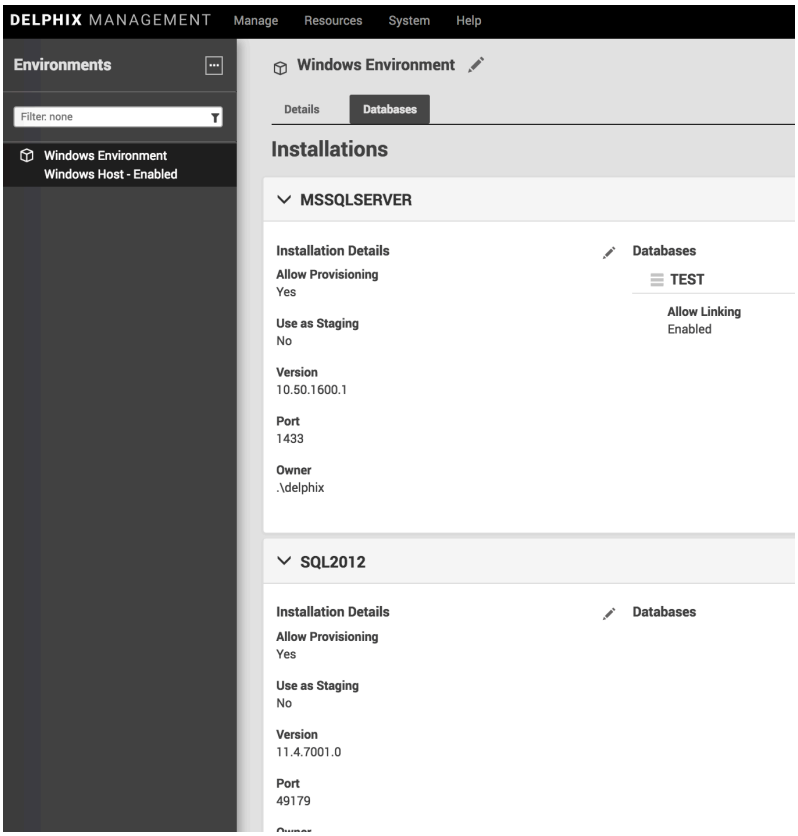
Upgrading MS Sql

When we Provision/Refresh an MS Sql virtual database, Delphix can use native MS Sql capabilities to upgrade the database. Here is an example of this process.

Infrastructure used for this upgrade:

- Delphix Engine (5.2.2.0).
- Windows Server 2012 R2 with MS Sql 2008 R2 and MS Sql 2012 installed[1]

We discover this environment and we verify that we can see both MS Sql installations available:

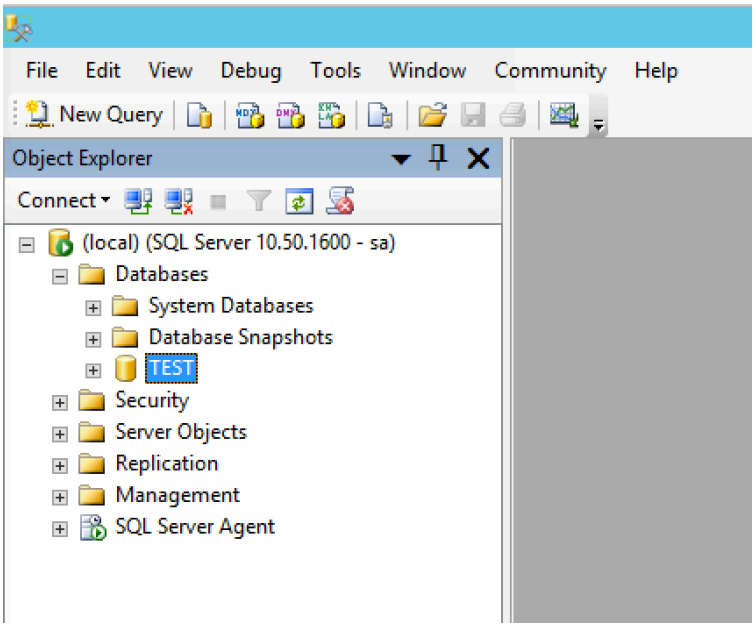


[1] Note that we will use the same Windows Server Machine as Source, Staging and Target for demonstration purposes. In a real implementation the best practice is to have dedicated servers for each one.

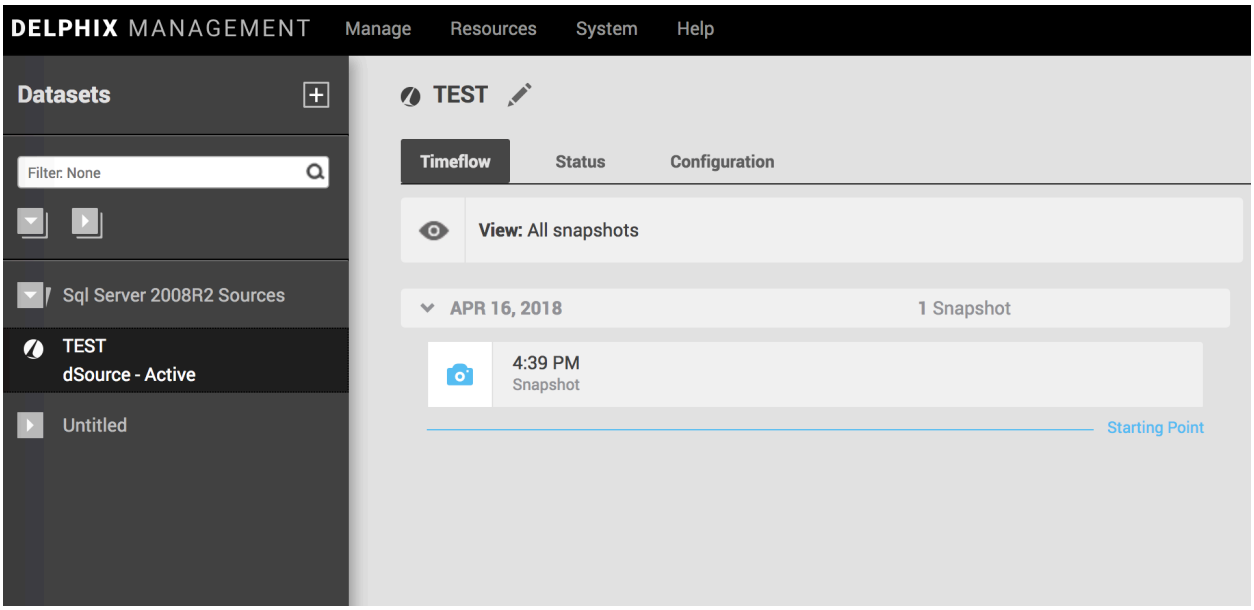
We can see 2 installations:

- MSSQLSERVER (2008 R2)
- SQL2012

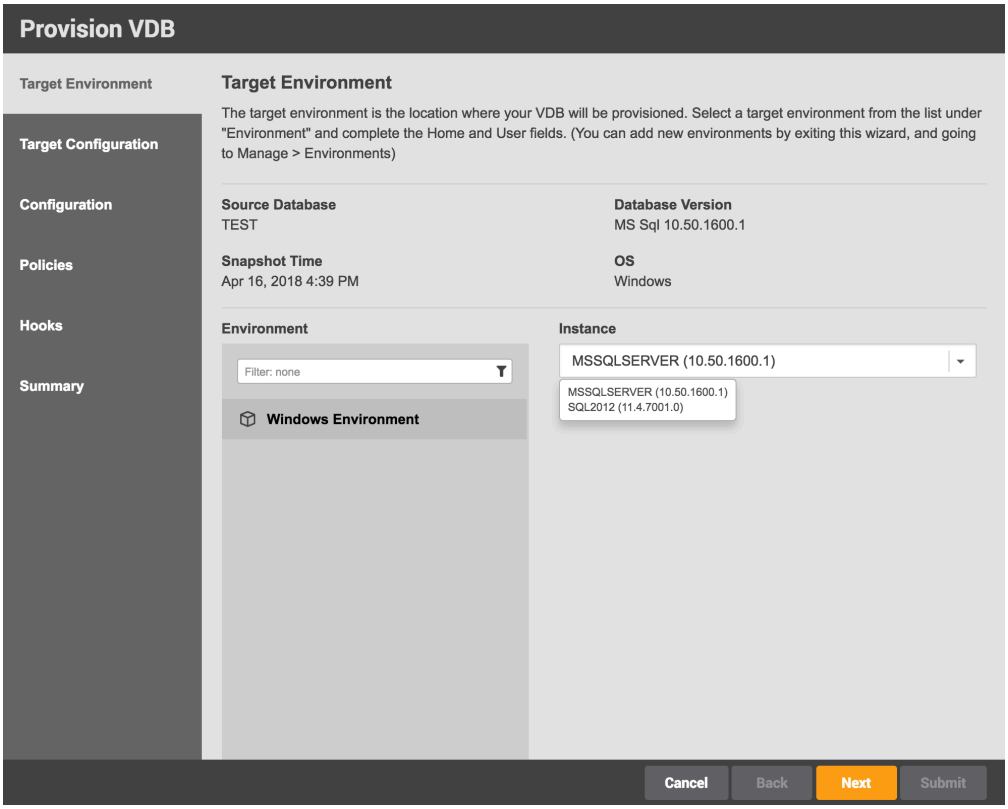
We have a MS Sql database called TEST running in MS Sql 2008 R2 :



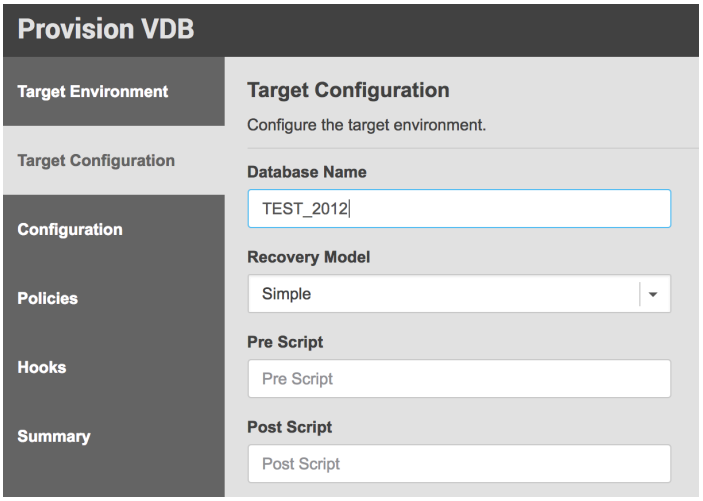
We create TEST dSource:



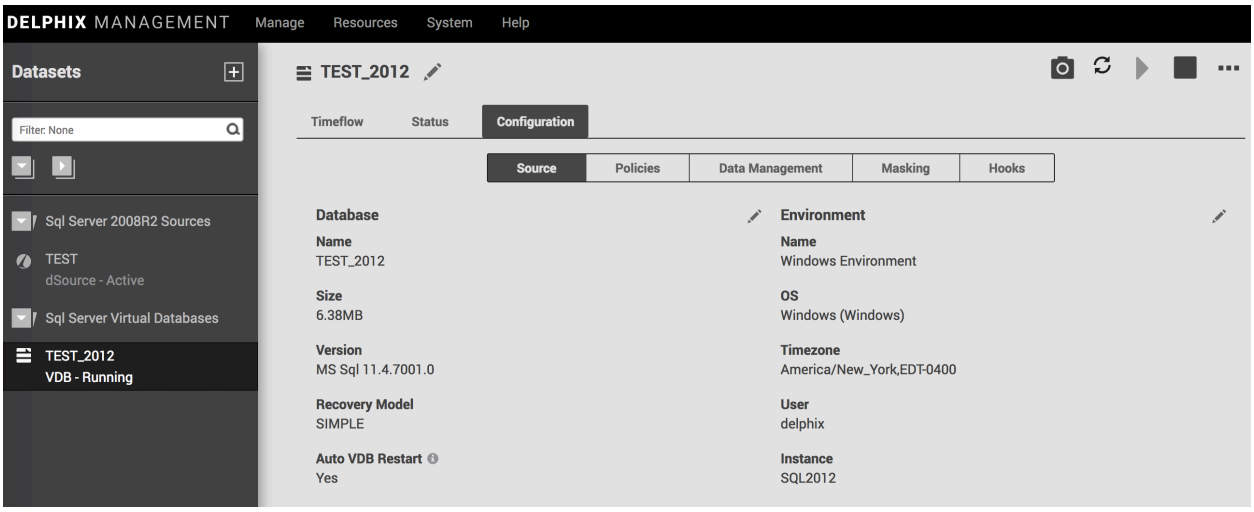
Now we are ready to provision a VDB. We select the Environment and when we click on Instance, we will see all the available options on the server:



As we have both 2008 R2 and 2012, we can now select which one we want to provision the database to. If we select 2008 R2 it will be a simple provision on the same version. If we select 2012 we will not only create a virtual copy of the TEST database, but this database will also be upgraded to MS Sql 2012:

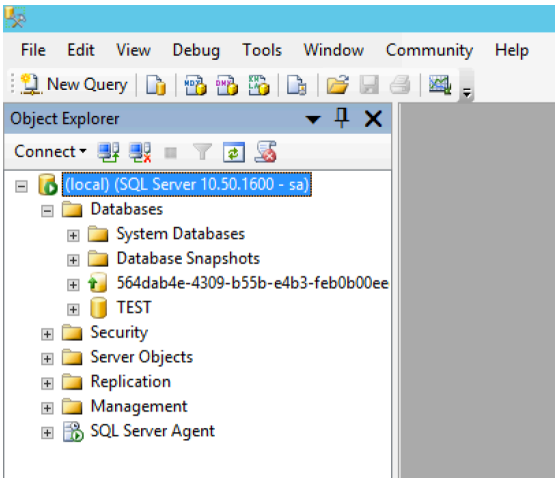


We complete the provision wizard and when it's completed we will have our virtual database running in MS Sql 2012:

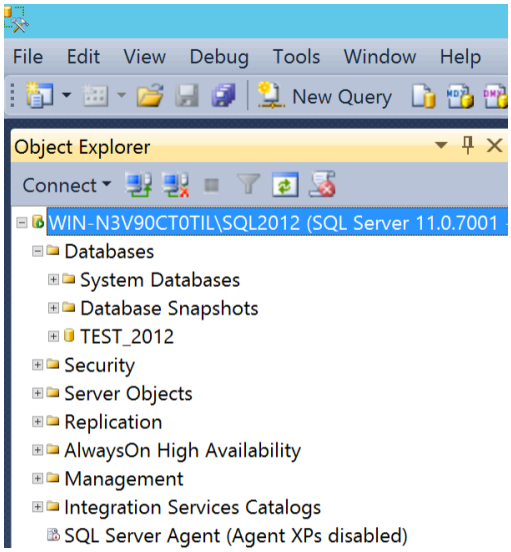


On the image above we can see that version is MS Sql 11.4.7001.0 that is 2012.

It's important to mention that the staging database will be running on MS Sql 2008 R2:



And to see the virtual database, TEST_2012 we will have to connect to Sql Server 2012:


File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
WIN-N3V90CT0TIL\SQL2012 (SQL Server 11.0.7001
Databases
System Databases
Database Snapshots
TEST_2012
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)

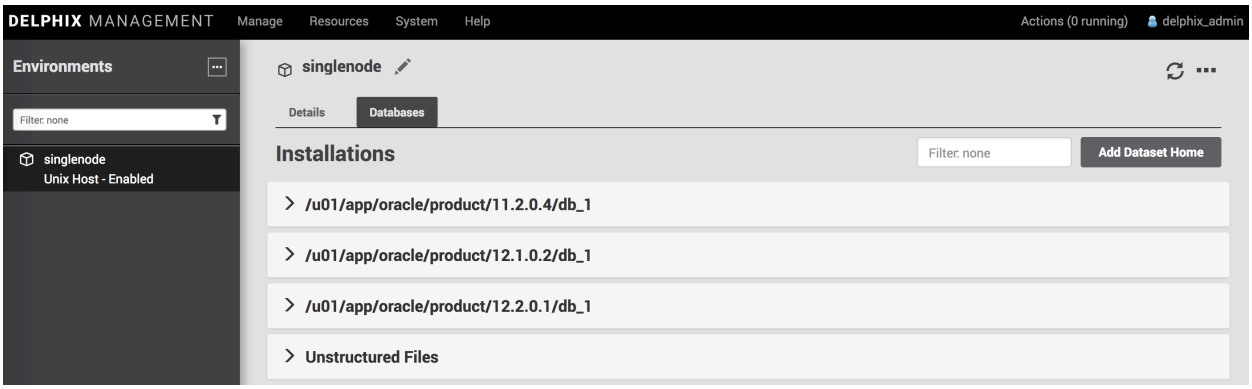
Upgrading Oracle

When we Provision/Refresh an Oracle Database, we can use a hook to invoke the Oracle tools that will help us to automate the upgrade of the virtual copy (dbua). The upgrade process can take same so that's why when we upgrade Oracle dataabases we use a similar approach as when we mask. We provision one VDB and upgrade it through a hook, and when this VDB is ready, we provision the VDBs for the final users from this one. Upgrade once, provision many. Here is an example of this process.

Infrastructure used for this upgrade:

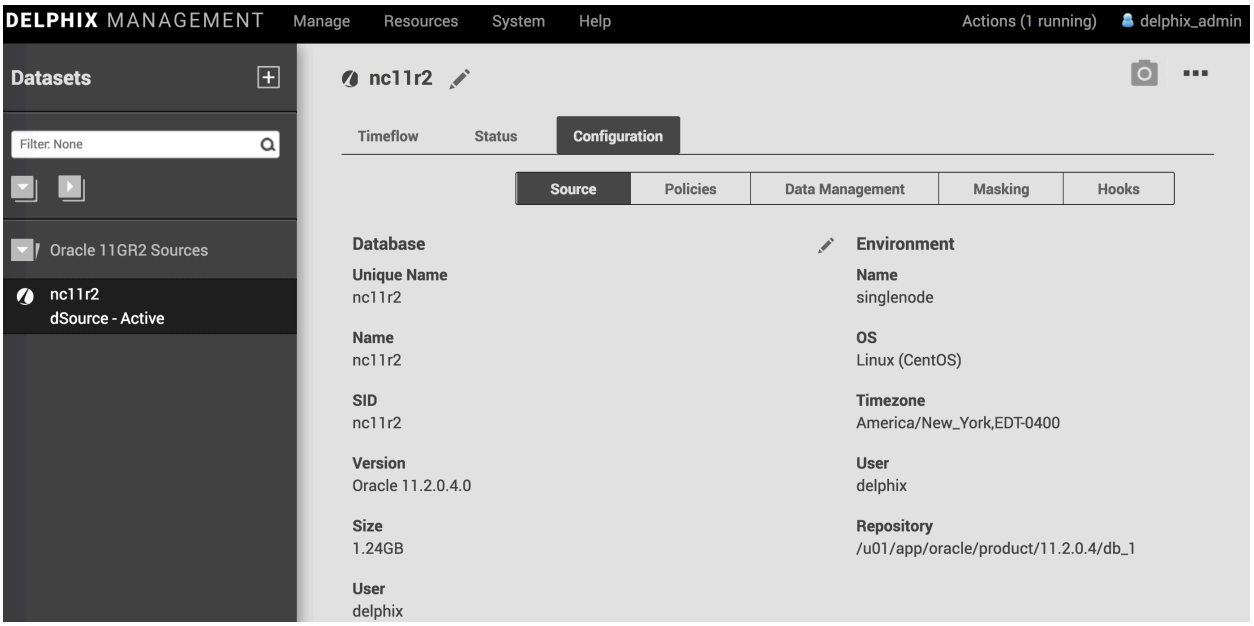
- Delphix Engine (5.2.2.0).
- Centos 7.4 Server with Oracle 11.2.0.4, 12.1.0.2 and 12.2.0.1 installed[2]

First we need to confirm that in our Environment we can see all the Oracle versions available:



[2] Note that we will use the same Centos Server as Source and Target for demonstration purposes. In a real implementation the best practice is to have dedicated servers.

We now confirm that the source has been properly ingested in Delphix:



Our Source database is called nc11r2 and it's running in 11.2.0.4. We are now ready to provision a VDB, but first let's review the script we will use to orchestrate the upgrade through a hook.

This is the script we will be using, that will be located in /home/oracle/upgrd.sh:

```
#/bin/ksh
#================================================================================
#  File:     upgrd.sql
#  Type:   UNIX/Linux korn-shell script
#  Date:   24-Mar 2016
#  Author: Delphix
#
#  Licensed under the Apache License, Version 2.0 (the "License");
#  you may not use this file except in compliance with the License.
#  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
#  Unless required by applicable law or agreed to in writing, software
#  distributed under the License is distributed on an "AS IS" BASIS,
#  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#  See the License for the specific language governing permissions and
#  limitations under the License.
#
#  Copyright (c) 2016 by Delphix. All rights reserved.
#
#  Description:
#
#             Shell-script intended to orchestrate a basic Oracle database upgrade
#             from 10g or 11g to a higher version.  For upgrades to Oracle12c, a
#             non-CDB database is created.
#
#  Usage:
#
```

```
#		upgrd.sh <new-ORACLE_HOME> <dlpx-Engine> <dlpx-EnvName> [ <ORACLE_BASE> ]
#
#		where:
#			<new-ORACLE_HOME>	to which this VDB is being upgraded
#			<dlpx-Engine>		Delphix Engine on which this VDB resides
#			<dlpx-EnvName>		Delphix Environment on which this VDB resides
#			<ORACLE_BASE>		under which this VDB presently resides
#
#  Modifications:
#		TGorman 24-Mar 2016		v1.0 first test version
#===============================================================================
_dlpxUser=ora12102
#
#-------------------------------------------------------------------------------
# Validate command-line parameters specified...
#-------------------------------------------------------------------------------
case $# in
	4)	_newOraHome=$1
		_dlpxEngine=$2
		_dlpxEnvName=$3
		export ORACLE_BASE=$4
		;;
	3)	_newOraHome=$1
		_dlpxEngine=$2
		_dlpxEnvName=$3
		;;
	*)	echo "Usage: \"upgrd.sh <new-ORACLE_HOME> <dlpx-Engine> <dlpx-EnvName> [
<ORACLE_BASE> ]\"; aborting..."
		echo "	<new-ORACLE_HOME> to which this VDB is being upgraded"
		echo "	<dlpx-Engine> Delphix Engine on which this VDB resides"
		echo "	<dlpx-EnvName> Delphix Environment on which this VDB resides"
		echo "	<ORACLE_BASE> under which this VDB presently resides"
		exit 1
		;;
esac
#
#-------------------------------------------------------------------------------
# Verify that ORACLE_SID, ORACLE_BASE, and ORACLE_HOME are set...
#-------------------------------------------------------------------------------
if [[ "${ORACLE_SID}" = "" ]]
then
	echo "ORACLE_SID is not set; aborting..."
	exit 1
fi
if [[ "${ORACLE_HOME}" = "" ]]
then
	echo "ORACLE_HOME is not set; aborting..."
	exit 1
fi
if [[ "${ORACLE_BASE}" = "" ]]
then
	echo "ORACLE_BASE is not set; aborting..."
	exit 1
fi
if [ ! -d ${ORACLE_BASE} ]
then
	echo "ORACLE_BASE directory \"${ORACLE_BASE}\" not found; aborting..."
	exit 1
fi
if [ ! -d ${_newOraHome} ]
then
	echo "New ORACLE_HOME directory \"${_newOraHome}\" not found; aborting..."
	exit 1
fi
#
#-------------------------------------------------------------------------------
# Verify how the upgrade is going to be performed...
```

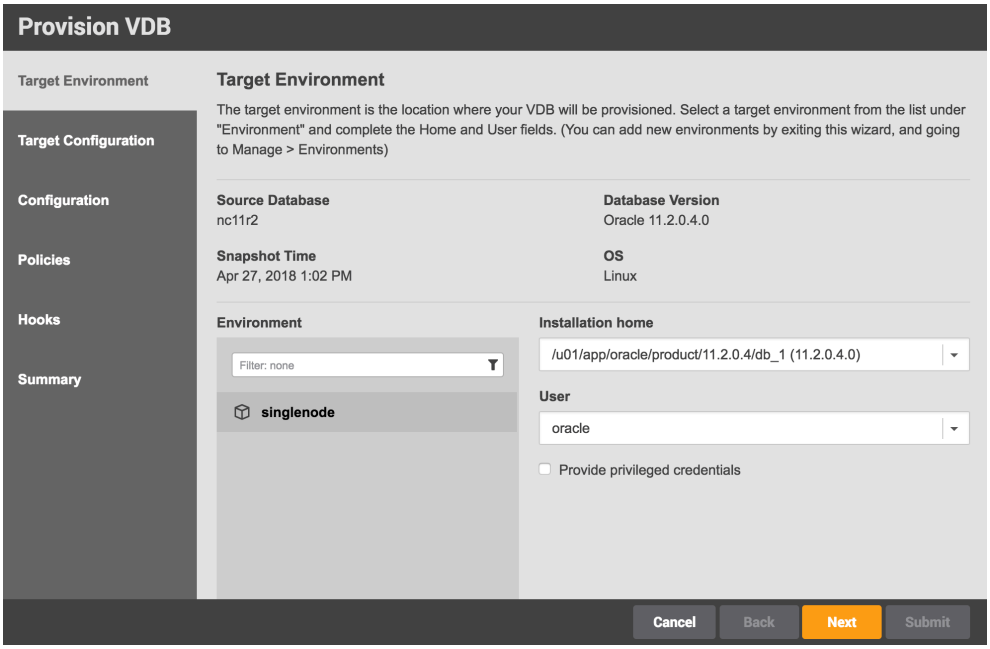
```
#-------------------------------------------------------------------------------
echo ""
echo "upgrading ORACLE_SID \"${ORACLE_SID}\""
echo "from ORACLE_BASE \"${ORACLE_BASE}\", ORACLE_HOME \"${ORACLE_HOME}\", LD_LIBRARY_PATH
\"${LD_LIBRARY_PATH}\""
echo "to ORACLE_HOME \"${_newOraHome}\""
echo ""
#
#-------------------------------------------------------------------------------
# Save a copy of the Oracle initialization parameter file to "/tmp"...
#-------------------------------------------------------------------------------
cp ${ORACLE_HOME}/dbs/init${ORACLE_SID}.ora /tmp
if (( $? != 0 ))
then
        echo "copy of init${ORACLE_SID}.ora from \"${ORACLE_HOME}/dbs\" to \"/tmp\" failed; aborting..."
        exit 1
fi
#
#-------------------------------------------------------------------------------
# If the SQL script "emremove.sql" exists in the new Oracle version's ORACLE_HOME,
# then run it to remove the Enterprise Manager schema...
#-------------------------------------------------------------------------------
if [ -f ${_newOraHome}/rdbms/admin/emremove.sql ]
then
        sqlplus / as sysdba << __EOF__
whenever oserror exit failure
start ${_newOraHome}/rdbms/admin/emremove.sql
exit success
__EOF__
        if (( $? != 0 ))
        then
                echo "Running \"${_newOraHome}/rdbms/admin/emremove.sql\" failed; aborting..."
                exit 1
        fi
fi
#
#-------------------------------------------------------------------------------
# Save "old" settings of ORACLE_HOME and ORACLE_BASE and set "new" settings for
# ORACLE_HOME and PATH to the upgraded version...
#-------------------------------------------------------------------------------
_oldOraBase=${ORACLE_BASE}
_oldOraHome=${ORACLE_HOME}
export ORACLE_HOME=${_newOraHome}
export PATH=${ORACLE_HOME}/bin:${PATH}
export LD_LIBRARY_PATH=${ORACLE_HOME}/lib
#
#-------------------------------------------------------------------------------
# Copy saved-off copy of the Oracle initialization parameter file to the
# "dbs" subdirectory within the new ORACLE_HOME...
#-------------------------------------------------------------------------------
cp /tmp/init${ORACLE_SID}.ora ${ORACLE_HOME}/dbs
if (( $? != 0 ))
then
        echo "copy of init${ORACLE_SID}.ora from \"/tmp\" to \"${ORACLE_HOME}/dbs\" failed; aborting..."
        exit 1
fi
#
#-------------------------------------------------------------------------------
# Call the Oracle DBUA utility to perform the database upgrade...
#-------------------------------------------------------------------------------
${ORACLE_HOME}/bin/dbua  \
        -silent \
        -sid ${ORACLE_SID} \
        -oracleHome ${_oldOraHome} \
        -oracleBase ${_oldOraBase} \
        -autoextendFiles \
        -recompile_invalid_objects true \
```

```
        -degree_of_parallelism 4
if (( $? != 0 ))
then
        echo "DBUA failed; aborting..."
        exit 1
fi
echo "silent DBUA completed successfully"
#
```

We verify that the script is available in /home/oracle on the target server:

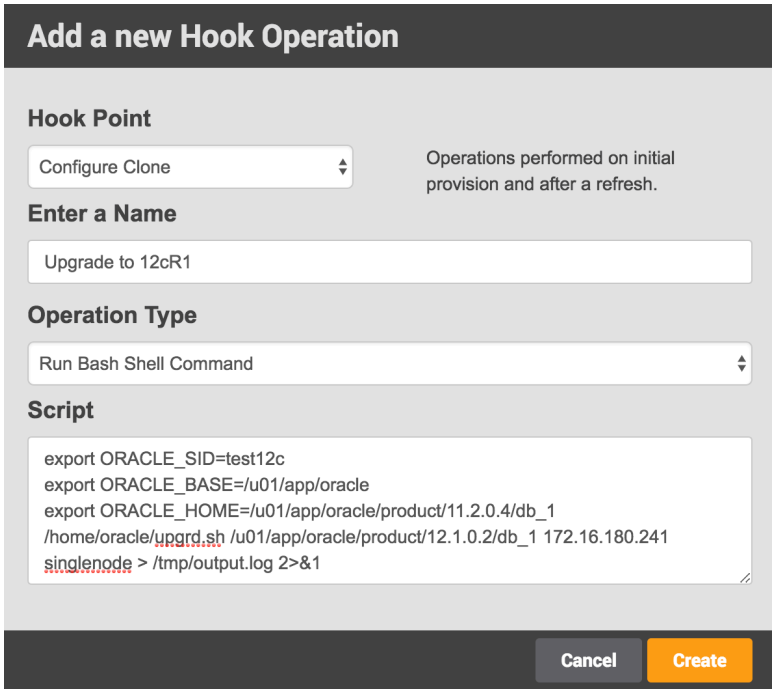
```
[oracle@singlenode ~]$ pwd
/home/oracle
[oracle@singlenode ~]$ ls -lrth upgrd.sh
-rw-r--r--. 1 oracle oinstall 6.1K Apr 27 13:08 upgrd.sh
[oracle@singlenode ~]$
```

We can now start the provision wizard. I am going to use Oracle user to simplify the utilization of dbua. If you want to use delphix os user, you may need to modify some privileges to be able to run dbua with delphix user:

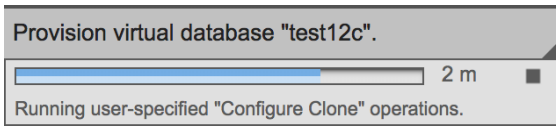


We will call the virtual database we want to provision and upgrade test12c. We do a standard provision proccess but we add our hook in configure clone. We select the Configure Clone hook because we want this upgrade process to run on provision but also on refresh. On the hook, we must add the script with the 3 parameter the scripts needs to be executed:

<new-ORACLE_HOME>	to which this VDB is being upgraded
<dlpx-Engine>	Delphix Engine on which this VDB resides
<dlpx-EnvName>	Delphix Environment on which this VDB resides



We can now submit the provision job. On the Actions panel you can monitor the provision status, and you will notice that it will spend quite some time in *Running user-specified "Configure Clone" operations*. While you see this message, the upgrade will be running:

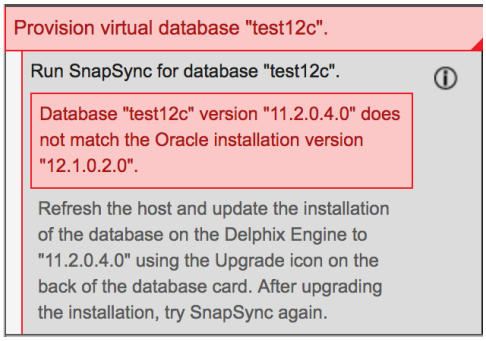


and you can test the status tailing the log we specified in the hook:

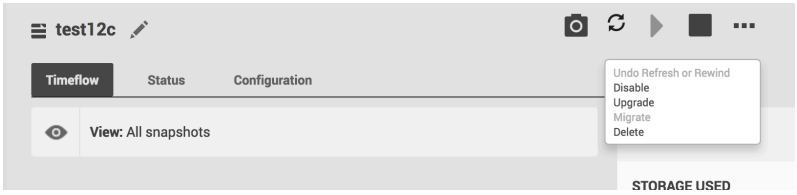
```
[oracle@singlenode ~]$ cat /tmp/output.log

upgrading ORACLE_SID "test12c"
from ORACLE_BASE "/u01/app/oracle", ORACLE_HOME "/u01/app/oracle/product/11.2.0.4/db_1", LD_LIBRARY_PATH
"/u01/app/oracle/product/11.2.0.4/db_1/lib"
to ORACLE_HOME "/u01/app/oracle/product/12.1.0.2/db_1"


SQL*Plus: Release 11.2.0.4.0 Production on Fri Apr 27 13:25:24 2018

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


Connected to:
```

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

SQL> SQL> old  69:     IF (upper('&LOGGING') = 'VERBOSE')
new  69:     IF (upper('VERBOSE') = 'VERBOSE')

PL/SQL procedure successfully completed.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Log files for the upgrade operation are located at: /u01/app/oracle/cfgtoollogs/dbua/test12c/upgrade1
Performing Pre Upgrade
4% complete
20% complete
Performing RDBMS Upgrade
20% complete
```

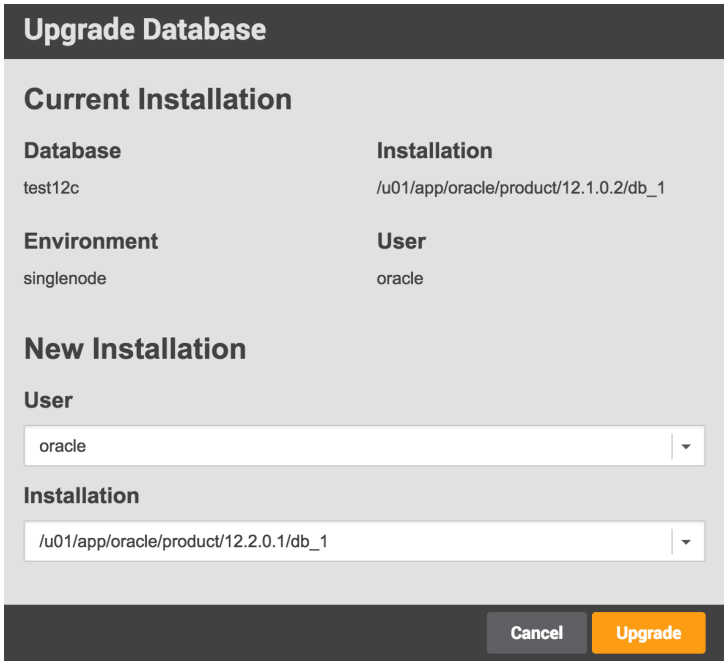
When the provision is complete we should have a virtual database running in 12cR1. The last step of provisioning a VDB is to take a snapshot, and we will receive a message stating that this database is in 12cR1, and in order to do that snapshot we need to tell Delphix that the database is now 12cR1:



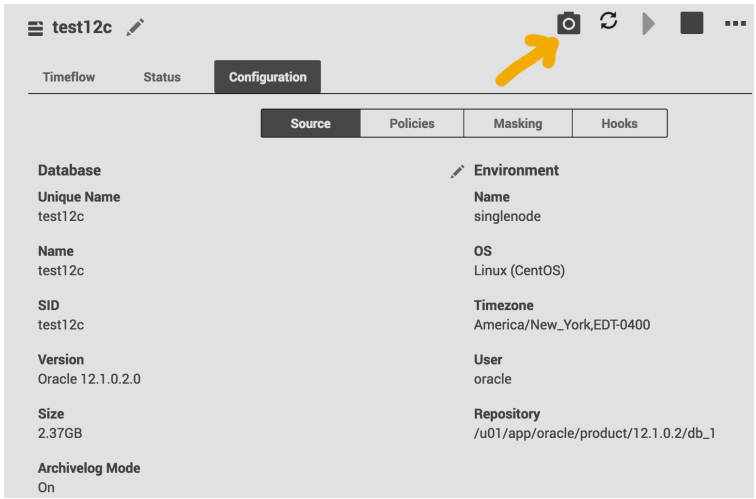
To change the metadata in Delphix and reflect the new version of the database we have to do click the Upgrade button on the VDB:



Once there we need to reflect that test12c database has to use the proper set of binaries, in this case 12cR1:



Click on Upgrade and the database repository will be now showing the proper 12cR1 set of binaries. Now we can run the snapshot:



After this latest snapshot the Virtual Database is complete. We can now provision 12c VDBs from this one.

This last 2 steps, Upgrade and Snapshot can be also run from a CLI script if we want to automate the whole process. We can't run these 2 steps from the upgrd.sh script, because Delphix will send an error message as we are trying to do 2 Delphix actions in the same object (you can't snapshot/upgrade a VDB while that VDB is being created).

Here is an example of CLI script to upgrade the repository to 12cR1:

```
ssh delphix_admin@172.16.180.241 "/sourceconfig select nc11r2; update; set
repository=\"singlenode/'/u01/app/oracle/product/12.1.0.2/db_1'\"; commit; exit"
```

Where 172.16.180.241 is the IP of the Delphix Engine.

Apendix 1: Oracle upgrade execution log

```
[oracle@singlenode ~]$ cat /tmp/output.log

upgrading ORACLE_SID "test12c"
from ORACLE_BASE "/u01/app/oracle", ORACLE_HOME "/u01/app/oracle/product/11.2.0.4/db_1", LD_LIBRARY_PATH
"/u01/app/oracle/product/11.2.0.4/db_1/lib"
to ORACLE_HOME "/u01/app/oracle/product/12.1.0.2/db_1"


SQL*Plus: Release 11.2.0.4.0 Production on Fri Apr 27 13:25:24 2018

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


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

SQL> SQL> old  69:     IF (upper('&LOGGING') = 'VERBOSE')
new  69:     IF (upper('VERBOSE') = 'VERBOSE')

PL/SQL procedure successfully completed.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Log files for the upgrade operation are located at: /u01/app/oracle/cfgtoollogs/dbua/test12c/upgrade1
Performing Pre Upgrade
4% complete
20% complete
Performing RDBMS Upgrade
20% complete
20% complete
21% complete
21% complete
22% complete
22% complete
22% complete
22% complete
23% complete
23% complete
23% complete
23% complete
```

23% complete
24% complete
24% complete
24% complete
24% complete
24% complete
25% complete
25% complete
25% complete
25% complete
26% complete
26% complete
26% complete
27% complete
27% complete
27% complete
27% complete
27% complete
28% complete
28% complete
28% complete
28% complete
29% complete
29% complete
29% complete
30% complete
30% complete
31% complete
31% complete
31% complete
32% complete
32% complete
33% complete
33% complete
33% complete
34% complete
34% complete
35% complete
35% complete
35% complete
36% complete
36% complete
37% complete
37% complete

```
37% complete
38% complete
38% complete
38% complete
39% complete
39% complete
40% complete
Performing Post Upgrade
42% complete
44% complete
60% complete
Configuring Database with Enterprise Manager
62% complete
64% complete
80% complete
Generating Summary
Database upgrade has been completed successfully, and the database is ready to use.
100% complete
Check the log file "/u01/app/oracle/cfgtoollogs/dbua/logs/silent.log_1524849989005" for upgrade details.
silent DBUA completed successfully
[oracle@singlenode ~]$
```