xTTS Automation

Overview :

This script build as a wrapper on top of the method described on oracle documentation. There are many manual steps need to be performed to achieve minimal downtime for cross platform migration. xTTS automation scripts used to migrate Oracle 11g/12c databases on different OS platform in standard and fully automated manner with minimal user input. It is intended to save a lot of manual time to start and forget it and also ensure each migration is done exactly in the same way.

Prerequisites:

1. Create directory structure as below :

```
This script assume directory structure as below

XTTS_HOME
  |
  |____ bin
  |      |____ xtt.properties
  |      |____ source_capture.sh
  |      |____ target_apply.sh
  |      |____ xttx_cutover_script_generator.sh
  |      |____ source_capture_status.txt
  |      |____ target_apply_status.txt
  |      |____ other xTTS binary files
  |
  |____ cfg
  |      |____ auto_xtts_config.txt
  |
  |____ cutover
         |____ Cutover Scripts will be generated in this folder
```

2. Identify Tablespaces part of this migration
3. Identify Schemas part of this migration
4. Create/Update **auto_xtts_config.txt** file with below details:

```
[oracle@ip-172-31-77-87 cfg]$ cat auto_xtts_config.txt

# Schema names part of xtts migration in CSV format
SCHEMAS=SCHEMA1,SCHEMA2,SCHEMA3,...,SCHEMAn

# Source and target Oracle SID
SRC_ORA_SID=orcl
TGT_ORA_SID=orcl
```

```
# Source and target Oracle Home Path
TGT_ORA_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
SRC_ORA_HOME=/u01/app/oracle/product/12.1.0/dbhome_1

# Source and target DB host name, this will be use for SCP
SOURCE_DB_HOST=ec2-18-232-51-153.compute-1.amazonaws.com
TARGET_DB_HOST=target-db.compute-1.amazonaws.com

# Source and target DB, datapump directory locations
SOURCE_DUMP_DIR=/home/oracle/xtts/datapump
TARGET_DUMP_DIR=/home/oracle/xtts/datapump

# Source and target DB, xTTS home path as shown previos section
SOURCE_XTTS_HOME=/home/oracle/xtts
TARGET_XTTS_HOME=/home/oracle/xtts

# Number of parallel session use for coping backup files
PARALLEL_COPY_CTN=4

# Mail id to notfy
NOTIFY_LIST=someone@organization.com

# Version of the script used ( as of now only 3 supported )
SCRIPT_VERSION=3
```

5. Create/Update **xtts.properties** file with below parameters.

```
platformid=4
tablespaces=TBS1,TBS2,...,TBSn
dfcopydir=<<Full backup DIR at Source host>>
backupformat=<<Incremental backup DIR at Source host>>

stageondest=<<Full backup DIR at Target host>>
backupondest=<<Incremental backup DIR at Target host>>
storageondest=<<Datafile DIR at Target host>>

parallel=8
rollparallel=4
```

6. Create below directories in source and target DB as shown below :

```
# Source DB Host

dfcopydir=/u20/oradata/SCDEP/xtt/CKMDBP/fullbkp
backupformat=/u19/oradata/SCDEP/xtt/CKMDBP/incrbkp

mkdir -p ${dfcopydir} ${backupformat}

# Target DB Host
stageondest=/u04/orabkp/CKMDBP/xtt/fullbkp
backupondest=/u04/orabkp/CKMDBP/xtt/incrbkp
storageondest=/u05/oradata/CKMDBP

```

```
mkdir -p ${stageondest} ${backupondest} ${storageondest}
```

Getting Started :

1. Initiate backup on Source

```
As Oracle user create a screen session to avoid lost host connection

screen -S xtts_migration_src -h 99999

${XTTS_HOME}/bin/source_capture.sh --srcfullbackup ${XTTS_HOME}

This will perform below actions in high level:

   1. Create required Directory on source DB Host
   2. Create datapump_dir 'XTTS_DIR' on database
   3. Initiate Full Database Backup
   4. Calculate Backup size
   5. Capture datafile count
   6. Add a CRON entry on source DB host that will be scheduled
      once in a day to take incremental backup
   7. Update Source Capture Status and send mail
```

2. Initiate Recovery on Target DB host

```
Start this process after you receive success mail from step 1.

As Oracle user create a screen session to avoid lost host connection

screen -S xtts_migration_tgt -h 99999

${XTTS_HOME}/bin/target_apply.sh --tgtfullapply ${XTTS_HOME}

This will perform below actions in high level:

   1. Create required Directory on Target DB Host
   2. Create datapump_dir 'XTTS_DIR' on database
   3. Fetch required files from Source DB host
   4. Split backup files for parallel pull
   5. Pull backups from source in parallel
   6. Validate backup size and datafile counts
   7. Convert pulled backup as per target endian format
   8. Add a CRON entry on target DB host that will be scheduled
      once in a day to recover incremental backup pulled from source DB
   9. Update Target Apply Status and send mail
```

3. At this point of time incremental backup at source and incremental backup apply at target runs automatically,

but check below files to understand the time taken by backup and restore operations

```
Source :

${XTTS_HOME}/bin/source_capture_status.txt

Target :

${XTTS_HOME}/bin/target_apply_status.txt
```

4. Before cutover day change CRON schedule, so that before cutover window once backup and recovery complete and during the cutover window no backup and recovery invoked. CRON entries can be removed/commented
5. During Cutover window run below command at source DB

```
${XTTS_HOME}/bin/xttx_cutover_script_generator.sh ${XTTS_HOME}

This will create cutover related scripts under ${XTTS_HOME}/cutover and transfer
to the Target DB under ${XTTS_HOME}/cutover

00_both_readme.txt
01_src_tablespace_ro.sh
02_src_final_incremental_backup.sh
03_src_tts_exp.sh
04_src_obj_exp.sh
05_src_db_stat_exp.sh
06_tgt_final_incremental_apply.sh
07_tgt_tts_imp.sh
08_tgt_obj_imp.sh
09_tgt_tablespace_rw.sh
10_tgt_db_stat_imp.sh
11_tgt_recompile_objects.sh
12_both_validate_tbs_obj.sql
13_tgt_rename_df_combined.sh
14_tgt_shutdown_db.sh
15_tgt_move_datafiles.sh
16_tgt_mount_db.sh
17_tgt_rename_datafiles.sql
18_tgt_open_db.sh
obj_metadata_expdp.par
obj_metadata_impdp.par
tts_metadata_expdp.par
tts_metadata_impdp.par

Read 00_both_readme.txt for mode details about the scripts before executing

File naming standard :

   A. Files need to executed in the order as numbered
   B. File name contained 'src' need to be executed on source DB host
      File name contained 'tgt' need to be executed on target DB host
      File name contained 'both' need to be executed on source and target DB host
```

6. Execute generated commands as per their naming standard and order.

High Level Implementation :

This is a set of three scripts to cover the cross platform migration end-to-end with very little manual effort.

1. source_capture.sh
2. target_apply.sh
3. xttx_cutover_script_generator.sh

Improvement/Challenges :

1. EC2 instance, by default do not enable port 25 for mail communication. This is critical for timely updating DBA if there is any issue in script execution. https://aws.amazon.com/premiumsupport/knowledge-center/ec2-port-25-throttle/ . To overcome this limitation we may need to explore SNS as an option. **(Done)**
2. HP datafile naming standard is `{DataFileMountDir}/{DATABASENAME}{tablespace_name}_{counter}.dbf` but xTTS script by default using datafile naming convention`{DataFileMountDir}/{TABLESPACENAME}_{fileNumber}.dbf`. **(Fixed)**
3. Test need to be done for datafile addition scenario **(Fixed, manual intervention required)**
4. Is there any way we can push file from source to Target **(Fixed)**
5. If there is any failure stop next schedule **(Done)**
6. Enable logging for all scripts generated by cutover script **(Done)**
7. Match backup size between source and target DB **(Done)**
8. Target apply iteration mismatch should not create lock **(Done)**

Known Issues:

1. If datafiles are added to tablespace part of migration, since last incremental backup and/or a new tablespace name is added to the xtt.properties, the script will fail with below message in mail and incremental backup will stop here.

```
2020-07-16 15:39:02 | Current Iteration Number : 3 |
2020-07-16 15:39:02 | Initiating Incremental backup on Source DB |
2020-07-16 15:39:30 | Incremental backup Failed with Exit Status : 25, please check lo
```

Fix:

```
Log in to the source database host as 'oracle' user and open below file
${XTTS_HOME}/bin/FAILED and follow the instuctions.
```

```
  Please Do the following:
--------------------------
 1. Copy fixnewdf.txt from source to destination temp dir

    scp ${SRC_XTTS_HOME}/bin/fixnewdf.txt oracle@${TGT_DB_HOST}:${TGT_XTTS_HOME}/bin/

 2. Copy new datafile copies
    XTTS2_TS_37.tf,XTTS3_TS_36.tf
    from /u01/app/oracle/orabkp/fullbkp to the <stage_dest> in destination

    scp ${DFCOPYDIR}/<<bak_df>>.tf oracle@${TGT_DB_HOST}:${DEST_DIR}/

 3. On Destination, run $ORACLE_HOME/perl/bin/perl xttdriver.pl --fixnewdf

    export TMPDIR=${SRC_XTTS_HOME}/bin
    export ORACLE_HOME=<<>>
    export ORACLE_SID=<<>>
    $ORACLE_HOME/perl/bin/perl xttdriver.pl --fixnewdf

 4. Delete FAILED and source_capture.sh.lock file on source DB host

    rm ${SRC_XTTS_HOME}/bin/FAILED
    rm ${SRC_XTTS_HOME}/bin/source_capture.sh.lock
```