10 Shell Scripts

TO AUTOMATE PROCESSES

Caution: Please use the commands with care, try them on test environments first.

1. Monitor goldengate process

The following script is used to monitor goldengate processes like extract and replicat. And in case extract or replicat is down, it will send alert to the respective email ids.

SCRIPT PREPARATION:

First create a shell script file and name it gg_alert.sh

Then, give it the necessary priviledges. View your file using:

cat gg_alert.sh

```
#!/bin/bash
EMAIL_LIST="support@dbaclass.com"

OIFS=$IFS
IFS="
"
NIFS=$IFS

function status {
OUTPUT=`$GG_HOME/ggsci << EOF
info all
exit
EOF`
}
function alert {
for line in $OUTPUT
do
if [[ $(echo "${line}"|egrep 'STOP|ABEND' >/dev/null;echo $?) = 0 ]]
then
GNAME=$(echo "${line}" | awk -F" " '{print $3}')
GSTAT=$(echo "${line}" | awk -F" " '{print $2}')
GTYPE=$(echo "${line}" | awk -F" " '{print $1}')
case $GTYPE in
"MANAGER")
cat $GG_HOME/dirrpt/MGR.rpt | mailx -s "${HOSTNAME} - GoldenGate ${GTYPE}
${GSTAT}" $NOTIFY ;;
```

```
"EXTRACT"|"REPLICAT")
cat $GG_HOME/dirrpt/"${GNAME}".rpt |mailx -s "${HOSTNAME} - GoldenGate
${GTYPE} ${GNAME} ${GSTAT}" $EMAIL_LIST ;;
esac
fi
done
}

export GG_HOME=/goldengate/install/software/gghome_1
export ORACLE_HOME=/oracle/app/oracle/product/12.1.0/db_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
status
alert
```

Finally configure the script in crontab with 30 min interval.

```
00,30 * * * * /home/goldengate/gg_alert.sh > /home/goldengate/gg_alerts.log
```

2. Monitor lag in standby datbase using dgmgrl

Below script is helpful in monitoring lag in standby database and send mail to DBAs in case the lag is increasing. For the script to work, make sure dataguard broker is enabled between primary and standby database.

SCRIPT PREPARATION:

PRIMARY DB UNIQUE_NAME – > PRIMDB
STANDBY DB UNIQUE_NAME -> STYDB

cat /home/oracle/dgmgrl_standby_lag.sh

```
#!/bin/bash
export ORACLE_HOME=/oracle/app/oracle/product/12.1.0/dbhome_1
export ORACLE_SID=primdb
export PATH=$ORACLE_HOME/bin:$PATH
echo -e "show database stydb"|${ORACLE_HOME}/bin/dgmgrl sys/orcl1234 >
DB_DG_DATABASE.log
cat /home/oracle/DB_DG_DATABASE.log  | grep "Apply Lag"  >
FILTERED_DB_DG_DATABASE.log
time_value=`cut -d " " -f 14 FILTERED_DB_DG_DATABASE.log`
time_param=`cut -d " " -f 15 FILTERED_DB_DG_DATABASE.log`
if [[ "$time_param" == "minutes" && "$time_value" -ge 1 ]]
then
mailx -s "DREAIDB LAG is in minutes "  suppor@dbaclass.com<DB_DG_DATABASE.log
else
if [[ "$time_param" == "seconds" && "$time_value" -ge 30 ]]
then
mailx -s "DREAIDB LAG is in seconds
```

```
"   support@dbaclass.com<DB_DG_DATABASE.log
else
if  [[ "$time_param" == "hour(s)" && "$time_value" -ge 1 ]]
then
mailx -s "DREAIDB LAG is in hours "   support@dbaclass.com
<DB_DG_DATABASE.log
fi
fi
fi
```

Now configure the the script in crontab

```
00,10,20,30,40,50 * * * * /home/oracle/dgmgrl_standby_lag.sh >
/tmp/dg_lag.log
```

3. Delete old archives using RMAN

If the requirement is to delete archive log backups automatically (without taking backup), then below shell script can be configured in crontab.

prepare the shell script.

cat rman_arch_del.sh

```
#!/bin/bash
export ORACLE_HOME=/oracle/app/oracle/product/12.1.0.2.0
export ORACLE_SID=PARIS12C
export PATH=$ORACLE_HOME/bin:$PATH

delBackup () {
rman log=/home/oracle/arch_del.log << EOF
connect target /
DELETE noprompt ARCHIVELOG ALL COMPLETED BEFORE 'sysdate-1';
CROSSCHECK ARCHIVELOG ALL;
DELETE EXPIRED ARCHIVELOG ALL;
exit
EOF
}
# Main

delBackup
```

Now configure in crontab:

```
00 22 * * * /u01/app/oracle/rman_arch_del.sh > /tmp/rmanarch.log
```

4. Monitoring blocking sessions

Below is the shell script, to be configured in crontab, which will send mail incase of blocking session observed in the database .
In the mail body it will contain the blocking sessions details also.

1. Prepare the blocker.sql file.[for blocking sessions more than 10 seconds)

```
set feed off
set pagesize 200
set lines 299
col event for a31
SELECT
s.inst_id,
s.blocking_session,
s.sid,
s.serial#,
s.seconds_in_wait,
s.event
FROM
gv$session s
WHERE
blocking_session IS NOT NULL and s.seconds_in_wait > 10;
```

2. Shell script.(/home/oracle/monitor/blocker.sh)

You need to define the ORACLE_HOME,ORACLE_SID respectively.

```
export ORACLE_HOME=/oracle/app/oracle/product/12.1.0/dbhome_1
export ORACLE_SID=ORCL
export PATH=$ORACLE_HOME/bin:$PATH
logfile=/home/oracle/monitor/block_alert.log
sqlplus -s "/as sysdba" > /dev/null << EOF
spool $logfile
@/home/oracle/monitor/blocker.sql
spool off
exit
EOF
count=`cat $logfile|wc -l`
if [ $count -ge 1 ];
 then mailx -s "BLOCKING SESSION REPORTED IN PROD DB ( > 10 SEC) "
support@dbaclass.com < $logfile
fi
```

3. configure in crontab(every one minute)

```
* * * * * /home/oracle/monitor/blocker.sh > /tmp/block.log
```

5. Monitor asm diskgroup usage

The following is a shell script that will trigger a mail alert, if the utilization of the asm diskgroup reached 90 percent.

1. Below is the shell script.

Make sure to update ORACLE_HOME, ORACLE_SID inside the shell script.

cat /export/home/oracle/asm_dg.sh

```
export ORACLE_HOME=/oracle/app/oracle/product/12.1.0.2/dbhome_1
export ORACLE_SID=PRODDB1
export PATH=$ORACLE_HOME/bin:$PATH
logfile=/export/home/oracle/asm_dg.log
sqlplus -s "/as sysdba" > /dev/null << EOF spool $logfile
SET LINESIZE 150
SET PAGESIZE 9999
SET VERIFY off
COLUMN group_name
FORMAT a25 HEAD 'DISKGROUP_NAME'
COLUMN state FORMAT a11 HEAD 'STATE'
COLUMN type FORMAT a6 HEAD 'TYPE'
COLUMN total_mb FORMAT 999,999,999 HEAD 'TOTAL SIZE(GB)'
COLUMN free_mb FORMAT 999,999,999 HEAD 'FREE SIZE (GB)'
COLUMN used_mb FORMAT 999,999,999 HEAD 'USED SIZE (GB)'
COLUMN pct_used FORMAT 999.99 HEAD 'PERCENTAGE USED'

SELECT distinct name group_name , state state , type type ,
round(total_mb/1024) TOTAL_GB , round(free_mb/1024) free_gb ,
round((total_mb - free_mb) / 1024) used_gb ,
round((1- (free_mb / total_mb))*100, 2) pct_used from
v$asm_diskgroup where round((1- (free_mb / total_mb))*100, 2) > 90 ORDER BY
name;
spool off
exit
EOF
count=`cat $logfile|wc -l`
#echo $count
if [ $count  -ge 4 ];
 then
  mailx -s "ASM DISKGROUP REACHED 90% UTILIZATION" support@dbaclass.com <
$logfile
fi
```

2. Give proper permission:

```
chmod 755 /export/home/oracle/asm_dg.sh
```

3. Configure in crontab:

```
0,15,30,45 * * * * /export/home/oracle/asm_dg.sh
```

6. To report failed login attempt in oracle

Configure a shell script in crontab, that will send alert to DB support Team in case of any invalid login attempts in the database.

1. First, enable audit for create session

```
SQL> audit create session;

Audit succeeded.
```

2. Final shell script

Below script for any invalid login attempts in last 15 minutes.

cat /export/home/oracle/invalid_log.sh

```
export ORACLE_HOME=/oracle/app/oracle/product/12.1.0/dbhome_1
export ORACLE_SID=SBIP18DB
export PATH=$ORACLE_HOME/bin:$PATH
logfile=/export/home/oracle/test.log
sqlplus -s "/as sysdba" > /dev/null << EOF
spool $logfile
set pagesize 1299
set lines 299
col username for a15
col userhost for a13
col timestamp for a39
col terminal for a23
SELECT username,userhost,terminal,to_char(timestamp,'DD/MM/YY HH24:MI:SS' )
"TIMESTAMP" ,
CASE
when returncode=1017 then 'INVALID-attempt'
when returncode=28000 then 'account locked'
end "FAILED LOGIN ACTION"
FROM dba_audit_session where timestamp > sysdate-1/9and returncode in
(1017,28000);
spool off
exit
```

```
EOF
count=`cat $logfile|wc -l`
#echo $count
if [ $count  -ge 4 ];
 then
  mailx -s "INVALID ATTEMPS IN DB  " support@dbaclass.com < $logfile
fi
```

3. provide proper permission:

```
chmod 755 invalid_log.sh
```

4. Configure in crontab:

```
0,15,30,45 * * * * /export/home/oracle/invalid_log.sh
```

7. A script for file system alert

Below is script to notification when a mount point or filesystem usage crosses a threshold value.

For solaris

```
#!/bin/sh

df -h | egrep -v '/system|/platform|/dev|/etc|lib' | awk '{print $6 " "
$5}'|cut -d% -f1|while read fs val

do

if [ $val -ge 90 ]
then
echo "The $fs usage high $val% \n \n \n `df -h $fs`" | mailx -s "Filesystem
$fs Usage high on Server `hostname`" support@dbaclass.com

fi
done
```

Put in crontab:

```
00 * * * * /usr/local/scripts/diskalert.sh
```

For monitoring zpool usage in solaris:

```
zpool list | awk '{print $5}'| grep -v CAP | cut -d% -f1| while read val

do
```

```
if [ $val -ge 80 ]
then
echo "The $fs usage high $val% \n \n \n `df -h $fs`" | mailx -s "Filesystem
$fs Usage high on Server `hostname`" rpatro.c@stc.com.a

fi
done
```

Put in crontab as below:

```
00 * * * * /usr/local/scripts/zpoolusage.sh
```

8. Alert log rotation script in oracle

Alert log size will grow in Oracle database from day to day. So for housekeeping, we need to move the existing alert log to a backup location and compress there. Upon moving the alert log, the database will create a fresh alert log automatically.

1. **Below is the shell script.**

We need to define the **ORACLE_HOME** in the script. and **ORACLE_SID** will be passed as an argument while running the script.

```
# $Header: rotatealertlog.sh
#
*=============================================================================
========+
# | AUTHOR : DBACLASS SUPPORT TEAM
# | |
#
+=============================================================================
========+
# |
#!/bin/bash
echo ========================
echo Set Oracle Database Env
echo ========================

ORACLE_SID=$1; export ORACLE_SID
ORACLE_HOME=/oracle/app/oracle/product/12.1.0.2/dbhome_1
ORACLE_BASE=/oracle/app/oracle; export ORACLE_BASE
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib; export LD_LIBRARY_PATH
PATH=$ORACLE_HOME/bin:$PATH;export PATH
TO_DATE="20`date +%y%m%d`"; export TO_DATE

echo =======
echo Extract Alert log location
```

```
echo =======
export VAL_DUMP=$(${ORACLE_HOME}/bin/sqlplus -S /nolog <<EOF
conn /as sysdba
set pages 0 feedback off;
prompt
SELECT value from v\$parameter where NAME='core_dump_dest';
exit;
EOF
)
export LOCATION=`echo ${VAL_DUMP} | perl -lpe'$_ = reverse' |awk '{print
$1}'|perl -lpe'$_ = reverse'`
export ALERTDB=${LOCATION}/alert_$ORACLE_SID.log
export ELOG=$( echo ${ALERTDB} | sed s/cdump/trace/)

echo =======
echo Compress current
echo =======

if [ -e "$ELOG" ] ; then
 mv ${ELOG} ${ELOG}_${TO_DATE};
 gzip ${ELOG}_${TO_DATE};
 > ${ELOG}
else
 echo not found
fi

exit
```

2. **Configure in crontab:**

SCHEDULE – Weekly once

Here, we have passed the **ORACLE_SID** (PRODDB) as **argument**

```
00 22 * * 5 /u01/app/oracle/dbscripts/rotatealertlog.sh PRODDB
```

9. Monitoring Tablespace

Below script can be configured in crontab to send a notification to the support DBAs in case tablespace usage crosses a threshold.

1. First, make the below .sql file, which will be used inside the shell script.

In this script we have defined the threshold as 90%. You can change it as per your requirement.

```
cat /export/home/oracle/Housekeeping/scripts/tablespace_alert.sql
```

```
set feedback off
set pagesize 70;
set linesize 2000
set head on
COLUMN Tablespace        format a25 heading 'Tablespace Name'
COLUMN autoextensible         format a11              heading 'AutoExtend'
COLUMN files_in_tablespace    format 999             heading 'Files'
COLUMN total_tablespace_space format 99999999 heading 'TotalSpace'
COLUMN total_used_space       format 99999999 heading 'UsedSpace'
COLUMN total_tablespace_free_space format 99999999 heading 'FreeSpace'
COLUMN total_used_pct              format 9999      heading '%Used'
COLUMN total_free_pct              format 9999     heading '%Free'
COLUMN max_size_of_tablespace      format 99999999 heading 'ExtendUpto'
COLUM total_auto_used_pct         format 999.99      heading 'Max%Used'
COLUMN total_auto_free_pct         format 999.99      heading 'Max%Free'
WITH tbs_auto AS
     (SELECT DISTINCT tablespace_name, autoextensible
                FROM dba_data_files
               WHERE autoextensible = 'YES'),
     files AS
     (SELECT   tablespace_name, COUNT (*) tbs_files,
               SUM (BYTES/1024/1024) total_tbs_bytes
          FROM dba_data_files
      GROUP BY tablespace_name),
     fragments AS
     (SELECT   tablespace_name, COUNT (*) tbs_fragments,
               SUM (BYTES)/1024/1024 total_tbs_free_bytes,
               MAX (BYTES)/1024/1024 max_free_chunk_bytes
          FROM dba_free_space
      GROUP BY tablespace_name),
     AUTOEXTEND AS
     (SELECT   tablespace_name, SUM (size_to_grow) total_growth_tbs
          FROM (SELECT   tablespace_name, SUM (maxbytes)/1024/1024
size_to_grow
                    FROM dba_data_files
                   WHERE autoextensible = 'YES'
                GROUP BY tablespace_name
                UNION
                SELECT   tablespace_name, SUM (BYTES)/1024/1024 size_to_grow
                    FROM dba_data_files
                   WHERE autoextensible = 'NO'
                GROUP BY tablespace_name)
      GROUP BY tablespace_name)
SELECT c.instance_name,a.tablespace_name Tablespace,
       CASE tbs_auto.autoextensible
          WHEN 'YES'
             THEN 'YES'
          ELSE 'NO'
       END AS autoextensible,
       files.tbs_files files_in_tablespace,
       files.total_tbs_bytes total_tablespace_space,
       (files.total_tbs_bytes - fragments.total_tbs_free_bytes
       ) total_used_space,
       fragments.total_tbs_free_bytes total_tablespace_free_space,
       round((  (  (files.total_tbs_bytes - fragments.total_tbs_free_bytes)
           / files.total_tbs_bytes
```

```
          )
        * 100
       )) total_used_pct,
       round(((fragments.total_tbs_free_bytes / files.total_tbs_bytes) * 100
       )) total_free_pct
  FROM dba_tablespaces a,v$instance c , files, fragments, AUTOEXTEND,
tbs_auto
WHERE a.tablespace_name = files.tablespace_name
   AND a.tablespace_name = fragments.tablespace_name
   AND a.tablespace_name = AUTOEXTEND.tablespace_name
   AND a.tablespace_name = tbs_auto.tablespace_name(+)
and (((files.total_tbs_bytes - fragments.total_tbs_free_bytes)/
files.total_tbs_bytes))* 100 > 90
order by total_free_pct;
```

2. Now prepare the shell script:

At the beginning of the script, we need to define the env variables like ORACLE_HOME, PATCH, LD_LIBRARY_PATH, ORACLE_SID.

Below is the final script(tablespace_threshold.ksh)

```
cat /export/home/oracle/Housekeeping/scripts/tablespace_threshold.ksh
```

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export ORACLE_SID=PRODDB
cd /export/home/oracle/Housekeeping/scripts
logfile=/export/home/oracle/Housekeeping/scripts/Tablespace_alert.log
cnt1=`ps -ef|grep pmon|grep $ORACLE_SID|wc -l`
if [ $cnt1 -eq 1 ];
then
sqlplus -s "/as sysdba" > /dev/null << EOF
spool $logfile
@/export/home/oracle/Housekeeping/scripts/tablespace_alert.sql
spool off
exit
EOF
# If there are more then these two lines in the output file, mail it.
count=`cat $logfile|wc -l`
#echo $count
if [ $count  -ge 4 ];
 then
  mailx -s "TABLESPACE ALERT FOR PROD DB  " support@dbaclass.com <$logfile
fi
fi
```

3. Now configure in crontab:

```
0,15,30,45 * * * *
/export/home/oracle/Housekeeping/scripts/tablespace_threshold.ksh >
/export/home/oracle/Housekeeping/logs/ts_alert.log  2>&1
```

10. A script for monitoring Alert log

Configure a shell script to monitor alert log for all the databases on a server once in every 15 min.And in the case of any ORA- error mail to the DBA TEAM.

Below script is prepared using the ADRCI utility of oracle 11g. It will monitor alert log for all the databases having same oracle base.

SCRIPT:(Adrci_alert_log.ksh)

```
########################################################
###### ALERT LOG CHECKING VIA ADRCI
######  Author - DBACLASS ADMIN
########################################################

LOG_DIR=/export/home/oracle/Housekeeping/logs/alert_log_check_daily.txt
adrci_homes=( $(adrci exec="show homes" | egrep -e rdbms ))
echo '##############################' > $LOG_DIR
echo '############################ALERT LOG OUTPUT FOR LAST 15 MINUTES
############################' >> $LOG_DIR
echo '##############################' >> $LOG_DIR

for adrci_home in ${adrci_homes[@]}
do

echo ' '>>$LOG_DIR
echo '################################################' >> $LOG_DIR
echo '########################################' >> $LOG_DIR
echo ' '>>$LOG_DIR
echo $adrci_home' Alert Log' >> $LOG_DIR
adrci exec="set home ${adrci_home}; show alert -p \\\"message_text like
'%ORA-%' and originating_timestamp > systimestamp-1/96\\\"" -term >> $LOG_DIR

done
num_errors=`grep  -c 'ORA' $LOG_DIR`
if [ $num_errors != 0 ]
then

mailx -s "ORA- error found in alert Log of the server  " support@dbaclass.com
<$LOG_DIR

fi
```

Give 755 permission to the script

```
chmod 755  Adrci_alert_log.ksh
```

Configure the script in crontab:

```
0,15,30,45 * * * *
/export/home/oracle/Housekeeping/scripts/Adrci_alert_log.ksh >
/export/home/oracle/Housekeeping/logs/error_alert.log 2>&1
```