

```

# get the last host of the cluster from the key_scan.sh script
# where all the members are listed
LASTHOST=`/usr/bin/tail -1 /home/grid/key_scan.sh | /bin/awk -F " " '{print $2}'`
# get my own hostname
ME=`/bin/hostname`
# only if I am the last host start the configuration of the GI
if [ "${LASTHOST}" == "${ME}" ]
then
    # add the hosts keys for ssh
    /bin/su - grid -c "/home/grid/key_scan.sh"
    /bin/cp /home/grid/key_scan.sh /var/tmp/
    /bin/su - oracle -c "/var/tmp/key_scan.sh"
    # configure GI for cluster operations on this node
    /opt/oracle/product/crs/11.2.0.4/crs/install/rootcrs.pl
    # create the environment for the grid user
    /bin/su - grid -c "echo \"ORACLE_HOME=/opt/oracle/product/crs/11.2.0.4\" >> ~/.bashrc"
    /bin/su - grid -c "echo \"ORACLE_SID=+ASM1\" >> ~/.bashrc"
    /bin/su - grid -c "echo \"PATH=\$ORACLE_HOME/bin:\$PATH\" >> ~/.bashrc"
    /bin/su - grid -c "echo \"export ORACLE_HOME ORACLE_SID PATH LD_LIBRARY_PATH\" >> ~/.bashrc"
    # create the environment for the oracle user
    /bin/su - oracle -c "echo \"ORACLE_HOME=/opt/oracle/product/base/11.2.0.4\" >> ~/.bashrc"
    /bin/su - oracle -c "echo \"ORACLE_SID=racdb1\" >> ~/.bashrc"
    /bin/su - oracle -c "echo \"PATH=\$ORACLE_HOME/bin:\$PATH\" >> ~/.bashrc"
    /bin/su - oracle -c "echo \"export ORACLE_HOME ORACLE_SID PATH LD_LIBRARY_PATH\" >> ~/.bashrc"
    # get my ASM instance number
    ASMINSTNUM=`ps -ef | grep asm_pmon | grep -v grep | awk -F "_" '{print $3}' | awk -F "M" '{print $2}'`
    # get the other members of the cluster
    OTHERHOSTS=`/bin/cat /home/grid/key_scan.sh | /bin/grep -v ${ME} | /bin/awk -F " " '{print $2}'`
    # the ASM instance on the next host will be 2
    i=2
    for x in ${OTHERHOSTS}
    do
        ASMSID="+ASM"${i}
        DBSID="racdb"${i}
        /bin/su - grid -c "/usr/bin/ssh ${x} /home/grid/key_scan.sh"
        /bin/su - grid -c "/usr/bin/ssh -tt ${x} sudo /opt/oracle/product/crs/11.2.0.4/crs/install/rootcrs.pl"
        # environment for the grid user
        /bin/su - grid -c "/usr/bin/ssh ${x} echo \"ORACLE_HOME=/opt/oracle/product/crs/11.2.0.4\" \>\> ~/.bashrc"
        /bin/su - grid -c "/usr/bin/ssh ${x} echo \"ORACLE_SID=${ASMSID}\" \>\> ~/.bashrc"
        /bin/su - grid -c "/usr/bin/ssh ${x} echo \"PATH='\$ORACLE_HOME'/bin:'\$PATH'\" \>\> ~/.bashrc"
        /bin/su - grid -c "/usr/bin/ssh ${x} echo \"LD_LIBRARY_PATH='\$ORACLE_HOME'/lib:'\$LD_LIBRARY_PATH'\" \>\> ~/.bashrc"
        /bin/su - grid -c "/usr/bin/ssh ${x} echo \"export ORACLE_HOME ORACLE_SID PATH LD_LIBRARY_PATH\" \>\> ~/.bashrc"
        # environment for the oracle user
        /bin/su - oracle -c "/usr/bin/ssh ${x} echo \"ORACLE_HOME=/opt/oracle/product/base/11.2.0.4\" \>\> ~/.bashrc"
        /bin/su - oracle -c "/usr/bin/ssh ${x} echo \"ORACLE_SID=${DBSID}\" \>\> ~/.bashrc"
        /bin/su - oracle -c "/usr/bin/ssh ${x} echo \"PATH='\$ORACLE_HOME'/bin:'\$PATH'\" \>\> ~/.bashrc"
        /bin/su - oracle -c "/usr/bin/ssh ${x} echo \"LD_LIBRARY_PATH='\$ORACLE_HOME'/lib:'\$LD_LIBRARY_PATH'\" \>\> ~/.bashrc"
        /bin/su - oracle -c "/usr/bin/ssh ${x} echo \"export ORACLE_HOME ORACLE_SID PATH LD_LIBRARY_PATH\" \>\> ~/.bashrc"
        let i=i+1
    done
done

```

```

# correct asm listener
/bin/su - grid -c "srvctl add listener"
/bin/su - grid -c "srvctl add listener -l listener_asm -p 1526"
/bin/su - grid -c "srvctl modify asm -l listener_asm"
/bin/su - grid -c "srvctl remove listener"
/bin/su - grid -c "srvctl start listener -l listener_asm"
# add a listener for the database
/bin/su - grid -c "srvctl add listener"
/bin/su - grid -c "srvctl start listener -l listener"
# create data and fra diskgroups
/bin/su - grid -c "asmcmd mkgg '<dg name=\"data\" redundancy=\"external\"><disk string=\"/dev/sdc1\"/><a name=\"compatible.asm\" value=\"11.2\"/> <a
name=\"compatible.rdbms\" value=\"11.2\"/></dg>' "
/bin/su - grid -c "asmcmd mkgg '<dg name=\"fra\" redundancy=\"external\"><disk string=\"/dev/sdd1\"/><a name=\"compatible.asm\" value=\"11.2\"/> <a
name=\"compatible.rdbms\" value=\"11.2\"/></dg>' "
# mount the diskgroups
/bin/su - grid -c "asmcmd mount data"
/bin/su - grid -c "asmcmd mount fra"
# mount the diskgroups on the other hosts
for x in ${OTHERHOSTS}
do
    /bin/su - grid -c "/usr/bin/ssh ${x} asmcmd mount data"
    /bin/su - grid -c "/usr/bin/ssh ${x} asmcmd mount fra"
done
# needed for dbca
/bin/chmod 770 /opt/oracle/product/base/cfgtoollogs
# add the rac resource
/bin/su - oracle -c "srvctl add database -d racdb -o \$ORACLE_HOME -n racdb -a data,fra -c RAC"
# dbca will fail at the end because the database is already registered in GI, but that does not matter
# will switch the database to rac after that
/bin/su - oracle -c "dbca -silent -createDatabase -templateName General_Purpose.dbc -gdbName racdb -sid racdb1 -sysPassword admin123 -systemPassword
admin123 -emConfiguration NONE -redoLogFileSize 16 -storageType ASM -diskGroupName DATA -recoveryGroupName FRA -characterSet AL32UTF8
-registerWithDirService false -memoryPercentage 10 -databaseType OLTP"
# add the instance to GI
/bin/su - oracle -c "srvctl add instance -d racdb -i racdb1 -n ${ME}"
i=2
# add all other instances to GI
for x in ${OTHERHOSTS}
do
    DBSID="racdb"${i}
    /bin/su - oracle -c "srvctl add instance -d racdb -i ${DBSID} -n ${x}"
    let i=i+1
done
# create a temporary sql file for setting the RAC parameters
/bin/echo "alter system set cluster_database=TRUE scope=spfile;" > /var/tmp/switch_to_rac.sql
/bin/echo "alter system set remote_listener='lab-scan:1521' scope=spfile;" >> /var/tmp/switch_to_rac.sql
/bin/echo "alter system set instance_number=1 scope=spfile sid='racdb1';" >> /var/tmp/switch_to_rac.sql
/bin/echo "alter system set thread=1 scope=spfile sid='racdb1';" >> /var/tmp/switch_to_rac.sql
/bin/echo "alter system set undo_tablespace=UNDOTBS1 scope=spfile sid='racdb1';" >> /var/tmp/switch_to_rac.sql
# all instance specific parameters
i=2

```

```

STEP=10
for x in ${OTHERHOSTS}
do
    DBSID="racdb"${i}
    /bin/echo "create undo tablespace undotbs${i} datafile '+data' size 64m;" >> /var/tmp/switch_to_rac.sql
    /bin/echo "alter system set instance_number=${i} scope=spfile sid='${DBSID}';" >> /var/tmp/switch_to_rac.sql
    /bin/echo "alter system set thread=${i} scope=spfile sid='${DBSID}';" >> /var/tmp/switch_to_rac.sql
    /bin/echo "alter system set undo_tablespace=UNDOTBS2 scope=spfile sid='${DBSID}';" >> /var/tmp/switch_to_rac.sql
    # static at the moment: 3 groups per thread
    for (( y=1; y<=3; y++ ))
    do
        let group=${i}+${STEP}+${y}
        /bin/echo "alter database add logfile thread ${i} group ${group} size 16M;" >> /var/tmp/switch_to_rac.sql
    done
    /bin/echo "alter database enable public thread ${i};" >> /var/tmp/switch_to_rac.sql
    let i=i+1
    let STEP=STEP+10
done
# execute the temporary sql and shutdown the instance afterwards
/bin/su - oracle -c "sqlplus / as sysdba <<EOF
shutdown abort
startup
start /var/tmp/switch_to_rac.sql
shutdown immediate;
EOF"
# remove the file
/bin/rm -f /var/tmp/switch_to_rac.sql
# copy password file and spfile to the other hosts
i=2
for x in ${OTHERHOSTS}
do
    /bin/su - oracle -c "/usr/bin/scp /opt/oracle/product/base/11.2.0.4/dbs/initracdb1.ora ${x}:/opt/oracle/product/base/11.2.0.4/dbs/initracdb${i}.ora"
    /bin/su - oracle -c "/usr/bin/scp /opt/oracle/product/base/11.2.0.4/dbs/orapwracdb1 ${x}:/opt/oracle/product/base/11.2.0.4/dbs/orapwracdb${i}"
    let i=i+1
done
# start the complete rac database
/bin/su - oracle -c "srvctl start database -d racdb"
fi
# remove the entry from rc.local. we do not want it to be executed anymore
sed -i '$d' /etc/rc.local

```