**Automate Startup/Shutdown of Oracle Database On RHEL 7**

Please follow below all required steps from the beginning which has been segregated in three steps to verify the issue.

**1. Create a "scripts" directory:**

  mkdir -p /home/oracle/scripts

**2. Create an environment file called "setEnv.sh".**

**\*\*Change Oracle Home and Base location according to your environment.**

cat > /home/oracle/scripts/setEnv.sh <<EOF  
# Oracle Settings

export TMP=/tmp

export TMPDIR=$TMP

export ORACLE\_BASE=/u03/app/oracle

export ORACLE\_HOME=/u03/app/oracle/product/19.0.0

export ORACLE\_SID= oemtest

export PATH=/usr/sbin:/usr/local/bin:$PATH

export PATH=$ORACLE\_HOME/bin:$PATH

export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:/lib:/usr/lib

export CLASSPATH=$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib  
EOF

Add a reference to the "setEnv.sh" file at the end of the "/home/oracle/.bash\_profile" file if you want the settings to be applied for a normal login. The profile will not be set during the start/stop of a service, so this is not necessary for the automatic start/stop functionality.

# echo ". /home/oracle/scripts/setEnv.sh" >> /home/oracle/.bash\_profile

**3. Create a "start\_all.sh" and "stop\_all.sh" script that can be called from a startup/shutdown service. Make sure the ownership and permissions are correct.**

cat > /home/oracle/scripts/start\_all.sh <<EOF  
#!/bin/bash  
. /home/oracle/scripts/setEnv.sh

export ORAENV\_ASK=NO  
. oraenv  
export ORAENV\_ASK=YES

dbstart \$ORACLE\_HOME  
EOF

cat > /home/oracle/scripts/stop\_all.sh <<EOF  
#!/bin/bash  
. /home/oracle/scripts/setEnv.sh

export ORAENV\_ASK=NO  
. oraenv  
export ORAENV\_ASK=YES

dbshut \$ORACLE\_HOME  
EOF

chown -R oracle.oinstall /home/oracle/scripts  
chmod u+x /home/oracle/scripts/\*.sh

**DB should be able to start/stop the database with the following scripts run from the "oracle" user.**

$ ~/scripts/start\_all.sh  
$ ~/scripts/stop\_all.sh

**4. Once the instance is created, edit the "/etc/oratab" file setting the restart flag for each instance to 'Y'.**

UK1TST01:/oracle/app/oracle/product/12.1.0.2/dbhome\_1:Y

LISTENER:/oracle/app/oracle/product/12.1.0.2/dbhome\_1:Y

**5. Creating Linux Services from ROOT user:**

**\*\* Check the Oracle user group for the below file then we have use the same group in dbora.service script:**

ls -ltr /home/oracle/scripts/start\_all.sh

vi /lib/systemd/system/dbora.service <<EOF

[Unit]  
Description=The Oracle Database Service  
After=syslog.target network.target

[Service]  
# systemd ignores PAM limits, so set any necessary limits in the service.  
# Not really a bug, but a feature.  
# https://bugzilla.redhat.com/show\_bug.cgi?id=754285  
LimitMEMLOCK=infinity  
LimitNOFILE=65535

#Type=simple  
# idle: similar to simple, the actual execution of the service binary is delayed  
# until all jobs are finished, which avoids mixing the status output with shell output of services.  
RemainAfterExit=yes  
User=oracle  
**Group=oinstall**  
ExecStart=/home/oracle/scripts/start\_all.sh  
ExecStop=/home/oracle/scripts/stop\_all.sh

[Install]  
WantedBy=multi-user.target

EOF

**6. Reload systemctl so it can see the new service.**

 # systemctl daemon-reload

**7. Start the service and enable so it is automatically restarted on reboot.**

# systemctl start dbora.service  
# systemctl enable dbora.service

# systemctl status dbora.service 🡨---------- Server should be ACTIVE now

**8. Check database status: DB should be running**

# ps -ef|grep pmon; ps -ef|grep tns

**9. Bring down the database using systemctl**

# systemctl stop dbora.service

# systemctl status dbora.service 🡨---------- Server should be inactive

**9. Check database status: DB should be down**

# ps -ef|grep pmon; ps -ef|grep tns

**10. Start the service and check the database status. DB should be running**

# systemctl start dbora.service  
# systemctl status dbora.service

# ps -ef|grep pmon; ps -ef|grep tns

**11. Ask Linux team to reboot the host**

**12. Check the service status and database status once server will be up.**

# systemctl status dbora.service

# ps -ef|grep pmon; ps -ef|grep tns

**Artifacts:**



