**Quick Commands for RAC**

[Overview of Basic RAC Commands](http://www.pafumi.net/RAC_Commands.html#Overview_of_Basic_RAC_Management)  
[Basic RAC Management Commands](http://www.pafumi.net/RAC_Commands.html#Basic_RAC_Management_Commands)  
[Starting the Oracle RAC 10g Environment](http://www.pafumi.net/RAC_Commands.html#Starting_the_Oracle_RAC_10g_Environment)  
[Stopping the Oracle RAC 10g Environment](http://www.pafumi.net/RAC_Commands.html#Stopping_the_Oracle_RAC_10g_Environment)  
[Verify and Explore the RAC Cluster and RAC Database Environment](http://www.pafumi.net/RAC_Commands.html#Verify_and_Explore_the_RAC_Cluster_and)  
  
  
**RAC Architecture Picture**  
A diagram of a computer

Description automatically generated  
  
  
**Auto Start/Stop**  
The Oracle Cluster Services handle the automatic start and stop of the Cluster, Listener, ASM and the Database.  
For the Grid Control Agent, the auto start- stop is handled via the /etc/init.d/dbora script.  On server boot the dbora script starts the agent with /home/oracle/scripts/start\_agent. On server shutdown the dbora script calls /home/oracle/scripts/stop\_agent

**Overview of Basic RAC Management Commands**

The commands we will use are listed below. Remember that this document is a quick reference, and not an exhaustive list of all commands for managing your RAC environment.

|  |  |
| --- | --- |
| **Cluster Related Commands** | |
| crs\_stat -t | Shows HA resource status (hard to read) |
| crsstat | Ouptut of crs\_stat -t formatted nicely |
| ps -ef|grep d.bin | crsd.bin evmd.bin ocssd.bin |
| crsctl check crs | CSS,CRS,EVM appears healthy |
| crsctl stop crs | Stop crs and all other services |
| crsctl disable crs\* | Prevents CRS from starting on reboot |
| crsctl enable crs\* | Enables CRS start on reboot |
| crs\_stop -all | Stops all registered resources |
| crs\_start -all | Starts all registered resources |

\* These commands update the file /etc/oracle/scls\_scr/<node>/root/crsstart which contains the string “enable” or “disable” as appropriate.

|  |  |
| --- | --- |
| **Database Related Commands** | |
| srvctl start instance -d <db\_name> -i <inst\_name> | Starts an instance |
| srvctl start database -d <db\_name> | Starts all instances |
| srvctl stop database -d <db\_name> | Stops all instances, closes database |
| srvctl stop instance -d <db\_name> -i <inst\_name> | Stops an instance |
| srvctl start service -d <db\_name> -s <service\_name> | Starts a service |
| srvctl stop service -d <db\_name> -s <service\_name> | Stops a service |
| srvctl status service -d <db\_name> | Checks status of a service |
| srvctl status instance -d <db\_name> -i <inst\_name> | Checks an individual instance |
| srvctl status database -d <db\_name> | Checks status of all instances |
| srvctl start nodeapps -n <node\_name> | Starts gsd, vip, listener, and ons |
| srvctl stop nodeapps -n <node\_name> | Stops gsd, vip and listener |

There are three main background processes you can see when doing a ps –ef|grep d.bin.  They are normally started by init during the operating system boot process.  They can be started and stopped manually by issuing the command /etc/init.d/init.crs {start|stop|enable|disable}

1. /etc/rc.d/init.d/init.evmd
2. /etc/rc.d/init.d/init.cssd
3. /etc/rc.d/init.d/init.crsd

Once the above processes are running, they will automatically start the following services in the following order if they are enabled.  This list assumes you are using ASM and have a service set up for TAF/load balancing.

1. The nodeapps (gsd, VIP, ons, listener) are brought online.
2. The ASM instances are brought online.
3. The database instances are brought online.
4. Any defined services are brought online.

**Basic RAC Management Commands**

Now that we know the dependency tree and have some commands at our disposal, let’s have a look at them one at a time, starting with the cluster commands and processes.

**crs\_stat -t**

This command shows us the status of each registered resource in the cluster. I generally avoid this command because its output is hard to read since the names are truncated as you can see in the sample output below.  You can download a helpful script called crsstat from <http://www.dbspecialists.com/specialists/specialist2007-05.html> to make it easy on your eyes.  
Usually located under  
/u01/crs/oracle/product/10.2.0/crs/bin/crs\_stat -t

[oracle@green ~]$ crs\_stat -t  
Name Type Target State Host  
------------------------------------------------------------  
ora....SM1.asm application ONLINE ONLINE green  
ora....EN.lsnr application ONLINE ONLINE green  
ora.green.gsd application ONLINE ONLINE green  
ora.green.ons application ONLINE ONLINE green  
ora.green.vip application ONLINE ONLINE green  
ora.....RAC.cs application ONLINE ONLINE red  
ora....cl1.srv application ONLINE ONLINE green  
ora....cl2.srv application ONLINE ONLINE red  
ora.orcl.db application ONLINE ONLINE red  
ora....l1.inst application ONLINE ONLINE green  
ora....l2.inst application ONLINE ONLINE red  
ora....SM2.asm application ONLINE ONLINE red  
ora....ED.lsnr application ONLINE ONLINE red  
ora.red.gsd application ONLINE ONLINE red  
ora.red.ons application ONLINE ONLINE red  
ora.red.vip application ONLINE ONLINE red  
[oracle@green ~]$

**crsstat**

The output of this script is much better. You can learn more about this script and download it at <http://www.dbspecialists.com/specialists/specialist2007-05.html>.

[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm ONLINE ONLINE on green  
ora.green.LISTENER\_GREEN.lsnr ONLINE ONLINE on green  
ora.green.gsd ONLINE ONLINE on green  
ora.green.ons ONLINE ONLINE on green  
ora.green.vip ONLINE ONLINE on green  
ora.orcl.RAC.cs ONLINE ONLINE on red  
ora.orcl.RAC.orcl1.srv ONLINE ONLINE on green  
ora.orcl.RAC.orcl2.srv ONLINE ONLINE on red  
ora.orcl.db ONLINE ONLINE on red  
ora.orcl.orcl1.inst ONLINE ONLINE on green  
ora.orcl.orcl2.inst ONLINE ONLINE on red  
ora.red.ASM2.asm ONLINE ONLINE on red  
ora.red.LISTENER\_RED.lsnr ONLINE ONLINE on red  
ora.red.gsd ONLINE ONLINE on red  
ora.red.ons ONLINE ONLINE on red  
ora.red.vip ONLINE ONLINE on red  
[oracle@green ~]$

**ps -ef|grep d.bin**

We can use this command to verify that the CRS background processes are actually running.  It is implicit that they are running if the crs\_stat command and crsstat script work. If they do not work, you will want to verify the background processes are really running.

[root@green ~]# ps -ef|grep d.bin  
oracle    5335  3525  0 Jul11 ?        00:00:05 /u01/app/oracle/product/10.2.0/crs/bin/evmd.bin  
root      5487  3817  0 Jul11 ?        00:00:00 /u01/app/oracle/product/10.2.0/crs/bin/crsd.bin reboot  
oracle    5932  5392  0 Jul11 ?        00:00:00 /u01/app/oracle/product/10.2.0/crs/bin/ocssd.bin  
root     30486 30177  0 18:23 pts/1    00:00:00 grep d.bin  
[root@green ~]#

**crsctl check crs**

This command verifies that the above background daemons are functioning.

[oracle@green ~]$ crsctl check crs  
CSS appears healthy  
CRS appears healthy  
EVM appears healthy  
[oracle@green ~]$

**crsctl stop crs**

We’ll need to be logged onto the server as the root user to run this command. It will stop all HA resources on the local node, and it will also stop the above mentioned background daemons.

[oracle@green ~]$ crsctl stop crs  
Insufficient user privileges.  
[oracle@green ~]$ su  
Password:  
[root@green oracle]# crsctl stop crs  
Stopping resources. This could take several minutes.  
Successfully stopped CRS resources.  
Stopping CSSD.  
Shutting down CSS daemon.  
Shutdown request successfully issued.  
[root@green oracle]#

**crsctl disable crs**

This command will prevent CRS from starting on a reboot.  Note there is no return output from the command.

[root@green oracle]# crsctl disable crs  
[root@green oracle]#

We did a reboot after this and verified that CRS did not come back online because we wanted to do some operating system maintenance. Let’s check the status by running some of the commands we’ve just discussed.

[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
error connecting to CRSD at [(ADDRESS=(PROTOCOL=ipc)(KEY=ora\_crsqs))] clsccon 184  
  
[oracle@green ~]$ crsctl check crs  
Failure 1 contacting CSS daemon  
Cannot communicate with CRS  
Cannot communicate with EVM  
  
[oracle@green ~]$ ps -ef|grep d.bin  
oracle 6149 5582 0 15:54 pts/1 00:00:00 grep d.bin  
[oracle@green ~]$

Everything appears to be down on this node as expected.

Now let’s start everything back up. We will need to be root for this, unless you have been given permissions or sudo to run crsctl start crs.

[root@green oracle]# crsctl start crs  
Attempting to start CRS stack  
The CRS stack will be started shortly  
[root@green oracle]#

After a few minutes the registered resources for this node should come online. Let’s check to be sure:

[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm ONLINE ONLINE on green  
ora.green.LISTENER\_GREEN.lsnr ONLINE ONLINE on green  
ora.green.gsd ONLINE ONLINE on green  
ora.green.ons ONLINE ONLINE on green  
ora.green.vip ONLINE ONLINE on green  
ora.orcl.RAC.cs ONLINE ONLINE on red  
ora.orcl.RAC.orcl1.srv ONLINE ONLINE on green  
ora.orcl.RAC.orcl2.srv ONLINE ONLINE on red  
ora.orcl.db ONLINE ONLINE on red  
ora.orcl.orcl1.inst ONLINE ONLINE on green  
ora.orcl.orcl2.inst ONLINE ONLINE on red  
ora.red.ASM2.asm ONLINE ONLINE on red  
ora.red.LISTENER\_RED.lsnr ONLINE ONLINE on red  
ora.red.gsd ONLINE ONLINE on red  
ora.red.ons ONLINE ONLINE on red  
ora.red.vip ONLINE ONLINE on red  
[oracle@green ~]$

Let’s not forget to enable CRS on reboot:

[root@green oracle]# crsctl enable crs

**crs\_stop -all**

This is a handy script that stops the registered resources and leaves the CRS running.  This includes all services in the cluster, so it will bring down all registered resources on all nodes.

[oracle@green ~]$ crs\_stop -all  
Attempting to stop `ora.green.gsd` on member `green`  
Attempting to stop `ora.orcl.RAC.orcl2.srv` on member `red`  
Stop of `ora.orcl.TEST.orcl1.srv` on member `green` succeeded.  
Attempting to stop `ora.orcl.RAC.orcl1.srv` on member `green`  
Attempting to stop `ora.green.ons` on member `green`  
Attempting to stop `ora.orcl.RAC.cs` on member `red`  
Stop of `ora.green.gsd` on member `green` succeeded.  
Stop of `ora.orcl.RAC.orcl1.srv` on member `green` succeeded.  
Stop of `ora.orcl.RAC.orcl2.srv` on member `red` succeeded.  
Stop of `ora.orcl.TEST.orcl2.srv` on member `red` succeeded.  
Stop of `ora.green.ons` on member `green` succeeded.  
--snip--  
CRS-0216: Could not stop resource 'ora.orcl.orcl2.inst'.  
[oracle@green ~]$

Occasionally you will get the CRS-0216 error message shown above.  This is usually bogus, but you should re-check with crsstat and ps –ef|grep smon or similar to be sure everything has died off.

Let’s verify that crs\_stop -all worked as expected:

[root@green oracle]# crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm OFFLINE OFFLINE  
ora.green.LISTENER\_GREEN.lsnr OFFLINE OFFLINE  
ora.green.gsd OFFLINE OFFLINE  
ora.green.ons OFFLINE OFFLINE  
ora.green.vip OFFLINE OFFLINE  
ora.orcl.RAC.cs OFFLINE OFFLINE  
ora.orcl.RAC.orcl1.srv OFFLINE OFFLINE  
ora.orcl.RAC.orcl2.srv OFFLINE OFFLINE  
ora.orcl.db OFFLINE OFFLINE  
ora.orcl.orcl1.inst OFFLINE OFFLINE  
ora.orcl.orcl2.inst OFFLINE OFFLINE  
ora.red.ASM2.asm OFFLINE OFFLINE  
ora.red.LISTENER\_RED.lsnr OFFLINE OFFLINE  
ora.red.gsd OFFLINE OFFLINE  
ora.red.ons OFFLINE OFFLINE  
ora.red.vip OFFLINE OFFLINE  
[root@green oracle]#

Let’s move on to working with srvctl and managing individual resources. We will begin with the crs background daemons already running, and all registered resources being offline from the last step above.  We will first start the nodeapps, then the ASM instances, followed by the database instances, and lastly the services for TAF and load balancing.  This is the dependency order in our particular environment.  You may or may not have ASM or TAF and load balancing services to start in your environment.

**srvctl start nodeapps -n (node)**

This will bring up the gsd, ons, listener, and vip. The same command can shut down the nodeapps by replacing start with stop.

[oracle@green ~]$ srvctl start nodeapps -n green  
[oracle@green ~]$ srvctl start nodeapps -n red

Now we will check with crsstat again to be sure the nodeapps have started.

[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm OFFLINE OFFLINE  
ora.green.LISTENER\_GREEN.lsnr ONLINE ONLINE on green  
ora.green.gsd ONLINE ONLINE on green  
ora.green.ons ONLINE ONLINE on green  
ora.green.vip ONLINE ONLINE on green  
ora.orcl.RAC.cs OFFLINE OFFLINE  
ora.orcl.RAC.orcl1.srv OFFLINE OFFLINE  
ora.orcl.RAC.orcl2.srv OFFLINE OFFLINE  
ora.orcl.db OFFLINE OFFLINE  
ora.orcl.orcl1.inst OFFLINE OFFLINE  
ora.orcl.orcl2.inst OFFLINE OFFLINE  
ora.red.ASM2.asm OFFLINE OFFLINE  
ora.red.LISTENER\_RED.lsnr ONLINE ONLINE on red  
ora.red.gsd ONLINE ONLINE on red  
ora.red.ons ONLINE ONLINE on red  
ora.red.vip ONLINE ONLINE on red  
[root@green oracle]#

Now we need to start our ASM instances before we bring up our database and services.

**srvctl start asm -n (node)**

This will bring up our ASM instances on nodes green and red. Again, the same command will stop the ASM instances by replacing start with stop.

[oracle@green ~]$ srvctl start asm -n green  
[oracle@green ~]$ srvctl start asm -n red  
[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm ONLINE ONLINE on green  
ora.green.LISTENER\_GREEN.lsnr ONLINE ONLINE on green  
ora.green.gsd ONLINE ONLINE on green  
ora.green.ons ONLINE ONLINE on green  
ora.green.vip ONLINE ONLINE on green  
ora.orcl.RAC.cs OFFLINE OFFLINE  
ora.orcl.RAC.orcl1.srv OFFLINE OFFLINE  
ora.orcl.RAC.orcl2.srv OFFLINE OFFLINE  
ora.orcl.db OFFLINE OFFLINE  
ora.orcl.orcl1.inst OFFLINE OFFLINE  
ora.orcl.orcl2.inst OFFLINE OFFLINE  
ora.red.ASM2.asm ONLINE ONLINE on red  
ora.red.LISTENER\_RED.lsnr ONLINE ONLINE on red  
ora.red.gsd ONLINE ONLINE on red  
ora.red.ons ONLINE ONLINE on red  
ora.red.vip ONLINE ONLINE on red  
[oracle@green ~]$

Now let’s bring up our orcl1 and orcl2 instances, and verify they are up with crsstat. Once more we can replace start with stop and shutdown an individual instance if we so choose.

**srvctl start instance -d (database) -I (instance)**

[oracle@green ~]$ srvctl start instance –d orcl –i orcl1  
[oracle@green ~]$ srvctl start instance -d orcl -i orcl2  
[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm ONLINE ONLINE on green  
ora.green.LISTENER\_GREEN.lsnr ONLINE ONLINE on green  
ora.green.gsd ONLINE ONLINE on green  
ora.green.ons ONLINE ONLINE on green  
ora.green.vip ONLINE ONLINE on green  
ora.orcl.RAC.cs OFFLINE OFFLINE  
ora.orcl.RAC.orcl1.srv OFFLINE OFFLINE  
ora.orcl.RAC.orcl2.srv OFFLINE OFFLINE  
ora.orcl.db ONLINE ONLINE on red  
ora.orcl.orcl1.inst ONLINE ONLINE on green  
ora.orcl.orcl2.inst ONLINE ONLINE on red  
ora.red.ASM2.asm ONLINE ONLINE on red  
ora.red.LISTENER\_RED.lsnr ONLINE ONLINE on red  
ora.red.gsd ONLINE ONLINE on red  
ora.red.ons ONLINE ONLINE on red  
ora.red.vip ONLINE ONLINE on red  
[oracle@green ~]$

**srvctl start service -d (database) -s (service)**

Now we will finish up by bringing our load balanced/TAF service named RAC online.

[oracle@green ~]$ srvctl start service -d orcl -s RAC  
[oracle@green ~]$ crsstat  
HA Resource Target State  
----------- ------ -----  
ora.green.ASM1.asm ONLINE ONLINE on green  
ora.green.LISTENER\_GREEN.lsnr ONLINE ONLINE on green  
ora.green.gsd ONLINE ONLINE on green  
ora.green.ons ONLINE ONLINE on green  
ora.green.vip ONLINE ONLINE on green  
ora.orcl.RAC.cs ONLINE ONLINE on red  
ora.orcl.RAC.orcl1.srv ONLINE ONLINE on green  
ora.orcl.RAC.orcl2.srv ONLINE ONLINE on red  
ora.orcl.db ONLINE ONLINE on red  
ora.orcl.orcl1.inst ONLINE ONLINE on green  
ora.orcl.orcl2.inst ONLINE ONLINE on red  
ora.red.ASM2.asm ONLINE ONLINE on red  
ora.red.LISTENER\_RED.lsnr ONLINE ONLINE on red  
ora.red.gsd ONLINE ONLINE on red  
ora.red.ons ONLINE ONLINE on red  
ora.red.vip ONLINE ONLINE on red  
[oracle@green ~]$

There we have it; all of our resources are now online. The next steps would be to verify you can connect via SQL\*Plus or your favorite application.

**Starting the Oracle RAC Environment**  
The first step is to start the node applications (Virtual IP, GSD, TNS Listener, and ONS).  
Once the node applications are successfully started, then bring up the ASM instance.  
Finally, bring up the Oracle instance (and related services) and the Enterprise Manager Database console.  
  
srvctl start nodeapps -n <hostname1>  
srvctl start nodeapps -n <hostname2>  
srvctl start asm -n <hostname1>  
srvctl start asm -n <hostname2>  
srvctl start instance -d db\_name -i instancename1  
srvctl start instance -d db\_name -i instancename2  
srvctl start database -d <database name>  
srvctl start service -d <database name> -s <service name>  
emctl start dbconsole  
crs\_stat -t  
  
  
  
**Stopping the Oracle RAC Environment**  
The first step is to stop the Oracle instance.  
Once the instance (and related services) are down, then bring down the ASM instance.  
Finally, shut down the node applications (Virtual IP, GSD, TNS Listener, and ONS).  
  
emctl stop dbconsole  
srvctl stop service -d <database name> -s <service name>  
srvctl stop instance -d db\_name -i instancename1  
srvctl stop instance -d db\_name -i instancename2  
   or   $ srvctl stop database -d db\_name -o immediate   to bring all instances down  
rvctl stop listener -n hostname1  
rvctl stop listener -n hostname2  
srvctl stop asm -n <hostname1>  
srvctl stop asm -n <hostname2>  
srvctl stop nodeapps -n <hostname1>  
srvctl stop nodeapps -n <hostname2>  
crs\_stat -t  
  
Execute **crsctl**on the command line to check out all the available options.  
  
  
**Grid Control Agent**  
Set environment to ‘agent’ using . oraenv  
$ORACLE\_HOME/bin/emctl start agent  
$ORACLE\_HOME/bin/emctl stop agent  
  
  
  
**Verify and Explore the RAC Cluster and RAC Database Environment**  
  
Now that you have successfully installed a virtual two-node RAC database, it’s time to do a little exploration of the environment you have just set up.  
This section provides several srvctl commands and SQL queries that can be used to validate your RAC configuration  
There are five node-level tasks defined for SRVCTL:

* Adding and deleting node level applications.
* Setting and unsetting the environment for node-level applications.
* Administering node applications.
* Administering ASM instances.
* Starting and stopping a group of programs that includes virtual IP addresses, listeners, Oracle Notification Services, and Oracle Enterprise Manager agents *(for maintenance purposes)*.

**Check the status of application resources**  
  
**crs\_stat -t**  
Name           Type           Target    State     Host  
------------------------------------------------------------  
ora.devdb.db   application    ONLINE    ONLINE    rac1  
ora....b1.inst application    ONLINE    ONLINE    rac1  
ora....b2.inst application    ONLINE    ONLINE    rac2  
ora....SM1.asm application    ONLINE    ONLINE    rac1  
ora....C1.lsnr application    ONLINE    ONLINE    rac1  
ora.rac1.gsd   application    ONLINE    ONLINE    rac1  
ora.rac1.ons   application    ONLINE    ONLINE    rac1  
ora.rac1.vip   application    ONLINE    ONLINE    rac1  
ora....SM2.asm application    ONLINE    ONLINE    rac2  
ora....C2.lsnr application    ONLINE    ONLINE    rac2  
ora.rac2.gsd   application    ONLINE    ONLINE    rac2  
ora.rac2.ons   application    ONLINE    ONLINE    rac2  
ora.rac2.vip   application    ONLINE    ONLINE    rac2  
  
**Status of all instances and services  
srvctl status database -d devdb**  
Instance devdb1 is running on node rac1  
Instance devdb2 is running on node rac2  
  
**Status of a single instance  
srvctl status instance -d devdb -i devdb2**Instance devdb2 is running on node rac2  
  
**Status of node applications on a particular node**  
**srvctl status nodeapps -n rac1**  
VIP is running on node: rac1  
GSD is running on node: rac1  
Listener is running on node: rac1  
ONS daemon is running on node: rac1  
  
**srvctl status nodeapps -n rac2**  
VIP is running on node: rac2  
GSD is running on node: rac2  
Listener is running on node: rac2  
ONS daemon is running on node: rac2  
  
**Status of an ASM instance  
srvctl status asm -n rac1**  
ASM instance +ASM1 is running on node rac1.  
  
**srvctl status asm -n rac2**  
ASM instance +ASM2 is running on node rac2.     
  
**List all configured databases  
srvctl config database**  
devdb  
  
**Display configuration for our RAC database**  
**srvctl config database -d devdb**  
rac1 devdb1 /u01/app/oracle/product/10.2.0/db\_1  
rac2 devdb2 /u01/app/oracle/product/10.2.0/db\_1  
  
  
**Display the configuration for node applications - (VIP, GSD, ONS, Listener)**  
**srvctl config nodeapps -n rac1 -a -g -s -l**  
VIP exists.: /rac1-vip/192.168.2.31/255.255.255.0/eth0  
GSD exists.  
ONS daemon exists.  
Listener exists.  
  
  
**Display the configuration for the ASM instance(s)**  
**srvctl config asm -n rac1**  
+ASM1 /u01/app/oracle/product/10.2.0/db\_1  
  
  
**Check the status of Oracle Clusterware**  
rac1->**crsctl check crs**  
CSS appears healthy  
CRS appears healthy  
EVM appears healthy  
  
rac2-> **crsctl check crs**  
CSS appears healthy  
CRS appears healthy  
EVM appears healthy  
  
  
**List the RAC instances**

**SELECT inst\_id, instance\_number inst\_no, instance\_name inst\_name, parallel,  
 status, database\_status db\_status, active\_state state, host\_name host  
FROM gv$instance  
ORDER BY inst\_id;**  
 INST\_ID INST\_NO INST\_NAME PAR STATUS DB\_STATUS STATE HOST  
-------- -------- ---------- --- ------- ------------ --------- --------  
 1 1 devdb1 YES OPEN ACTIVE NORMAL rac1  
 2 2 devdb2 YES OPEN ACTIVE NORMAL rac2

**Check connectivity**  
Verify that you are able to connect to the instances and service on each node.  
**sqlplus system@devdb1  
sqlplus system@devdb2  
sqlplus system@devdb**  
  
  
**Check database configuration**rac1->**export ORACLE\_SID=devdb1**rac1->**sqlplus / as sysdba  
show sga**  
Total System Global Area  209715200 bytes  
Fixed Size                  1218556 bytes  
Variable Size             104859652 bytes  
Database Buffers          100663296 bytes  
Redo Buffers                2973696 bytes  
  
**Check all data files which are in the disk group**

**select name from v$datafile  
union  
select member from v$logfile  
union  
select name from v$controlfile  
union  
select name from v$tempfile;**  
  
NAME  
-------------------------------------------  
++RECOVERYDEST/devdb/controlfile/current.256.578676737  
++RECOVERYDEST/devdb/onlinelog/group\_1.257.578676745  
++RECOVERYDEST/devdb/onlinelog/group\_2.258.578676759  
++RECOVERYDEST/devdb/onlinelog/group\_3.259.578682963  
++RECOVERYDEST/devdb/onlinelog/group\_4.260.578682987  
++DG1/devdb/controlfile/current.256.578676735  
++DG1/devdb/datafile/example.263.578676853  
++DG1/devdb/datafile/indx.270.578685723  
++DG1/devdb/datafile/sysaux.261.578676829  
++DG1/devdb/datafile/system.259.578676767  
++DG1/devdb/datafile/undotbs1.260.578676809  
++DG1/devdb/datafile/undotbs1.271.578685941  
++DG1/devdb/datafile/undotbs2.264.578676867  
++DG1/devdb/datafile/undotbs2.272.578685977  
++DG1/devdb/datafile/users.265.578676887  
++DG1/devdb/datafile/users.269.578685653  
++DG1/devdb/onlinelog/group\_1.257.578676739  
++DG1/devdb/onlinelog/group\_2.258.578676753  
++DG1/devdb/onlinelog/group\_3.266.578682951  
++DG1/devdb/onlinelog/group\_4.267.578682977  
++DG1/devdb/tempfile/temp.262.578676841

**select file\_name,bytes/1024/1024 from dba\_data\_files;**  
FILE\_NAME                                   BYTES/1024/1024  
------------------------------------------- ---------------  
+DG1/devdb/datafile/users.259.606468449                   5  
+DG1/devdb/datafile/sysaux.257.606468447                240  
+DG1/devdb/datafile/undotbs1.258.606468449               30  
+DG1/devdb/datafile/system.256.606468445                480  
+DG1/devdb/datafile/undotbs2.264.606468677               25  
  
**select group#, type, member, is\_recovery\_dest\_file  
  from v$logfile  
  order by group#;**  
GROUP# TYPE    MEMBER                                              IS\_  
------ ------- --------------------------------------------------- ---  
     1 ONLINE  +RECOVERYDEST/devdb/onlinelog/group\_1.257.606468581 YES  
     1 ONLINE  +DG1/devdb/onlinelog/group\_1.261.606468575          NO  
     2 ONLINE  +RECOVERYDEST/devdb/onlinelog/group\_2.258.606468589 YES  
     2 ONLINE  +DG1/devdb/onlinelog/group\_2.262.606468583          NO  
     3 ONLINE  +DG1/devdb/onlinelog/group\_3.265.606468865          NO  
     3 ONLINE  +RECOVERYDEST/devdb/onlinelog/group\_3.259.606468875 YES  
     4 ONLINE  +DG1/devdb/onlinelog/group\_4.266.606468879          NO  
     4 ONLINE  +RECOVERYDEST/devdb/onlinelog/group\_4.260.606468887 YES  
  
rac1-> **export ORACLE\_SID=+ASM1**  
rac1-> **sqlplus / as sysdba**  
SQL> **show sga**  
Total System Global Area   92274688 bytes  
Fixed Size                  1217884 bytes  
Variable Size              65890980 bytes  
ASM Cache                  25165824 bytes  
  
SQL> **show parameter asm\_disk**  
NAME                           TYPE        VALUE  
------------------------------ ----------- ------------------------  
asm\_diskgroups                 string      DG1, RECOVERYDEST  
asm\_diskstring                 string  
  
**select group\_number, name, allocation\_unit\_size alloc\_unit\_size, state,  
         type, total\_mb, usable\_file\_mb  
  from v$asm\_diskgroup;**  
                       ALLOC                        USABLE  
 GROUP                  UNIT                 TOTAL    FILE  
NUMBER NAME             SIZE STATE   TYPE       MB      MB  
------ ------------ -------- ------- ------ ------ -------  
     1 DG1           1048576 MOUNTED NORMAL   6134    1868  
     2 RECOVERYDEST  1048576 MOUNTED EXTERN   2047    1713  
  
  
**select name, path, header\_status, total\_mb free\_mb, trunc(bytes\_read/1024/1024) read\_mb,  
         trunc(bytes\_written/1024/1024) write\_mb  
from v$asm\_disk;**  
NAME  PATH       HEADER\_STATU    FREE\_MB    READ\_MB   WRITE\_MB  
----- ---------- ------------ ---------- ---------- ----------  
VOL1  ORCL:VOL1  MEMBER             3067        229       1242  
VOL2  ORCL:VOL2  MEMBER             3067        164       1242  
VOL3  ORCL:VOL3  MEMBER             2047         11        354  
  
  
  
**Check flash recovery area space usage**  
 **select \* from v$recovery\_file\_dest;**  
NAME          SPACE\_LIMIT SPACE\_USED SPACE\_RECLAIMABLE NUMBER\_OF\_FILES  
------------- ----------- ---------- ----------------- ---------------  
+RECOVERYDEST  1572864000  331366400                 0               7  
  
**select \* from v$flash\_recovery\_area\_usage;**  
FILE\_TYPE    PERCENT\_SPACE\_USED PERCENT\_SPACE\_RECLAIMABLE NUMBER\_OF\_FILES  
------------ ------------------ ------------------------- ---------------  
CONTROLFILE                 .97                         0               1  
ONLINELOG                    20                         0               6  
ARCHIVELOG                    0                         0               0  
BACKUPPIECE                   0                         0               0  
IMAGECOPY                     0                         0               0  
FLASHBACKLOG                  0                         0               0