Assumption:

Clusterware	Oracle RAC
Version	11g R2
SCAN Properties: **Name** **DNS/GNS**	 x39-3001.xyz.com DNS
Node Names	Linux-223 Linux-224
DB name	XLM
Instance Names:	XLM1 XLM2

Solution:

1. Add PATH to .bash_profile.

vi ~/.bash_profile

```
export PATH=$PATH:$HOME/bin:/sbin:/u01/app/oracle/product/11.2.0.3/db_1/bin:/u01/app/11.2.0.3/grid/bin
```

Note: This is to find cluster utilities like crsctl, srvctl etc.

2. re-execute .bash_profile

```
. ~/.bash_profile
```

3. Set the DB environment

```
. oraenv
XLM
```

4. Create a new database user for our testing

```
sqlplus / as sysdba
create user Afshan identified by abc123;
grant dba to Afshan;
```

5. Verify all instances are up

```
srvctl status database -d XLM
```

6. Add a new service

```
srvctl add service -d XLM -s XLM_n1pn2p -r XLM1,XLM2 -P BASIC -e SELECT
```

Note: This will add a new service called XLM_n1pn2p. I am using n1pn2p because both node1 is preferred and node2 is also preferred.

7. Start the service

```
srvctl enable service -d XML
srvctl start service -d XML
```

8. Update the tnsnames.ora file with the connect identifier.

cd $ORACLE_HOME/network/admin
vi tnsnames.ora

```
XML =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = x39-3001.xyz.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = x39_n1pn2p)
    )
  )
```

9. Test the failover.

To test the failover, perfom the following steps.

A	1. Login to any other db server where oracle client is installed. 2. This server needs to be on the same network as the cluster you want to test. 3. set oracle env to any database that is running on server a. . oraenv 4. cd $ORACLE_HOME/network/admin 5. vi tnsnames.ora → and add the tnsentry created in step7 above. 6. Verify tnsping is working for XML database a. tnsping XML 7. Once the tnsping is confirmed, proceed to the next steps
B	While still on testing db server, login to SQL Plus with the DB user we created earlier. sqlplus Afshan@XML Verify the instance you are connected to select instance_name from v$instance; Assuming that it shows that you are connected to instance-1 XML1 Our goal is to shutdown instance-1 so that this connection should failover to instance-2

C	Open a new putty to node2 and run the following SQL to check which node “Afshan” is connected and if it has failed over gracefull.
	SQL is:
	`set linesize 200`
	`set pagesize 200`
	`col username format a12`
	`col machine format a20`
	`Select`
	`inst_id,`
	`username,`
	`machine,`
	`failover_type,`
	`failover_method,`
	`failed_over`
	`from`
	`gv$session`
	`where`
	`username='AFSHAN'`
	`order by 1,machine;`

D	Open a new putty and connect to any node to stop instance-1.

	srvctl stop instance -d XML -i XML1
E	Go back to putty opened in 9C and re-execute the SQL in the buffer. You should see the connection has failed over.

Hope this helps.

Moid Muhammad