

Configuring Active-Passive service on 2 node RAC cluster

Oracle RAC (2 nodes) supports active-passive mode enabling 1 node to "do the work" whilst node 2 remains comparatively idle.

During a failure of the active database instance, users failover to the passive instance. This feature can be configured by setting the following database initialisation parameter to 1 on both instances:

```
active_instance_count = 1
```

When you set this parameter to 1, the first instance you start up becomes the primary instance and accepts client connections.

The second instance starts up as a secondary instance and can accept client connections only if the first instance fails.

In such an event, the secondary instance becomes the primary instance.

In Oracle 11g the active-passive mode for 2 node RAC can be configured using a database service.

This is particularly useful when using OEM Grid Control, where both instances can be "seen" and evaluated.

When configuring active-passive using initialisation parameters, the passive node appears as down.

Configure a database service to have instance 1 as "preferred" and instance 2 as "available" using the Server Control utility.

Logon to either node as oracle user and execute the following commands:

```
srvctl add service -d <db_name> -s <service_name> -r <instance1> -a <instance2>
srvctl start service -d <db_name> -s <service_name>
```

E.g

```
srvctl add service -d MYDB -s ACTIVE_SRV -r MYDB1 -a MYDB2
srvctl start service -d MYDB -s ACTIVE_SRV
```

The above command may fail.

E.g.

```
PRKP-1030: Failed to start the service ACTIVE_SRV.
CRS-1007: Failed after successful dependency consideration
CRS-0223: Resource 'ora.MYDB.ACTIVE_SRV.cs' has placement error.
```

This issue is described in Metalink Note 372145.1 service start fails with CRS-1030 CRS-1007 CRS-0223

Bug.7560908/6674075 (96) CAN'T START SERVICES ,IF ITS INSTANCES REGISTERED WITH NODE NAME IN UPPER CASE

A workaround (when using uppercase hostnames)

```

srvctl add service -d <db_name> -s <service_name> -r <instance1>
srvctl start service -d <db_name> -s <service_name>
srvctl add service -d <db_name> -s <service_name> -u -a <instance2>

```

E.g

```

srvctl add service -d MYDB -s ACTIVE_SRV -r MYDB1 -a MYDB2
srvctl start service -d MYDB -s ACTIVE_SRV
srvctl add service -d MYDB -s ACTIVE_SRV -u -a MYDB2

```

The -u switch allows the service to be updated.

To see configuration of service:

```

srvctl config service -d <db_name> -s <service_name>

```

The `srvctl start service ..` command translates into an `ALTER SYSTEM SET service_names = '<service_name list>' SCOPE=MEMORY;` command, where the service name is appended to the existing list of service names. This can be seen the alert log of the database instance.

To enable transparent application failover (TAF) for your client ensure you add the following to your

`$ORACLE_HOME/network/admin/tnsnames.ora` file.

```

<tns alias> =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = <node a VIP>) (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP) (HOST = <node b VIP>) (PORT = 1521))
    (LOAD_BALANCE = no)
    (FAILOVER = true)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <service_name>)
      (FAILOVER_MODE= (TYPE=SELECT) (METHOD=BASIC))
    )
  )

```

E.g.

```

MYDB_ACTIVE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = RAC1-VIP) (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP) (HOST = RAC2-VIP) (PORT = 1521))
    (LOAD_BALANCE = no)
    (FAILOVER = true)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ACTIVE_SRV)
      (FAILOVER_MODE= (TYPE=SELECT) (METHOD=BASIC))
    )
  )

```

N.B.

Ensure the client's `tnsnames.ora` entry references the hostname (`hostname-vip`) and the `hosts` file contains the relevant entries. This is important for failover, as the server sends back the hostname to the client on connection failure for client redirection.

If the tnsnames.ora entry contains IP addresses and no entries exist in the hosts file, the hostname-vip cannot be resolved and the connection will fail to ORA-12545 “Connect failed because target host or object does not exist”. Furthermore, the ACTIVE_SRV service will NOT move back to the original instance following crash recovery. This has to be done manually.

The following example shows how to relocate a database service from one instance to another:

```
$ srvctl status service -d MYDB -s ACTIVE_SRV
Service ACTIVE_SRV is running on instance(s) MYDB2

$ sqlplus sys/password@MYDB_ACTIVE as sysdba

SQL> select instance_name from v$instance;

INSTANCE_NAME
-----
MYDB2

SQL> exit

$ srvctl relocate service -d MYDB -s ACTIVE_SRV -i MYDB2 -t MYDB1

$ srvctl status service -d MYDB -s ACTIVE_SRV
Service ACTIVE_SRV is running on instance(s) MYDB1

$ sqlplus sys/password@MYDB_ACTIVE as sysdba

SQL> select instance_name from v$instance;

INSTANCE_NAME
-----
MYDB1

SQL> exit
```