Cluster MYSQL for HA using Oracle CRS

Oracle clusterware CRS can be used to cluster for high availability (HA) non-Oracle resources. This wiki page describes the steps to set up HA mysql databases in a failover configuration using Oracle 10gR2 CRS.

Oracle CRS 10gR2 allows to define generic cluster resources. An implementation for HA mysql is described here. The macro steps are:

- a 2-node cluster with shared (SAN) storage is used
- oracle CRS and mysql are installed on both nodes
- 3 cluster resource profiles are defined: VIP, filesystem, mysql application
- the resources are created with dependencies, so that CRS will take care of running them on only 1 node at a time
- mysql startup and shutdown will be handled by the clusterware: **crs_start mysql1vip mysql1 fs_data1**
- monitor with crs_stat -t (or **crsstat.sh**)

Implementation steps

Software Installation

- a 2-node cluster is allocated with shared storage. CRS installation as in Installation_verbose
  - ♦ the main difference is that with configure mirroring (RAID 1+0) at the array level
- a filesystem for mysql is created (/data1) on shared storage.
  - ♦ the filesystem (/data1) can be mounted by all cluster nodes (but only one node at a time)
  - ♦ make sure that the filesystem is commented out in /etc/fstab (it will be mounted using cluster resources)
- mysql is installed locally on both nodes, a DB is create on /data1
  - ♦ run **/sbin/chkconfig mysql off**
  - ♦ make sure that mysql is shut down before starting this configuration

Service Configuration

- run the following commands as root (some of those can be run from an upriviledged user if needed)
- clean up the services configured by default

```
srvctl stop nodeapps -n <nodename>
crs_unregister ora.<nodename>.ons
crs_unregister ora.<nodename>.gsd
crs_unregister ora.<nodename>.vip
```

- configure the VIP
  - ♦ create the VIP for mysql1 + register it + set it to run as root + allow oracle to start it + startit

The command below does not work properly for CRS 10.2.0.3. Check the bug 413077.1 on Metalink for a workaround

```
crs_profile -create mysql1vip -t application -a $ORA_CRS_HOME/bin/usrvip -o oi=eth0,ov=<VIP_IP_AD
crs_register mysql1vip
crs_setperm mysql1vip -o root
crs_setperm mysql1vip -u user:oracle:r-x #note: you can use mysql user instead of oracle
crs_start mysql1vip
```

- write action script for filesystem /data1
  - ♦ see: action_PDB_data1.scr: action script for Oracle CRS - filesystem handler

- create profile and register action script for filesystem /data1

```
crs_profile -create fs_data1 -t application -d "Filesystem data1" -r mysql1vip -a $ORA_CRS_HOME/c
crs_register fs_data1
crs_setperm fs_data1 -o root
crs_setperm fs_data1 -u user:oracle:r-x
crs_start fs_data1
```

- ♦ copy the contents of $ORA_CRS_HOME/crs/public and $ORA_CRS_HOME/crs/profile to the other cluster node
- write action script for mysql instances
  - ♦ see: action_PDB_mysql1.scr: action script for Oracle CRS - mysql instance handler
- create profile and register action script for mysql

```
crs_profile -create mysql1 -t application -d "Mysql Instance 1" -r "mysql1vip fs_data1" -a /ORA/d
crs_register mysql1
crs_setperm mysql1 -o root
crs_setperm mysql1 -u user:oracle:r-x
crs_start mysql1
```

- ♦ copy the contents of $ORA_CRS_HOME/crs/public and $ORA_CRS_HOME/crs/profile to the other cluster node
- stop and start the db with
  - ♦ **crs_stop mysql1vip mysql1 fs_data1**
  - ♦ **crs_start mysql1vip mysql1 fs_data1**

NOTE: when shutting down services, stop commands can fail (for example umount can fail in unmounting the filesystem). The corresponding resources will then go in UNKNOWN state. Take special care in resolving these issues manually before restarting the services. With careless operations it is possible to mount /data1 on 2 nodes, something that will result in data corruption

---

This topic: PSSGroup > MysqlOraCRS
Topic revision: r4 - 2007-09-24 - JacekWojcieszuk