

High Availability Documentation on OAR - Admin Guide



Joris Brémond - joris.bremond@gmail.com

Laboratoire d'Informatique de Grenoble
Bat. ENSIMAG - antenne de Montbonnot
ZIRST 51, avenue Jean Kuntzmann
38330 MONTBONNOT SAINT MARTIN

Authors: LIG laboratory
Organization: LIG laboratory
Status: Testing

Copyright: licenced under the GNU GENERAL PUBLIC LICENSE

July 23, 2009

Contents

1	Introduction	1
1.1	Require	1
1.2	General functioning	1
2	Installation	2
2.1	DRBD	2
2.2	Heartbeat-2	2
3	Configuration	3
3.1	System configuration	3
3.2	Heartbeat Configuration	3
3.2.1	Authentication file : <i>/etc/ha.d/authkeys</i>	3
3.2.2	Daemon configuration file : <i>/etc/ha.d/ha.cf</i>	4
3.2.3	Ressource description file : <i>/var/lib/heartbeat/crm/cib.xml</i>	4
3.2.4	Make OAR Service LSB compatible	6
3.3	DRBD Configuration	6
3.3.1	synchronization type	6
3.3.2	Startup action	6
3.3.3	Disk action	7
3.3.4	Synchronization settings	7
3.3.5	Host section	7
3.3.6	Handlers	7
3.3.7	Launch DRBD	8
3.4	Database Configuration	8
3.4.1	Mysql	9
3.4.2	Postgresql	9
3.5	OAR Configuration	9
3.6	monitoring utils	9
4	known problems	10
4.1	Split brain	10
4.2	check disk	10
5	Test	11
A	Heartbeat example configuration	12
A.1	High Availability on OAR with 2 nodes configuration, postgresql database	12
A.2	High Availability on OAR with 4 nodes configuration	13

Abstract

This documentation explain how to configure High Availability on OAR. This solution provide fault tolerance on both OAR server and database (mysql or postgresql).

Chapter 1

Introduction

This documentation explain how to configure a high availability cluster with OAR. We will see to different configurations :

- OAR-server daemon and database on the same server. So we have two servers, the master OAR-server and the backup. (insert schema)
- OAR-server daemon and database on different server. We have four servers, the master OAR-server, the master database, the backup OAR-server, and the backup database. (insert schema)

1.1 Require

This documentation don't describe how to configure and install OAR. So before install OAR High Availability solution, install OAR with the database of your choice (mysql or postgresql). OAR-server and database must be also installed on the backup(s) server(s), with the same daemon configuration (oar.conf, my.conf or pg.conf). Nevertheless, it's not necessary that we configure OAR (oarnodesetting, etc.) because it's data incorporated in database and DRBD will share it. So we will have the same configurations in both nodes. Before continue, please stop oar-server and database daemon.

1.2 General functioning

For provide OAR High Availability, we use different produces :

- The first one is **Heartbeat** version 2. Heartbeat is a daemon that provides fault tolerance. It manage and monitor services, like Virtual IP, filesystem, etc. When Heartbeat detect problems in resource, it can migrate this one on other server. It can manage groups of resources and define rules and priority on each server.
- **DRBD**. DRBD refers to block devices designed as a building block to form high availability clusters. This is done by mirroring a whole block device via an assigned network. DRBD can be understood as network based raid-1. In OAR high availability solution, we will use DRBD for duplicate database. (insert schema)
- **VirtualIP** : This package is installed with heartbeat, and managed by it. It provide a virtual IP address. Thereby, client permanently know the elected server.

Chapter 2

Installation

2.1 DRBD

In this guide, we use DRBD-8.

- On debian :

The package name is drbd8-utils. It's easier to use module-assitant for install it:

```
apt-get install drbd8-utils module-assistant
module-assistant auto-install drbd8
```

Module Assitant require kernel header. You can install it with :

```
apt-get install linux-headers -`uname -r`
```

Now DRBD is operational.

2.2 Heartbeat-2

Heartbeat has two version. We will use the version 2 because we can monitor each resource and detect service crash.

- On debian:

The package name is heartbeat-2.

```
apt-get install heartbeat-2
```

You can also install the GUI package on the your computer for heartbeat remote configuration.

```
apt-get install heartbeat-2-gui
```

Chapter 3

Configuration

For illustrate example, the nodes names are :

- 2 nodes configurations :

- node1 (master)
- node2 (backup)

- 4 nodes configurations :

- node1 (master oar)
- node2 (master database)
- node3 (backup oar)
- node4 (backup database)

3.1 System configuration

It's really recommended that we have two different network between each Heartbeat server. Indeed, heartbeat use the network for check its neighbour. With two networks, we can for example avoid split-brain problem. We can use one Ethernet network and one serial network, ore two Ethernet network. With a solution like this, we can easily cut the Ethernet network without interfere with heartbeat functioning. For Heartbeat configuration, we need one or two network address (Virtual IP). If we are in configuration 2 nodes (OAR-server and database on the same nodes), we need one IP address, and two IP if we are in 4 nodes configurations. This virtual IP must be free and accessible on the network.

DRBD need a low-level storage for write data. It could be an entire disk, a partition, or a virtual partition (in a file). So before configure DRBD, create a free partition.

If you want create a partition in a file (/image.img), you can follow this commands

```
SIZE=200          #Size in Mega-bytes
dd if=/dev/zero of=/image.img bs=1M count=$SIZE
losetup /dev/loop/0 /image.img
mkfs -t ext3 /dev/loop/0
shred -zvf -n 1 /dev/loop/0
```

3.2 Heartbeat Configuration

There are three files to configure heartbeat-2 :

3.2.1 Authentication file : */etc/ha.d/authkeys*

The authkeys configuration file contains information for Heartbeat to use when authenticating cluster members. It cannot be readable or writable by anyone other than root. For sha authentication, write ::

```
auth 1
1 sha1 MyPassword
```

You can also use **md5** or **crc**.
Set file right:

```
chmod 0600 /etc/ha.d/authkeys
```

3.2.2 Daemon configuration file : */etc/ha.d/ha.cf*

The ha.cf file is one of the more important files. It lists the cluster nodes, the communications topology, and which features of the configuration are enabled.

- - An example with 2 nodes configuration and two Ethernet interfaces ::

```
logfile /var/log/ha-log
debugfile /var/log/ha-debug
#use_logd on
udpport 694
keepalive 1 # 1 second
deadtime 10
initdead 80
ucast eth1 IPNODE1
ucast eth1 IPNODE2
ucast eth0_rename IPNODE1
ucast eth0_rename IPNODE2
node node1
node node2
crm yes
```

- - For 4 nodes configurations ::

```
logfile /var/log/ha-log
debugfile /var/log/ha-debug
#use_logd on
udpport 694
keepalive 1 # 1 second
deadtime 10
initdead 80
ucast eth1 IPNODE1
ucast eth1 IPNODE2
ucast eth1 IPNODE3
ucast eth1 IPNODE4
ucast eth0_rename IPNODE1
ucast eth0_rename IPNODE2
ucast eth0_rename IPNODE3
ucast eth0_rename IPNODE4
node node1
node node2
node node3
node node4
crm yes
```

details:

keepalive : x second between each beat

deadtime : after x second without heartbeat, the nodes is declared dead

initdead : set the time that it takes to declare a cluster node dead when Heartbeat is first started

ucast: [interface] [ip node1] network for send/receive heartbeat. You can also use multicast(mcast) or broadcast(bcast) address

node : the node names

crm : yes for use heartbeat version 2

3.2.3 Ressource description file : */var/lib/heartbeat/crm/cib.xml*

This file specifie the services manage by heartbeat. We can find three sections :

- The first one is the cluster property. It describe general cluster behavoir. We can specify different options :
 - **symmetric-cluster** : Is resource can run on any node by default ?
 - **no-quorum-policy** : Action to do when cluster detect quorum
 - **default-resource-stickiness** : How much does ressource prefer to stay where it is

- **default-resource-failure-stickiness** : How many failure before migration
- **stonith-enabled** : If true, heartbeat kill the node which have resource failure. You must create a STONITH device before active it.
- **tonith-action** : Action to do : reboot or poweroff
- **startup-fencing** : STONITH unseen nodes if true (default)
- **stop-orphan-resources** : If true, heartbeat should deleted resource be stoped
- **stop-orphan-actions** : If true, should deleted action be cancelled
- **remove-after-stop** : Remove resources from heartbeat after they are stopped, default = false
- **short-resource-names** : true
- **transition-idle-timeout** : Provides the default global timeout for actions. Any action which has a defined timeout automatically uses the action-specific timeout
- **default-action-timeout** : How long heartbeat wait for actions to compete
- **is-managed-default** : Heartbeat can start or stop resources as required
- **cluster-delay** : Round trip delay over the network
- **pe-error-series-max** : The number of PE inputs resulting in ERRORS to save. -1 = unlimited
- **pe-warn-series-max** : The number of PE inputs resulting in WARNINGS to save
- **pe-input-series-max** : The number of other PE inputs to save

- The second part is the resources properties. We can defined group of resources, and configure them.

- **Group definition** : you can create group with the directive :

```
<group id="group1">
```

Now, you can add resources in this group.

- **Resource definition** : There is two different type of ressource : OCF or LSB.
OCF example :

```
<primitive class="ocf" id="resource_name" provider="heartbeat" type="resource_name">
```

LSB example :

```
<primitive class="lsb" id="oar-server" provider="heartbeat" type="oar-server">
```

OCF (Open Cluster Framework) Resource Agents is provided by heartbeat. You can find resource like IPaddr2 (for virtual IP), DRBDdisk for DRBD,etc. You can also add ressource with LSB standard (Linux Standards Base). Generaly, it is the main type of resource. The location of the resource we can add is on `/etc/ha.d/resource.d` and `/etc/init.d/`. When you defined the resource, the type is the name of the resource. Example, if you want to add mysql (`/etc/init.d/mysql`), you must write : `type="mysql"`. For "id", you can enter the name of your choice.

- **Resource operations** : Some operations can be added on resources. In our case, you can add resource monitoring. Example :

```
<operations>
  <op id="VirtualIP-mon" interval="5s" name="monitor" timeout="5s" />
</operations>
```

When the timeout is attained, heartbeat detect resource crash.

- **Resources attribute** : Sometimes, resources required additional parameters. You can enter them with the attribute directive :

```
<instance_attributes id="VirtualIP_inst_attr">
  <attributes>
    <nvpair id="VirtualIP_attr_0" name="ip" value="172.16.16.220" />
    <nvpair id="VirtualIP_attr_1" name="nic" value="eth1" />
    <nvpair id="VirtualIP_attr_2" name="cidr_netmask" value="24" />
  </attributes>
</instance_attributes>
```


- The last part is resources constraint. You can add constraint on resource or resource group, like location constraint. In our case, you can specify node priority, for example if you want resources run in priority on master.

```
<constraints>
  <rsc_location id="rsc_location_group_1" rsc="group_1">
    <rule id="preferred_location_group_1" score="100">
      <expression attribute="#uname" id="preferred_location_group_1_expr" operation="eq"
        value="NODE-NAME" />
    </rule>
  </rsc_location>
</constraints>
```

- Example of heartbeat configuration cib.xml file : see appendix A

Now, you must set rights on the cib.xml file, and give it to its owner :

```
chown hacluster:haclient /var/lib/heartbeat/crm/cib.xml
chmod 0600 /var/lib/heartbeat/crm/cib.xml
```

3.2.4 Make OAR Service LSB compatible

The OAR service (/etc/init.d/oar-server) is currently not compatible with LSB standard. It haven't the status function. So heartbeat can't monitor it. For add status function on oar-server, edit /etc/init.d/oar-server file, and add :

```
status)
    if [ -f $PIDFILE ]
    then
        PID=`head -n 1 $PIDFILE`
        if [ `ps -ef | grep -v grep | grep $PID | wc -l` -ge 1 ]
        then
            echo "OAR is running"
            exit 0
        else
            echo "OAR is not running"
            exit 2
        fi
    else
        echo "OAR is not running"
        exit 2
    fi
;;
```

3.3 DRBD Configuration

In DRBD configuration file, we can specify a lots of parameters, which describe resources, synchronization type, synchronization rate, etc.

3.3.1 synchronization type

With have three synchronization type in DRBD :

- **Protocol A** : write ACK (on master) is send when data was transmitted on master disk and sent to slave.
 - **Protocol B** : write ACK (on master) is send when data was transmitted on master disk and received by slave.
 - **Protocol C** : write ACK (on master) is send when data was transmitted on master disk and slave disk.
- Very recommended !**

3.3.2 Startup action

When DRBD is launched, we can do actions. For example, inform it to wait x seconds :

```
startup {
    wfc-timeout x;
}
```

3.3.3 Disk action

The directive “disk” can add action to do for disk. It’s recommended to add this one :

```
disk {
    on-io-error detach;
}
```

With the detach option, the node drops its backing device on I/O error, and continues in diskless mode.

3.3.4 Synchronization settings

You can precise the rate of synchronization, the hot-area size :

```
syncer {
    rate 700000K;
    al-extents 257;
}
```

700000K is the maximum value accepted by DRBD.

1 extend corresponds to 4 Mbytes, so with 257, we have 1 GBytes.

3.3.5 Host section

One resource must have two host section. A host section describe on each node which DRBD partition and disk is use. You must also specify neighbor ip and port :

```
on node1
    device DRBDPARTITION;
    disk DISKPARTITION;
    address node2:DRBDPORT;
    meta-disk internal;
}

on node2
    device DRBDPARTITION;
    disk DISKPARTITION;
    address node1:DRBDPORT;
    meta-disk internal;
}
```

3.3.6 Handlers

DRBD provide an handler which can execute actions on evenements. For the problem of split brain (see 4.2), we can for example write in log :

```
handlers {
    split-brain "echo Splitbraindetected >> /var/log/drbd-log"
    pri-lost-after-sb "echo pri-lost-after-sb >> /var/log/drbd-log"
}

net {
    #cram-hmac-alg \ "sha1\";
    #shared-secret \ "123456\";
    after-sb-0pri discard-younger-primary;
    after-sb-1pri consensus;
    after-sb-2pri call-pri-lost-after-sb;
    #rr-conflict violently;
}
```

In the net directive, you can specify different policies when you detect split brain.

after-sb-0pri : Split brain has just been detected, but at this time the resource is not in the Primary role on any host.

- **disconnect** : Do not recover automatically, call split-brain handler script, drop the connection and continue in disconnected mode.
- **discard-younger-primary**. Discard and roll back the modifications made on the host which assumed the Primary role last.
- **discard-least-changes**. Discard and roll back the modifications on the host where fewer changes occurred.

- **discard-zero-changes.** If there is any host on which no changes occurred at all, simply apply all modifications made on the other and continue.

after-sb-1pri : Split brain has just been detected, and at this time the resource is in the Primary role on one host.

- **disconnect**
- **consensus :** Apply the same recovery policies as specified in after-sb-0pri. If a split brain victim can be selected after applying these policies, automatically resolve. Otherwise, behave exactly as if disconnect were specified.
- **call-pri-lost-after-sb.** Apply the recovery policies as specified in after-sb-0pri. If a split brain victim can be selected after applying these policies, invoke the pri-lost-after-sb handler on the victim node.
- **discard-secondary.** Whichever host is currently in the Secondary role, make that host the split brain victim.

after-sb-2pri : Split brain has just been detected, and at this time the resource is in the Primary role on both hosts. This option accepts the same keywords as after-sb-1pri except discard-secondary and consensus.

3.3.7 Launch DRBD

Before launch DRBD between the two host, you can prepare the device. The first things to do, is to load DRBD module :

```
modprobe drbd
```

After that, you must verify if the drbd device exist. Look at `/dev/drbd0`. If this device don't exist, create it with :

```
mknod /dev/drbd0 b 147 0
```

Now, it is the time to create meta data and launch DRBD :

```
drbdadm create-md all
drbdadm up all
```

For mount DRBD device, you must create a folder, `/mnt/drbddata` for example :

```
mkdir /mnt/drbddata
```

DRBD is practically operational. You have to defined a master, `node1` in our example, so enter this command onmy on this node :

```
drbdadm -- --overwrite-data-of-peer primary all
mkfs -t ext3 /dev/drbd0
mount /dev/drbd0 /dev/drbddata
```

The “all” option mean that `node1` is the master of all resources defined in `drbd.conf`. The `mkfs` command prepare file system on drbd partition, and mount this partition.

Now DRBD is operational, all data put in `/dev/drbddata` will be synchronized.

3.4 Database Configuration

Now, it is the time to configure database. The `datadir` will be changed into the disk partition defined in DRBD (see 3.3.5).

3.4.1 Mysql

For change datadir in mysql, open the file */etc/mysql/my.cnf*, search the primitive **datadir**, and replace the value by the partition defined in DRBD. For example, we can enter :

```
datadir = /mnt/drbbdata
```

You must copy all old mysql data in new datadir, and change owner :

```
cp -r /var/lib/mysql /mnt/drbbdata  
chown -R mysql:mysql /mnt/drbbdata/mysql
```

If you want remove mysql access, don't forget to comment the **bind-address** in the my.cnf.

3.4.2 Postgresql

It is almost the same in Postgres. Open the file */etc/postgres/8.3/main/postgresql.conf*, and search `data_directory` and change value in :

```
data_directory = 'mnt/drbbdata/main'
```

Like mysql, if you want remote access, change **listen_addresses = 'localhost'** in **listen_addresses = ***

3.5 OAR Configuration

3.6 monitoring utils

Chapter 4

known problems

4.1 Split brain

4.2 check disk

`tune2fs -c 0 DISKPARTITION`

Chapter 5

Test

Appendix A

Heartbeat example configuration

A.1 High Availability on OAR with 2 nodes configuration, postgresql database

```
<cib admin_epoch="0" have_quorum="true" ignore_dtd="false" num_peers="1" cib_feature_revision="
2.0" generated="false" epoch="4" num_updates="4" cib_last_written="Mon Jul 20 15:22:44
2009" ccm_transition="1">
<configuration>

  <crm-config>
    <cluster_property_set id="cib-bootstrap-options">
      <attributes>
        <nvpair id="cib-bootstrap-options-symmetric-cluster" name="symmetric-cluster" value=
"true"/>
        <nvpair id="cib-bootstrap-options-no-quorum-policy" name="no-quorum-policy" value="
stop"/>
        <nvpair id="cib-bootstrap-options-default-resource-stickiness" name="default-
resource-stickiness" value="0"/>
        <nvpair id="cib-bootstrap-options-default-resource-failure-stickiness" name="default
-resource-failure-stickiness" value="0"/>
        <nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="
false"/>
        <nvpair id="cib-bootstrap-options-stonith-action" name="stonith-action" value="
reboot"/>
        <nvpair id="cib-bootstrap-options-startup-fencing" name="startup-fencing" value="
true"/>
        <nvpair id="cib-bootstrap-options-stop-orphan-resources" name="stop-orphan-resources"
value="true"/>
        <nvpair id="cib-bootstrap-options-stop-orphan-actions" name="stop-orphan-actions"
value="true"/>
        <nvpair id="cib-bootstrap-options-remove-after-stop" name="remove-after-stop" value=
"false"/>
        <nvpair id="cib-bootstrap-options-short-resource-names" name="short-resource-names"
value="true"/>
        <nvpair id="cib-bootstrap-options-transition-idle-timeout" name="transition-idle-
timeout" value="5min"/>
        <nvpair id="cib-bootstrap-options-default-action-timeout" name="default-action-
timeout" value="20s"/>
        <nvpair id="cib-bootstrap-options-is-managed-default" name="is-managed-default"
value="true"/>
        <nvpair id="cib-bootstrap-options-cluster-delay" name="cluster-delay" value="60s"/>
        <nvpair id="cib-bootstrap-options-pe-error-series-max" name="pe-error-series-max"
value="-1"/>
        <nvpair id="cib-bootstrap-options-pe-warn-series-max" name="pe-warn-series-max"
value="-1"/>
        <nvpair id="cib-bootstrap-options-pe-input-series-max" name="pe-input-series-max"
value="-1"/>
        <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="2.1.3 - node:
552305612591183b1628baa5bc6e903e0fle26a3"/>
      </attributes>
    </cluster_property_set>
  </crm-config>

  <resources>
    <group id="group_1">
      <primitive class="ocf" id="VirtualIP" provider="heartbeat" type="IPAddr2">
        <operations>
          <op id="VirtualIP_mon" interval="5s" name="monitor" timeout="5s"/>
        </operations>
      </primitive>
    </group>
  </resources>
</cib>
```

```

    </operations>
    <instance_attributes id="VirtualIP_inst_attr">
      <attributes>
        <nvpair id="VirtualIP_attr_0" name="ip" value="172.16.16.220"/>
        <nvpair id="VirtualIP_attr_1" name="nic" value="eth1"/>
        <nvpair id="VirtualIP_attr_2" name="cidr_netmask" value="24"/>
      </attributes>
    </instance_attributes>
  </primitive>
  <primitive class="heartbeat" id="DRBD-disk" provider="heartbeat" type="drbddisk">
    <operations>
      <op id="DRBD-disk_mon" interval="120s" name="monitor" timeout="60s"/>
    </operations>
    <instance_attributes id="DRBD-diskinst_attr">
      <attributes>
        <nvpair id="DRBD-disk_attr_1" name="1" value="mysql"/>
      </attributes>
    </instance_attributes>
  </primitive>
  <primitive class="ocf" id="Filesystem" provider="heartbeat" type="Filesystem">
    <operations>
      <op id="Filesystem_mon" interval="120s" name="monitor" timeout="60s"/>
    </operations>
    <instance_attributes id="Filesystem_inst_attr">
      <attributes>
        <nvpair id="Filesystem_attr_0" name="device" value="/dev/drbd0"/>
        <nvpair id="Filesystem_attr_1" name="directory" value="/mnt/drbddata"/>
        <nvpair id="Filesystem_attr_2" name="fstype" value="ext3"/>
      </attributes>
    </instance_attributes>
  </primitive>
  <primitive id="Postgres" class="lsb" type="postgresql-8.3" provider="heartbeat">
    <operations>
      <op id="Postgres_mon" interval="120s" name="monitor" timeout="60s"/>
    </operations>
    <meta_attributes id="Postgres_meta_attrs">
      <attributes>
        <nvpair id="Postgres_metaattr_target_role" name="target_role" value="started"/>
      </attributes>
    </meta_attributes>
  </primitive>
  <primitive class="lsb" id="oar-server" provider="heartbeat" type="oar-server">
    <operations>
      <op id="oar-server_mon" interval="120s" name="monitor" timeout="60s"/>
    </operations>
  </primitive>
  <meta_attributes id="group_1_meta_attrs">
    <attributes>
      <nvpair id="group_1_metaattr_ordered" name="ordered" value="true"/>
    </attributes>
  </meta_attributes>
</group>
</resources>

<constraints>
  <rsc_location id="rsc_location_group_1" rsc="group_1">
    <rule id="prefered_location_group_1" score="100">
      <expression attribute="#uname" id="prefered_location_group_1_expr" operation="eq"
        value="node1"/>
    </rule>
  </rsc_location>
</constraints>

</configuration>
</cib>

```

A.2 High Availability on OAR with 4 nodes configuration

```

<cib generated="false" admin_epoch="0" have_quorum="false" ignore_dtd="false" num_peers="1"
  cib_feature_revision="2.0" epoch="72" num_updates="1" cib_last_written="Wed_Jul_22_13
:16:03_2009" ccm_transition="1">
  <configuration>

    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <attributes>
          <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="2.1.3-node:
552305612591183b1628baa5bc6e903e0f1e26a3"/>

```



```

<nvpair id="cib-bootstrap-options-symmetric-cluster" name="symmetric-cluster" value="
false" />
<nvpair id="cib-bootstrap-options-no-quorum-policy" name="no-quorum-policy" value="
ignore" />
<nvpair id="cib-bootstrap-options-default-resource-stickiness" name="default-
resource-stickiness" value="0" />
<nvpair id="cib-bootstrap-options-default-resource-failure-stickiness" name="default
-resource-failure-stickiness" value="0" />
<nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="
false" />
<nvpair id="cib-bootstrap-options-stonith-action" name="stonith-action" value="
reboot" />
<nvpair id="cib-bootstrap-options-startup-fencing" name="startup-fencing" value="
true" />
<nvpair id="cib-bootstrap-options-stop-orphan-resources" name="stop-orphan-resources"
value="true" />
<nvpair id="cib-bootstrap-options-stop-orphan-actions" name="stop-orphan-actions"
value="true" />
<nvpair id="cib-bootstrap-options-remove-after-stop" name="remove-after-stop" value="
false" />
<nvpair id="cib-bootstrap-options-short-resource-names" name="short-resource-names"
value="true" />
<nvpair id="cib-bootstrap-options-transition-idle-timeout" name="transition-idle-
timeout" value="5min" />
<nvpair id="cib-bootstrap-options-default-action-timeout" name="default-action-
timeout" value="20s" />
<nvpair id="cib-bootstrap-options-is-managed-default" name="is-managed-default"
value="true" />
<nvpair id="cib-bootstrap-options-cluster-delay" name="cluster-delay" value="60s" />
<nvpair id="cib-bootstrap-options-pe-error-series-max" name="pe-error-series-max"
value="-1" />
<nvpair id="cib-bootstrap-options-pe-warn-series-max" name="pe-warn-series-max"
value="-1" />
<nvpair id="cib-bootstrap-options-pe-input-series-max" name="pe-input-series-max"
value="-1" />
</attributes>
</cluster-property-set>
</crm-config>

<resources>
<group id="Database-servers">
<meta-attributes id="Database-servers.meta.attrs">
<attributes>
<nvpair name="target_role" id="Database-servers.metaattr.target_role" value="
started" />
<nvpair id="Database-servers.metaattr.ordered" name="ordered" value="true" />
<nvpair id="Database-servers.metaattr.collocated" name="collocated" value="true" />
</attributes>
</meta-attributes>
<primitive class="ocf" type="IPaddr2" provider="heartbeat" id="VirtualIP-database">
<operations>
<op id="IPaddr2.mon" interval="5s" name="monitor" timeout="5s" />
</operations>
<instance-attributes id="VirtualIP-database.instance.attrs">
<attributes>
<nvpair id="89b87cae-b271-4d55-8881-c89508ad8451" name="ip" value="172.16.16.221"
/>
<nvpair id="bff53b6b-a11d-4cb0-8db3-24ef1b82fbb0" name="nic" value="eth1" />
<nvpair id="d6dc4cd1-b057-48be-91c7-3b733e066269" name="cidr_netmask" value="24"
/>
</attributes>
</instance-attributes>
<meta-attributes id="VirtualIP-database.meta.attrs">
<attributes>
<nvpair name="target_role" id="VirtualIP-database.metaattr.target_role" value="
started" />
</attributes>
</meta-attributes>
</primitive>
<primitive id="DRBD-disk" class="heartbeat" type="drbdisk" provider="heartbeat">
<operations>
<op id="DRBD-disk.mon" interval="120s" name="monitor" timeout="60s" />
</operations>
<instance-attributes id="DRBD-disk.inst.attr">
<attributes>
<nvpair id="DRBD-disk.attr.1" name="1" value="mysql" />
</attributes>
</instance-attributes>
<meta-attributes id="DRBD-disk.meta.attrs">
<attributes>
<nvpair id="DRBD-disk.metaattr.target_role" name="target_role" value="started" />
</attributes>
</meta-attributes>
</primitive>

```

```

<primitive id="Filesystem" class="ocf" type="Filesystem" provider="heartbeat">
  <operations>
    <op id="Filesystem_mon" interval="120s" name="monitor" timeout="60s"/>
  </operations>
  <instance-attributes id="Filesystem-instance-attrs">
    <attributes>
      <nvpair id="29b7a79c-ad8f-47ef-91dd-610292acac10" name="device" value="/dev/
drbd0"/>
      <nvpair id="639a3cc9-8df6-497a-b874-f04b72da5577" name="directory" value="/mnt/
drbdata"/>
      <nvpair id="1e59d136-4bc7-4f6c-9eb4-a62e06ec48d1" name="fstype" value="ext3"/>
    </attributes>
  </instance-attributes>
  <meta-attributes id="Filesystem-meta-attrs">
    <attributes>
      <nvpair id="Filesystem-metaattr-target-role" name="target-role" value="started"/>
    </attributes>
  </meta-attributes>
</primitive>
<primitive id="MysqlHA" class="lsb" type="mysqlha" provider="heartbeat">
  <operations>
    <op id="MysqlHA_mon" interval="120s" name="monitor" timeout="60s"/>
  </operations>
  <meta-attributes id="MysqlHA-meta-attrs">
    <attributes>
      <nvpair id="MysqlHA-metaattr-target-role" name="target-role" value="started"/>
    </attributes>
  </meta-attributes>
</primitive>
</group>
<group id="OAR-servers">
  <meta-attributes id="OAR-servers-meta-attrs">
    <attributes>
      <nvpair id="OAR-servers-metaattr-target-role" name="target-role" value="stopped"/>
      <nvpair id="OAR-servers-metaattr-ordered" name="ordered" value="true"/>
      <nvpair id="OAR-servers-metaattr-collocated" name="collocated" value="true"/>
    </attributes>
  </meta-attributes>
  <primitive id="VirtualIP-OAR-server" class="ocf" type="IPaddr2" provider="heartbeat">
    <operations>
      <op id="VirtualIP-OAR-server_mon" interval="5s" name="monitor" timeout="5s"/>
    </operations>
    <instance-attributes id="VirtualIP-OAR-server-instance-attrs">
      <attributes>
        <nvpair id="bb615f47-387e-46cb-945f-8757d494791b" name="ip" value="172.16.16.220
"/>
        <nvpair id="6fel6a0e-c414-4a5b-8e5f-f0c48b369770" name="nic" value="eth1"/>
        <nvpair id="b25acc28-5186-4a22-9433-5347017cdc71" name="cidr_netmask" value="24"
"/>
      </attributes>
    </instance-attributes>
    <meta-attributes id="VirtualIP-OAR-server-meta-attrs">
      <attributes>
        <nvpair id="VirtualIP-OAR-server-metaattr-target-role" name="target-role" value=
"started"/>
      </attributes>
    </meta-attributes>
  </primitive>
  <primitive id="OAR-server" class="lsb" type="oar-server" provider="heartbeat">
    <operations>
      <op id="OAR-server_mon" interval="120s" name="monitor" timeout="60s"/>
    </operations>
    <meta-attributes id="OAR-server-meta-attrs">
      <attributes>
        <nvpair id="OAR-server-metaattr-target-role" name="target-role" value="started"/>
      </attributes>
    </meta-attributes>
  </primitive>
</group>
</resources>

<constraints>
  <rsc.location id="location-Database" rsc="Database-servers">
    <rule id="prefered_location-Database" score="0">
      <expression attribute="#uname" id="f968a0dd-59dd-4d02-b54a-36c7139afa13" operation="
ne" value="node1"/>
      <expression attribute="#uname" id="0ef6e38b-6a34-4e23-bc22-9dccfacc95b6" operation="
ne" value="node3"/>
    </rule>
  </rsc.location>
  <rsc.location id="location-OAR" rsc="OAR-servers">
    <rule id="prefered_location-OAR" score="0">

```

```

        <expression attribute="#uname" id="4d2c21d8-aea7-460b-bfeb-cd6d22e4a17e" operation="
            ne" value="node2"/>
        <expression attribute="#uname" id="3292b1b4-04cb-4650-b6e0-2e4428c837a8" operation="
            ne" value="node4"/>
    </rule>
</rsc_location>
<rsc_location id="location_Master-OAR" rsc="OAR-servers">
    <rule id="prefered_location_Master-OAR" score="INFINITY">
        <expression attribute="#uname" id="37a60f84-c7ed-4e4a-835a-55382961c990" operation="
            eq" value="node1"/>
    </rule>
</rsc_location>
<rsc_location id="location_Master-Database" rsc="Database-servers">
    <rule id="prefered_location_Master-Database" score="INFINITY">
        <expression attribute="#uname" id="37a78f84-c7ed-4e4a-835a-54382451c990" operation="
            eq" value="node2"/>
    </rule>
</rsc_location>
</constraints>

</configuration>
</cib>

```