High-Availability Cluster

ICD - Infraestruturas e Centros de Dados

2020/2021

The main goal of this guide is to understand how to use and configure an High-Availability Cluster in Linux systems.

For the exercises described next, the following tools must be installed,

- VirtualBox - https://www.virtualbox.org

while useful documentation is available at:

- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_reference/index

Steps

VM Deployment and Configuration

1. Reuse the DRBD server VMs exposing an ISCSI target service (Guide 3).

2. At the iSCSI client VM (IP - 10.0.0.5/24) change the hostname (nmtui) to *cl1*.

3. Install the following packages:

   ```
   dnf config-manager --set-enabled HighAvailability
   yum install pcs pacemaker fence-agents-all
   yum install postgresql-server.x86_64
   ```

4. Update (keep the lines already at the file) the */etc/hosts* file to include:

   ```
   10.0.0.5 cli1
   10.0.0.6 cli2
   ```

5. Ensure that the firewall and Selinux services are disabled:

   systemctl stop firewalld and *systemctl disable firewalld*

   Edit file /etc/selinux/config and change to SELINUX=disabled

6. Clone this machine and create a new one with hostname *cli2* and IP *10.0.0.6/24*.

7. Launch both VMs and test that iSCSI and multipath are working correctly.

Cluster Setup

1. At both VMs:

   define the cluster's user password with the command *passwd hacluster* (e.g., password: pass123456).

   start and enable the pcs daemon with *systemctl start pcsd.service* and *systemctl enable pcsd.service*.

   enable the corosync and pacemaker deamons with *systemctl enable corosync* and *systemctl enable pacemaker*.

   disable stonith with the command *pcs property set stonith-enabled=false*.

2. At *cli1* VM:

   authenticate the cluster with the command *pcs host auth cli1 cli2* (username: hacluster and password: pass123456).

   start the cluster with the command *pcs cluster setup my_cluster -- start cli1 cli2*.

Testing

Database Initialisation

1. At both VMs create the folder /mnt/db (*mkdir /mnt/db*).

2. At *cli1* VM initialise and configure a postgreSQL database whose content should be stored at the multipath device:

   Note 1: Do not use sudo while running the commands with the postgres user.

   Note 2: The two postgres configurations are just for testing purposes and are not secure.

```
mount /dev/mapper/mpatha /mnt/db/ (the device mapper name might be different)
mkdir /mnt/db/PG
chown postgres /mnt/db/PG
sudo -i -u postgres
initdb --locale $LANG -E UTF8 -D /mnt/db/PG/
/mnt/db/PG/postgresql.conf -> listen_addresses='*'  (also uncomment the line)
/mnt/db/PG/pg_hba.conf -> host  all  all 0.0.0.0/0 trust
exit
umount /mnt/db
```

Cluster Console and Resources

1. Access the cluster web interface from your browser (https://10.0.0.5:2224).

2. Login and add the cluster (Manage Clusters - Add Existing).

3. Go to the cluster ("my_cluster") resource configurations, and add the following resources:

    (a) a Virtual IP address resource (ofc:heartbeat IPaddr2) with the following configurations:

    > IP - 10.0.0.10
    > nic - enp0s8
    > cidr_netmask - 24

    (b) a Postgres resource (ofc:heartbeat psgsql) with the following configurations:

    > pgdata - /mnt/db/PG
    > config - /mnt/db/PG/postgresql.conf

    (c) a Filesystem resource (ofc:heartbeat Fylesystem) with the following configurations:

    > device - /dev/mapper/mpatha (the device mapper name might be different)
    > directory - /mnt/db/
    > fstype - xfs

    (d) The postgres resource can only start after the filesystem is enabled (Resource Ordering Preference).

4. Group all these resources into a single group and, at the group configurations, add different priorities for the resource location (Resource Location Preferences).

5. Enable the group resource and check that all resources are running at the cluster node with the higher priority (may need to use the refresh button of the web interface).

DB access

1. Access the database with the psql client (either installed at the host or on another VM)

    > psql -U postgres -h 10.0.0.10

    you can also check at the respective *cli* VM that the services are indeed running (filesystem, postgres and virtual ip). Also, note that in the other VM the services are not deployed.

2. At the group configuration (Resource Location Preferences) swap the priorities between client nodes. Check that the resources were migrated to the other cluster node (may need to use the refresh button of the web interface).

3. Access the database with the psql client. Again, you can also check at the respective *cli* VM that the services are indeed running (filesystem, postgres and virtual ip).

4. Power off the cluster server where the resources are allocated. Check that the resources were migrated to the other cluster node.

5. Access the database with the psql client.

Learning outcomes   Experiment High-availability clusters deployment and configuration. Assess how to provide resilient services on a cluster deployment.