Compute HA and Live Migration Guide

Step-by-step guide to achieve compute node HA using Pacemaker on Canonical (Ubuntu 14.04)

Table of Contents

Assumptions

This guide makes the following assumptions:

1. OpenStack has been setup with HA (3 node controller HA cluster) using pacemaker & corosync deployed through Juju (Maas)
2. Working OpenStack cloud
3. Ufw in disabled or state & ufw reset has been performed
4. Iptables stopped and flushed

Software Versions

The version of software packages used in this guide are as follows:

1. Linux:
    a. Ubuntu 14.04.4 LTS
2. OpenStack:
    a. Kilo Release (2015.1.3)
3. Clustering:

```
pacemaker: 1.1.12-rc4+git-9290c5545a-dg1~trusty
pacemaker-cli-utils: 1.1.12-rc4+git-9290c5545a-dg1~trusty
corosync: 2.3.3-1ubuntu3
libcorosync-common4: 2.3.3-1ubuntu3
resource agents: 1:3.9.3+git20121009-3ubuntu2
pacemaker-remote: 1.1.12-rc4+git-9290c5545a-dg1~trusty
fence-agents: 3.1.5-2ubuntu4
```

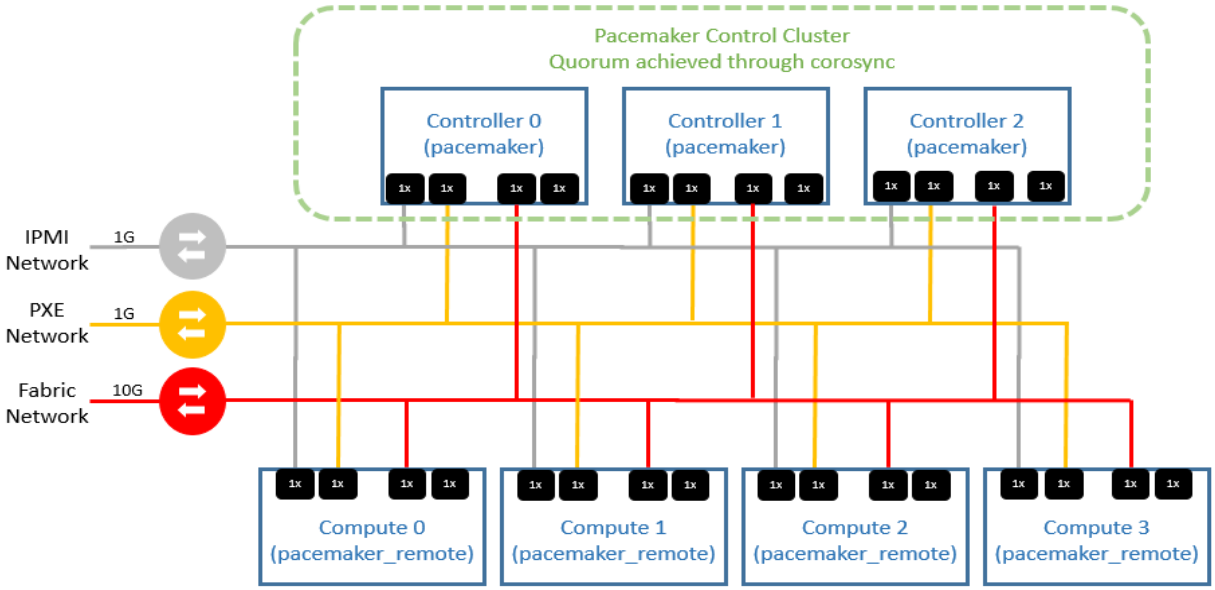
Requirements for VM HA:

Here are the basic requirements for VM HA:

1. Pacemaker 1.1.12rc4 & pacemaker remote agents configured on compute nodes
2. Shared Storage for evacuations and live migration
3. IPMI tool
4. CRMSH
5. Nova Credentials

Cluster Setup for Compute HA

For this section we will be using the below network diagram:



Now that we have a base system ready we now go ahead and prepare the cluster for Compute Node HA. Follow these steps in exact this order.

Setting up shared storage

Here are the steps to follow for setting up shared storage:

1. To run the commands from root user:
   ```
   sudo su
   ```

2. Go to Controller1 and install NFS server:
   ```
   apt-get install nfs-server -y
   ```

3. Install NFS common libraries on Controller2 & Controller3 & Compute nodes:
   ```
   apt-get install nfs-common -y
   ```

4. Create a directory /storage on all controllers
   ```
   mkdir /storage
   ```

5. Add the following to /etc/exports on Controller1:
   ```
   /storage <CIDR of Fabric
   Network>(rw,fsid=0,insecure,no_subtree_check,async,no_root_squash)

   E.g:
   /storage
   172.30.3.0/24(rw,fsid=0,insecure,no_subtree_check,async,no_root_squash)
   ```

6. Start NFS service on controller1 & export the directory:
   ```
   service nfs-kernel-server start
   service rpcbind start
   exportfs -avr
   ```

7. Add to /etc/fstab on Controller2 & Controller3:
   Reference command:
   ```
   <IP of Controller1 from Fabric Network>:/storage /storage nfs auto 0 0
   ```

   For Example:
   ```
   172.30.3.206:/storage /storage nfs auto 0 0
   ```

8. Edit /etc/fstab on compute nodes: (IP on controller1)
   ```
   172.30.3.206:/storage /var/lib/nova/instances nfs auto 0 0
   ```

9. Mount the directory on Controller2 & Controller3 & compute nodes:
   ```
   mount -a -v
   ```

10. Edit the nova lxc config files on all three controllers (Change the juju machine & lxc ID as per the setup):
    Reference Command:
    ```
    nano /var/lib/lxc/juju-machine-<Juju-Controller-ID>-lxc-3/config
    ```

For Example: Controller1 with juju machine id=3

```
nano /var/lib/lxc/juju-machine-3-lxc-3/config
```

And add the below line just above the “lxc.tty = 4” Line:

```
lxc.mount.entry = /storage var/lib/nova/instances none bind,optional 0
0
```

11. Restart the nova lxc in order for the changes to take effect:
    Check the lxc names on controller nodes:

    ```
    lxc-ls –f
    ```

    Restart nova LXC on the controller nodes one by one with the following command:

    ```
    lxc-stop -n juju-machine-<juju-controller-id>-lxc-3
    ```

    E.g. Controller1 with machine id=3

    ```
    lxc-stop -n juju-machine-3-lxc-3
    lxc-start -n juju-machine-3-lxc-3 -d
    ```

    E.g. Controller2 with machine id=6

    ```
    lxc-stop -n juju-machine-6-lxc-3
    lxc-start -n juju-machine-6-lxc-3 -d
    ```

    E.g. Controller2 with machine id=9

    ```
    lxc-stop -n juju-machine-9-lxc-3
    lxc-start -n juju-machine-9-lxc-3 -d
    ```

12. Change the permissions of the /var/lib/nova directory on all nova LXCs & compute nodes:
    Note1: Use juju ssh from MAAS to login to LXCs
    Note2: Use “juju stat --format=tabular | grep nova-cloud-controller/ | awk '{print $5}'” to enlist all nova LXCs

    Command for changing permissions:

    ```
    chown -R nova:nova /var/lib/nova/
    ```

13. Restart libvirt & nova-compute services on the compute nodes

    ```
    service libvirt-bin restart
    service nova-compute restart
    ```

14. Test changes up till now. Create VMs on each compute node and verify that the VMs are being created: (change image ID, Net-id and availability zone according to your setup)

```
nova boot vm1 --flavor <flavor-id> --image <image-id> --nic net-
id=<net-id> --availability-zone nova:<name of compute node>
```

For example:

```
nova boot vm1 --flavor 1 --image f0240527-a9a4-4c63-817b-0b3e114bfe43 -
-nic net-id=1da8b4cd-56aa-4193-acd5-6d75b936523d --availability-zone
nova:compute2-b1
```

15. On all Nova LXCs & Compute nodes:

```
ufw reset
ufw disable
iptables -F
```

Installing and setting up Pacemaker

Follow the below steps to setup pacemaker on Nova LXCs:

1. To run the commands from root user:
   ```
   sudo su
   ```

2. Add repo for pacemaker 1.1.12rc4 version on Nova LXCs:
   ```
   add-apt-repository  ppa:david-gabriel/ppa
   apt-get update
   ```

3. Install pacemaker 1.1.12rc4 on Nova LXCs:
   ```
   apt-get install pacemaker pacemaker-dbg pacemaker-cli-utils booth-
   pacemaker pacemaker-dev pacemaker-mgmt pacemaker-mgmt-client pacemaker-
   mgmt-dev pacemaker-remote -y
   ```

4. Install fence-agents on Nova LXCs:
   ```
   apt-get install fence-agents -y
   ```

5. Install pcs on Nova LXCs:
   ```
   apt-get install git -y
   git clone https://github.com/feist/pcs.git
   cd pcs
   make
   make install
   ```

6. Configure pcs by editing file `/usr/lib/python2.7/dist-packages/pcs/settings.py` in the following way (on all Nova LXCs).

   Replace this:
   ```
   from pcs.settings_default import *
   pengine_binary = "/usr/lib/x86_64-linux-gnu/pacemaker/pengine"
   crmd_binary = "/usr/lib/x86_64-linux-gnu/pacemaker/crmd"
   cib_binary = "/usr/lib/x86_64-linux-gnu/pacemaker/cib"
   stonithd_binary = "/usr/lib/x86_64-linux-gnu/pacemaker/stonithd"
   pcsd_exec_location = "/usr/share/pcsd/"
   ```

   With:
   ```
   from pcs.settings_default import *
   pengine_binary = "/usr/lib/pacemaker/pengine"
   crmd_binary = "/usr/lib/pacemaker/crmd"
   cib_binary = "/usr/lib/pacemaker/cib"
   stonithd_binary = "/usr/lib/pacemaker/stonithd”
   pcsd_exec_location = "/usr/share/pcsd/"
   ```

7. Reboot all Nova LXCs from their respective consoles (Do this one Nova LXC at a time):
   ```
   reboot
   ```

8. Add the following in the default section in the nova configuration files on all nova LXCs:
   ```
   nano /etc/nova/nova.conf
   ```
   [default]
   …..
   service_down_time = 35
   report_interval = 5

9. Restart all nova services on all Nova LXCs:
   ```
   service nova-conductor status
   service nova-conductor restart
   service nova-consoleauth restart
   service nova-cert restart
   service nova-scheduler restart
   ```

10. Add the following in the default section in the nova configuration files on all Neutron LXCs (neutron LXC id: 5):
    ```
    nano /etc/neutron/neutron.conf
    ```
    [default]
    …..
    agent_down_time = 40

11. Restart neutron-server service on all Neutron LXCs
    ```
    service neutron-server restart
    ```

Install & Setting up Pacemaker Remote

Follow the below steps to install and configure pacemaker remote on all compute nodes:

1. To run the commands from root user:
   ```
   sudo su
   ```

2. Add repo for pacemaker 1.1.12rc4 version on Nova LXCs and compute nodes:
   ```
   add-apt-repository  ppa:david-gabriel/ppa
   apt-get update
   ```

3. Install pacemaker-remote, fence-agents and ipmitool on compute nodes:
   ```
   apt-get install pacemaker-remote -y
   apt-get install fence-agents -y
   apt-get install ipmitool -y
   ```

4. Start the pacemaker-remote daemon on Compute Nodes:
   ```
   service pacemaker_remote start
   update-rc.d pacemaker_remote defaults
   ```

Configuring pacemaker cluster

Follow the below steps to configure the pacemaker cluster; these can be run in any of the Nova LXC (Remember this LXC for later use):

1. To run the commands from root user:
   ```
   sudo su
   ```

2. Make the cluster non-symmetric:
   ```
   pcs property set symmetric-cluster=false
   ```

3. Clean up all Resources:
   ```
   pcs resource cleanup
   ```

4. Verify that all resource nodes and remote nodes are online and all resources have started:
   ```
   pcs status
   crm status
   crm_mon
   pcs constraint show --full
   ```

5. Create directory /etc/vm-ha & /opt/vm-ha on any one of the nova-lxc:
   ```
   mkdir /etc/vm-ha
   mkdir /opt/vm-ha
   ```

6. Download the vm-ha folder to the MAAS node and copy the folder to the above nova-lxc.
   Download the vm-ha folder from the following link and copy it to the MaaS node:
   https://drive.google.com/drive/u/0/folders/0B7Qz_nyVryPySGZQQkJjd0pLN3c
   From the MAAS node execute the below commands:
   ```
   juju scp -- -r vm-ha 7/lxc/3:/home/ubuntu
   ```

   Note: In the above command 7 refers to the machine-id of the controller

7. Place the scripts in the following directories (from the folder where files are placed):
   In the Nova LXC (7/lxc/3 in this case), execute the following commands
   ```
   cd /home/ubuntu/vm-ha
   mv daemonize.sh evacuate_setup.sh evacuate.sh /opt/vm-ha/
   mv vm-ha.conf /etc/vm-ha/
   ```

8. Make the scripts executable:
   ```
   chmod x /opt/vm-ha/daemonize.sh
   chmod x /opt/vm-ha/evacuate_setup.sh
   chmod x /opt/vm-ha/evacuate.sh
   ```

9. Edit the vm-ha configuration file /etc/vm-ha/vm-ha.conf and change the following fields
   Note: The information can be viewed from nova.rc file in the MAAS node:

```
OS_USERNAME= <username>
OS_PASSWORD= <password>
OS_TENANT_NAME= <tenant name>
OS_AUTH_URL= <keystone auth-url>
OS_REGION_NAME= <region name>
EVACUATION_TARGET= <hostname of reserved host>
COMPUTE_NODES= <hostnames of compute nodes>
```

10. Add the IPMI information to the end of the /etc/vm-ha/vm-ha.conf file in the following format, one per line: Test this with creating a new IPMI user:

```
<compute-hostname> <ipmi-ip> <ipmi-username> <ipmi-password>
<compute-hostname> <ipmi-ip> <ipmi-username> <ipmi-password>
```

Adding remote nodes in pacemaker cluster

Follow the below steps to add pacemaker remote nodes in the pacemaker cluster:

1. To run the commands from root user:
   ```
   sudo su
   ```

2. On Nova LXCs and compute nodes:
   ```
   mkdir /etc/pacemaker
   ```

3. On compute1 generate the auth key for pacemaker & pacemaker-remote authentication:
   ```
   dd if=/dev/urandom of=/etc/pacemaker/authkey bs=4096 count=1
   ```

4. scp authkey to MAAS node:
   ```
   scp /etc/pacemaker/authkey <plumgrid@172.30.1.6>:/home/plumgrid
   ```

5. From MAAS node scp the same authkey to all nova LXCs and remaining compute nodes.:
   ```
   juju scp authkey <3>/lxc/3:/home/ubuntu
   juju scp authkey <6>/lxc/3:/home/ubuntu
   juju scp authkey <9>/lxc/3:/home/ubuntu
   juju scp authkey <5>:/home/ubuntu
   ```

6. From LXCs & remaining compute nodes copy the authkey to /etc/pacemaker directory:
   ```
   cp authkey /etc/pacemaker
   ```

7. On the Nova LXC selected for vm-ha service hosting run the script evacuate_setup.sh in order to add pacemaker remote nodes and stonith resrouce for compute nodes:
   ```
   cd /opt/vm-ha/
   ./evacuate_setup.sh <machine-id-of-controller1> <machine-id-of-
   controller2> <machine-id-of-controller3>
   ```

   For example the machine IDs are 5, 7 & 8:
   ```
   ./evacuate_setup.sh 5 7 8
   ```

Starting the VM-HA service

In order to start the vm-ha service, follow the below steps (on the Nova LXC selected earlier e.g. 7/lxc/3):

1. To run the commands from root user:
   ```
   sudo su
   ```

2. Install daemon service:

```
apt-get install daemon
```

3. Initialize the vm-ha service
```
cd /opt/vm-ha/
./daemonize.sh vm-ha evacuate.sh
```

4. Start the vm-ha service
```
service vm-ha start
```

5. Verify service vm-ha is running:
```
service vm-ha status
```

Testing Manual Evacuation

In order to test manual evacuation at this stage follow the below steps:

1. To run the commands from root user:
```
sudo su
```

2. Spawn VM on compute2:
```
nova boot vm1 --flavor <flavor-id> --image <image-id> --nic net-
id=<net-id> --availability-zone nova:compute2
```

3. Check the location of VMs:
   ```
   nova list --fields name,status,task_state,host
   ```

4. On compute2 stop the nova-compute service:
   ```
   service nova-compute stop
   ```

5. On Nova LXC wait for the nova-compute service on compute2 to go down:
   ```
   nova service-list
   ```

6. On nova LXC execute the following command:
   Reference Command:
   ```
   nova host-evacuate --target <target-host> <source-host> --on-shared-storage
   ```

   For example:
   ```
   nova host-evacuate --target compute1-t4 compute2-b1 --on-shared-storage
   ```

7. Check the location of VMs after evacuations:
   ```
   nova list --fields name,status,task_state,host
   ```

Testing Automatic Evacuation

In order to test manual evacuation at this stage follow the below steps on the selected Nova LXC:

1. To run the commands from root user:
   ```
   sudo su
   ```

2. Spawn VM on compute2 from the nova LXC:
   ```
   nova boot vm1 --flavor <flavor-id> --image <image-id> --nic net-id=<net-id> --availability-zone nova:compute2
   ```

3. Check the location of VMs:
   ```
   nova list --fields name,status,task_state,host
   ```

4. Execute the below command to constantly monitor the host status of the VMs running on compute2 node:
   ```
   watch nova list --fields name,status,task_state,host
   ```

5. For checking the logs of the vm-ha service, enter the following command in a separate terminal window of the same nova LXC:
   tail -f /var/log/vm-ha/vm-ha.log

6. Ssh to compute2 node and stop the pacemaker-remote service by executing the below command:
   ```
   service pacemaker_remote stop
   ```

   Note: wait for about 60 seconds for the evacuation to complete. Keep looking at the host column from the command at step #4.

Cluster Administration

Some important commands for managing the cluster are as follows:

1. Checking the status of pcs cluster
   ```
   pcs status
   pcs cluster status
   pcs status nodes
   crm_mon
   ```

2. To remove any failed actions from pcs:
   ```
   pcs resource cleanup
   pcs resource cleanup <resource_name>
   ```

3. Cluster authentication:
   ```
   pcs cluster auth
   pcs cluster auth <nodes>
   ```

Node IPMI Settings

In order to view IPMI settings for any node use the following commands:

1. To view settings of any node, ssh into that node and run the following command:

```
ipmitool -I lanplus -H 172.20.6.225 -U plumgrid -P plumgrid lan print 1
```

2. To change IP address, netmask, gateway of the node:

```
ipmitool lan set 1 ipsrc static
ipmitool lan set 1 ipaddr x.x.x.x
ipmitool lan set 1 netmask x.x.x.x
ipmitool lan set 1 defgw ipaddr x.x.x.x
ipmitool lan set 1 arp respond on
ipmitool lan set 1 auth ADMIN MD5
ipmitool lan set 1 access on
```

Enabling Live Migrations

Before enabling live migrations make sure that shared storage is set up between the compute nodes. Follow the below steps in order to enable live migration:

1. On the compute nodes edit /etc/libvirt/libvirtd.conf

   ```
   before : #listen_tls = 0
   after : listen_tls = 0

   before : #listen_tcp = 1
   after : listen_tcp = 1

   add: auth_tcp = "none"
   ```

2. Edit /etc/init/libvirt-bin.conf

   ```
   before : exec /usr/sbin/libvirtd -d
   after : exec /usr/sbin/libvirtd -d -l
   ```

3. Modify /etc/default/libvirt-bin

   ```
   before :libvirtd_opts=" -d"
   after :libvirtd_opts=" -d -l"
   ```

4. Restart libvirtd service on compute nodes:

   ```
   service libvirtd restart
   ```

5. Perform live migration with the following command:

   ```
   nova live-migration <instance_uuid> <compute-node>
   ```

Useful Information

1. Adding/Removing remote nodes from the pacemaker cluster
   a. When you want to remove a remote node from pacemaker cluster, you can do that with the following command:

      ```
      pcs cluster remote-node remove <name-of-remote-node>
      ```

   b. When you want to add the node back into the pacemaker cluster, we use the command:

      ```
      pcs resource create <name-of-remote-node> ocf:pacemaker:remote op monitor interval=20
      ```

   c. The above command can give you an error:
      Error: unable to remove: cannot find remote-node 'overcloud-compute-2'

   d. This happens due to traces of the node still present in the database, check with:

      ```
      cibadmin --query | grep <name-of-remote-node>
      ```

e. In order to completely remove the node, use the following command:

```
crm_node --force -R <name-of-remote-node>
cibadmin --delete --xml-text '<node_state id="<name-of-remote-node>"/>'
cibadmin --delete --xml-text '<nvpair id="status-1-last-failure-<name-of-remote-node>"/>'
cibadmin --delete --xml-text '<nvpair id="status-2-last-failure-<name-of-remote-node>"/>'
```

Source: https://github.com/feist/pcs/issues/78

2. In order to remove Stonith levels for compute nodes, use the following command:

```
pcs stonith  level remove 1 <name-of-remote-node>
```

→ pcs stonith level remove <level> [node id] [stonith id]

3. In the case where a VM goes to an error state while the task status is still “Spawning” and the compute node nova-compute.log shows the following error:

   InternalServerError: PLUMgrid Plugin Error: Error connection to HTTPS server

   Here is how you solve this:
   Check the status of plumgrid service on all the nodes (controllers + compute)

```
service plumgrid status
```

   If the service is inactive on any node, perform the following steps

```
service plumgrid stop
service plumgrid start
```

   `service nova-compute restart` (on compute nodes ONLY)

4. Before maintaining the cluster (software upgrades and configuration changes which impacts the cluster resources) be sure to always put the cluster in maintenance mode.So that all the resources will be tagged as un-managed by pacemaker. Which means , Pacemaker monitoring will be turned off and no action will be taken by cluster until you remove the maintenance mode. This is one of the useful feature to upgrade the cluster components and perform the other resource changes. The command is:

```
pcs property set maintenance-mode=true
pcs property set maintenance-mode=false
```

   (removes maintenance mode)
   OR

```
pcs property unset maintenance-mode
```

5. For removing any resource from the pacemaker cluster use the following command:

```
pcs resource delete <resource-name>
pcs constraint location remove <name>
pcs constraint order remove <name>
pcs constraint colocation remove <name>
pcs constraint remove <name>
pcs stonith delete <name>
pcs stonith level remove <level>
```

6. To enable logging in pacemaker. Edit the file /etc/sysconfig/pacemaker and add the line: PCMK_debug=yes