Step by Step Multiple KVM Host/Ubuntu Server 16.04

Preface: I am not a writer by any means what so ever, So if you see a mistake or you want to help with this document let me know. I would appreciate the help. So we will be doing a multiple server install of KVM on Ubuntu Server 16.04 step by step.

Prerequisites: At least 2 physical servers with at least 8 gb of Ram, 4 CPUs, w/1 hd, and 1 nic
ideal hardware would be 16 gb of ram, 8 cpus, 2 hd/ssd, 2 nics.

My setup is 4 servers in total. However the 4th server I will not be able cluster in a failover configuration. All of your servers should be Intel or AMD, The same technology(brand if you will) on CPUs is preferred makes it easier to do Live migrations.

Software: Ubuntu Server 16.04 Default install, qemu, libvirt, KVM, virt-manager. Bridge-utils

We need to remove the default LXD and snapd installation. Not all server administrators want to use containers. The Ubuntu folks decided to force this on us as opposed to giving us a choice.

Note: Anything in Bold Italics you should type at the command prompt.

A – Base Server Prep

1 – ***apt-get remove –purge lxd snapd*** ← This removes it from the server and cleans it up as well. (16.04 Only at this point)

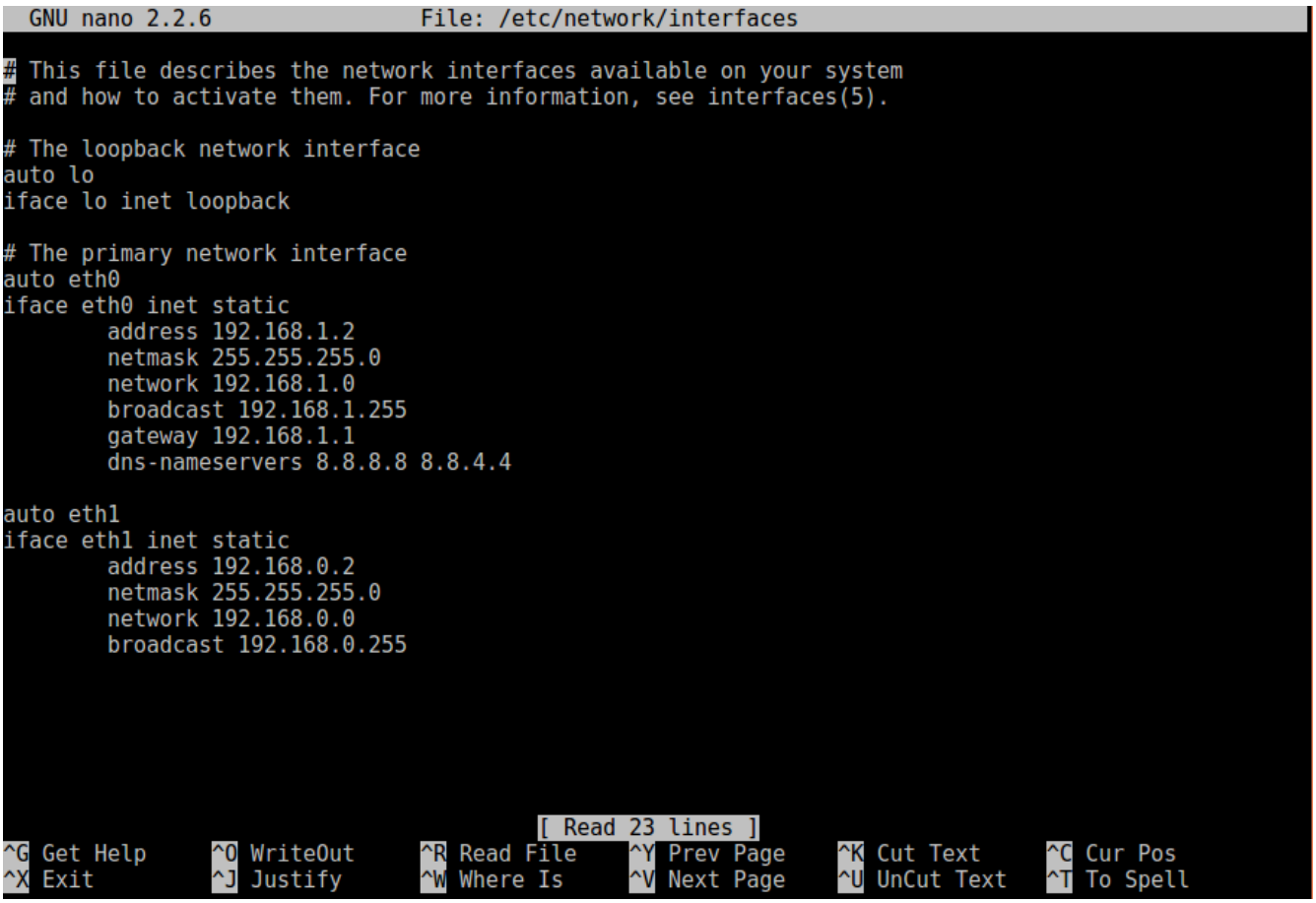
Set static ip to primary nic – always make backup copies of files before editing (*cp /etc/network/interfaces /etc/network/interfaces.orig*) you can type ***ip link*** at prompt to find your device names.

2 – ***nano -c /etc/network/interfaces***

Original interfaces file set for dhcp

```
  GNU nano 2.5.3                        File: /etc/network/interfaces.orig

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp1s0
iface enp1s0 inet dhcp
```

3 – Edit the file to look like the screen capture below, please notice that dhcp has been changed to static.



We have to set the ip to the hosts file for the server
nano -c /etc/hosts

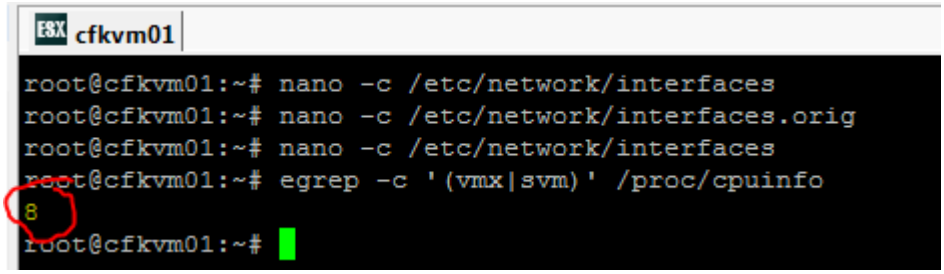
Change the /etc/hosts and put the ip of your server in
192.168.xxx.xxx server1.example.com server1

Change the /etc/hostname to the fqdn of the server
nano -c /etc/hostname

Change the hostname to match the server
server1.example.com

Let's check to make sure the system supports Virtualization
4 - ***egrep -c '(vmx|svm)' /proc/cpuinfo***

If the output is 0 then Virtualization will not work on your system



Let's install some tools we will need:
5 - ***apt-get -y install ntp ntpdate nano wget htop rkhunter cpu-checker***
choose **local** only at prompt

Now let's install KVM on your system (this is a single server install)

6 - ***apt-get -y install qemu qemu-kvm libvirt-bin bridge-utils***

Choose ***Local only*** on the postfix config

You should always avoid logging in remotely as root, So we need to add our current user to libvirtd group and to the kvm group as well.

7 - ***adduser `id -un` libvirtd***
8 - ***adduser `id -un` kvm***

Now let's check to make sure it was successfully installed.

9 - ***virsh -c qemu:///system list***

It should display something like this:

```
root@server1:~# virsh -c qemu:///system list
 Id Name                 State
----------------------------------

root@server1:~#
```

We need to setup a network bridge so our VMs can communicate with the other hosts as if they are physical servers on the network.

10 – ***nano -c /etc/network/interfaces***

Before the modification, my file looks as follows:

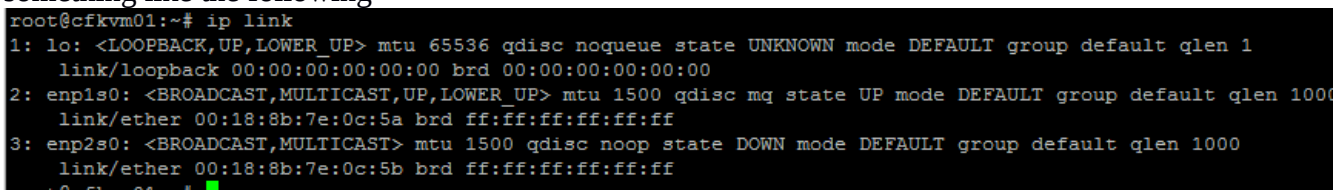
```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.0.100
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        dns-nameservers 8.8.8.8 8.8.4.4
```

I change it so that it looks like this:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet manual


auto br0
iface br0 inet static
        address 192.168.0.100
        network 192.168.0.0
        netmask 255.255.255.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        dns-nameservers 8.8.8.8 8.8.4.4
        bridge_ports eth0
        bridge_fd 9
        bridge_hello 2
        bridge_maxage 12
        bridge_stp off
```

Be sure to use your ip scheme not mine.

11 - Use your nic name (if not eth0) you can find it by typing ***ip link*** at the prompt, you will see something like the following



Notice that mine is enp1s0 instead of eth0

Now restart the network service:
12 - ***/etc/init.d/networking restart*** or ***systemctl restart networking.service***
13 - ***reboot***

Now lets create a script to run this rootkit sniffer on a daily basis

14 – ***nano -c /etc/cron.daily/rkhunter-cron.sh***

#!/bin/sh
(
rkhunter --versioncheck
rkhunter --update
rkhunter -c -sk –cronjob
) | mail -s 'RKHunter Daily Check' your_email@your.domain

15 – ***chmod +x /etc/cron.daily/rkhunter-cron.sh***

Let's turn off remote root access

16 – ***nano -c /etc/ssh/ssh_config***

at the bottom of the file add the following:
Securing remote access
PermitRootLogin no

17 – ***service ssh restart***

Let's update the OS and it's packages

18 – ***apt-get update***

19 – ***apt-get upgrade***
20 - type ***Y*** at the prompt

Now let’s use bash instead of dash, answer no at prompt
21 - ***dpkg-reconfigure dash***

Now, I will turn off the AppArmor (Equivalent of SELinux) and the firewall because I am already behind a firewall. I don't see the sense in being double firewalled and it may block something for the HA KVM Servers and cause VMs not to work.

22 - ***service apparmor stop***
23 - ***update-rc.d -f apparmor remove***
24 - ***apt-get -y remove apparmor apparmor-utils***

Now we turn off the firewall

25 – ***service ufw stop***
26 – ***ufw disable***

Let's reboot the server so that starts fresh with all the software and configuration

27 – ***reboot***

Storage:

If you have a second drive or storage let's do that now

28 – ***fdisk -l***
You should see something like the following

```
root@cfkvm04:~# fdisk -l

Disk /dev/sda: 250.1 GB, 250059350016 bytes
255 heads, 63 sectors/track, 30401 cylinders, total 488397168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000ef4df

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048      499711      248832   83  Linux
/dev/sda2          501758   488396799   243947521    5  Extended
/dev/sda5          501760   488396799   243947520   8e  Linux LVM

Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders, total 1953525168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk identifier: 0x8338ebee

   Device Boot      Start         End      Blocks   Id  System

Disk /dev/mapper/cfkvm04--vg-root: 232.6 GB, 232628682752 bytes
255 heads, 63 sectors/track, 28282 cylinders, total 454352896 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/cfkvm04--vg-root doesn't contain a valid partition table

Disk /dev/mapper/cfkvm04--vg-swap_1: 17.2 GB, 17167286272 bytes
255 heads, 63 sectors/track, 2087 cylinders, total 33529856 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

notice the /dev/sdb, I have already put a partition and formatted with ext4, so all have to do is mount it.

29 – ***mkdir -p /mnt/VMs*** # Create a mount point – Directory if you will
30 - ***mount /dev/sdb1 /mnt/VMs*** # Mount the second drive to the mount point just created
31 – ***mount -a*** # This reloads all the mounts in fstab file.
32 – ***nano -c /etc/fstab***
You will see something similar to the following:

/etc/fstab: static file system information.
#
Use 'blkid' to print the universally unique identifier for a
device; this may be used with UUID= as a more robust way to name devices

that works even if disks are added and removed. See fstab(5).
#
<file system> <mount point> <type> <options> <dump> <pass>
/dev/mapper/cfkvm04--vg-root / ext4 errors=remount-ro 0 1
# /boot was on /dev/sda1 during installation
UUID=5ddbe69c-88b9-4a83-84b2-82d1532d72b9 /boot ext2 defaults 0 2
/dev/mapper/cfkvm04--vg-swap_1 none swap sw 0 0

 # Mounting the Second Drive #
/dev/sdb1 /mnt/VMs ext4 defaults 0 1

Go to the bottom and add your drive and mount point. This will let it automatically load/mount when system starts.

I have a QNAP TS-412 NAS on my network, and I have a NFS Share I need that has my ISOs on it.
33 - ***apt-get -y install nfs-common*** # this is all I need to set my server up as a NFS Client
34 - ***mkdir -p /mnt/ISOs*** # Making the mount point for my NFS Share
35 - ***mount 1.2.3.4:/ISOs /mnt/ISOs*** # Now I have connected to my NAS and mounted the share locally on my server. Note: 1.2.3.4 Put your NAS Server ip where this is. Now like you did earlier you can add this to your fstab to where it automounts at server startup. If you need help email me and I can help you with it.

NOTE: *If you are only doing a single server setup then you can stop here.*

B - HA Configuration and Live Migration for KVM

Lets set up the share so we can do Live Migration, I prefer to use the /mnt area so I know where all of my network mounts/shares are located.

29 - ***apt-get install nfs-common (***should already be done if you are following this***)***
30 – ***mkdir -p /mnt/dir***
at this point you can go to your management client and start vmm (virt-manager)

Next we should add this to where it is loaded on server startup
31 – nano -c /etc/fstab
32 – ***ip_of_nas:/dir/to/mount /mnt/dir nfs defaults 0 0***

Appendix: Last but not least we need to manage our Virtual Infrastructure, with VMM (Virtual Machine Manager) otherwise known as virt-manager. There are several other ways to do this but this is the easiest to get you up and running. Opennebula, oVirt, ConVIRT, Openstack and others exist.

Note: This is mainly for the GUI people out there, it is very user friendly.

Okay lets get started:

Installing Virt-Manager on a remote Linux Desktop (mine is Ubuntu Desktop 16.04) to manage KVM

- ***apt-get update***
- ***apt-get upgrade***
- ***apt-get install virt-manager***
- ***apt-get install ssh-askpass-gnome ssh-askpass***

The above allows you to connect remotely to the KVM server and create VMs and such.

- ***reboot***

Next you want to start the virt-manager GUI

Do a search on your desktop the icon is in the upper right corner, type Virtual and VMM should show click it and off we go.

Alternative Method:
CLI - In a terminal do this → ***virt-manager –no-fork*** (but keep terminal open and watch because passwords will apear there.)

This should do it for your virtualization setup and home lab, you can contact me on my site → www.coopfire.com, I have a contact form and live chat as well. If you would like to email me you can do so here: mcooper@coopfire.com

Thank you for reading and please support the Open Source Community that you are a part of….