

Step by Step Single KVM/Virtual Ubuntu Server 14.04

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 single 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.

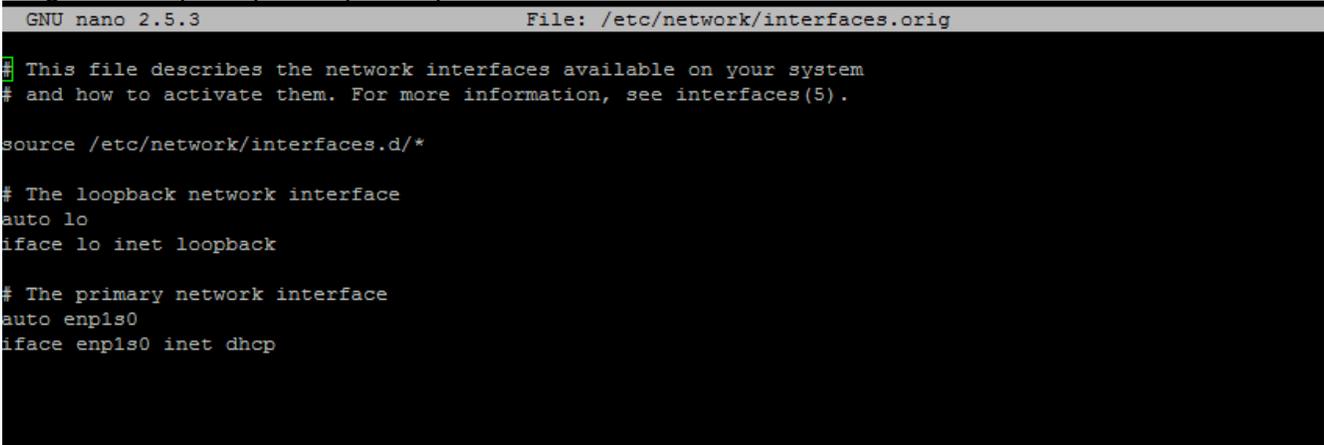
A – Base Server Prep

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.

```
GNU nano 2.2.6      File: /etc/network/interfaces
# 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 192.168.1.2
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-nameservers 8.8.8.8 8.8.4.4

auto eth1
iface eth1 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255

[ Read 23 lines ]
^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page      ^U UnCut Text    ^T To Spell
```

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

```
ESX cfkvm01
root@cfkvm01:~# nano -c /etc/network/interfaces
root@cfkvm01:~# nano -c /etc/network/interfaces.orig
root@cfkvm01:~# nano -c /etc/network/interfaces
root@cfkvm01:~# egrep -c '(vmx|svm)' /proc/cpuinfo
8
root@cfkvm01:~#
```

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

```
root@cfkvm01:~# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 00:18:8b:7e:0c:5a brd ff:ff:ff:ff:ff:ff
3: enp2s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:18:8b:7e:0c:5b brd ff:ff:ff:ff:ff:ff
```

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***