0.1 Configure Virtual Network

In the previous step we created the system on which we will use to create our virtual environment.

We will use VERY commonly used tools called QEMU, KVM and Libvirt to enable a virtual network and virtual machines under outernetwork1. In fact, the very same open source tools (QEMU, KVM and Libvirt) are used by Cloudstack.

# Install tools

You will need to sudo most all of these commands as you are changing the system configuration.

First make sure that the apt repository system indexes are up to date:

apt update

Now install the following packages using apt install:

qemu-kvm -- software that provides hardware emulation for the KVM hypervisor

libvirt-bin -- software for managing virtualization platforms

bridge-utils -- a set of command-line tools for configuring ethernet bridges

virtinst -- a set of command-line tools for creating virtual machines

# Delete the default KVM network

In your first net-list below, you should see the default network.

virsh net-list --all

virsh net-undefine default

virsh net-destroy default

virsh net-list --all

In the last net-list above, you should not see the default network.

# Create a NAT network

You are going to create a NAT network that defines a subnet 172.16.10.0; which conveniently matches the Cloudstack documentation for how to configure the network.

Create a file called nat1.xml with this content

<network>

<name>nat1</name>

<forward mode='nat'>

<nat>

<port start='80' end='65535'/>

</nat>

</forward>

<bridge name='nat1' stp='on' delay='0'/>

<ip address='172.16.10.1' netmask='255.255.255.0'>

</ip>

</network>

And now run these commands:

virsh net-define nat1.xml

virsh net-autostart nat1

virsh net-start nat1

Run another virsh net-list --all to see your nat1 network in an active state.

# Prepare SSH

You will want to be able to access this VM using SSH (with a password). To do so

Use the passwd command to set a password for your Google login account:

passwd <google login name>

Edit /etc/ssh/sshd\_config so that PasswordAuthentication has the value “yes”.

Restart the sshd service (to read the change you just made to sshd\_config). You use systemctl command to do this.

# Set up the network

Add a rule to pass things heading to port 8080 on to innerhost

iptables -t nat -I PREROUTING -p tcp -i ens4 --dport 8080 -j DNAT --to 172.16.10.2:8080

Install a tool to persist your iptables changes across system boots:

apt install iptables-persistent

when prompted during the install answer yes

This tool creates files at /etc/iptables/rules.v4 and rules.v6. If you make changes to the iptables rules you can do

iptables-save > /etc/iptables/rules.v4

to store the rules. They will be reset when you boot.

NOTE: libvirt sets its own firewall rules every time you boot. The iptables-persistent tools apply their changes BEFORE libvirt applies its changes -- thus, the libvirt rules prevail.