Mohammed Aboullaite | 26 Nov 2017 | 3 min (757 words)

KVM/libvirt: Forward Ports to guests with Iptables

I've been playing in the past few weeks with kvm/qemu and libvirt to set up some VMs. If you've been using libvirt before, you'll for sure agree that Libvirt is particularly awesome when it comes to managing virtual machines, their underlying storage and networks. However, if you happen to use NAT-ed networking and want to allow external access to services offered by your VMs, you've got to do some manual work. In this post, I'll try to demystify the process of NAT port forwarding to VMs.

Assign static IP address to VM

First, we need to list available networks. to do so, we use the following command:

```
$ virsh net-list
Name                 State      Autostart     Persistent
----------------------------------------------------------
xhubnet              active     yes           yes
```

To see the `xhubnet` network information:

```
    $ virsh net-dumpxml xhubnet
    <network>
    <name>xhubnet</name>
    <uuid>2ba36c4f-46c9-4767-a9fc-6808cf001a19</uuid>
```

```
    <forward mode='nat'>
        <nat>
        <port start='1024' end='65535'/>
        </nat>
    </forward>
    <bridge name='virbr1' stp='on' delay='0'/>
    <mac address='52:54:00:8e:b5:88'/>
    <domain name='xhubnet'/>
    <ip address='192.168.111.1' netmask='255.255.255.0'>
        <dhcp>
        <range start='192.168.111.120' end='192.168.111.254'/>
        </dhcp>
    </ip>
    </network>
```

Where it's clearly indicated that the DHCP range is between 192.168.111.120 and 192.168.111.254.

N[illegible] first step toward setting up a public IP to our VM, is to get it MAC addre[illegible] Again this is easy to achieve using `virsh` CLI:

```
$ virsh dumpxml VM_NAME | grep -i '<mac'
  <mac address='52:54:00:e2:95:c6'/>
```

The last step is to edit the network config to add new VM config using the following command:

```
$ virsh net-edit xhubnet
```

and add this line after the `<range ...>` section:

```
<host mac='52:54:00:e2:95:c6' name='VM_NAME' ip='192.168.111.36'/>
```

To make changes effective, we need to start DHCP service using:

```
$ virsh net-destroy xhubnet
$ virsh net-start xhubnet
```

Restarting DHCP service should be enough, However in some cases you need to restart the `libvirt-bin` service:

```
virsh shutdown VM_NAME
/etc/init.d/libvirt-bin restart
$ virsh start VM_NAME
$ ping -a 192.168.111.36
```

Configuring the Firewall to port forward

By default, guests that are connected via a virtual network with `<forward mode='nat'/>` can make any outgoing network connection they like. Incoming connections are allowed from the host, and from other guests connected to the same libvirt network, but all other incoming connections are blocked by iptables rules.
If we would like to make a service that is on a guest behind a NATed virtual network publicly available, we need to setup the necessary iptables rules to forward incoming connections to the host on any given port.

Let's suppose for example that we need to forward the forward incoming connections to port 9867 on host machine to port 22 on guest machine. below are the needed rules to achieve that:

```
# connections from outside
$ iptables -I FORWARD -o virbr1 -d  192.168.111.36 -j ACCEPT
$ iptables -t nat -I PREROUTING -p tcp --dport 9867 -j DNAT --to 192.168.111.

# Masquerade local subnet
$ iptables -I FORWARD -o virbr1 -d  192.168.111.36 -j ACCEPT
$ iptables -t nat -A POSTROUTING -s 192.168.111.0/24 -j MASQUERADE
$ iptables -A FORWARD -o virbr1 -m state --state RELATED,ESTABLISHED -j ACCEF
$ iptables -A FORWARD -i virbr1 -o eth0 -j ACCEPT
$ iptables -A FORWARD -i virbr1 -o lo -j ACCEPT
```

w irbr1 is the interface in `192.168.111.0/24` subnet and `eth0` is interface wi ic .

Now that we have set up port forwarding, we can save this to our permanent rule set and load the rule set:

```
$ service netfilter-persistent save
$ service netfilter-persistent reload
```

Now, test that your VM is accessible through your firewall's public IP address:

```
$ ssh user@PUBLIC_IP -p 9867
```

This should work!