NASA hw4

B04902045 孫凡耘

April 30, 2017

System Administration

1

使用與 lab 相同的環境。
CentOS-7-x86_64-Minimal-1611.iso
1. setup network with nmtui
2. yum install -y virt-install qemu-kvm libvirt
3. start and enable libvirt
$ systemctl start libvirtd
$ systemctl enable libvirtd
4. Download CentOS ISO
$ curl [url] -o centos.iso
$ mv centos.iso /var/lib/libvirt/images

Task 1

Enable ”Virtualize Intel VT-x/EPT or AMD-V/RVI” in VM settings

Verify Intel VT CPU virtualization extensions on a Linux
Type the following command as root to verify that host cpu has support for Intel VT technology

```
$ grep --color vmx /proc/cpuinfo
```

If the output has the vmx flags, then Intel CPU host is capable of running hardware virtualization.
P.S. The /proc/cpuinfo file has information about the CPU.

< 方法一 >
You can specify a user which will have access to libvirt’s setup (and thus creating VMs and pretty much running virsh commands) by adding the users to the libvirtd and kvm groups on the host.

```
$ sudo usermod -a -G libvirt <username>
$ sudo usermod -a -G kvm <username>
```

Change group ownership and permissions on the default images directory:

```
$ sudo chown root:libvirt /var/lib/libvirt/images
$ sudo chmod g+rw /var/lib/libvirt/images
```

after doing that, I was able to create VM with virt-install using non-root user, and only this certain user can create a VM without sudo.

You can also use policykit to manage access, the procedure is described in the libvirt Wiki: `http://wiki.libvirt.org/page/SSHPolicyKitSetup`
Connect to VM host without root permission

```
virsh -c qemu+ssh://user@IP/system
user can be root or non-root user
```

2

# for graphical console

```
yum install virt-viewer

sudo virt-install -n VM_NAME --os-type=generic --ram=256 --vcpus=2 \
    --disk path=/dev/nasavg/lv1 \
    --graphics=vnc,password=MY_PASSWORD,listen=0.0.0.0,port=5903 \
    --location /var/lib/libvirt/images/centos.iso \
   --extra-args="ks=anaconda-ks.cfg console=tty0 console=ttyS0,115200n8" \
    --network bridge:virbr0
```

parameters explained:

-n NAME
–**os-type**=OS_TYPE
–**ram**=MEMORY
–**vcpus**=Number of virtual processors to configure for the guest
–**disk path**=A path to some storage media to use
P.S. /dev/volume_group/logical_volume is the lvm of VM host
–**graphics=vnc,password=MY_PASSWORD,listen=0.0.0.0,port=5903**
Set up a virtual console in the guest and export it as a VNC server in the host. connect the VM to a VNC server listening on every interface (0.0.0.0) on port 5903 with password
–**location**
Specifies the installation source for guest virtual machine kernel+initrd pair(or ISO).
–**extra-args**=additional kernel command-line arguments to pass to the installer when performing a guest installation from a kernel+initrd.

ks=kickstart file path
P.S. I modified the kickstart file anaconda-ks.cfg
–**network**=NETWORK
The virbr0, or ”Virtual Bridge 0” interface is used for NAT .
It is provided by the libvirt library, and virtual environments sometimes use it to connect to the outside network.

reference: `https://velenux.wordpress.com/2013/01/04/creating-a-new-libvirt-virtual-machine-fr`
Problem encountered:

```
failed to initialize KVM: Permission denied
```

Resolved by:

```
rmmod kvm_intel
rmmod kvm
modprobe kvm
modprobe kvm_intel
```

# Another way to use password to protect my vnc session

```
virsh edit centos
```

Edit it as follow:

```
<graphics type='vnc' port='-1' autoport='yes' listen='192.168.1.5' \
passwd='PASSWORD-HERE' keymap='en-us'/>
```

reference: `https://www.cyberciti.biz/faq/linux-kvm-vnc-for-guest-machine`

3

virsh console: Access a virtual machine’s console over a serial connection
To do this on a physical machine, one has to do:
1. Configure console access on the target machine
2. On Linux: dterm, Minicom, picocom, Screen, Serialclient, For windows: gtkterm, Cutecom, putty
For detailed instructions: `https://wiki.archlinux.org/index.php/working_with_the_serial_console`
source: `http://keywordsuggest.org/gallery/762307.html`
P.S.
virsh ttyconsole <my VM name>
If the output is shown(mine is /dev/pts/0), it indicates the Guest has a console device already.

4

1. Add console=ttyS0 to line in /boot/grub2/grub.cfg
   should look something like this

```
linux16 /vmlinuz-3.10.0-123.el7.x86_64 \
root=UUID=ba0f2424-e66e-4862-90ff-7dccb63339c2 ro rd.lvm.lv=centos/swap \
vconsole.font=latarcyrheb-sun16 rd.lvm.lv=centos/root crashkernel=auto  \
vconsole.keymap=us LANG=en_US.UTF-8 console=ttyS0
```

2. Allow login into ttyS0

```
$ echo "ttys0" >> /etc/securetty
```

3. assert VM guest console type(should be serial)

```
<console type='pty' tty='/dev/pts/6'>
    <source path='/dev/pts/6'/>
    <target type='serial' port='0'/>
    <alias name='serial0'/>
</console>
```

4. start ttyS0

```
$ sudo start ttyS0
or
$ systemctl enable serial-getty@ttyS0.service
$ systemctl start serial-getty@ttyS0.service
```

5. login with

```
$ virsh console generic
```

Above should do it but if you want to see even the grub2 menu options you can try the following: in guest VM in /etc/default/grub replace

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
#GRUB_TERMINAL=console
```

by

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0"
GRUB_TERMINAL="serial console"
```

and

```
$ update-grub
```

can login to your VM with the command virsh console automatically after the vm boots. reference: http://blog.ls-al.com/kvm-virsh-console-on-centos-7/
reference: https://mcdee.com.au/kvm-virsh-console-access-centos/
reference: https://serverfault.com/questions/364895/virsh-vm-console-does-not-show-any-output

5

Bridge to virbr0(NAT)
P.S. Do *NOT* attempt to attach a physical device to 'virbr0' - this is only for NAT connectivity
Execute the following command on VM host to bridge the VM to VM_host through virbr0(NAT)

```
$ virsh attach-interface --domain generic --type bridge --source virbr0
```

configure network with nmtui on VM guest(Activate the wired connection)
reference: https://kashyapc.fedorapeople.org/virt/add-network-card-in-guest.txt
Bridging for VM
Use nmtui
1. add a bridge connection named br0
2. Add ens33 as slaves of br0
3. reboot
我們在這一步可以看到ens33下面的ip的設定了，br0下面有ip的設定
4. execute the following command

```
$ virsh attach-interface --domain generic --type bridge --source br0
```

5. Inside the vm of vm, dhclient the interface（attach之後跑出來的)

```
$ sudo dhclient ens9
```

6. Done!
這時可以看到vm host跟vm in vm有相同個網段，有gui那臺centos也可以ping到vm中的vm
reference: http://pominglee.blogspot.tw/2014/03/linux.html

6

1. virsh list
2. virsh undefine VM_NAME
3. virsh domiflist VM_NAME
4. virsh detach-interface domain type
5. virsh edit