

VNC and Initial VM Build with VMBuilder

Contents

1 Overview	1
2 Using VNC Server and Client	2
2.1 Start vncserver on your MacMini	2
2.2 Install and Start a VNC client on your local machine	2
2.3 Start the libvrit graphical interface tool virt-manager	3
3 Build an initial virtual machine using VMBuilder	3
3.1 Using vmbuilder	3
4 Managing your VM	4
4.1 Manage your VM using virt-manager	4
4.2 Manage your VM using the command line and virsh	5

1 Overview

This lab will step you through the following:

- Starting an X-Server on your MacMini. We will use the VNC Server software.
- Connecting to your MacMini's desktop from a client machine using VNC client software.
- Start and use virt-manager
- Building a single, initial virtual machine using VMBuilder
- Managing your virtual machine with virt-manager (GUI)
- Managing your virtual machine from the command line

Beware that some commands are typed into your MacMini and some are typed into a virtual machine: take a note of the prompt you see before typing a command.

```
nsrc@kitX:~$      << the MacMini's operating system
```

```
sysadm@fun:~$     << the VM's operating system
```

2 Using VNC Server and Client

2.1 Start vncserver on your MacMini

Select one person from your team to log in on the MacMini and do the following to start the vncserver X-Server Desktop software.

NOTE Please do *not* do this as the ‘root’ user!

```
nsrc@kitX:~$ id          << check that you are logged in as user 'nsrc'
nsrc@kitX:~$ vncserver :1
```

For the requested password use the classroom admin password.

You will see several files being created and then the vncserver software will be running and ready to use.

2.2 Install and Start a VNC client on your local machine

Now select one person in your group that will run an X Window Desktop from your MacMini for the remaining exercises. Once you have done that in a web browser go to:

<http://kit1.lab.nsrc.org/downloads/vnc/>

You will see three directories:

```
Linux
Macintosh
Windows
```

Select the directory appropriate for your machine, then download the software. If you are using Linux you may already have a VNC viewer available to use (called “Remmina”), or you may be able to install the software more simply through your packaging system. If you need help installing and running the software ask your instructor for assistance.

Once you have TigerVNC installed run the program and in the “VNC server:” box enter:

```
x.x.x.x:1
```

replacing x.x.x.x with your MacMini’s IP address or DNS hostname. If you are connected onto the private network behind your MacMini then you can enter

```
s1.ws.nsrc.org:1
```

When prompted for a password use the classroom admin password.

You will now have an X Window Desktop appear on your screen with one open terminal.

2.3 Start the libvrit graphical interface tool virt-manager

In the open terminal that appears in your X Window Desktop (known as an “xterm”) type the command:

```
nsrc@kitX:~$ virt-manager
```

You will see a new window appear that is empty. Leave this window open as we build and manage our virtual machines.

If, for some reason, your X Window Desktop crashes or becomes unresponsive you should log in on your MacMini and do the following:

```
nsrc@kitX:~$ vncserver -kill :1
nsrc@kitX:~$ vncserver :1
```

And, then, reconnect using TigerVNC to your X Window Desktop.

3 Build an initial virtual machine using VM-Builder

3.1 Using vmbuilder

VMBuilder has already been installed and configured on your MacMini, so we just need to use the command properly. Here we are going to build a virtual machine with the following properties:

- It will be a KVM virtual machine (VM)
- It will be running ubuntu
- Name the VM whatever you want `<SOME_FUN_NAME>` (without spaces or odd characters). Be sure you replace “`<SOME_FUN_NAME>`” with other text, like “zombierobots”
- The VM will have 512MB of RAM
- The size of the VM will be 20G
- The VMs data file (known as a “backing store”) will reside in the `/data/vm/` directory

Now in the xterm in your X Window Desktop type the following commands. The `vmbuilder` command is a bit long and it should be on one line.

```
nsrc@kitX:~$ cd
nsrc@kitX:~$ sudo mkdir /data/vm
nsrc@kitX:~$ sudo vmbuilder kvm ubuntu --hostname <SOME_FUN_NAME> --mem 512 --debug --roots
```

After the VM completes building it should appear in the `virt-manager` window in your X Window Desktop.

4 Managing your VM

4.1 Manage your VM using `virt-manager`

- In the `virt-manager` window on your X Window Desktop highlight the Virtual Machine and then click the Run option above.
- Immediately after click on the Open button - you can watch as your VM boots.
- At the log in prompt you can log in to the VM using:

```
username: sysadm
password: nsrc+ws
```

Once you are logged in obtain the VM's IP address:

```
sysadm@fun:~$ ifconfig
```

Make a note of the IP address.

Now let's show how much memory is in use by typing:

```
sysadm@fun:~$ free
```

You see something like 507260 total memory (512MB).

Now shut it down:

```
sysadm@fun:~$ sudo halt -p
```

Remember, the sudo password will be the password for the sysadm user.

Now you can watch the VM reboot in the console window. Notice how quickly it reboots!

Now shut down the VM by pressing the ShutDown button at the top of the console window. To close the console window press the “X” on the upper-right corner of the window.

4.2 Manage your VM using the command line and virsh

Now either using the xterm in your X Window Desktop or by opening an ssh session to your MacMini we will manage your VM from the Linux command line using the “virsh” utility that is part of libvirt. Everyone in the group can do this part.

Show all your created and available VMs to run:

```
nsrc@kitX:~$ virsh list --all
```

You should just see your VM’s name and “shut off” in the right column of output.

Let’s have a look at the configuration file for your VM that describes how it is configured. The file is in XML and you can see this by typing:

```
nsrc@kitX:~$ virsh dumpxml <SOME_FUN_NAME>
```

The probably scrolled by quickly on the screen, so to review the contents a bit more slowly do:

```
nsrc@kitX:~$ virsh dumpxml <SOME_FUN_NAME> | less
```

Press the space bar to page and “q” to quit when you reach the end of the file.

Now we are going to edit the XML configuration file directly and update the amount of RAM allocated to the VM. Only ONE person should do this.

```
nsrc@kitX:~$ virsh edit <SOME_FUN_NAME>
```

By default this places you in the vi editor. There are two lines near the top of the file we wish to change. These look like:

```
<memory unit='KiB'>524288</memory>
<currentMemory unit='KiB'>524288</currentMemory>
```

Let's add 512MB of RAM to the VM by changing these lines to look like:

```
<memory unit='KiB'>1048576</memory>
<currentMemory unit='KiB'>1048576</currentMemory>
```

Now save and exit from the file (in vi that's ESC, then :wq).

You could have changed both these parameters from the command line using virsh commands, and you can even change a VM's memory allocation while it's running using the command (*don't do this now*):

```
nsrc@kitX:~$ virsh setmem <SOME_FUN_NAME> 1048576 --config --live
```

But, to change memory permanently the VM must be stopped.

Now to start your VM from the command line do:

```
nsrc@kitX:~$ virsh start <SOME_FUN_NAME>
```

You can see that the VM is running from the command line:

```
nsrc@kitX:~$ virsh list --state-running
```

Or, you can probably see this in your virt-manager GUI tool in your X Window Desktop.

Now you can SSH to the VM, but let's open a console from the command line by typing:

```
nsrc@kitX:~$ virsh console <SOME_FUN_NAME>
```

Press and log in as user sysadm again.

Let's see how much RAM the VM now has:

```
sysadm@fun:~$ free
```

You should see 1026020 total Memory available. That is pretty cool! You just added 512MB to a virtual machine in a few simple steps. You can make other changes to your VMs using this method as well.

Now let's shut down your VM to finish this session. You can either highlight the VM in your virt-manager GUI and press the "Shutdown" button or in your console session you can issue a halt command, but let's exit from the console and use virsh to shut down your VM:

```
sysadm@fun:~$ exit
```

At the "login:" prompt press the keys CTRL and] (normally represented as "CTRL-]") to exit from the console session, then at the command line type:

```
nsrc@kitX:~$ virsh shutdown <SOME_FUN_NAME>
```

And you should see the message "Domain is being shutdown" - Wait a few seconds then type:

```
nsrc@kitX:~$ virsh list --all
```

and see if your VM says "shut off" next to it. You can see this in virt-manager as well.