libvirt and vmbuilder exercise

Contents

1 Objective

You are going to use vmbuilder to build an Ubuntu VM image automatically, and manage it using the "virsh" component of libvirt.

2 Using vmbuilder

2.1 Check the configuration

The package "python-vmbuilder" should already be installed on your machine. Have a look at its top-level configuration file:

```
cat /etc/vmbuilder.cfg
```

This defines how the VM will be built:

- what architecture we are building (i386 = 32-bit, x86_64 = 64-bit)
- where to download packages from (mirror, security-mirror). Note that we are pointing to apt.ws.nsrc.org:3142 which is the apt-cacher running on your server.
- which version of Ubuntu we are building (suite, flavour)
- which extra packages to include
- ssh key to include
- initial username and password. Notice that these are different to the credentials you use to login to your server. Generally, the host server password is private and not known to the users of the VMs.

There are also file snippets and templates under `/etc/vmbuilder/` which are used during the build. You normally do not need to change these.

2.2 Build a VM

Everyone in your group will ssh into the server and build a VM; it's fine for you to do them at the same time. Use the following name for your VM:

- 1st person in the group: noc.ws.nsrc.org
- 2nd person in the group: noc2.ws.nsrc.org
- 3rd person in the group: noc3.ws.nsrc.org
- . . . etc

To build your VM, run the following command. If you are not the first person in the group then modify the two places where "noc.ws.nsrc.org" appears.

```
sudo vmbuilder kvm ubuntu --hostname noc.ws.nsrc.org --mem 512 --debug \
    --rootsize 20480 --dest /var/lib/libvirt/images/noc.ws.nsrc.org
```

This builds a VM which will initially have 512MB RAM allocated when it runs, and a root disk which can grow up to 20GB.

You should see progress messages scrolling up the screen. The first build may be slow as the packages are being downloaded for the first time.

> If you want to confirm that your apt-cacher is being used, you can do `tail -f /var/log/apt-cacher-ng/apt-cacher.log` in another session (ctrl-C to stop)
>
> If vmbuilder is running very slowly, you can do the "virt-manager and VNC" exercise at the end of the handout while you wait.

Check that no errors are shown at the end of the build. You should see something like:

```
...
2013-11-11 10:30:21,390 DEBUG   : Calling deploy method in VMBuilder.plugins.ubuntu.distro p
2013-11-11 10:30:21,391 DEBUG   : No such method
2013-11-11 10:30:21,391 DEBUG   : Calling deploy method in context plugin VMBuilder.plugins.
2013-11-11 10:30:21,391 DEBUG   : ['rm', '-rf', '--one-file-system', '/tmp/tmpjC8QLO']
```

If there is a problem, try to work out what is wrong from the error message, and ask for help if you need it.

3 Managing your VM

3.1 Start

Try the following commands

```
virsh list         # shows running VMs (should be empty)
virsh list --all   # shows all VMs, including stopped ones
```

Now start the VM that you built. Remember to change "noc.ws.nsrc.org" to the name of your VM.

```
virsh start noc.ws.nsrc.org
virsh list         # show running VMs
```

Is your virtual machine running? A low-level way of checking is to see if libvirt has started a "kvm" process

```
ps auxwww | grep kvm     # look for your KVM process
```

3.2 Attach to the console

Since you don't know what IP address your VM has got (if any), you need another way to connect to it. KVM provides two different ways:

- an emulated serial port
- an emulated VGA screen

We are going to use the serial port, since you can easily do this using your ssh session.

```
virsh console noc.ws.nsrc.org
```

Hit Enter a few times. You should get a login prompt. Login using the username and password which the VM was built with.

Type "ifconfig eth0" to see what IP address has been picked up by the VM.

3.3 Set up networking

While you are still attached to the console, edit `/etc/network/interfaces` to replace DHCP with a static IP address.

```
auto eth0
iface eth0 inet static
        address 10.10.0.XXX
        netmask 255.255.255.0
        gateway 10.10.0.254
        dns-nameservers 10.10.0.241
        # Disable UDP checksum offloading on virtio; it breaks when
        # packets traverse Dynamips
        post-up ethtool --offload eth0 tx off
```

Choose your IP address as follows:

- noc.ws.nsrc.org: 10.10.0.250
- noc2.ws.nsrc.org: 10.10.0.249
- noc3.ws.nsrc.org: 10.10.0.248
- . . . etc

Note that the cursor keys in the editor may not work exactly as expected over a serial console - try to manage. Alternatively, since you know the machine's temporary (DHCP-assigned) IP address, you can ssh to that.

3.4 Shutdown and restart

One way to shutdown the VM is to type `halt -p` within the VM. But you need to be very sure you are typing this in the VM and not the host server!

A safer way is to send a shutdown signal from the host.

- If you are still attached to the virtual console, disconnect by pressing `ctrl` and `]`
- Now issue the shutdown command to your VM

  ```
  virsh shutdown noc.ws.nsrc.org
  virsh list  # keep repeating this until your VM is no longer shown as running
  ```

- Now restart your VM

  ```
  virsh start noc.ws.nsrc.org
  ```

Check you can ping your VM on the static IP address you configured. It will probably take around 10-15 seconds before it starts to respond.

If your machine does not respond, then go back in using the virtual console and correct the problem.

3.5 ssh directly into the VM

Now you can confirm that you have ssh access into your VM. You should be able to ssh directly from your laptop to 10.10.0.XXX, where this is the IP address you assigned to your VM.

Once logged in, you are going to do some basic system administration on the VM.

```
sudo apt-get update
sudo apt-get install apache2
```

Edit `/var/www/index.html` and put your own “hello world” type message in it.

Now point your laptop’s web browser at http://10.10.0.XXX/ and check you can view this page. Check the pages for the other users in your group too.

Congratulations, you have installed a complete Ubuntu virtual machine with webserver!

3.6 Changing VM parameters

Let's say you decided that your noc.ws.nsrc.org needs more than 512MB of RAM to run. You can change the memory allocation in libvirt's XML configuration file for that VM.

First, have a look at the existing XML:

```
virsh dumpxml noc.ws.nsrc.org
```

Now use the command to edit it:

```
virsh edit noc.ws.nsrc.org
```

Look for this section near the top:

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

The first number is the *maximum* amount of memory which this VM can use, and the second is the *current* amount of memory allocated to this VM.

Change both these numbers to 1048576 (1GB), then shutdown and restart the VM.

ssh into the VM and type "free" to see how much memory is visible.

You can even change the memory used by a VM while it is running - this is called "hot-plug memory" - up to the maximum configured. The command is:

```
virsh setmem noc.ws.nsrc.org --size 400000
```

Type "free" again within the VM to see the change. It happens immediately.

3.7 Set up ssh key access

Of course, you don't want to type your password every time you connect to the VM using ssh.

To prevent this, you can put your public key into `/home/sysadm/.ssh/authorized_keys` in the VM. You'll have to create the .ssh directory if it doesn't already exist.

You can also allow root login by putting your public key into `/root/.ssh/authorized_keys` in the VM. This is very convenient for managing the VM, especially using automated scripts. Then try ssh'ing in as root.

To make this easier in future, put your public key into `/etc/vmbuilder/misc/authorized_keys` on the host. Then vmbuilder will include it in every VM it builds from now on.

4 OPTIONAL EXERCISES

4.1 virt-manager and VNC

You can try out the graphical interface which virt-manager provides.

- Login to your server as the "nsrc" user
- Type "vncserver". If you have never run this before, you will be prompted to choose a password to protect your desktop. Use the instructor password.
- On your laptop, download and run a vnc viewer application.
  - For Windows: real VNC
  - For Mac: chicken VNC
  - For Linux desktop: gvncviewer
- Using vnc viewer, connect to your server (10.10.0.241) on port 5901. You should get a desktop with a terminal window.
- Type "virt-manager" into the terminal window. This should start the manager which shows your VMs and allows you to control them. Move and resize it to fit better.
- Disconnect your VNC session and reconnect; you should get the desktop exactly as you left it.

4.2 Graphical VNC console onto your VM

For some VMs, like Windows VMs, a serial console may not be sufficient.

You can get a graphical console indirectly by using virt-manager. Alternatively, you can expose the VM's graphical console directly to the network. You need to configure libvirt to allow VNC to listen on external interfaces, not just the loopback interface.

To do this:

- virsh shutdown VNMAME
- wait for it to stop running (virsh list)
- virsh edit VMNAME

This will open up the XML definition. Find this section:

Find this section:

```
<graphics type='vnc' port='-1' autoport='yes' listen='127.0.0.1'>
  <listen type='address' address='127.0.0.1'/>
</graphics>
```

Change “127.0.0.1” to “0.0.0.0” and set a password, e.g.

```
<graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0' passwd='secret'>
  <listen type='address' address='0.0.0.0'/>
</graphics>
```

- virsh start VMNAME
- virsh vncdisplay VMNAME

This will return colon and a number, e.g. “:0”. Add 5900 to this number to get the VNC port number.

- Use your laptop VNC client to connect to your server (10.10.0.241) on the port you have calculated. Enter the VNC password.

4.3 Discarding a virtual machine

For reference: if you wanted to discard a virtual machine permanently, these are the steps you would have to take.

```
virsh destroy VMNAME                        # (if it's currently running)
virsh undefine VMNAME                       # delete the XML
sudo rm -rf /var/lib/libvirt/images/VMNAME  # delete the disk image
```