Xen Installation

Dom0 -> Ubuntu 14.04.5 - Trusty Tahr LTS DomU -> Ubuntu 14.04.4 - Trusty Tahr

Installing Dom0

During the Dom0 installation, use 'Install using LVM(Logical Volume Management)' option for Ubuntu 14.04. The disk space will be assigned by default. We can resize the partitions using LVM later. For the time being, accept the default sizes and finish the installation of Dom0. The ideal size of Dom0 is ideally between 5 to 10GB.

Installing Xen hypervisor

Install the 64-bit hypervisor. It also works with 32-bit dom0 kernel while allowing you to install 64-bit guests.

```
sudo apt-get install xen-hypervisor-amd64
```

Reboot the system .

```
sudo Reboot
```

To verify if the installation has succeeded, use the command

```
sudo xl list
```

The Domain-0 information will appear for a successful installation.

Network Configuration

The following configuration steps is for a wired interface. We will use the bridged network configuration.

1. Disable the Network Manager We are going to manually configure the network settings and hence we disable the network manager which might be automatically configuring the network settings for us. This step will cause temporary loss of internet connection and hence one has to be physically connected to the machine.

   ```
   sudo stop network-manager
   echo "manual" | sudo tee /etc/init/network-manager.overri
   ```

2. Use bridge-utils

   ```
   sudo apt-get install bridge-utils
   ```

   For a bridged setup, we are required to assign ip address to the bridged interface. Make the necessary changes to /etc/network/interfaces file to get the ip either through dhcp or my using static ip. For our case, we are going to assign a static ip address 192.168.1.1.

   ```
   sudo vim /etc/network/interfaces
   ```

   ```
   auto lo eth0 xenbr0
   iface lo inet loopback

   iface eth0 inet manual
   ```

```
allow-hotplug xenbr0
iface xenbr0 inet static
 address 192.168.1.111
 netmask 255.255.255.0
 gateway 192.168.1.1
 bridge-ports eth0
```

The above configuration will create a linux bridge named xenbr0 which will be connecting to interface eth0. eth0 will have the assigned ip address.

NOTE: Make sure that you are in fact connected to the router using eth0 (or the interface you have in your file) case you have multiple ethernet ports.

Setting up SSH on the server

Use apt-get to install SSH server.

```
sudo apt-get install openssh-server
```

Now restart the ssh server using

```
sudo /etc/init.d/ssh restart
```

In case you don't have a ssh key on your system, use the following command to generate one:

```
ssh-keygen -t rsa -C "name@server_ip"
```

Following the instructions or keep pressing enter to save the key to the default settings.

To reconnect to a system you have previously been connected to and now has a new identification, use

```
ssh-keygen -R name@server_ip
```

This will delete the old hash associated with the server ip address and create a new one.

Resizing the root logical volume

When you use the LVS option during installation, two partitions are created: root and swap_1. To see how much physical memory is allocated to each partition, type

```
sudo lvs
```

You will notice there is nearly no free space available in these two logical volumes. So, you will have to free up space to create new logical volumes. Reducing the size of swap_1 can be done in a single line. We do it before installing the VMs to avoid any data loss that might occur to VMs or files residing in this logical volume in case they are installed before resizing.

```
sudo lvresize -L -120G ubuntu-vg/swap_1
```

This will reduce the size of swap_1 by 120 GB. IN case you want to expand the logical volume(lv), drop the - sign.

The tricky part however is shrinking the root lv. Since, it it where your file system is residing, it is bad idea to use lvresize directly on it. We need to shrink the filesystem before we can safely shrink the logical volume.

```
sudo vgchange -a y
sudo e2fsck -f /dev/ubuntu-vg/root
sudo resize2fs /dev/ubuntu-vg/root 40G         //reduce file
sudo lvreduce -L 50G /dev/ubuntu-vg/root      //the root lv
```

Check if everything is working fine by rebooting the system.

Creating Virtual Machines

For this part we will need physical and logical volume. As the name suggests, physical volume is a partition on the hard disk or hard disk itself. Volume group is a aggregation of physical volumes combined to form one large "virtual physical" volume. Logical volumes are the "pseudo/virtual/logical" partition residing in volume group.Read more about it here

As we have already partitioned the disk using LVM,it has already created a volume group for us. To list the volume groups, use the following command.

```
sudo vgs
```

For each of the virtual machine installations, we are going to create one logical volume.

```
sudo lvcreate -L 10G -n vm-1 /dev/<volume-group>
```

This will create a logical volume named vm-1 of size 10GB living in volume group name you specify.

To list the logical volumes, use

```
sudo lvs
```

Installing a HVM guest VM

Download the guest operating system using wget. I am using Ubuntu 14.04 as the guest OS.

```
sudo wget http://releases/ubuntu.com/14.04/ubuntu-14.04.4-de
```

create a guest config file /etc/xen/vm-1.cfg

```
builder = "hvm"
name = "vm-1"
memory = "2048"
vcpus = 2
vif = ['']
disk = ['phy:/dev/<VG>/vm-1,hda,w','file:/root/ubuntu-14.04.
vnc = 1
boot="dc"
```

The vnc option specifies if the file should use vnc. vnc. For vnc=1, an external vncviewer is connected. The server will listen on ADDR (default 127.0.0.1) on port N+5900. N defaults to the domain id. If vncunused=1, the server will try to find an arbitrary unused port above 5900.

Create the virtual machine using

```
sudo xl create /etc/xen/vm-1.cfg
```

To view the VM, you can use either vncviewer or xtightvncviewer. xtightvncviewer is a lot faster. You will have to install vncviewer/xtightvncviewer using apt-get.

```
sudo apt-get install vncviewer
```

```
vncviewer localhost:0
```

or

```
xtightvncviewer localhost:0
```

You might have to install some java dependencies on the server to use vnc. Also, if the port 0 does not work, try using a different port number.

We have not set any password for vnc in the config file, so click enter when prompted for passcode.

Install the VM you would install on your system. You can either an LVM o non LVM method for disk partition. Once, the OS has been installed, when prompted to reboot, hit enter. You will notice that the system fails to shut down. This kind of behaviour is expected.

Shut down the vnc window. Go to the terminal of your server, and destroy the VM.

```
sudo xl destroy vm-1
```

Now go back to your VM config file /etc/xen/vm-1.cfg and commented out and add lines as shown below:

```
builder = "hvm"
name = "vm-1"
memory = "1024"
vcpus = 2
vif = ['']
#disk = ['phy:/dev/<VG>/vm-1,hda,w','file:/root/ubuntu-14.04
disk = ['phy:/dev/<VG>/vm-1,hda,w']
vnc = 1
boot="c"
#boot="dc"
```

Now start the VM again and your VM is ready.

```
sudo xl create /etc/xen/vm-1.cfg
```