**Gowamr1**

**Virtualisation IN720 2018**

**Assignment 3: Xen**



Mark Gowans

Table of Contents

[Set up a volume for a guest image 1](#_Toc529986807)

[Obtain files necessary for running a Linux installer 2](#_Toc529986808)

[Download Kernels. 2](#_Toc529986809)

[Make Config File 3](#_Toc529986810)

[Boot a guest running the installer and carry out the installation 5](#_Toc529986811)

[Perform post installation finalisation of the guest image 8](#_Toc529986812)

[Modify Config 8](#_Toc529986813)

[Restart VM 9](#_Toc529986814)

[Prepare volumes for cloned guests 10](#_Toc529986815)

[Stop VM 10](#_Toc529986816)

[Make New Volumes 11](#_Toc529986817)

[Write Xen configuration files for the cloned guests 11](#_Toc529986818)

[Create the guest domains 13](#_Toc529986819)

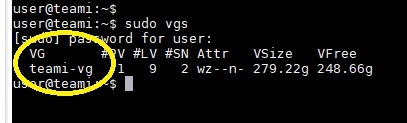
[Basic functionality of the running guests 13](#_Toc529986820)

# Set up a volume for a guest image

First you want to find the name of our volume group.

Enter the command

vgs



then create a new volume for our guest

Enter the command

sudo lvcreate -L 10G -n guest\_volume /dev/<name of volume group (VG) from last command>

This creates a volume named guest\_volume

# Obtain files necessary for running a Linux installer

We need to download the netboot installer for Ubuntu 14.04

Make a directory to hold the downloaded files

mkdir -p /var/lib/xen/images/ubuntu-netboot/trusty

then make that new directory the working directory

cd /var/lib/xen/images/ubuntu-netboot/trusty

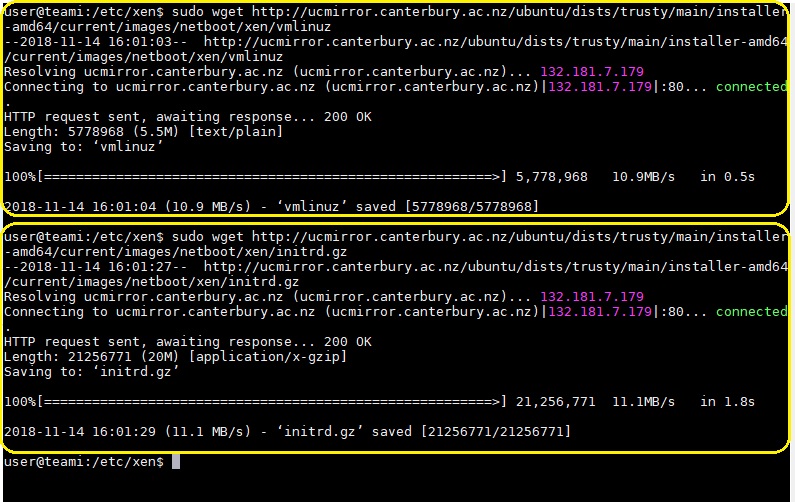
## Download Kernels.

First one (remember this command is just one line).

After the command is entered, something will download.

wget http://ucmirror.canterbury.ac.nz/ubuntu/dists/trusty/main/installer-amd64/current/images/netboot/xen/vmlinuz

Second (again, one command and there will be a download)

wget <http://ucmirror.canterbury.ac.nz/ubuntu/dists/trusty/main/installer-amd64/current/images/netboot/xen/initrd.gz>

## Make Config File

We now need to make a config file for Xen to inform it how our VM should be set up. We can use files already made from the Xen installation

Change current directory to /etc/xen

cd /etc/xen

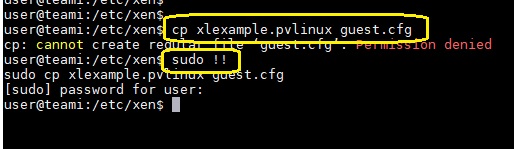
then copy a configuration file and rename the copy something that relates to our guest name.

sudo cp xlexample.pvlinux guest.cfg

If you find a command doesn’t work because of Permission denied, it is because sudo is needed in front of it. Just type

sudo !!

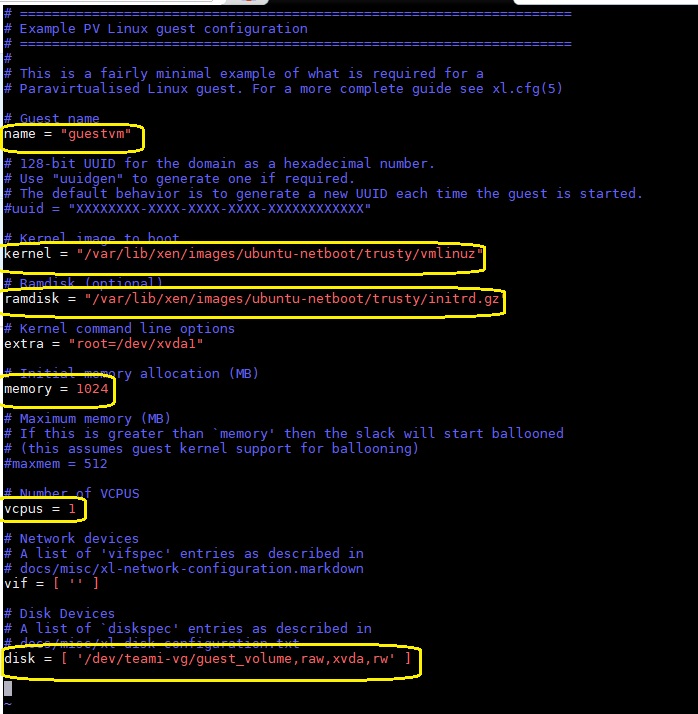
It reruns the last command with sudo at the front



Now we edit the file so Xen can run our VM

sudo vim guest.cfg

Press i to insert text.

* Set name to guestvm
* Set kernel to /var/lib/xen/images/ubuntu-netboot/trusty/vmlinuz (note that this kernel is from the file we downloaded earlier)
* Set ramdisk to /var/lib/xen/images/ubuntu-netboot/trusty/initrd.gz (also from the file we downloaded earlier)
* Set memory to 1024
* Set vcpus to 1
* Set disk to /dev/<name of volume group>/guest\_volume,raw,xvda,rw

Once these modifications have been done, press ESC, then :wq. This saves the changes and closes Vim.

# Boot a guest running the installer and carry out the installation

Now that our config file is set up, we need to run the VM.

sudo xl create -c /etc/xen/guest.cfg

Notice that the second part of that command is pointing to the config file we just copied and edited.

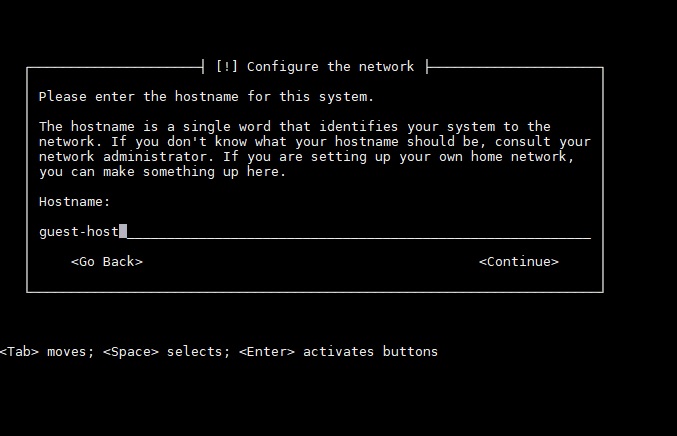
This creates our guest and opens up to its console.

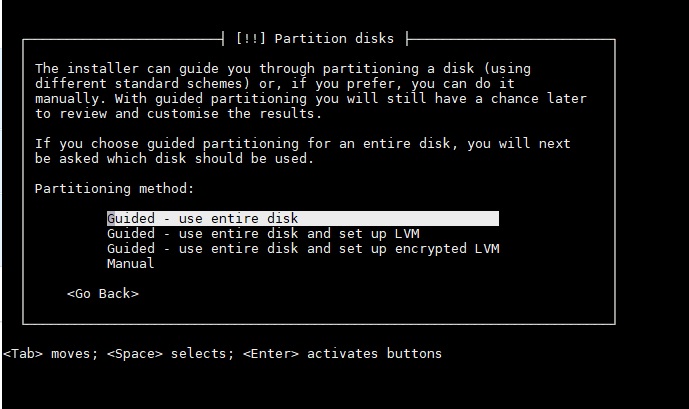
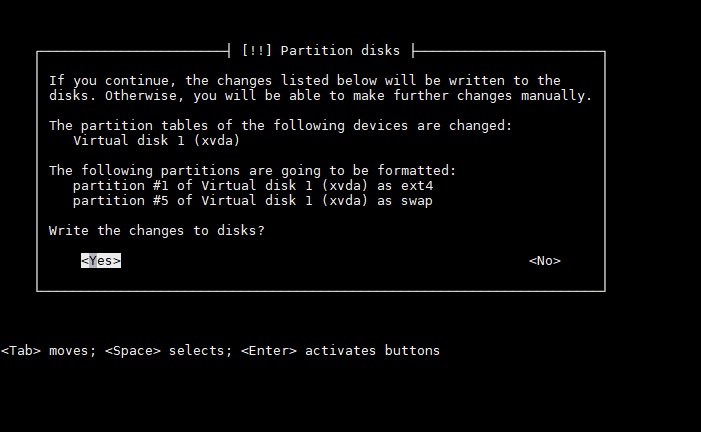
From this point you will see a normal Ubuntu installation dialogue. This is pretty standard, keep everything to the default. Country set as New Zealand, detect keyboard, etc.

Except User and Password. I suggest *user* and *P@ssw0rd* for User and Password. It is up to you but make sure you remember what you have set.

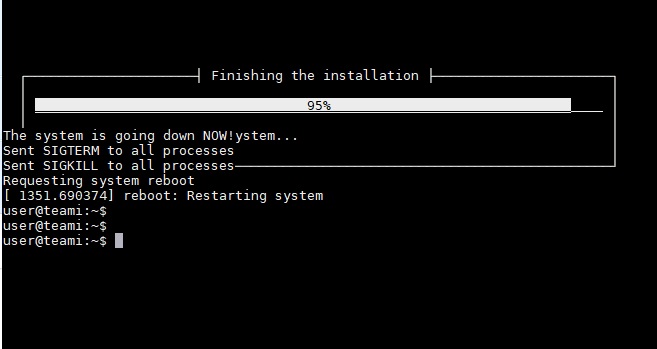
There are a couple of other things you will have to set:

Hostname – guest-host (note: you cannot have ‘\_’ in the name)



Partitions – use the entire disk

Once this has finished, the installer will reboot and you will be back in the Dom0 console.



# Perform post installation finalisation of the guest image

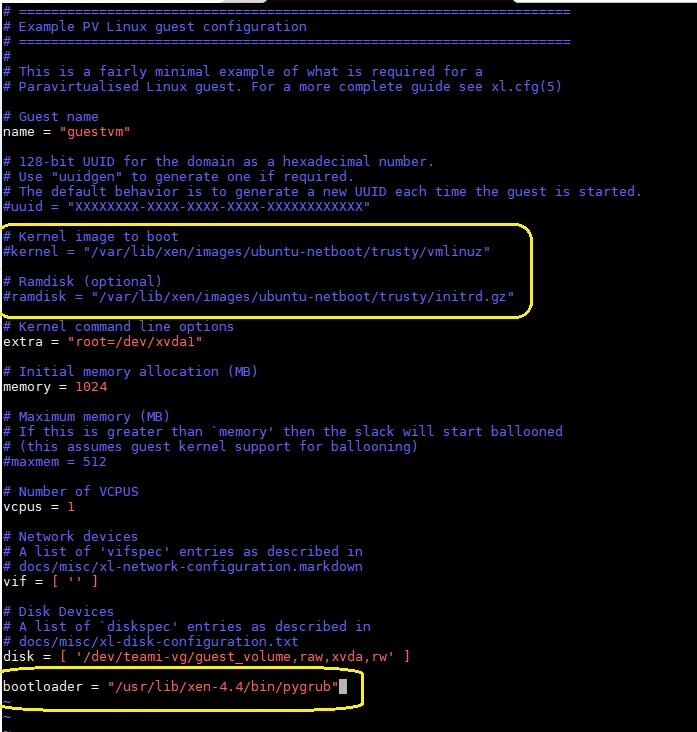
## Modify Config

Before we can use our guest we need to modify our config file again.

sudo vim /etc/xen/guest.cgf

Comment out, using #, the kernel and ramdisk lines (the ones we edited in the last step).

At the bottom of the file, add this line

bootloader = “/usr/lib/xen-4.4/bin/pygrub”

Press ESC, then :wq.

We have just told our VM to stop using the installer’s kernel and start using the Xen’s pygrub bootloader.

## Restart VM

Now we need to stop our VM and start it again.

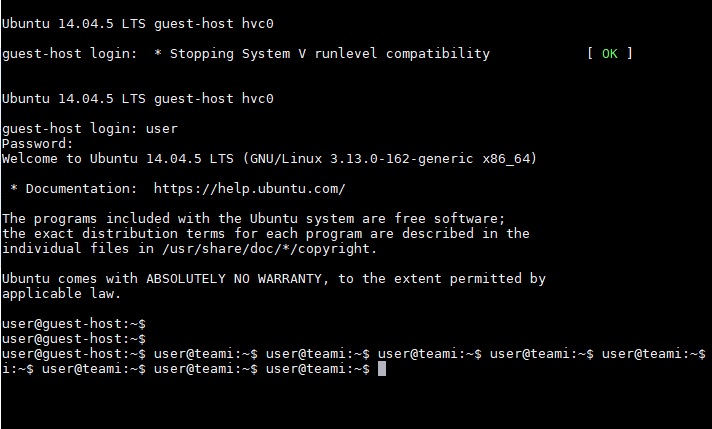
sudo xl shutdown guestvm

sudo xl create -c /etc/xen/guest.cfg

The screen may stop displaying text after about 30 seconds. Just press Enter a few times and the command prompt should show up.

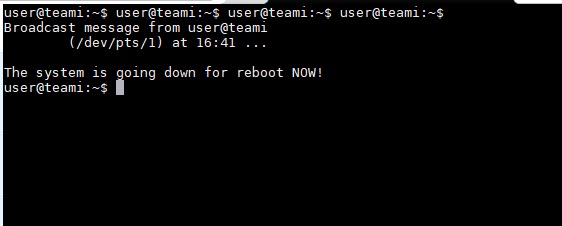
This will boot in to our new VM. Log in with the credentials you set during the installation.

If you want to leave the VM, just press and hold ctrl and ] although this may cause your command prompt to go weird.



You can still type but nothing shows. Just type

sudo reboot



And your system will reboot. It may take a couple of minutes after the boot.

Make sure the VM is all up to date.

sudo apt update

sudo adt dist-upgrade

Now your VM is all up to date! This is important because all the clones will be copied from this VM. It will save you individually updating all the clones after making them.

# Prepare volumes for cloned guests

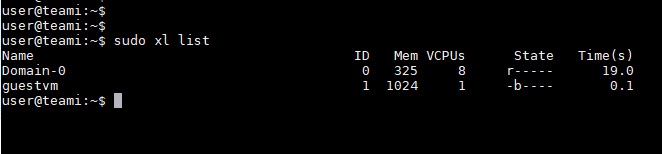
It can be rather laborious to do the past few steps every time we want to make a new guest. We will now explore a way to use what we have already done to make new guests each time.

## Stop VM

Make sure that the guest me made earlier is not running.

We will check this by showing a list of all VMs currently running.

sudo xl list



If you see guestvm in that list, you should remove it by stopping it

sudo xl shutdown guestvm

Check the VMs list again

sudo xl list

You should only see Dom0 in the list.

## Make New Volumes

Now we are ready to make new volumes for our clones

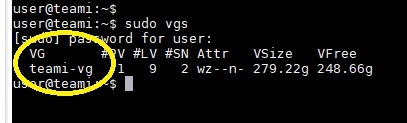
First we will take a snapshot of the system

modprobe dm-snapshot

Now we will look back at the first chapter of this tutorial and use very similar commands.

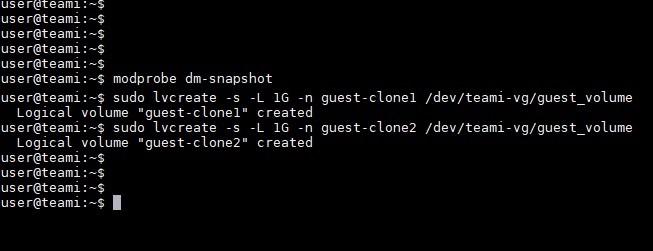
You will need to remember your volume group.

sudo vgs



sudo lvcreate -s -L 1G -n clone1 /dev/< name of volume group>/guest\_volume

sudo lvcreate -s -L 1G -n clone2 /dev/<name of volume group>/guest\_volume



We have just created two volumes based on our original volume. The new volumes are only 1GB each because they each use the files from the original VM. The new volumes only need to store the difference.

# Write Xen configuration files for the cloned guests

We need to have config files for both the new guests. We can copy the config file we made earlier and make some small modifications.

Make sure you are in the /etc/xen directory

cd /etc/xen

sudo cp guest.cfg guest-clone1.cfg

sudo cp guest.cfg guest-clone2.cfg

For each of the new config files we need to change some things.

The name in each config file needs to be unique. Example for clone1.cfg:

Set name – guest-clone1

Example for clone2.cfg

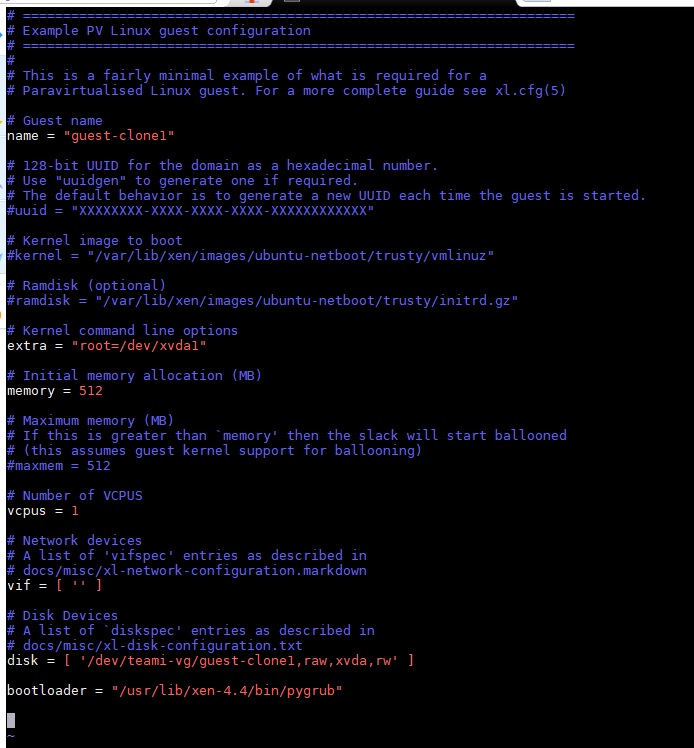
Set name – guest-clone2

We also need to change the starting memory for both config files

Set memory – 512

Set disk to the path of the clone volume /dev/<volume name>/guest-clone1,raw,xvda,rw

Here is an example of guest-clone1.cfg



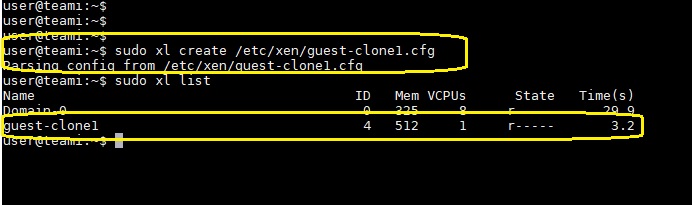
# Create the guest domains

We’re all set now to run our guest clones.

sudo xl create /etc/xen/guest-clone1.cfg

sudo xl create /etc/xen/guest-clone2.cfg

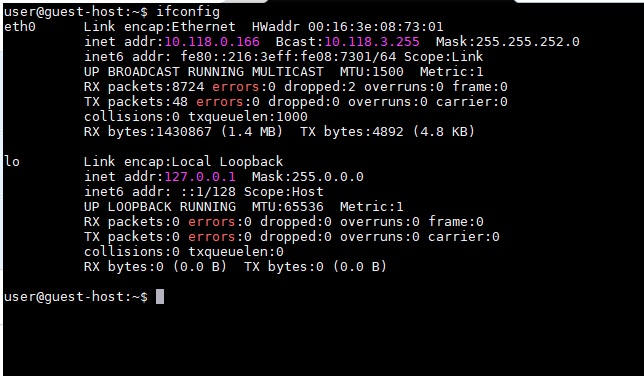
When we omit the ‘-c’ from the xl create command, it makes the VM but doesn’t boot to the console.



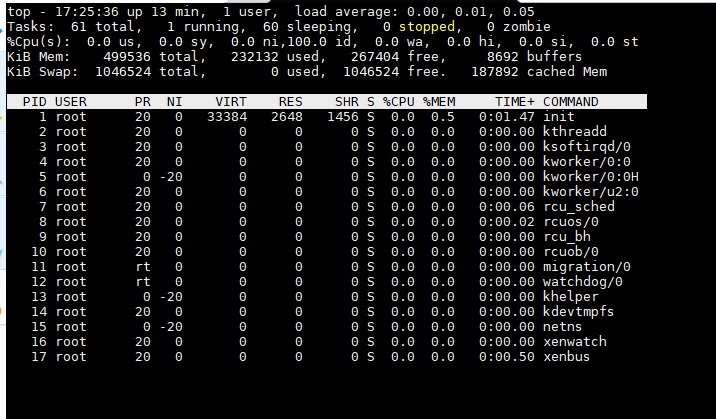
# Basic functionality of the running guests

Some basic commands to check that the VM clone is working like a real machine:

ifconfig – This command shows NIC information (same as ipconfig for Windows)



top – Shows running processes (same as Task Manager for Windows)



ping – used to check if devices can talk to other devices on networks. In this example, I pinged Google, which is an external address. This shows that the VM clone has internet access

