

BSc in Computer Engineering
CMP4103
Computer Systems and Network Security

Lecture 5(a)
Xen Hypervisor lab

Eng Diarmuid O'Briain, CEng, CISSP



Department of Electrical and Computer Engineering,
College of Engineering, Design, Art and Technology,
Makerere University

Copyright © 2017 Diarmuid Ó Briain

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table of Contents

1. GETTING XEN.....	5
2. INSTALL XEN.....	6
2.1 ENABLE VIRTUALISATION SUPPORT IN BIOS.....	6
2.2 INSTALL CITRIX XENSERVER.....	6
2.3 INSTALL XEN ON AN EXISTING SERVER.....	8
2.4 RUNNING XEN CLOUD PLATFORM.....	13
2.5 MANAGING THE XEN CLOUD PLATFORM.....	15
2.6 GRAPHICAL XEN MANAGERS.....	18
3. VIRTUAL MACHINES.....	20
3.2 ACCESSING THE GUI ON A VM.....	25
3.3 ACCESSING THE GUI OVER AN NON-SECURE NETWORK.....	29
3.4 XEN VM SNAPSHOTS.....	31
4. LAB EXERCISE.....	32

This page is intentionally blank

1. Getting Xen

The Xen hypervisor is available as source distribution from <http://www.xen.org>. It is more typical to get pre-built binaries pre-existing on Linux or BSD Unix distributions.

Another way to get Xen is via the XenServer ISO binary distribution that contains the Xen Hypervisor, a pre-packaged and configured Control Domain and the powerful Xen API toolstack and management API.

Installing Xen on a distribution typically requires the following basic steps:

- Install your preferred distribution.
- Install xen package(s) or meta-package.
- Check boot settings.
- Reboot.

After the reboot, your system will run your chosen Linux/Unix distribution as the Control Domain on top of the Xen Hypervisor. For detailed instructions relating your chosen distribution refer to the installation instructions on <http://www.xen.org>.

1.1.1 XCP on Debian based Linux

To install XCP on a Debian or Ubuntu based Linux distribution use the command:

```
debian ~ # apt-get install xcp-xapi
```

1.1.2 XCP on Fedora based Linux

To install XCP on a Fedora, Redhat based Linux distribution use the command:

```
fedora ~ # yum install xcp-xapi
```

1.1.3 XenServer

For the purposes of this course we will use the XenServer CentOS based ISO installation. This is a turnkey virtualisation solution that provides shrink wrapped virtualisation and cloud computing. XCP includes the Xen Hypervisor, the enterprise ready Xen API toolstack and with functionality such as :

- Capability to manage pools of host systems.
- Support for advanced storage repositories.
- Support for multi-tenancy.
- Support for SLA guarantees.
- Detailed metrics for consumption based charging and more.
- Pre-integrated network and disk functionality such as Open vSwitch.

2. Install Xen

2.1 Enable virtualisation support in BIOS

To support HVM guests, virtualisation extensions need to be enabled in the BIOS. If your system does not support such extensions Xen cannot virtualise unmodified operating systems, however PV VMs can be created.

In the BIOS the Virtualise option appears under “Advanced Chipset Features” as one of the following:

- Enable Virtualisation Technology.
- Enable Intel VT.
- Vanderpool Technology.

2.2 Install Citrix XenServer

Download the XenServer ISO binaries from the <http://www.xenserver.org> download page.

- Boot/install it on a fresh server.
- Configure the network from the very brief administrative interface that stays on the screen after installation.

```
Citrix | XenServer
```

```
Welcome to XenServer
```

```
To install or Upgrade press the <ENTER> Key. <ENTER>
```

```
[F1-Standard] [F2-Advanced] [F3-XCP Mode] Press F3
```

```
To install or upgrade, type install-xcp <ENTER>.
```

```
boot: install-xcp
```

```
Please select the keymap you would like to use:
```

```
[qwerty] uk --> OK
```

```
Installing XenServer will erase all data on the disks selected.
```

```
--> OK
```

```
CITRIX(R) LICENSE AGREEMENT
```

```
Accept EULA
```

```
System Hardware message --> OK
```

```
Virtual Machine Storage --> sda - XXX GB Drive
```

```
--> Enable thin provisioning

Select Installation Source --> Local media --> OK

Supplemental Packs --> No

Verify installation source --> Verify installation source

Verification Successful --> OK

Set Password --> <Password> Confirm --> <Password> --> OK

Networking --> eth0 --> OK

    Static Configuration --> IP Address: --> <IP address>
                                --> Subnet mask: --> <subnet mask>
                                --> Gateway: --> <IP address>

--> OK

Hostname & DNS --> Hostname --> <Hostname>
                                --> DNS Server 1 --> <IP address>
                                --> DNS Server 2 --> <IP address>
                                --> DNS Server 3 --> <IP address>
                                --> OK

Select Time Zone --> Europe --> Dublin --> OK

System Time --> Using NTP
                                --> 0.pool.ntp.org
                                --> 1.pool.ntp.org
                                --> 2.pool.ntp.org
                                --> OK

Confirm Installation --> Install XenServer
```

2.3 Install Xen on an existing Server

Sometimes it may be necessary to install Xen on a Debian or Fedora based server. Perhaps the server architecture is 32 bit, or the company standard is Debian, Ubuntu or Fedora and therefore it is decided that Dom0 should be that OS and not CentOS. Here is an example of the setup for the Debian OS.

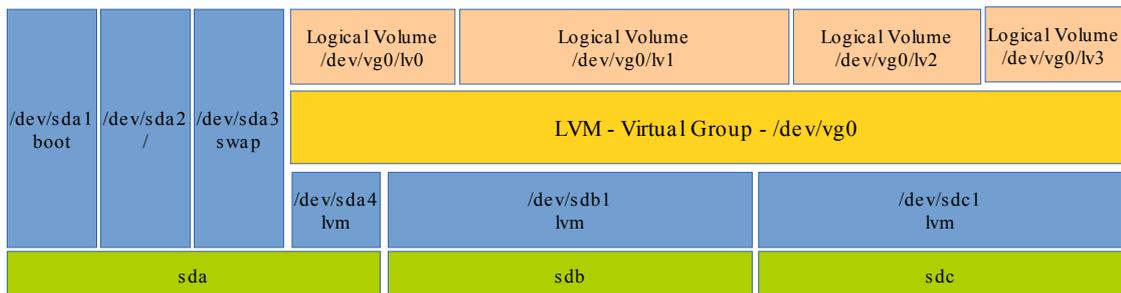
Install Debian on the Server. During the installation only install the basic system and the SSH Server by deselecting other options that appear by default like the Desktop Environment. Build with four partitions as follows:

```
sda1 - boot          200MB          (ext3)
sda2 - /             5GB           (ext3)
sda3 - swap
sda4 - LVM           (LVM) - Xen VMs will go here.
```

sda3 partition for **swap** space is determined by the amount of RAM on the Server. As a rule of thumb use as much **swap** as you have system memory.

sda4 partition is setup as a Logical Volume Manager (LVM) partition. This means that in the future it is possible to add additional drives under this logical partition.

2.3.1 Logical Volume Manager (LVM)



LVM is a logical volume manager for Linux. It manages disk drives and similar mass-storage devices. **Volume** refers to a disk drive or partition of a disk drive. It was written in 1998 by Heinz Mauelshagen, who based its design on that of the LVM in HP-UX.

LVM can be considered as a thin software layer on top of the hard disks and partitions, which creates an abstraction of continuity and ease-of-use for managing hard drive replacement, re-partitioning, and backup.

LVM is used for the following purposes:

- Managing large hard disk farms by allowing disks to be added and replaced without downtimes and services disruption, in combination with hot swapping.
- On small systems LVM allows file systems to be easily resized as needed.
- Performing consistent backups by taking snapshots of the logical volumes.
- Creating single logical volumes of multiple physical volumes or entire hard disks, allowing for dynamic volume resizing.

2.3.2 Configuring LVM

Initialise the LVM partition as an LVM physical volume for later use as part of an LVM logical volume.

```
dobriain@OB-Xen~$ sudo pvcreate /dev/sda4
```

Now create a volume group from the physical volume just initialised. It is possible to add further volumes to this volume group later.

The following command creates a volume group named `vg0` that contains physical volume `/dev/sda4` (more can be added to `vg0` later).

```
dobriain@OB-Xen~$ sudo vgcreate vg0 /dev/sda4
```

Within the volume group `vg0` create a logical volume that will be used by VMs later. Here we create a logical volumes of 20 GB called `lv0`.

```
dobriain@OB-Xen~$ sudo lvcreate -L 20G -n lv0 vg0
```

```
Logical volume "lv0" created
```

You can verify the LVM with the following LVM display commands.

```
dobriain@OB-Xen~$ sudo pvdisplay
--- Physical volume ---
PV Name           /dev/sda4
VG Name           vg0
PV Size           134.89 GiB / not usable 3.00 MiB
Allocatable       yes
PE Size           4.00 MiB
Total PE          34532
Free PE           34532
Allocated PE      0
PV UUID           rrL4Q5-RWIZ-wM5a-N7QL-M3to-BlzO-0wK0yF
```

```
dobriain@OB-Xen~$ sudo vgdisplay
--- Volume group ---
VG Name                vg0
System ID
Format                 lvm2
Metadata Areas        1
Metadata Sequence No  1
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                0
Open LV                0
Max PV                 0
Cur PV                1
Act PV                 1
VG Size                134.89 GiB
PE Size                4.00 MiB
Total PE               34532
Alloc PE / Size       0 / 0
Free PE / Size        34532 / 134.89 GiB
VG UUID                9gYEvb-Lhvy-BBry-vgBU-nqVs-fIeT-e00HaP

dobriain@OB-Xen~$ sudo lvdisplay
--- Logical volume ---
LV Path                /dev/vg0/lv0
LV Name                lv0
VG Name                vg0
LV UUID                irpKXm-pzKb-yvQA-t8Vw-gZ19-Ir9k-vRLL0k
LV Write Access        read/write
LV Creation host, time OB-Xen, 2013-10-27 20:14:19 +0000
LV Status               available
# open                 0
LV Size                20.00 GiB
Current LE              5120
Segments               1
Allocation              inherit
Read ahead sectors     auto
- currently set to    256
Block device           254:4
```

2.3.3 Setup a Linux Bridge for guest networking

Create a virtual switch within dom0 that takes packets from the virtual machines and forwards them onto the physical network so they can see the internet and other machines on your network.

```
dobriain@OB-Xen~$ sudo apt-get install bridge-utils
```

2.3.4 Setup Ethernet interface

Edit the /etc/network/interfaces file as follows and then reboot:

```
dobriain@OB-Xen~$ vi /etc/network/interfaces

# The loopback network interface

auto lo

iface lo inet loopback

# The primary network interface

iface eth0 inet manual

# The Xen Bridge interface

auto xenbr0

iface xenbr0 inet static

    address 75.144.153.179

    netmask 255.255.255.248

    gateway 75.144.153.177

    network 75.144.153.176

    broadcast 75.144.153.183

    bridge_ports eth0

:wq!
```

2.3.5 Install the Xen system

Install the XCP XAPI and all its dependencies, including the Xen Linux kernel and Xen Hypervisor. XCP requires *lspci*, part of the *pciutils* package, otherwise xcp-xapi will fail to start and give you a connection refused error.

```
dobriain@OB-Xen~$ sudo apt-get install pciutils
dobriain@OB-Xen~$ sudo apt-get install xcp-xapi
--> Configuring xcp-networkd
--> XCP networking manager: bridge
--> OK
```

Use apt-get update to get the latest builds.

```
dobriain@OB-Xen~$ sudo apt-get update
```

Set the Xen toolstack variable to get xcp-xapi to start.

```
dobriain@OB-Xen~$ sudo vi /etc/default/xen
TOOLSTACK="xapi"
:wq!
```

2.3.6 Modify Grub to boot XEN

Xen starts before the OS, which will now become Dom0. The GNU GRand Unified Bootloader (GRUB) must be altered to boot Xen before Debian Linux. GRUB configuration is stored in the */boot/grub/grub.cfg* file. It is edited using the scripts in */etc/grub.d/**. To change the order of the boot to move Xen to a higher priority than Debian Linux use the following command:

```
dobriain@OB-Xen~$ sudo dpkg-divert --divert /etc/grub.d/08_linux_xen
--rename /etc/grub.d/20_linux_xen
```

Reinitialise the *grub.cfg* file to take account of the change to the order.

```
dobriain@OB-Xen~$ sudo update-grub
```

2.3.7 Reboot the system

Reboot the system, xen should now boot and Debian Linux will run as Dom0.

```
dobriain@OB-Xen~$ sudo shutdown -r now
```

2.4 Running Xen Cloud Platform

2.4.1 The XCP Daemons

When XCP is running a number of daemons are active. Their start scripts are in */etc/init/d*.

xcp-xapi

XCP XenAPI server, which provides the control plane for XCP hosts and resource pools. To manage the xcp-xapi server, you should use either the XML-RPC based API, the **xe** command line tool, or a GUI such as OpenXenManager.

xcp-squeezed

XCP memory ballooning daemon, it takes requests from the XenAPI server (xapi) and can balloon domains up and down.

xcp-networkd

XCP network configuration daemon.

xcp-fe

XCP fork-and-exec daemon for **xapi**. When **xapi** requires another executable to be run, rather than forking and exec-ing itself, it uses this single-threaded daemon to do it.

xcp-v6d

XCP feature daemon. This daemon communicates with the xcp-xapi server over XML-RPC to enable or disable certain features of XCP.

2.4.2 Confirm that XCP Daemons are running

Check the XCP daemons are running.

```
dobriain@OB-Xen~$ sudo service xcp-xapi status
[ ok ] xapi is running.
dobriain@OB-Xen~$ sudo service xcp-squeezed status
[ ok ] squeezed is running.
dobriain@OB-Xen~$ sudo service xcp-networkd status
[ ok ] xcp-networkd is running.
dobriain@OB-Xen~$ sudo service xcp-fe status
[ ok ] xcp-fe is running.
```

Should there be a problem with one of the daemons, try reconfiguring the installed dpkg package. Here is an example of reconfiguring the *xcp-squeezed* and *xcp-xapi* packages as their daemons were failing to start.

```
dobriain@OB-Xen~$ sudo dpkg-reconfigure xcp-squeezed
```

```
dobriain@OB-Xen~$ sudo dpkg-reconfigure xcp-xapi
```

Check that control domain is up and running.

```
dobriain@OB-Xen~$ sudo xe vm-list
```

```
    uuid (RO): 60f2711d-afc7-5519-e8a7-ae25588032c5
    name-label (RW): Control domain on host: OB-Xen
    power-state (RO): running
```

2.5 Managing the Xen Cloud Platform

2.5.1 XE

List the templates available on the XCP Server.

```
dobriain@OB-Xen~$ sudo xe template-list | less

        uuid (RO): 53f9ae18-7766-d1d5-0f7a-89df13b3f470

        name-label (RW): Windows Server 2008 R2 (64-bit)

        name-description (RW): Clones of this template will automatically
        provision their storage when first booted and then reconfigure
        themselves with the optimal settings for Windows Server 2008 R2 (64-
        bit).

        uuid (RO): 53f9ae18 : 79c844d5-8192-c5fa-77a3-163f64096c15

        name-label (RW): Ubuntu Lucid Lynx 10.04 (32-bit)

        name-description (RW): Template that allows VM installation from
        Xen-aware Debian-based distros. To use this template from the CLI,
        install your VM using vm-install, then set other-config-install-
        repository to the path to your network repository, e.g.
        http://<server>/<path>

~ ~ ~
```

2.5.2 Create a Storage Repositories

Each VM to be created requires a Storage Repository (SR). The SR can come from a number of sources, however for the purpose of this document it will be aligned with the 20 GB logical volume *lv0* created earlier. When the **sr-create** command is run it will return the Universally Unique Identifier (UUID) for the new repository. This may take some time depending on the size of the logical volume.

```
dobriain@OB-Xen~$ sudo xe sr-create type=ext name-label=sr0 device-
config:device=/dev/vg0/lv0

4102b6f8-8656-91ec-eb12-7a8bb1410e72
```

Reviewing SRs

Review the SR list. This will show the newly created SR as well as the pre-built SR for XenServer Tools.

```
dobriain@OB-Xen~$ sudo xe sr-list

          uuid (RO): 1b759dfb-ed6e-73b4-2690-4d56cdaff3f0
      name-label (RW): XenServer Tools
name-description (RW): XenServer Tools ISOs
          host (RO): OB-Xen
          type (RO): iso
      content-type (RO): iso

          uuid (RO): 4102b6f8-8656-91ec-eb12-7a8bb1410e72
      name-label (RW): sr0
name-description (RW):
          host (RO): OB-Xen
          type (RO): ext
      content-type (RO):
```

2.5.3 Delete SRs

You may have recourse to delete an existing SR. This is the process for doing that.

Note: the SR UUID that must be deleted from the `xe sr-list` command as shown above. Lets take **4102b6f8-8656-91ec-eb12-7a8bb1410e72** as an example.

Before destroying the SR the Physical Block Device (PBD) through which hosts access SRs must be unplugged. Running the command below lists the associated PBD.

`xe pbd-list sr-uuid=< SR-UUID>`

```
dobriain@OB-Xen~$ sudo xe pbd-list sr-uuid=4102b6f8-8656-91ec-eb12-7a8bb1410e72

        uuid (RO): 7062087a-4019-47c2-b06e-910382c25136
    host-uuid (RO): 253be49e-cc08-6604-517c-8ab3721627d2
        sr-uuid (RO): 4102b6f8-8656-91ec-eb12-7a8bb1410e72
    device-config (MRO): device: /dev/vg0/lv3
    currently-attached (RO): true
```

Now unplug and destroy the PBD UUID (the first UUID in the previous command).

`xe pbd-unplug uuid=<PBD-UUID>`

`xe pbd-destroy uuid=<PBD-UUID>`

```
dobriain@OB-Xen~$ sudo xe pbd-unplug uuid=7062087a-4019-47c2-b06e-910382c25136
```

```
dobriain@OB-Xen~$ sudo xe pbd-destroy uuid=7062087a-4019-47c2-b06e-910382c25136
```

Now that the PBD is destroyed it is possible to remove the SR with the command.

`xe sr-forget uuid=<SR-UUID>`

```
dobriain@OB-Xen~$ sudo xe sr-forget uuid=4102b6f8-8656-91ec-eb12-7a8bb1410e72
```

2.6 Graphical Xen Managers

2.6.1 XenCenter

XenServer has a client software called XenCenter, this can be downloaded from <http://www.xenserver.org> download page. It provides a GUI-based administration console for remote management of XenServer XCP. It is only available for Microsoft Windows. To install download the latest MSI installer file. Double-click on it and follow the installation instructions.



2.6.2 OpenXenManager

For Linux there is the OpenXenManager which is an open-source multi-platform clone of XenCenter. Here are the install instructions using Debian based **apt-get** tool. You can similarly use **yum** on Fedora based systems.

Before installing it, install all necessary prerequisites first.

```
dobriain@OB-Xen~$ sudo apt-get install subversion
```

```
dobriain@OB-Xen~$ sudo apt-get install python-glade2
```

```
dobriain@OB-Xen~$ sudo apt-get install python-gtk-vnc
```

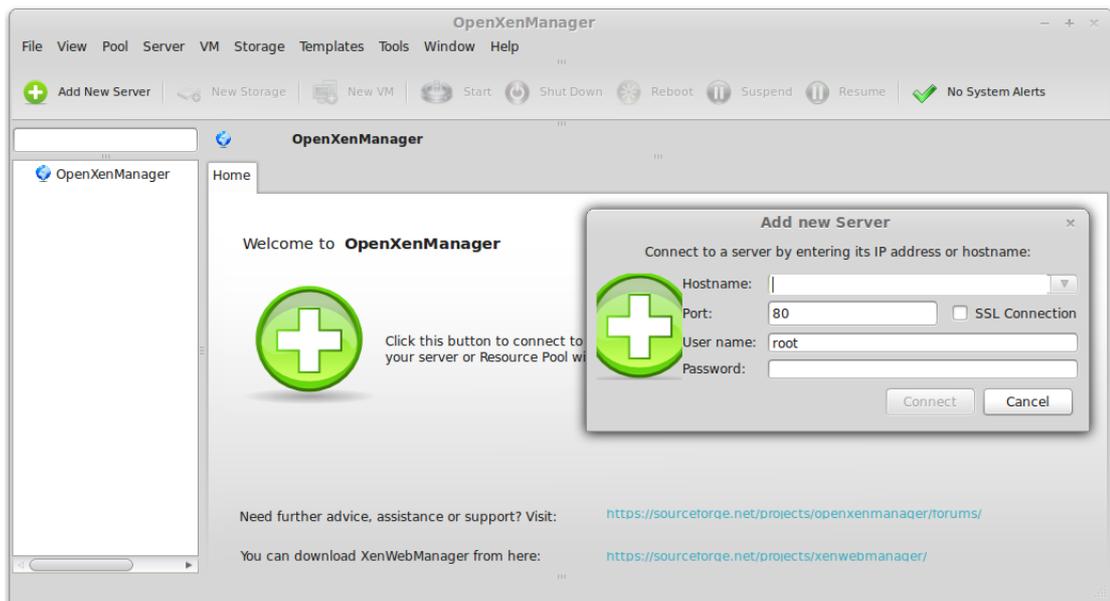
Install the OpenXenManager.

```
dobriain@OB-Xen~$ sudo apt-get install openxenmanager
```

To launch OpenXenManager, just type the following:

```
dobriain@OB-Xen~$ openxenmanager
```

The screenshot of OpenXenManager looks as follows.



3. Virtual Machines

3.1.1 Host memory

Before creating VMs it is important to know how much host memory is available. To determine this use the command:

```
dobriain@OB-Xen~$ sudo xe host-compute-free-memory  
367710208
```

It is important that this is not exceeded. XCP will never allow the sum of guests' dynamic memory minimum values to exceed the total host memory, after virtualisation overhead has been taken into account.

XCP requires that every VM has static and dynamic memory that satisfy the following:

$$0 \leq \textit{static min} \leq \textit{dynamic min} \leq \textit{dynamic max} \leq \textit{static max}$$

Typically for a Windows 7, 32 bit guest for example, a dynamic minimum of 1 GB and a maximum of 4 GB, while a Windows 7, 64 bit guest would require a dynamic minimum of 2 GB and a maximum of 32 GB. Linux can operate on much lower memory limits with 128 or 512 MB for the dynamic minimum and 32 GB for the dynamic maximum.

3.1.2 Create a VM

The XE commands involved in the creation of a VM are as follows.

General VM install information.

```
xe vm-install  
    template=<OS Template>  
    new-name-label=<VM unique name>  
    sr-name-label=<Storage repository label>
```

VM memory parameters.

```
xe vm-param-set uuid=<VM-UUID>  
    memory-static-max=<Memory>  
    memory-dynamic-min=<Memory>  
    memory-dynamic-max=<Memory>  
    memory-static-min=<Memory>
```

VM installation repository.

```
xe vm-param-set uuid=<VM-UUID>  
    other-config:install-repository=<OS Loc>
```

Kernel arguments.

```
xe vm-param-set uuid=<VM-UUID>
                    PV-args=auto-install/enable=true
                    hostname=<Hostname>
                    domain=<Local domain name>
```

Associate the network interface with the xenbr0 bridge interface created after installing the OS.

```
xe network-list bridge=xenbr0 -minimal
```

This returns a UUID, the Network UUID (Net-UUID).

```
xe vif-create
    vm-uuid=<VM-UUID>
    network-uuid=<Net-UUID>
    mac=random device=0
```

Now start the newly created VM.

```
xe vm-start uuid=<VM-UUID>
```

or

```
xe vm-start vm=UbuntuVM-1
```

3.1.3 VM creation script

Change the variables as necessary and the script does the rest.

```
#!/bin/bash
set -e          # Abort script at first error
set -x          # Print to stdout before executing and expand (Like -v but more
output)

# // Xen XCP VM Creation script - Diarmuid O'Briain //
# // Change the variables section as necessary //

# // Variables //
SRLABEL='sr0'          # Storage Repository
TEMPLATE='Ubuntu Lucid Lynx 10.04 (32-bit)' # 'xe template-list | less'
VMLABEL='UbuntuVM-1'  # VM Label and Hostname
MEMORY='512MiB'      # i.e. 512 MB
REPOSITORY='http://ie.archive.ubuntu.com/ubuntu/' # Online image
DOMAIN='obriain.com' # Domain

# ~~~~~ #

# // Install template //
VM=`xe vm-install template="$TEMPLATE" new-name-label="$VMLABEL" sr-name-
label="$SRLABEL"`

# // VM Parameters and Memory //
`xe vm-param-set uuid="$VM" memory-static-max="$MEMORY"`
`xe vm-param-set uuid="$VM" memory-dynamic-max="$MEMORY"`
`xe vm-param-set uuid="$VM" memory-dynamic-min="$MEMORY"`
`xe vm-param-set uuid="$VM" memory-static-min="$MEMORY"`
`xe vm-param-set uuid="$VM" other-config:install-repository="$REPOSITORY"`
`xe vm-param-set uuid="$VM" PV-args="auto-install/enable=true
hostname=$VMLABEL domain=$DOMAIN"`

# // Network interface //
NETWORK=`xe network-list bridge=xenbr0 --minimal`
VIF=`xe vif-create vm-uuid="$VM" network-uuid="$NETWORK" mac=random device=0`

# // Now start new VM //
set +x

echo ''
echo "Press any key to start Virtual Machine $VMLABEL"
echo ''

read -n 1
echo "Starting Virtual Machine $VMLABEL now, please wait..."
xe vm-start uuid="$VM"
```

3.1.4 List the installed VMs.

```
dobriain@OB-Xen~$ sudo xe vm-list
      uuid (RO): 60f2711d-afc7-5519-e8a7-ae25588032c5
      name-label (RW): Control domain on host: OB-Xen
      power-state (RO): running

      uuid (RO): 47b01617-b831-5c83-d4cb-db47f6a9c04e
      name-label (RW): Ubuntu VM 1
      power-state (RO): running
```

3.1.5 Removing a VM

If a VM has to be removed, it must be firstly shutdown. This can be forced from the command-line and then the VM can be destroyed as follows:

```
xe vm-destroy uuid=<VM-Name label>
```

```
xe vm-destroy uuid=<VM-UUID>
```

```
dobriain@OB-Xen~$ sudo xe vm-shutdown vm='UbuntuVM-1' force=true
```

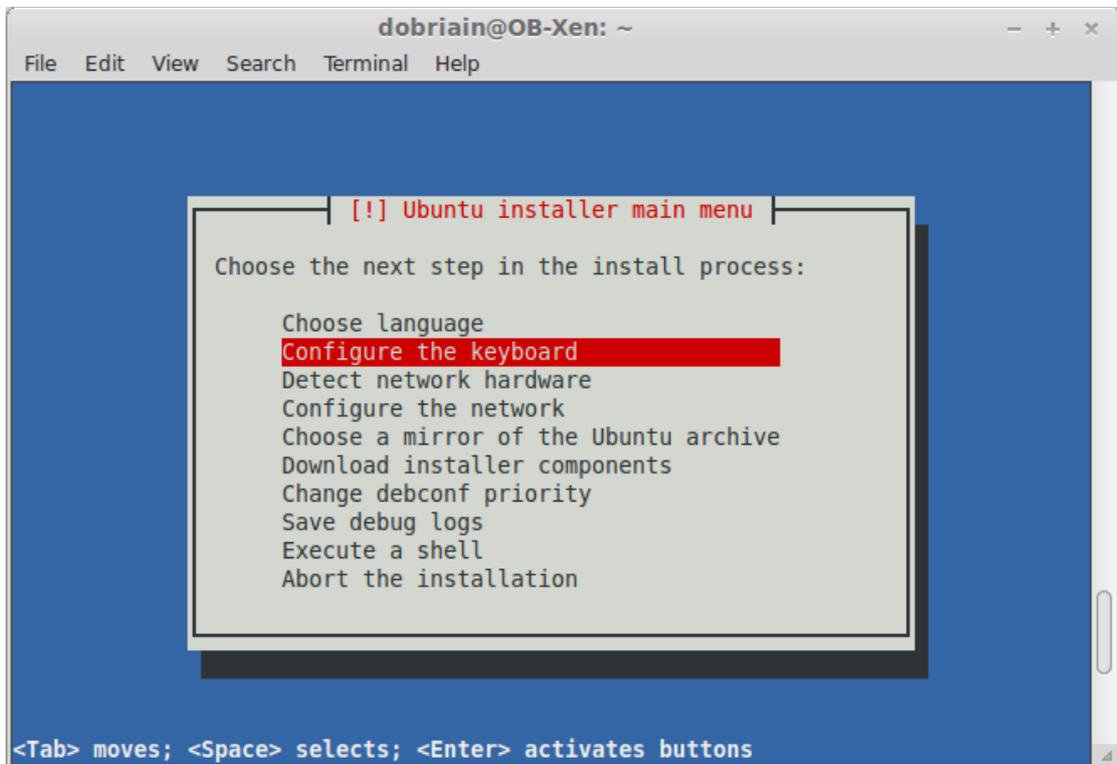
```
dobriain@OB-Xen~$ sudo xe vm-destroy uuid=47b01617-b831-5c83-d4cb-db47f6a9c04e
```

3.1.6 Connect to the Virtual Machine Console.

```
dobriain@OB-Xen~$ sudo xe console uuid=47b01617-b831-5c83-d4cb-db47f6a9c04e
```

Connecting to console on VM Ubuntu VM 1. Press Ctrl + ']' to quit.

The terminal now presents the Ubuntu installer menu.



As you go through the install, be prepared for waits between sections. When asked what to install, make sure to install at least the base system plus OpenSSH server. Later you can add further packages like GUI, LAMP Servers etc...

3.1.7 Xen Server tools

On the guest OS on the VM install the Xen Server tools. Here is an example on Debian or Ubuntu Linux.

```
dobriain@UbuntuVM-1~$ sudo apt-get install xen-tools
```

3.2 Accessing the GUI on a VM

Now that the VM is installed it maybe necessary to access the GUI on it. The simplest way to do that is to use Virtual Network Computing (VNC). VNC is a remote desktop sharing system that will allow connections to the GUI hosted on your VM.

TightVNC is a free remote control software package. With TightVNC, you can see the desktop of a remote machine and control it with your local mouse and keyboard.

TightVNC is:

- Free for both personal and commercial usage, with full source code available.
- Cross-platform, available for Linux, Unix and Microsoft Windows, with Java client included.
- Compatible with standard VNC software, conforming to Remote Frame Buffer (RFB) protocol specifications.

3.2.1 Install a GUI on the VM

The following command installs the Gnome 3/Unity desktop.

```
dobriain@UbuntuVM-1~$ sudo apt-get install gnome-desktop-environment
```

or for a slimmed down version without LibreOffice or some other programs use the ***–no-install-recommends*** switch.

With Linux there are a host of different Desktop Environments or Window Managers, to use like Gnome3, KDE, XFCE, Cinnamon, Unity, Mate, LXDE, Xmonad, Enlightenment, Window Maker Fluxbox, AmiWM, Sawfish, IceWM and TWM to name a few.

Install required fonts:

```
dobriain@UbuntuVM-1~$ sudo apt-get install xfonts-100dpi
```

```
dobriain@UbuntuVM-1~$ sudo apt-get install xfonts-100dpi-transcoded
```

```
dobriain@UbuntuVM-1~$ sudo apt-get install xfonts-75dpi
```

```
dobriain@UbuntuVM-1~$ sudo apt-get install xfonts-75dpi-transcoded
```

```
dobriain@UbuntuVM-1~$ sudo apt-get install xfonts-base
```

3.2.2 Install VNC

Before beginning, ensure that apt-get packages are upgraded with the following commands that update the package list and fetch and install the new versions:

```
dobriain@UbuntuVM-1~$ sudo apt-get update
```

```
dobriain@UbuntuVM-1~$ sudo apt-get upgrade
```

Install a TightVNC Server on the VM. Create a password and TightVNC server should create some configuration files and start up.

```
dobriain@UbuntuVM-1~$ sudo apt-get install tightvncserver
```

Start the TightVNC Server for the first time, it will copy config files and prompt you for a password (Note: no sudo, otherwise the server will present the login of the root user on the guest OS):

```
dobriain@UbuntuVM-1~$ tightvncserver :1
```

You will require a password to access your desktops.

```
Password: <Password>
```

```
Verify: <Password>
```

```
Would you like to enter a view-only password (y/n)? n
```

Stop the TightVNC Server, so it can be configured:

```
dobriain@UbuntuVM-1~$ tightvncserver -kill :1
```

3.2.3 X applications configuration file

The VNC *xstartup* file needs editing. This change sets the TightVNC Server to render the full Gnome3/Unity desktop. Copy the *xstartup* file to *xstartup.orig* and edit the *xstartup* file to replace its content with the following:

```
dobriain@UbuntuVM-1~$ cp ~/.vnc/xstartup ~/.vnc/xstartup.orig
dobriain@UbuntuVM-1~$ vi ~/.vnc/xstartup

#!/bin/sh

xrdb $HOME/.Xresources

xsetroot -solid grey

x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP
Desktop" &

gnome-session &

:wq!
```

3.2.4 Start the VNC server

Start the TightVNC Server again now that the changes are made with the desired resolution.

```
dobriain@UbuntuVM-1~$ tightvncserver -geometry 1024x768 :1
```

3.2.5 Connect to the VNC Server from a VNC Client

On the Linux client install the TightVNC Viewer.

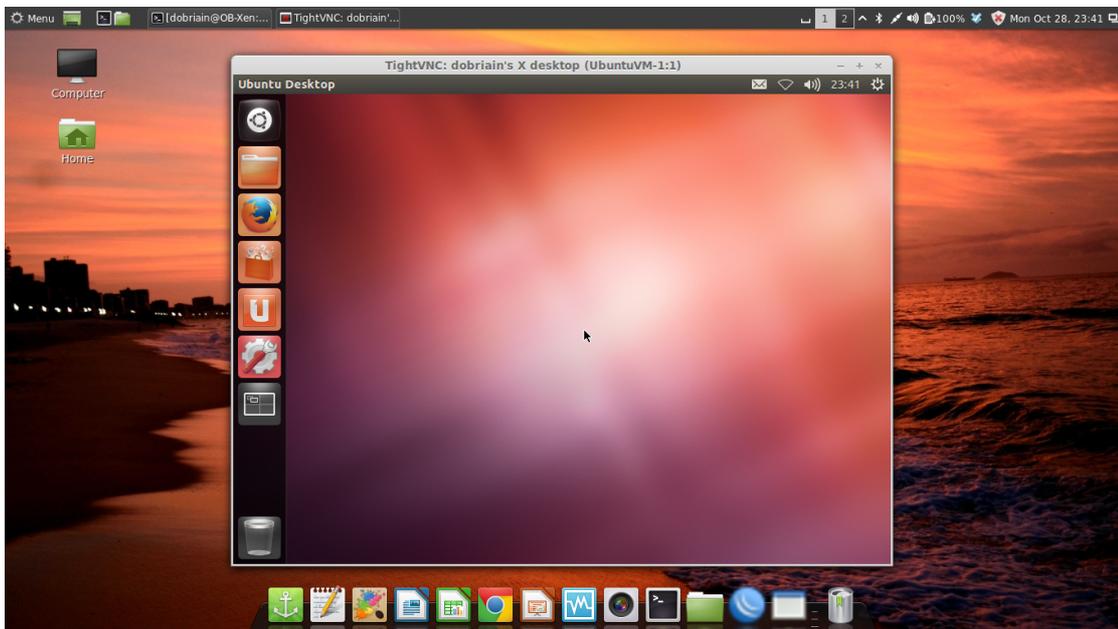
```
dobriain@desktop~$ sudo apt-get install xtightvncviewer
```

To connect to the server, use the the following command:

```
xtightvncviewer <vm-IP-address>:1
```

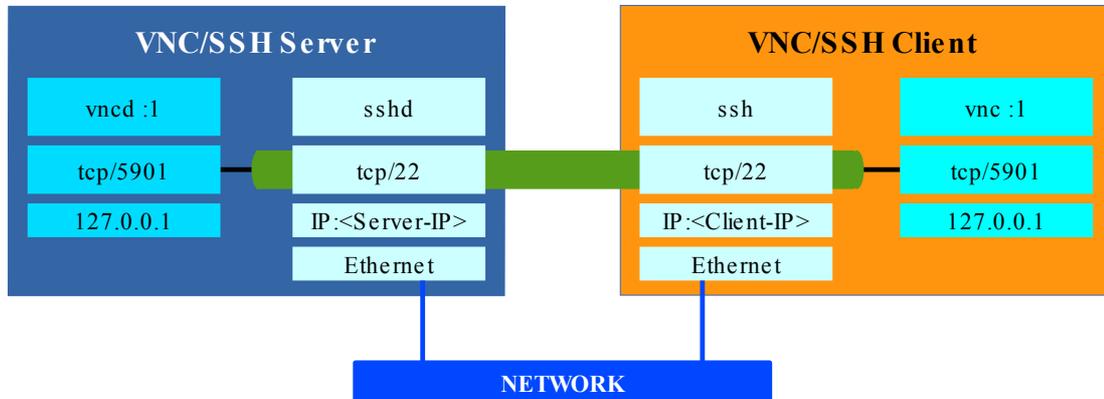
```
dobriain@desktop~$ xtightvncviewer 75.144.153.182:1
```

```
Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
Performing standard VNC authentication
Password: <VNC Password>
Authentication successful
Desktop name "dobriain's X desktop (UbuntuVM-1:1)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Warning: Cannot convert string "--helvetica-bold-r--*-16-*-*--*-*-*" to
type FontStruct
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```



3.3 Accessing the GUI over an non-secure network

It may be necessary to access the GUI over an non-secure network. In this case the use of SSH tunnelling can be employed to secure the link between the VNC Server and the client.



3.3.1 VNC Server ran as localhost only

On the Server stop the VNC Server and restart it with the switch **-localhost**. This prevents remote VNC Viewers accessing the VNC Server.

```
dobriain@UbuntuVM-1~$ tightvncserver -kill :1
```

```
dobriain@UbuntuVM-1~$ tightvncserver -geometry 800x600 -localhost :1
```

3.3.2 SSH connection

On the client open an SSH connection to the server but with some special switches.

```
dobriain@debian~$ ssh -C -L 5901:localhost:5901 dobriain@75.144.153.182
```

The **-C** switch enables compression of data between client and server.

The **-L 5901:localhost:5901** switch accepts connections from the localhost on port 5901 (this equates to the local display :1) and tunnels it to the SSH client side such that it is available locally on the client machine on its port 5901.

A **-v** switch can be used when establishing the first time as it gives verbose information on connection.

3.3.3 SSH connection

On the client run TightVNC viewer to access the 'local' VNC Server on port 5901 (This is in fact the tunnelled display).

```
dobriain@debian~$ xtightvncviewer -encodings "copyrect hextile" localhost:5901
```

VNC uses 'raw' pixel encoding by default as it gives better performance for local access. However in this case it is not really a local connection but a local connection redirected across an SSH tunnel. The **-encodings "copyrect hextile"** switch is more effective over the network.

The Copy Rectangle (CopyRect) encoding is efficient when something is being moved; the only data sent is the location of a rectangle from which data should be copied to the current location. Copyrect is also be used to efficiently transmit a repeated pattern. Hextile encoding splits rectangles up in to 16x16 tiles, which are sent in a predetermined order. Hextile encoding is usually the best choice for using in high-speed network environments (e.g. Ethernet local-area networks).

3.4 Xen VM Snapshots

3.4.1 Taking a VM snapshot

VM snapshot is a mechanism within XCP to 'copy' a VM, store it and potentially recover a failed VM at some future date from the snapshot.

```
xe vm-snapshot vm=<vm_name> new-name-label=<snapshot_name>
```

```
dobriain@OB-Xen~$ sudo xe vm-snapshot vm='UbuntuVM-1' new-name-label='UbuntuVM-1-Snapshot'  
1afb566f-20a9-fb6a-9e5e-5b8732dfeac2
```

3.4.2 Reviewing the list of Snapshots

Review the snapshot list:

```
dobriain@OB-Xen~$ sudo xe snapshot-list  
  
          uuid (RO): 1afb566f-20a9-fb6a-9e5e-5b8732dfeac2  
      name-label (RW): UbuntuVM-1-Snapshot  
 name-description (RW): Installed via xe CLI  
is-snapshot-from-vmpp (RO): false
```

3.4.3 Rolling back to a snapshot

Restoring a VM to snapshot state will not preserve the original VM UUID or MAC address.

To destroy the VM shown earlier.

```
dobriain@OB-Xen~$ sudo xe vm-shutdown vm='UbuntuVM-1' force=true  
  
dobriain@OB-Xen~$ sudo xe vm-destroy uuid=47b01617-b831-5c83-d4cb-db47f6a9c04e
```

Create a new VM from the snapshot:

```
xe vm-install new-name-label=<vm_name_label> template=<snapshot_name>
```

```
dobriain@OB-Xen~$ sudo xe vm-install new-name-label='UbuntuVM-1-NEW'  
template='UbuntuVM-1-Snapshot'
```

Start the VM

```
dobriain@OB-Xen~$ sudo xe vm-start vm='UbuntuVM-1-NEW'
```

4. Lab Exercise

Build a Xen Server on a server hardware platform using one of the following:

- XenServer
- Debian XCP-XAPI Project Kronos
- Fedora XCP-XAPI Project Zeus

Create one of the following guest OSs on a VM.

- Ubuntu
- Microsoft Windows 7

Establish a GUI on guest OS.

Establish remote access to guest OS GUI.