INR + LS Lab Assignment:
Virtualization*

Preparing the environment

Task 1. Make sure that either Ubuntu 16.04.3 64-bit is installed on your server. Use the PXE service to start the installation process if needed. Partition your disks such that you have at least 200 GB of unpartitioned space.

Answer:

Since in the previous course CIA I did a lot of changes in the server configuration, so I preferred to reinstall Ubuntu 16.04.3 64-bit again, using PXE service, I don't want to dive into details about this process because it is obvious. I just rebooted my server and pressed F12 in order to use this service, after that I chose Ubuntu 16.04 64-bit to install, after that I followed screen instructions in order to install it. However, I took into accounts that the requirements here need to have at least 200 GB of space, so I installed it on the SSD and I left the 1 TB hard disk for this.

This is the unused 1 Tb hard disk:

```
kotaiba@bristol:~$ sudo lshw -C disk
  *-disk
       description: ATA Disk
       product: CT240BX200SSD1
       physical id: 0.0.0
       bus info: scsi@0:0.0.0
       logical name: /dev/sda
       version: .6
       serial: 1618F01B6E8F
       size: 223GiB (240GB)
       capabilities: partitioned partitioned:dos
       configuration: ansiversion=5 logicalsectorsize=512 sectorsize=4096
signature=4945aa38
  *-disk
       description: ATA Disk
       product: ST1000NM0011
       vendor: Seagate
       physical id: 0.0.0
       bus info: scsi@1:0.0.0
       logical name: /dev/sdb
       version: PA07
       serial: Z1N2BXWW
       size: 931GiB (1TB)
       capabilities: gpt-1.00 partitioned partitioned:gpt
       configuration: ansiversion=5 guid=dbf48f6e-
```

```
b000-4fa6-948f-6e761faf5fd8 logicalsectorsize=512 sectorsize=512
```

This is the local storage on my server:

```
kotaiba@bristol:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G     0  3.9G   0% /dev
tmpfs           796M  8.5M  787M   2% /run
/dev/sda1       213G  1.3G  201G   1% /
tmpfs           3.9G     0  3.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           3.9G     0  3.9G   0% /sys/fs/cgroup
```

Task 2. Briefly explain what you think is the main difference between a 32-bit and a 64-bit operating system.

Answer:

The 32-bit and 64-bit are two main categories of processors. These types of processors a computer has affects its overall performance and dictate what type of software it uses.

1- The first difference between these two processors types is the number of calculations per second they can perform, which affects the speed at which they can complete tasks. “64-bit processors can come in dual core, quad core, six core, and eight core versions for home computing. Multiple cores allow for an increased number of calculations per second that can be performed, which can increase the processing power and help make a computer run faster.” As result of this software that require many calculations to function smoothly and need more computation power can operate efficiently and faster on the multi-core 64-bit processors.

2- In addition to that, the second difference is that the 32-bit processor support maximum amount 3-4GB of memory (RAM). Whereas a 64-bit computer can support memory amounts over 4 GB. This feature is important for software programs used in graphic design and video editing and rendering.

Sources:

1- https://www.computerhope.com/issues/ch001498.htm

Task 3. Install the lvm2 package and create a physical volume using 100 GB of the 200 GB free space reserved before. On top of the physical volume create a volume group called VolumeGroupXen. Here you will store the virtual machine images. We will create the logical volumes later.

Hints: pvcreate, vgcreate, pvdisplay.

Answer:

First, let's install it:

```
kotaiba@bristol:~$ sudo apt-get install lvm2
```

After that, I wanted to see the current LVM table. It gave me this error:

```
kotaiba@bristol:~$ sudo sudo pvdisplay

  /run/lvm/lvmetad.socket: connect failed: No such file or directory
  WARNING: Failed to connect to lvmetad. Falling back to internal scanning.
```

I checked about this error online and I found the following command will solve this issue:

```
kotaiba@bristol:~$ sudo systemctl enable lvm2-lvmetad.service
Synchronizing state of lvm2-lvmetad.service with SysV init with
/lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable lvm2-lvmetad
kotaiba@bristol:~$ sudo systemctl enable lvm2-lvmetad.socket
kotaiba@bristol:~$ sudo systemctl start lvm2-lvmetad.service
kotaiba@bristol:~$ systemctl start lvm2-lvmetad.service
```

Now, when I try to do that again, it works well but it gave me nothing:

```
kotaiba@bristol:~$ sudo pvdisplay
kotaiba@bristol:~$ echo $?
0
```

So, now as I mentioned above I installed the OS on the 250 GB SSD, so now I need to create a new volume for the 1 TB HDD in order to proceed in this assignment:

First, let's list the partition tables for my server:

```
kotaiba@bristol:~$ sudo fdisk -l
Disk /dev/sda: 223.6 GiB, 240057409536 bytes, 468862128 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x4945aa38

Device     Boot     Start       End   Sectors   Size Id Type
/dev/sda1  *         2048 452147199 452145152 215.6G 83 Linux
/dev/sda2       452149246 468860927  16711682     8G  5 Extended
/dev/sda5       452149248 468860927  16711680     8G 82 Linux swap / Solaris

Partition 2 does not start on physical sector boundary.


Disk /dev/sdb: 931.5 GiB, 1000204886016 bytes, 1953525168 sectors
Units: sectors of 1 * 512 = 512 bytes

```

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: DBF48F6E-B000-4FA6-948F-6E761FAF5FD8

Device     Start     End Sectors Size Type
/dev/sdb1   2048 2099199 2097152   1G FreeBSD ZFS
```

As we see it gave “Partition 2 does not start on physical sector boundary.” which means I need to use /dva/sda (using the HDD) for our new partition

Before we continue (I just noticed that there is one partition remaining from CIA “/dev/sdb1 2048 2099199 2097152 1G FreeBSD ZFS”, So I deleted it using fdisk (d function)

Now, let's create our new 200 GB partition:

```
kotaiba@bristol:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): m

Help:

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): n
Partition number (1-128, default 1): 1
First sector (34-1953525134, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-1953525134, default
1953525134): +200GB

Created a new partition 1 of type 'Linux filesystem' and of size 186.3 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 30
Changed type of partition 'Linux filesystem' to 'Linux LVM'.

Command (m for help): p
Disk /dev/sdb: 931.5 GiB, 1000204886016 bytes, 1953525168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: DBF48F6E-B000-4FA6-948F-6E761FAF5FD8

Device     Start       End   Sectors   Size Type
/dev/sdb1   2048 390627327 390625280 186.3G Linux LVM

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Now, let's use pvcreate to create the Physical Volume (LVM)

```
kotaiba@bristol:~$ sudo pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
```

Then, let's create the Volume Group – Use vgcreate

```
kotaiba@bristol:~$ sudo vgcreate VolumeGroupXen /dev/sdb1
  Volume group "VolumeGroupXen" successfully created
```

Let's check what we did:

```
kotaiba@bristol:~$ sudo lvmdiskscan
  /dev/sda1 [     215.60 GiB]
  /dev/sda5 [       7.97 GiB]
  /dev/sdb1 [     186.26 GiB] LVM physical volume
  0 disks
  2 partitions
  0 LVM physical volume whole disks
  1 LVM physical volume
```

```
kotaiba@bristol:~$ sudo !!
sudo vgdisplay
  --- Volume group ---
  VG Name               VolumeGroupXen
  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               186.26 GiB
  PE Size               4.00 MiB
  Total PE              47683
  Alloc PE / Size       0 / 0
  Free  PE / Size       47683 / 186.26 GiB
  VG UUID               xLCZfc-S3si-RszK-YvPm-0WkK-Lp6f-494NOO
```

It works, I got help from MARKO and ARNO (Credits for them)

Srouces:

1- https://www.digitalocean.com/community/tutorials/how-to-use-lvm-to-manage-storage-devices-on-ubuntu-16-04, 2- https://www.digitalocean.com/community/tutorials/an-introduction-to-lvm-concepts-terminology-and-operations#, 3-https://help.ubuntu.com/community/UbuntuDesktopLVM, 4- https://www.howtogeek.com/howto/40702/how-to-manage-and-use-lvm-logical-volume-management-in-ubuntu/, 5-https://www.liquidweb.com/kb/disk-partitioning-with-fdisk-2/, 6- https://www.cyberciti.biz/faq/linux-how-to-delete-a-partition-with-fdisk-command/, 7- For Troubleshooting refer to http://www.thegeekstuff.com/2010/08/how-to-create-lvm

Task 4. Install the xen-hypervisor-4.6-amd64 package and, if needed, configure the system such that the Xen kernel is booted by default. Check with dmesg whether the correct kernel booted after rebooting.

Hint: Like all Debian based systems, Ubuntu stores in /etc/default settings that the user is likely to change1.

Do not forget to run update-grub after changing the bootloader config files. You can test that your Xen installation is fully functional by issuing xl info and xl list.

Answer:

First, let's install the package:

```
kotaiba@bristol:~$ sudo apt-get install xen-hypervisor-amd64
```

Now, let's verify that the installation has succeeded:

```
First we reboot:
kotaiba@bristol:~$ sudo reboot
```

Using xl list:

```
kotaiba@bristol:~$ sudo xl list
Name                                        ID   Mem VCPUs  State    Time(s)
Domain-0                                     0  7921     4     r-----
6.6
```

Using xl info:

```
kotaiba@bristol:~$ sudo xl info
host                   : bristol
release                : 4.4.0-97-generic
version                : #120-Ubuntu SMP Tue Sep 19 17:28:18 UTC 2017
machine                : x86_64
nr_cpus                : 4
max_cpu_id             : 7
nr_nodes               : 1
cores_per_socket       : 2
threads_per_core       : 2
cpu_mhz                : 2294
hw_caps                :
bfebfbff:28100800:00000000:00007f00:77bae3ff:00000000:00000001:00000281
virt_caps              : hvm hvm_directio
total_memory           : 8161
free_memory            : 128
sharing_freed_memory   : 0
sharing_used_memory    : 0
outstanding_claims     : 0
free_cpus              : 0
xen_major              : 4
xen_minor              : 6
xen_extra              : .5
xen_version            : 4.6.5
xen_caps               : xen-3.0-x86_64 xen-3.0-x86_32p hvm-3.0-x86_32
hvm-3.0-x86_32p hvm-3.0-x86_64
```

```
xen_scheduler          : credit
xen_pagesize           : 4096
platform_params        : virt_start=0xffff800000000000
xen_changeset          :
xen_commandline        : placeholder
cc_compiler            : gcc (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
cc_compile_by          : stefan.bader
cc_compile_domain      : canonical.com
cc_compile_date        : Fri Oct 13 15:42:52 UTC 2017
xend_config_format     : 4
```

As we see above, the GRUB automatically chose to boot Xen first if Xen is installed and the system such that the Xen kernel is booted by default.

Sources:

1- https://help.ubuntu.com/community/Xen

Task 5. Install bridge-utils and use brctl to manually create a bridge named xenbr0. Do not add any interfaces to it; we will use routing instead of switching to connect the VMs to the Internet.

Answer:

First, let's install bridge-utils:

```
kotaiba@bristol:~$ sudo apt-get install bridge-utils
```

If we want to see the configured bridges no, it gave us it is empty:

```
kotaiba@bristol:~$ brctl show
bridge name bridge id        STP enabled    interfaces
```

Now we need to create a xenbr0 bridge using brctl command:

```
kotaiba@bristol:~$ sudo brctl addbr xenbr0
```

Test again:

```
kotaiba@bristol:~$ brctl show
bridge name bridge id        STP enabled    interfaces
xenbr0      8000.000000000000    no
```

Source:

1-
https://www.cyberciti.biz/faq/how-to-create-bridge-interface-ubuntu-linux/ubuntu-install-bridge-utils/

> Task 6. When creating the bridge, Linux will also create a network interface called xenbr0 that connects your server to that bridge. The IPv4 addresses to use for your VMs are those in the /28 subnet which is routed to your server (see SNE students mailing list). Assign the first free IPv4 address from your /28 subnet to this xenbr0 network interface using ifconfig. The first free address is your subnet is the address at the start of the range plus 1. The starting address of the range is reserved to act as a network address (e.g. 145.100.104.0) in currrent Internet practice. The last address in the range is reserved to act as the broadcast address (see RFC3021). The address you assign will act as the IP gateway address for your virtual machines.

Answer:

First, let's check the xenbr0 interface:

```
kotaiba@bristol:~$ ifconfig xenbr0
xenbr0    Link encap:Ethernet  HWaddr 0a:59:3d:42:07:dd
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Server Name	IP Address	IPV4 subnet (/28)	First IP	Last IP	Netmask	Broadcast
Bristol	145.100.104.163	145.100.108.80	145.100.108.81	145.100.108.94	255.255.255.240	145.100.108.95

Now, let's assign the first free IPv4 address from my /28 subnet to this xenbr0 network interface using ifconfig:

```
kotaiba@bristol:~$ sudo ifconfig xenbr0 inet 145.100.108.81 netmask
255.255.255.240
```

Show if it works:

```
kotaiba@bristol:~$ ifconfig xenbr0
xenbr0    Link encap:Ethernet  HWaddr 0a:59:3d:42:07:dd
          inet addr:145.100.108.81  Bcast:145.100.108.95
Mask:255.255.255.240
          inet6 addr: fe80::859:3dff:fe42:7dd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:648 (648.0 B)
```

Source:

1- https://www.tecmint.com/ifconfig-command-examples/

Task 7. Test whether you can reach the address on the bridge interface from outside of your machine. You may have to enable IPv4 and IPv6 routing (Hint: sysctl.conf). Make sure that you test using ping -n from your workstation or any other machine connected to the Internet. Note that reverse DNS for your /28 subnet is also delegated to you. Make sure you don’t have any firewall filters that prevent forwarding IP traffic.

Answer:

I did nothing for this question, I just pinged from my desktop (at home) to the xenbr0 address “145.100.108.81” and everything goes well:

```
kotaiba@kotaiba:~$ ping -n 145.100.108.81
PING 145.100.108.81 (145.100.108.81) 56(84) bytes of data.
64 bytes from 145.100.108.81: icmp_seq=1 ttl=58 time=13.7 ms
64 bytes from 145.100.108.81: icmp_seq=2 ttl=58 time=16.7 ms
64 bytes from 145.100.108.81: icmp_seq=3 ttl=58 time=19.8 ms
64 bytes from 145.100.108.81: icmp_seq=4 ttl=58 time=17.9 ms
^Z
[3]+  Stopped                 ping -n 145.100.108.81
```

Task 8. Edit /etc/network/interfaces such that xenbr0 will persist across system reboots. Use the Debian/Ubuntu way!

(Hint: man bridge-utils-interfaces)

Xen will detect the xenbr0 bridge and will automatically connect new virtual machines to it,creating a virtual Ethernet network.

Answer:

What I need to add to my /etc/network/interfaces file is auto xenbr0 before the iface xenbr0 inet static line in addition to the other lines, will bring it up on boot so it will persist across system reboots.

```
# xenbr0 network interface
auto xenbr0
iface xenbr0 inet static
    address 145.100.108.81
    netmask 255.255.255.240
    broadcast 145.100.108.95
```

Then, we restart the network:

```
kotaiba@bristol:~$ sudo ifdown --exclude=lo -a && sudo ifup --exclude=lo -a
```

Now, let's try to reboot and see what will happen:

```
kotaiba@bristol:~$ sudo reboot
```

It is not WORKING :D, let's try to troubleshoot.

Everything Disappear after I reboot, So I did again all the previous steps :'(

After reading the manual “ man bridge-utils-interfaces ”, I figured out that I forgot to add one extra line in the /etc/network/interfaces, so What I added is “bridge_ports all”:

What I added:

```
# xenbr0 network interface
auto xenbr0
iface xenbr0 inet static
    address 145.100.108.81
    netmask 255.255.255.240
    broadcast 145.100.108.95
    network 145.100.108.80
    bridge_ports all
```

Restart the network interfaces:

```
kotaiba@bristol:~$ sudo service networking restart
```

Then after I **REBOOT**, I test it again:

```
kotaiba@bristol:~$ ifconfig
eno1      Link encap:Ethernet  HWaddr d4:ae:52:bf:e4:da
          inet addr:145.100.104.163  Bcast:145.100.104.191
Mask:255.255.255.224
          inet6 addr: fe80::d6ae:52ff:febf:e4da/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1063 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:73365 (73.3 KB)  TX bytes:8137 (8.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

xenbr0    Link encap:Ethernet  HWaddr 16:cd:86:05:33:a1
          inet addr:145.100.108.81  Bcast:145.100.108.95
Mask:255.255.255.240
          inet6 addr: fe80::14cd:86ff:fe05:33a1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:648 (648.0 B)

kotaiba@kotaiba:~$ ping -n 145.100.108.81
PING 145.100.108.81 (145.100.108.81) 56(84) bytes of data.
64 bytes from 145.100.108.81: icmp_seq=1 ttl=58 time=17.1 ms
64 bytes from 145.100.108.81: icmp_seq=2 ttl=58 time=15.6 ms
64 bytes from 145.100.108.81: icmp_seq=3 ttl=58 time=13.8 ms
64 bytes from 145.100.108.81: icmp_seq=4 ttl=58 time=19.5 ms
^Z
[1]+  Stopped                 ping -n 145.100.108.81
```

And everything goes well now.

Source: 1- https://askubuntu.com/questions/73431/when-ubuntu-server-restarts-eth0-doesnt-come-back-up, 2- http://manpages.ubuntu.com/manpages/xenial/man5/bridge-utils-interfaces.5.html

Feedback on Task 8: “Please do not add any interfaces to your xenbr0 bridge. Use routing.”

Yes, you right I already fixed that with Arno (he told me to do that) but I forgot to update my Wiki page.

Fix:

```
# xenbr0 network interface
auto xenbr0
iface xenbr0 inet static
    address 145.100.108.81
    netmask 255.255.255.240
    broadcast 145.100.108.95
    network 145.100.108.80
    bridge_ports none
```

Task 9. Install xen-tools and use xen-create-image to create a Ubuntu virtual machine that has the following characteristics:

- Hostname: Guest-01
- RAM: 1024MB
- Disk size: 10GB
- Swap size: 1024MB
- VolumeGroup: VolumeGroupXen
- Distribution: Ubuntu Xenial (Hint: use the right mirror)
- Filesystem: ext3
- Virtual CPUs: 2
- IP: an IP from your own range

Hint: You might find useful to inspect the man page of xen-create-image and the contents of /etc/xen-tools. Customize xen-tools.conf to provide valid network settings so that every newly created virtual machine image will be able to use the bridge that you've created in the previous steps. Also make sure that you can set the root password interactively.

Answer:

First lets install the xen-tools package:

```
kotaiba@bristol:~$ sudo apt-get install xen-tools
```

In order to create a Ubuntu virtual machine with the previous characteristics:

```
kotaiba@bristol:/etc/xen-tools$ sudo xen-create-image --hostname=Guest-01 --
memory=1024MB --size=10GB --swap=1024MB --lvm=VolumeGroupXen --fs=ext3 --
vcpus=2 --ip=145.100.108.82

perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = "en_US:en",
    LC_ALL = (unset),
    LC_PAPER = "nl_NL.UTF-8",
    LC_ADDRESS = "nl_NL.UTF-8",
    LC_MONETARY = "nl_NL.UTF-8",
    LC_NUMERIC = "nl_NL.UTF-8",
    LC_TELEPHONE = "nl_NL.UTF-8",
    LC_IDENTIFICATION = "nl_NL.UTF-8",
    LC_MEASUREMENT = "nl_NL.UTF-8",
    LC_TIME = "nl_NL.UTF-8",
    LC_NAME = "nl_NL.UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to a fallback locale ("en_US.UTF-8").
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = "en_US:en",
    LC_ALL = (unset),
    LC_PAPER = "nl_NL.UTF-8",
    LC_ADDRESS = "nl_NL.UTF-8",
    LC_MONETARY = "nl_NL.UTF-8",
    LC_NUMERIC = "nl_NL.UTF-8",
    LC_TELEPHONE = "nl_NL.UTF-8",
    LC_IDENTIFICATION = "nl_NL.UTF-8",
    LC_MEASUREMENT = "nl_NL.UTF-8",
    LC_TIME = "nl_NL.UTF-8",
    LC_NAME = "nl_NL.UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to a fallback locale ("en_US.UTF-8").
WARNING:  No gateway address specified!
WARNING:  No netmask address specified!

General Information
--------------------
Hostname       :  Guest-01
Distribution   :  xenial
Mirror         :  http://archive.ubuntu.com/ubuntu
Partitions     :  swap            1024MB (swap)
                  /               10GB  (ext3)
Image type     :  full
Memory size    :  1024MB
Bootloader     :  pygrub

Networking Information
----------------------
IP Address 1   : 145.100.108.82 [MAC: 00:16:3E:8A:C4:65]


Creating swap on /dev/VolumeGroupXen/Guest-01-swap
Done

Creating ext3 filesystem on /dev/VolumeGroupXen/Guest-01-disk
Done
Installation method: debootstrap


Done

Running hooks
Done

No role scripts were specified.  Skipping

Creating Xen configuration file
Done
```

```
No role scripts were specified.  Skipping
Setting up root password
Generating a password for the new guest.
All done


Logfile produced at:
     /var/log/xen-tools/Guest-01.log

Installation Summary
---------------------
Hostname        :  Guest-01
Distribution    :  xenial
MAC Address     :  00:16:3E:8A:C4:65
IP Address(es)  :  145.100.108.82
SSH Fingerprint :  SHA256:X5rSvB6pHTvHwh1wNRmw5852KsShZsWTQqMmZuwzMic (DSA)
SSH Fingerprint :  SHA256:raygzKXmpDwWNulnQqkSATgdT3hgnXyTp/5b5nJKKzQ
(ECDSA)
SSH Fingerprint :  SHA256:nWdTsjbOFsOwGixcceUXMEJbEjwoJwg121JNYPkbhw4
(ED25519)
SSH Fingerprint :  SHA256:zH0HlY+ScUPkt10rpy2CNlB0vKj6WjukQdtNT3j2I+g (RSA)
Root Password   :  " cant put it here "
```

Before we continue, let me specify few stuff:

- It was really hard to answer this question :D .
- I used the second free IP from my subnet to this machine.
- I will specify the Distribution manually.
- I missed some of your requirements so I will ad it in Guest-01.cfg file.

I found a configuration file called Guest-01.cfg placed in /etc/xen :

```
kotaiba@bristol:/etc/xen$ cat Guest-01.cfg
#
# Configuration file for the Xen instance Guest-01, created
# by xen-tools 4.6.2 on Mon Oct 30 22:28:50 2017.
#

#
#  Kernel + memory size
#


bootloader = '/usr/lib/xen-4.6/bin/pygrub'

vcpus       = '2'
memory      = '1024'


#
#  Disk device(s).
```

```
#
root        = '/dev/xvda2 ro'
disk        = [
                  'phy:/dev/VolumeGroupXen/Guest-01-disk,xvda2,w',
                  'phy:/dev/VolumeGroupXen/Guest-01-swap,xvda1,w',
              ]


#
#  Physical volumes
#


#
#  Hostname
#
name        = 'Guest-01'

#
#  Networking
#
vif         = [ 'ip=145.100.108.82 ,mac=00:16:3E:8A:C4:65' ]

#
#  Behaviour
#
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
```

Now, I will add these stuff manually to the Guest-01.cfg file:

```
# Distribution

dist       = 'xenial'
mirror = "http://archive.ubuntu.com/ubuntu/dists/xenial/"

# Stuff that I added


boot = '1'
passwd = 'I can't tell :D'
```

Source:

1- http://www.virtuatopia.com/index.php/Building_a_Xen_Guest_Domain_using_Xen-Tools, 2- http://manpages.ubuntu.com/manpages/xenial/man8/xen-create-image.8.html, 3- http://www.virtuatopia.com/index.php/Building_a_Xen_Guest_Domain_using_Xen-Tools, 4- https://wiki.xenproject.org/wiki/Xen_Project_Beginners_Guide, 5- http://manpages.org/xen-create-image/8

Task 10. The MAC address starts with 00:16:3E. Explain why this prefix is used.

Answer:

The VM mac address is “mac=00:16:3E:8A:C4:65”.

According to their wiki “Assign a random address from within the space 00:16:3e:xx:xx:xx. 00:16:3e is an Organizationally unique identifier (OUI) assigned to the Xen project and which has been made available for Xen users for the purposes of assigning local addresses within that space.”

“A MAC address must be unique among all network devices (both physical and virtual) on the same local network segment. For this reason if you do not have your own OUI to use it is in general recommended to generate a random locally administered address rather than using the Xen OUI (the third option) since it gives 46 bits of randomness rather than 24 which significantly reduces the chances of a clash.”

Source:

1- https://wiki.xenproject.org/wiki/Xen_Networking

Task 11. Start the virtual machine and login to its console and test network connectivity Hint xl console hostname. Exit by hitting CTRL +]

Answer:

In order to start the VM, I already fixed the syntax error that it gave me, now we start it:

First, let's create it:

```
kotaiba@bristol:~$ sudo xl create /etc/xen/Guest-01.cfg -c
```

Now, we check:

```
kotaiba@bristol:~$ sudo !!
sudo xl list
Name                                        ID   Mem VCPUs  State    Time(s)
Domain-0                                     0  5989     4     r-----
309.2
Guest-01                                     1  1024     2     -b----
3.7
```

Connect to the VM consol, Credits to ARNO to help me in this:

```
kotaiba@bristol:~$ sudo xl consol Guest-01
```

```
Ubuntu 16.04 LTS Guest-01 hvc0

Guest-01 login: root
Password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-98-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@Guest-01:~#
```

Now,test network connectivity, **CREDITS TO ARNO**:

When I tried to check the connectivity on Guest-01 VM:

```
root@Guest-01:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:16:3e:8a:c4:65
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10080 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10080 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:745920 (745.9 KB)  TX bytes:745920 (745.9 KB)
```

After trying to figure out what is happening, I checked my /etc/network/interfaces file on my Guest-01 and i found that it missing the Netmask and gateway (it was a mistake when I created it I missed these parts):

I added the following:

```
# The primary network interface
auto eth0
iface eth0 inet static
 address 145.100.108.82
 netmask 255.255.255.240
 gateway 145.100.108.81
 broadcast 145.100.108.95
```

Although after adding the missing parts, I cant reach the internet, since I need to enable IPv4 forwarding on my server:

```
kotaiba@bristol:~$ sudo sysctl -w net.ipv4.ip_forward=1
[sudo] password for kotaiba:
net.ipv4.ip_forward = 1
```

After that, let's test:

```
root@Guest-01:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:3e:8a:c4:65
          inet addr:145.100.108.82  Bcast:145.100.108.95
Mask:255.255.255.240
          inet6 addr: fe80::216:3eff:fe8a:c465/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:688 (688.0 B)  TX bytes:508 (508.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1920 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1920 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:142080 (142.0 KB)  TX bytes:142080 (142.0 KB)

Let's try to ping:

root@Guest-01:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=60 time=0.571 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=60 time=0.668 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=60 time=0.600 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.571/0.613/0.668/0.040 ms
```

So, the network connectivity is working.

Task 12. Use xl to find information about the running VM and then stop it and start it again.

Answer:

List of running VM now:

```
kotaiba@bristol:~$ sudo xl list
Name                                        ID   Mem VCPUs  State    Time(s)
Domain-0                                     0  5989     4     r-----
429.6
Guest-01                                     1  1024     2     -b----
34.7
```

Stop and Start = reboot:

```
kotaiba@bristol:~$ sudo xl reboot -a
Rebooting domain 1
kotaiba@bristol:~$ sudo xl list
Name                                        ID   Mem VCPUs  State    Time(s)
Domain-0                                     0  5989     4     r-----
431.3
Guest-01                                     4  1024     2     r-----
1.0
```

I thing nothing happens (maybe because I answered the next question before that by adding boot= '1' which will reboot it after.

Before, we test that let's reboot the domain 0, so the others VM will reboot also and check again:

```
kotaiba@bristol:~$ sudo xl reboot 0
Rebooting domain 0

kotaiba@bristol:~$ sudo xl list
[sudo] password for kotaiba:
Name                                        ID   Mem VCPUs  State    Time(s)
Domain-0                                     0  7013     4     r-----
10.9
Guest-01                                     1  1024     2     -b----
0.1
```

Everything goes fine “that's weird”.

let's figure out why ? first I remover the boot ='1' now I will detroy and recreate it again from the Vm configuration file:

```
kotaiba@bristol:~$ sudo xl destroy 1

kotaiba@bristol:~$ sudo xl list
Name                                        ID   Mem VCPUs  State    Time(s)
Domain-0                                     0  7013     4     r-----
13.8

Recreate it again:
kotaiba@bristol:~$ sudo xl create /etc/xen/Guest-01.cfg -c
```

Now, let's test everything:

Reboot the Domain 0:

```
kotaiba@bristol:~$ sudo xl reboot 0
Rebooting domain 0

kotaiba@bristol:~$ sudo xl list
[sudo] password for kotaiba:
Name                                        ID   Mem VCPUs  State   Time(s)
Domain-0                                     0  7013     4     r-----
10.1
```

As we see if we list again, it will not boot by default, so now I will put it back again to answer the task 13.

Task 13. Configure your system such that Guest-01 is auto-started after a reboot

Answer:

In the Xen-Tools configuration settings. we add the boot. If boot is defined to be 1 then the new guest system will automatically boot once the creation process has completed. If password is set to 1 the creation process will prompt for a root password for the guest system prior to completion.

```
boot = '1'
```

Check Task 13, you will find full experiment and test.

Task 14. Briefly explain the following terms: DomU, Dom0, PCI pass-through. Which is which in your situation? Is PCI pass-through used, and if so for what?

Answer:

Answer this from the source.

- Dom0: is the initial domain started by the Xen hypervisor on boot. Dom0 is an abbreviation of “Domain 0” (sometimes written as “domain zero” or the “host domain”). Dom0 is a privileged domain that starts first and manages the DomU unprivileged domains.

The Xen hypervisor is not usable without Dom0. This is essentially the “host” operating system (or a “service console”, if you prefer). As a result, Dom0 runs the Xen management toolstack, and has special privileges, like being able to access the hardware directly.

- DomU: is the counterpart to Dom0; it is an unprivileged domain with (by default) no access to the hardware. It must run a Frontend Driver (To access devices that are to be shared between domains, like the disks and network interfaces) for multiplexed hardware it wishes to share with other domains. A DomU is started by running

```
xl create <xen-config-filename>
```

in Dom0. The kernel for a DomU comes from Dom0's filesystem, not from the filesystem exported to the DomU.

- PCI passthrough: allows you to give control of physical devices to guests: that is, you can use PCI passthrough to assign a PCI device (NIC, disk controller, HBA, USB controller, firewire controller, soundcard, etc) to a virtual machine guest, giving it full and direct access to the PCI device.

Yes, in my case it is PCI passthrough, because as the source explained above, it allows me to give the VM control of physical devices (e.g. NIC to get internet), in addition to that one of its potential uses, Passing through graphics cards to guests allows them full access to the 3D acceleration capabilities.

Source:

1- https://wiki.xenproject.org/wiki/DomU, 2-https://wiki.xenproject.org/wiki/Dom0, 3-https://wiki.xenproject.org/wiki/Xen_PCI_Passthrough

Task 15. Is Guest-01 a fully or a para-virtualized guest? Explain.

Answer:

Let's define first What is fully and para virtualized guest:

- Paravirtualization is virtualization in which the guest operating system (the one being virtualized) is aware that it is a guest and accordingly has drivers that, instead of issuing hardware commands, simply issue commands directly to the host operating system. This also includes memory and thread management as well, which usually require unavailable privileged instructions in the processor.
- Full Virtualization is virtualization in which the guest operating system is unaware that it is in a virtualized environment, and therefore hardware is virtualized by the host operating system so that the guest can issue commands to what it thinks is actual hardware, but really are just simulated hardware devices created by the host.

So if we want to check whether the Guest-01 full or para:

```
kotaiba@bristol:~$ sudo xl list --long
[
    {
        "domid": 0,
        "config": {
            "c_info": {
                "type": "pv",
                "name": "Domain-0"
            },
            "b_info": {
                "max_memkb": 4294967292,
                "target_memkb": 6133015,
                "sched_params": {

                },
                "type.pv": {

                },
                "arch_arm": {

                }
            }
        }
    },
    {
        "domid": 1,
        "config": {
            "c_info": {
                "type": "pv",
                "name": "Guest-01",
                "uuid": "cfee9802-62a5-4398-9962-dd81839cac2c",
                "run_hotplug_scripts": "True"
            },
            "b_info": {
                "max_vcpus": 2,
                "avail_vcpus": [
                    0,
                    1
                ],
                "numa_placement": "True",
                "max_memkb": 1048576,
                "target_memkb": 1048576,
                "video_memkb": 0,
                "shadow_memkb": 10240,
                "localtime": "False",
                "disable_migrate": "False",
                "blkdev_start": "xvda",
                "device_model_version": "qemu_xen",
                "device_model_stubdomain": "False",
                "sched_params": {

                },
                "claim_mode": "True",
                "event_channels": 1023,
                "cmdline": "root=/dev/xvda2 ro",
                "type.pv": {
                    "slack_memkb": 0,
                    "bootloader": "/usr/lib/xen-4.6/bin/pygrub",
                    "e820_host": "False"
                },
                "arch_arm": {

                }
            },
            "disks": [
                {
                    "pdev_path": "/dev/VolumeGroupXen/Guest-01-disk",
                    "vdev": "xvda2",
                    "format": "raw",
                    "readwrite": 1
                },
                {
                    "pdev_path": "/dev/VolumeGroupXen/Guest-01-swap",
                    "vdev": "xvda1",
                    "format": "raw",
                    "readwrite": 1
                }
            ],
            "nics": [
                {
                    "devid": 0,
                    "mac": "00:16:3e:8a:c4:65",
                    "ip": "145.100.108.82 "
                }
            ],
            "on_reboot": "restart",
            "on_crash": "restart"
        }
    }
]
```

As we see above, the Guest-01 is para-virtualized guest “pv”:

```
 "c_info": {
                "type": "pv",
                "name": "Guest-01",
                "uuid": "cfee9802-62a5-4398-9962-dd81839cac2c",
                "run_hotplug_scripts": "True"
            },
```

Source:

1-
https://stackoverflow.com/questions/21462581/what-is-the-difference-between-full-para-and-hardware-assisted-virtualiazation

Task 16. deboostrap, rinse and rpmstrap can be used to aid in the creation of virtual machine images. In fact, xen-create-image can use all of them under the hood. When would you use one over the others?

Answer:

In xen-create-image we specify the mechanism to be used to install the base operating system packages of the guest domain's root filesystem via the –install-method in the configuration file. For those using Debian or Ubuntu debootstrab should be specified. For systems which use the Red Hat Package Manager (RPM) such as Red Hat Linux, CentOS and Fedora use rpmstrap. Also, although Debian and Debian-alike distros are the preferred targets of xen-tools, it is possible to deal with RPM based distros, like OpenSuSE, fedora-core and Fedora itself, via rinse.

However, in our case I definitely used deboostrap since I want Ubuntu Xenial distribution so it is best practice and more convenient to use it.

Source:

1- http://www.virtuatopia.com/index.php/Building_a_Xen_Guest_Domain_using_Xen-Tools, 2- https://blog.xenproject.org/2013/01/24/using-xen-tools-on-fedora/

Task 17. User Mode Linux (?) is another approach to virtualization. Write a short paragraph highlighting at least two differences and two similarities between Xen and UML.

Answer:

- User-mode Linux is generally considered to have lower performance than Xen.
- A strength point in Xen is support for thread-local storage (TLS). This is now also supported in the latest UML kernels.
- Xen concentrates on virtualizing the whole machine, and thus all systems running on a Xen machine are really virtual machines.
- In UML, the host machine is not virtualized in any way, and only guest systems are true virtual machines. This allows UML guest direct access to host filesystems and hardware, where it is common to map a host directory (e.g. /uml/root → /).

Source:

1- https://en.wikipedia.org/wiki/User-mode_Linux

Task 18. How do you think that the virtual machine communicates with the outside network in your setup? Draw a simple network diagram showing at least the network cards, the bridges and any routers that might be present. Don’t forget to label everything with IP addresses and names.

Answer:

In my case as we saw through the Xen configuration uses bridging within the backend domain (typically domain 0) to allow the Guest-01 to appear on the network as individual hosts.

This peace of art inspired by ARNO setup in the Lab.

Feedback on Task 18: “Bridges do not have IP addresses. vif interfaces missing.”

Fix:

Source:

1- https://wiki.xenproject.org/wiki/Xen_Networking

Note for myself :D : If you want to troubleshoot or fix anything from the feedback refer to here: https://wiki.xenproject.org/wiki/Xen_Project_Beginners_Guide