Aia Al-Qasab

T5616SN

Advanced Enterprise Server Environment

Lab 7

**Lab 7. Linux Kernel Virtualization Module (KVM)**

March 2019

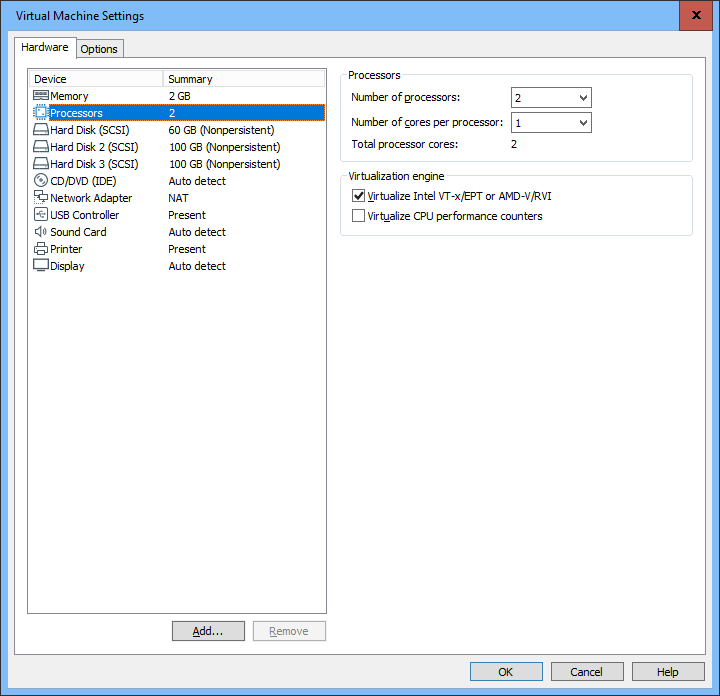


**Lab 7. Linux Kernel Virtualization Module (KVM)**

**Step 1. Installing and configuring the KVM virtualization to an Ubuntu Server**

- Open the **LinuxServer-01** VM settings and from the **Processors**, make sure that “**Virtualize Intel VT-x/EPT or AMD-V/RVI**” is selected. *In* ***your report****, explain why it is important to enable these technologies.*

**A: To run KVM, we need a processor that supports hardware virtualization, Intel and AMD both have developed extensions for their processors, deemed respectively Intel VT-x and AMD-V.**

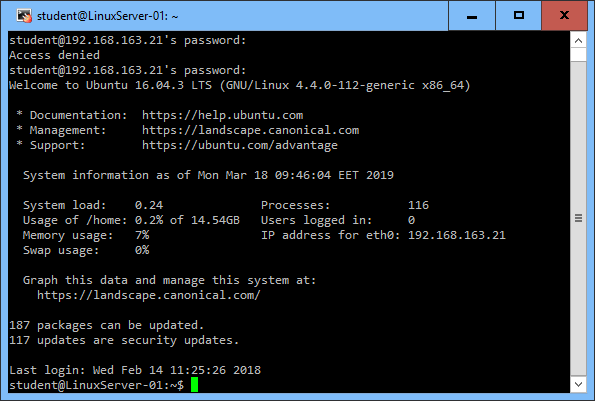


- Start the **LinuxServer-01** VM and minimize the window to the background.



- Open an **SSH** connection to the **LinuxServer-01** VM (the IP is 192.168.163.21) and login as **student** / **student**

o You could use the VM directly from the VMware Player window, but SSH is more natural way of accessing Linux servers remotely and additionally it allows you to easily copy-paste commands to the Shell.



- To start with, check the Wikipedia KVM page (https://en.wikipedia.org/wiki/Kernel-based\_Virtual\_Machine) and introduce yourself to the main structure of the KVM environment and components.

- This time, I don’t provide all the lab instructions ready but your task is to follow the instructions in the official documentation, so browse to https://help.ubuntu.com/community/KVM and start with the **Installation** step.

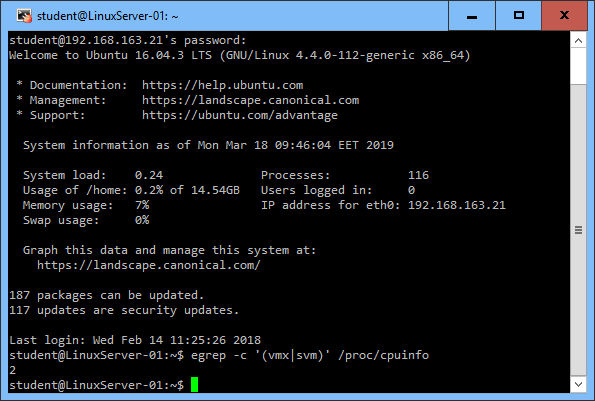
o You can also use other documentation – Google around (just be sure that the documentation is as up-to-date as possible).

o Read (and preferably understand) the instructions and install the KVM according to them.

**Installation:**

Run the KVM:

egrep -c '(vmx|svm)' /proc/cpuinfo

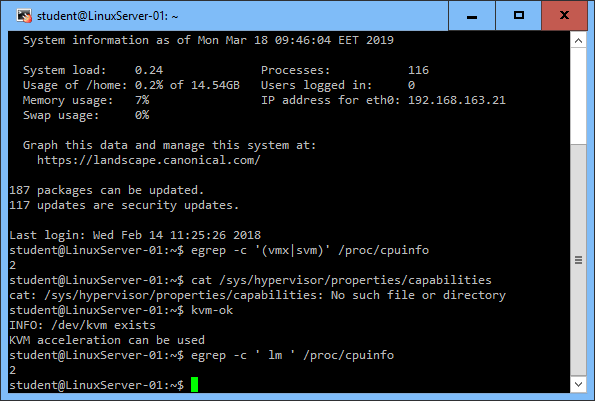


If **1** or more it does - but you still need to make sure that virtualization is enabled in the BIOS.

To check hvm flags in the output, we need to execute: kvm-ok,

the output : INFO: /dev/kvm exists KVM acceleration can be used

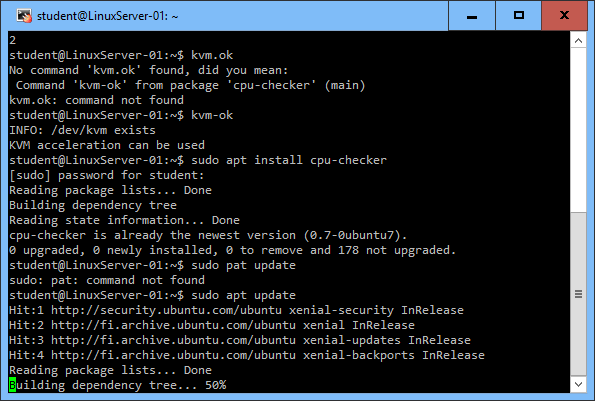
By running the command: egrep -c ' lm ' /proc/cpuinfo to use the kernel 64-bit



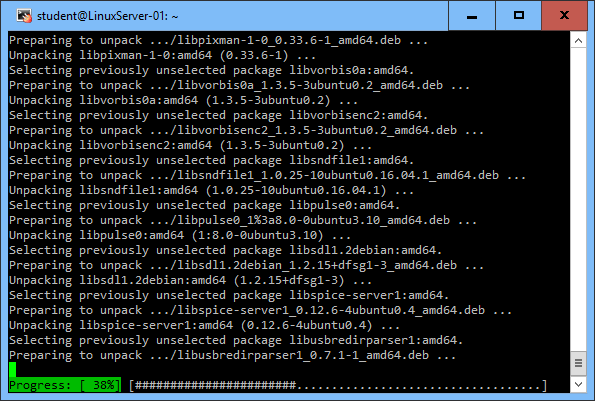
If **1** or higher, it is. Note: lm stands for Long Mode which equates to a 64-bit CPU.

To install KVM and its dependencies, we need to run the apt commands:

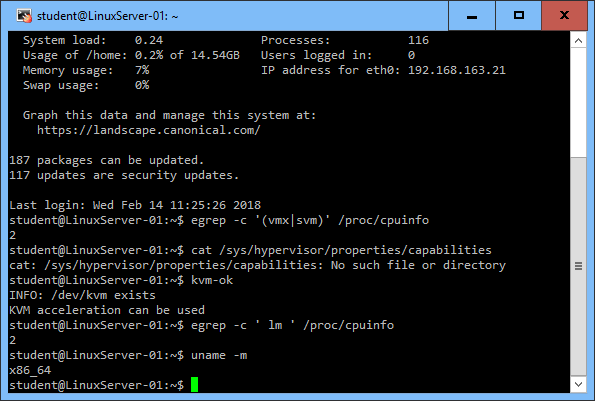
sudo apt update



sudo apt install qemu qemu-kvm



Once the packages are installed successfully, then our local user will be added to the group libvirtd.

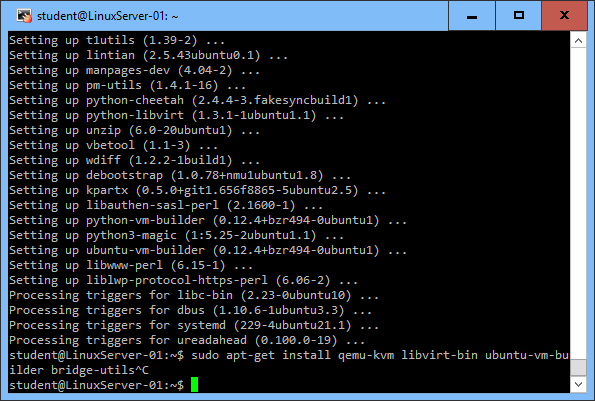


**x86\_64** indicates a running 64-bit kernel. If you use see i386, i486, i586 or i686, you're running a 32-bit kernel.

Enabling libvirtd services, by installing qemu and libvirtd packages in Ubuntu server then it will automatically start and enable libvirtd service.

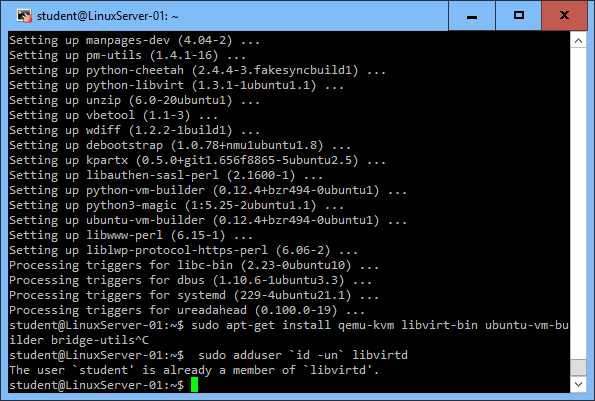
sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils

gemu-kvm is the backend

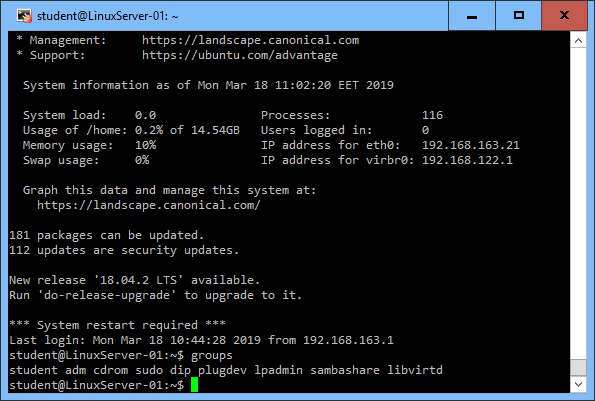


We need to ensure that our username is added to the group libvirtd:

sudo adduser `id -un` libvirtd

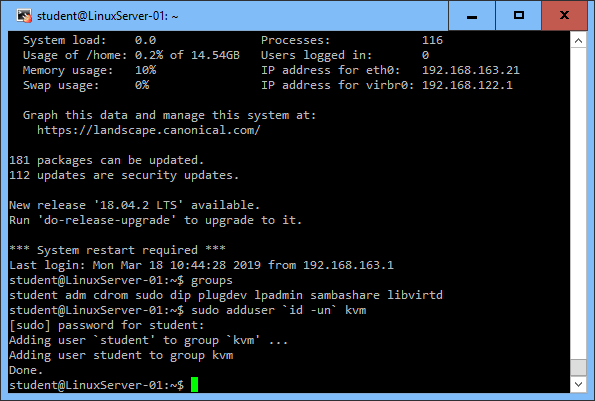


We need to relogin to let users becomes an effective member of the libvirtd group. This member can run the VM.

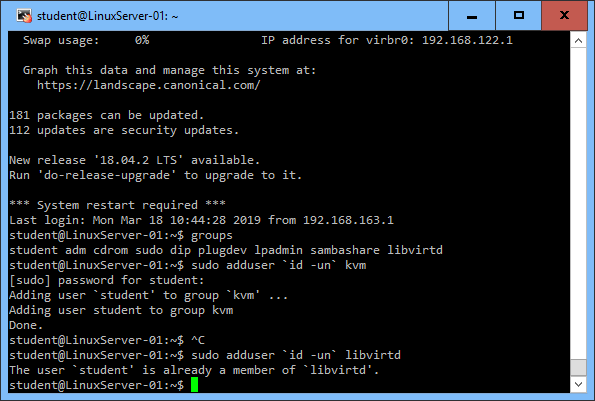


Adding a new user to the groups:

sudo adduser `id -un` kvm

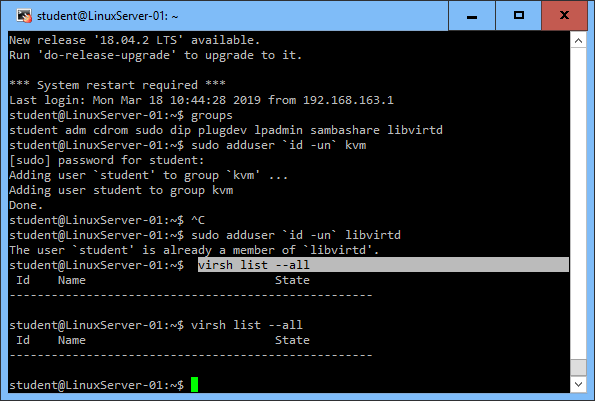


sudo adduser `id -un` libvirtd



We can test if the installation has been successful with the command:

virsh list –all



*In your* ***lab report****, shortly* ***describe the whole process of doing the lab*** *(with* ***screenshots*** *and/or copy-pasting commands and their output). You can write the report like a tutorial for a newbie: how to setup everything and work with the environment (or how not to* *).*

After the installation, check the **Networking** step in the KVM instructions

o If you prefer, you can use the default networking options but that will limit your connectivity from the VMs.

o If you wish a bit more challenge, you can configure the bridged mode to allow the VMs connect to the server NIC directly (allows you to connect to the VMs from outside).

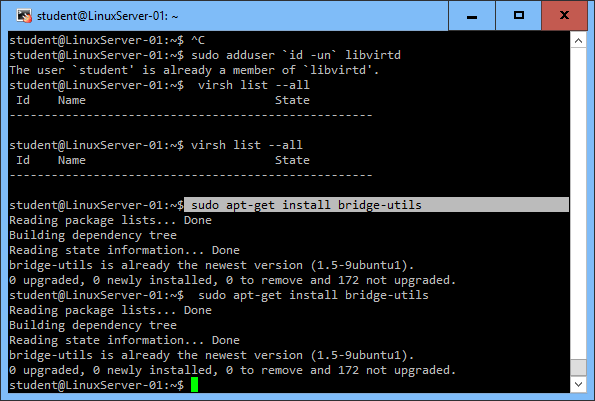
o If you are unsure, you can continue the lab with the defaults and configure bridging later if you have time.

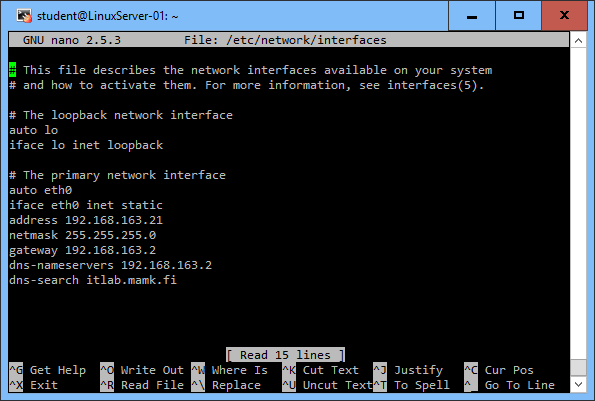
**Networking:**

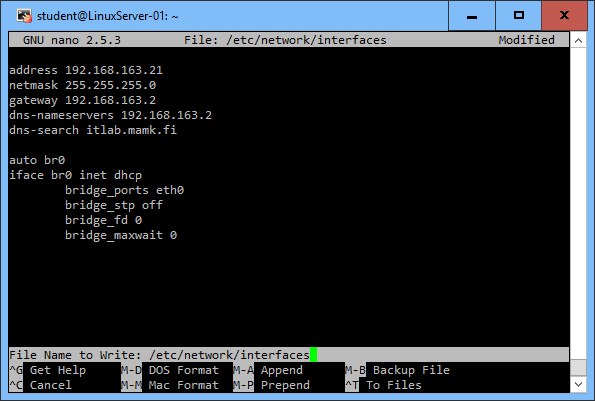
Networking bridge is required to access the KVM based virtual machines outside the KVM hypervisor or host, in Ubuntu the network is managed by netplan utility that’s installed with a file name “**/etc/netplan/50-cloud-init.yaml**” is created automatically..

Installing the package:

sudo apt-get install bridge-utils







**Step 2. Deploying and managing virtual machines in the KVM environment**

- Go to the **Guest Creation** step of the instructions and create at **least two virtual machines** with the methods you prefer.

o There are multiple different ways to accomplish this task and just use the documentation above (or Google for something else). Try for example the following tools (the tools need a lot of parameters, Google for instructions):

vmbuilder

virt-install

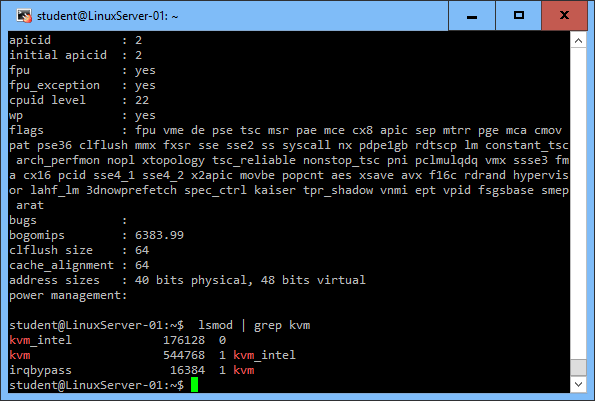
ubuntu-vm-builder

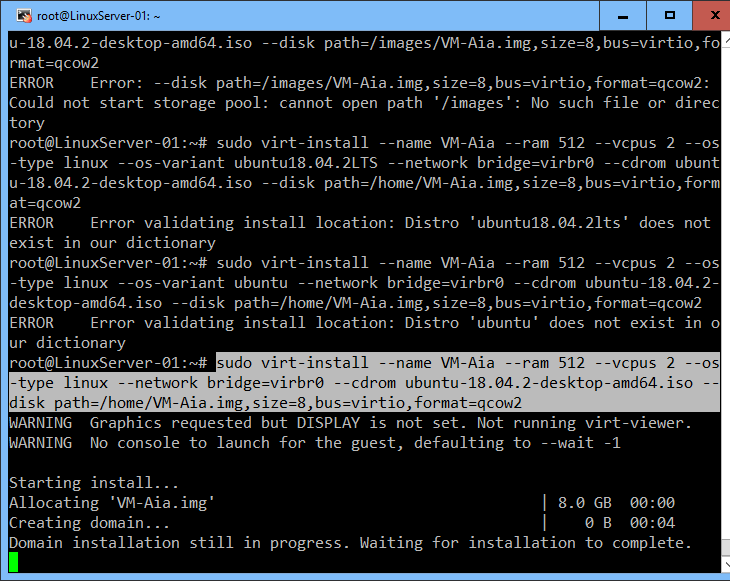
**Ubuntu –vm-builder is the best tool to use if we want to create VMs running, by the command:**

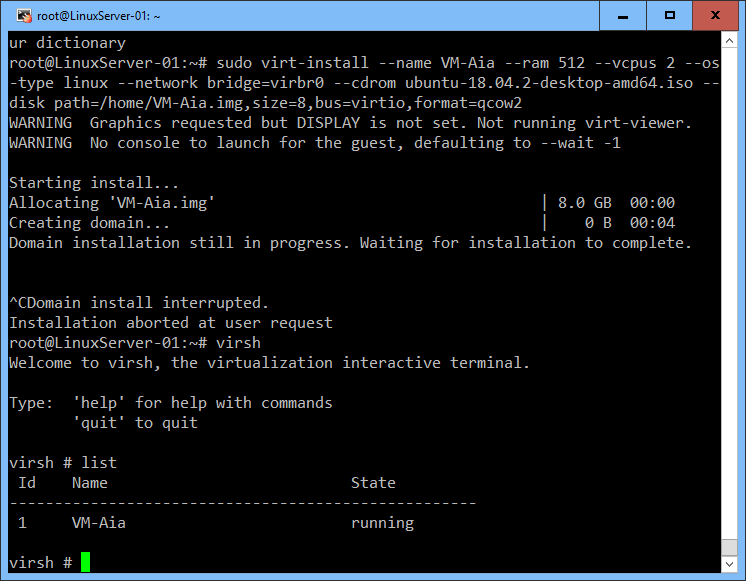
**sudo apt-get install ubuntu-vm-builder**

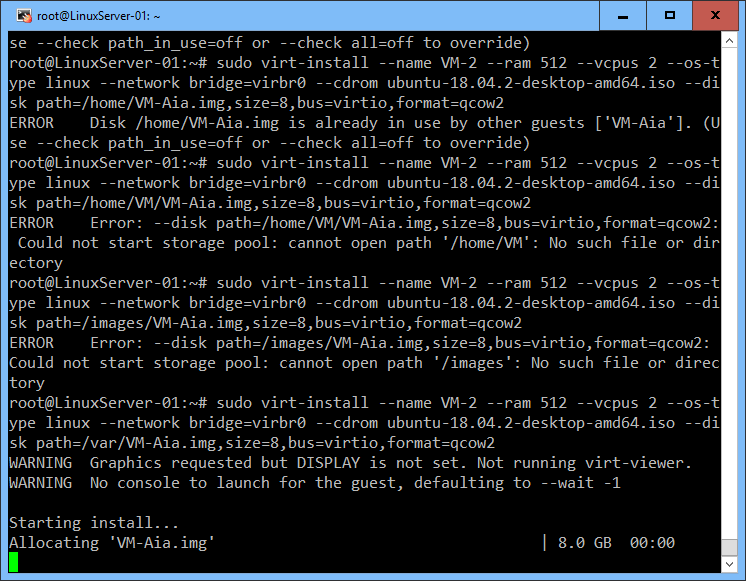
**lsmod | grep kvm**

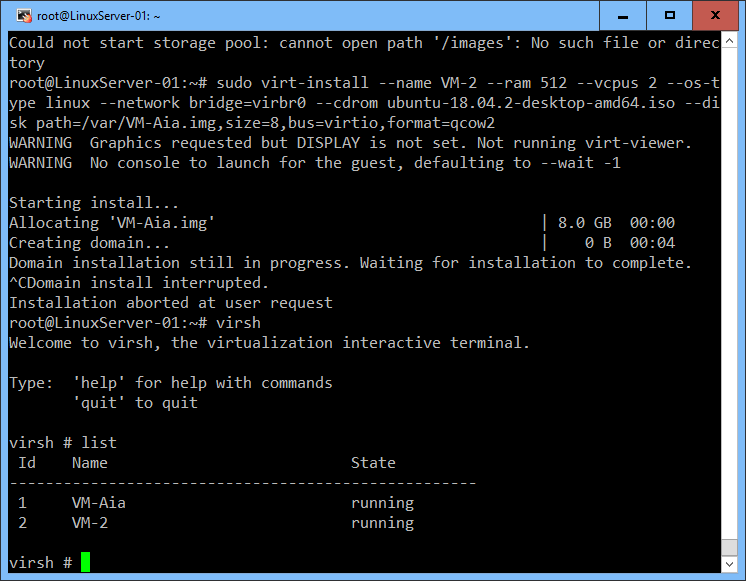
**to make sure that the needed kernel modules have been loaded.**



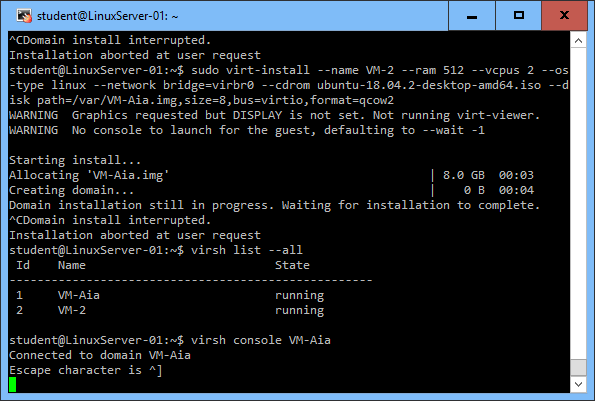




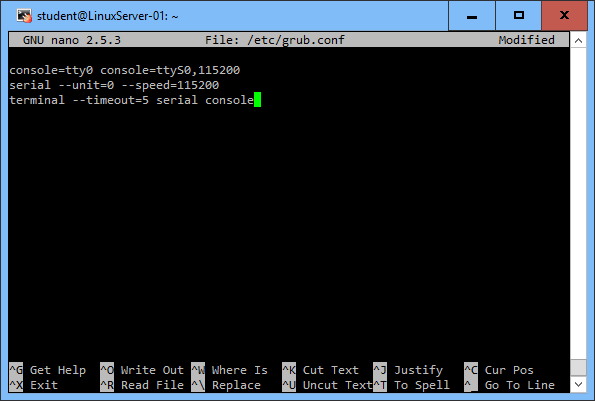




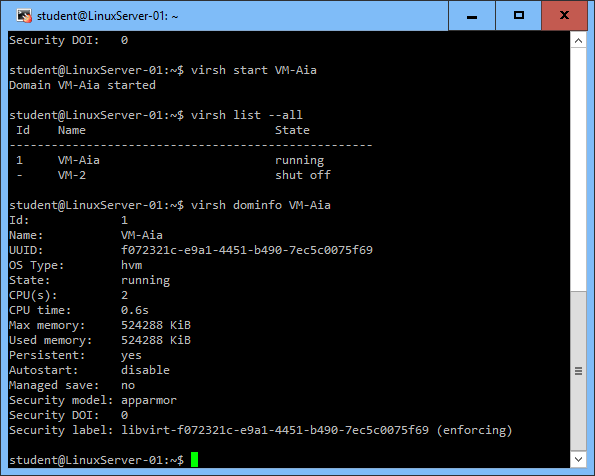
**To connect to the console of the VM use the command:$ virsh console VM-Aia**



**add the entries to the /etc/grub.conf file. and boot the system**



**to start and check the VM we need to input the following commands:**



- When you have some VMs in the system, practice using the **management tools**

o Again, use the instructions and Google for further information. There are plenty of management tools, for example **virt-manager** (a GUI tool), **virsh** (a Shell tool) and **Proxmox** (a web-based tool).

o See http://www.linux-kvm.org/page/Management\_Tools for more comprehensive list of the management tools.

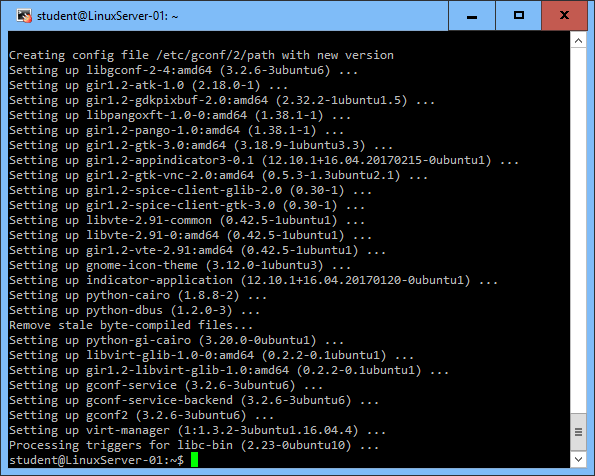
o *Experiment with the tools and explain the process in your* ***lab report****.*

At least, try out starting and stopping VMs, gathering information on the VMs (cpu, mem, storage, networking, …), investigating/changing VM configurations, etc.

o Also try opening a remote connection to you VMs

The method depends on the VM you are running. For a Linux VM, you would probably use SSH and for Windows VM you probably need a GUI to begin with (later, you can use the Remote Desktop/PowerShell)

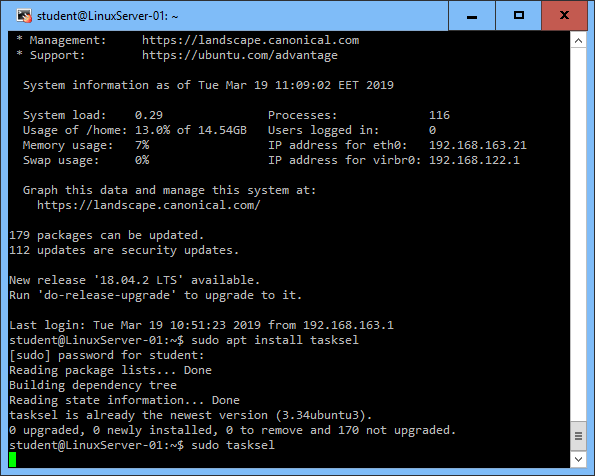
**To use the GUI tools of managing, we need to install the following command :** apt-get install virt-manager

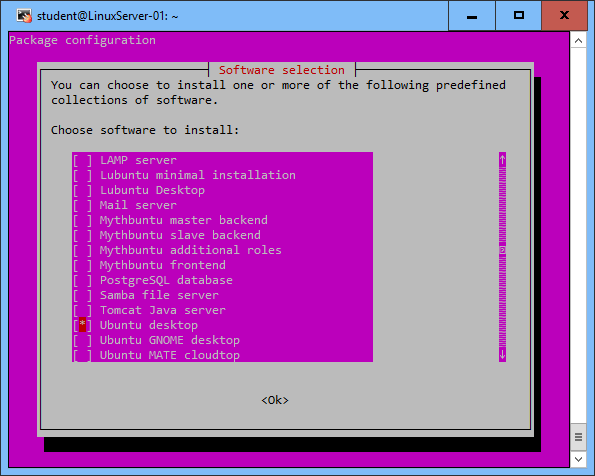


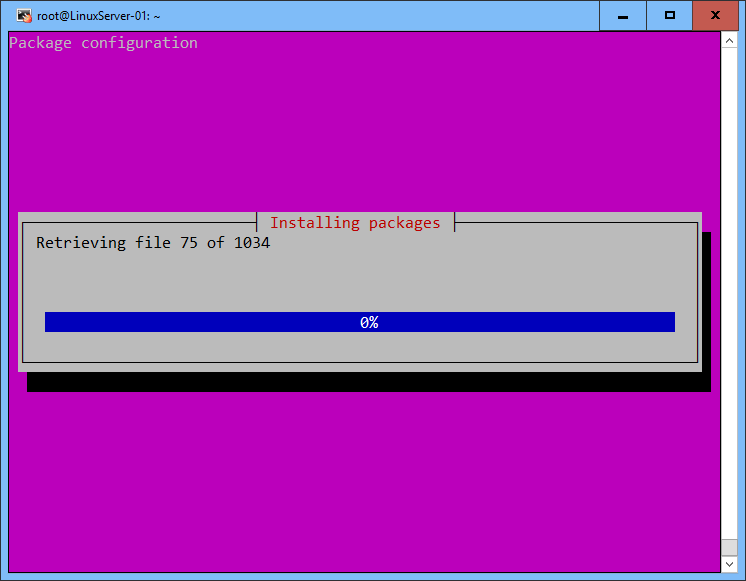
**To connect the powershell remotely by inserting the following command:**

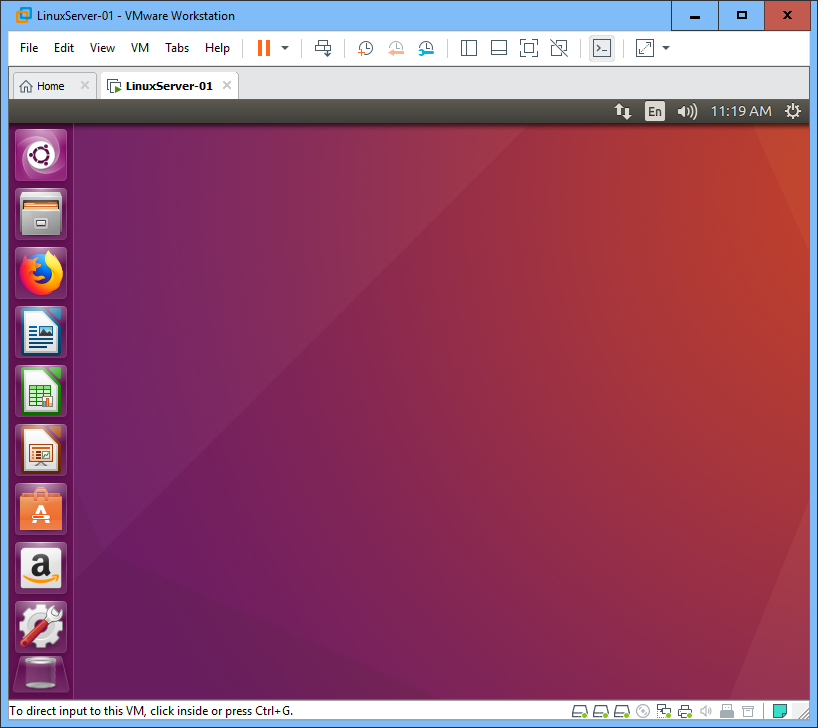
sudo apt-get install virt-manager ssh-askpass-gnome --no-install-recommends

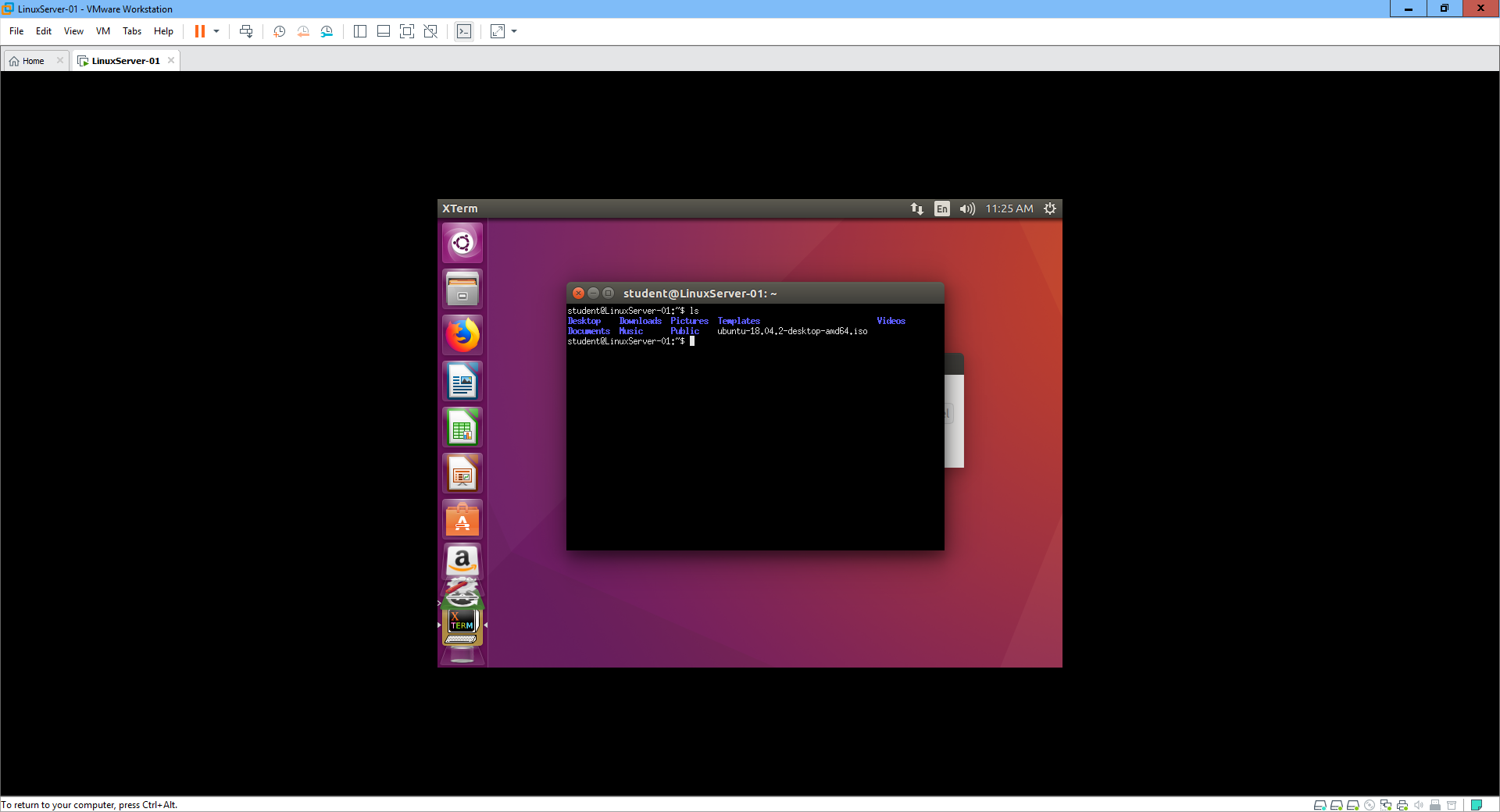
**Installing GUI management tools:**











Finally, describe **in your report** your personal experiences about doing the lab and your experiences about virtualization with Linux. Did you find the configurations and management complex thought the command line? Of course the GUI is available, if needed, but can you invent some practical examples when it might actually be useful to do these things directly in the command line?

**A: Linux virtualization uses to run one or more virtual machines on a physical computer that’s operated by the Linux open source operating system, it can be used for isolating specific apps, programming code or even an OS itself.**

**Powerful computers and hardware have made virtualization more practical and feasible for both desktop and server environments, helping to save power by consolidating several workspace on one system as well as maximizing the workload that the compute can handle. Virtualization is technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system. Software called a hypervisor connects directly to that hardware and allows you to split 1 system into separate, distinct, and secure environments known as virtual machines (VMs).** **Virtualizing resources lets administrators pool their physical resources, so their hardware can truly be commoditized.**