

Quick Introduction to installing and building a singularity container to run on the HPC

This tutorial takes you through the steps to install singularity and build a basic container which you can then run on your local machine or on the HPC. There are 5 main sections:

1. **Creating an Ubuntu Virtual Machine using Oracle's Virtual Box** – you can skip this step if you have a linux box that you have admin privileges on. If you have a windows or mac or prefer to keep your singularity environment separate from your linux environment then this step is important
2. **Install singularity on the linux box** – we will be installing singularity versions 3.0.0 and greater as these will be the only versions maintained on the HPC in the future. Ocelote does legacy Singularity 2.6 at the moment but this will be phased out soon.
3. **Build your first container from a definitions file** – we will build a basic singularity container using a definitions file which is like a recipe that automates the build of your container
4. Alternatively we will **build a container in a sandbox** – this gives you more flexibility to create containers especially in situations where its hard to automate the recipe or in situations where you are just exploring.
5. **Upload and test your containers on the HPC**

To start to build your own neuroimaging containers you can either use a sandbox approach which is described further below or look on github for examples of definitions files.

1. Create an Ubuntu Virtual Machine using Oracle's Virtual Box

- a. Download and Install Oracle VirtualBox for your platform (Latest version is 6.1.2)

<https://www.virtualbox.org/wiki/Downloads>

VirtualBox 6.1.2 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

- b. Download the extension pack for your version of Virtual Box – (if you already have a particular version of virtual box then downloads for older versions are available here

https://www.virtualbox.org/wiki/Download_Old_Builds

VirtualBox 6.1.2 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

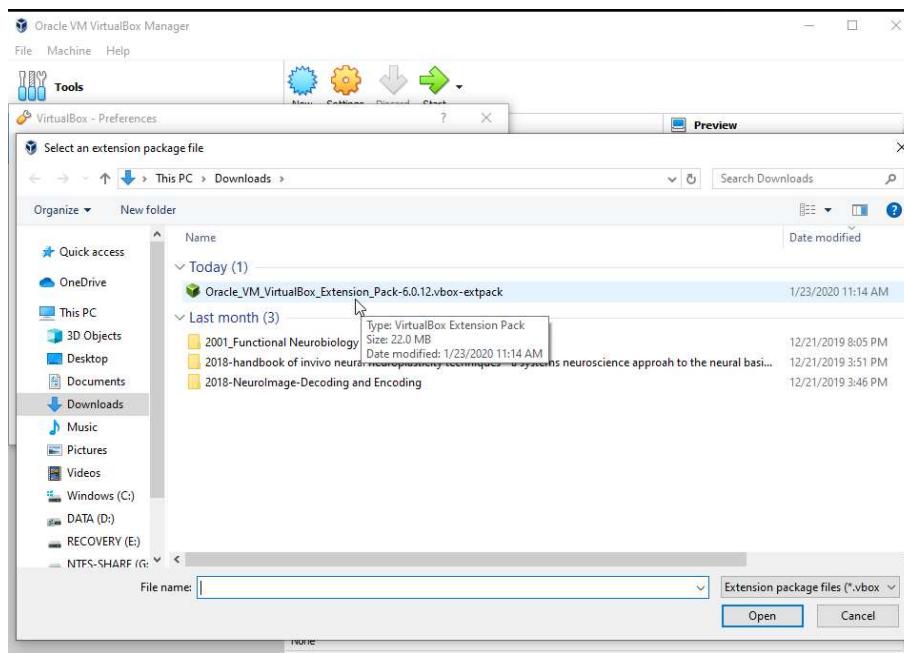
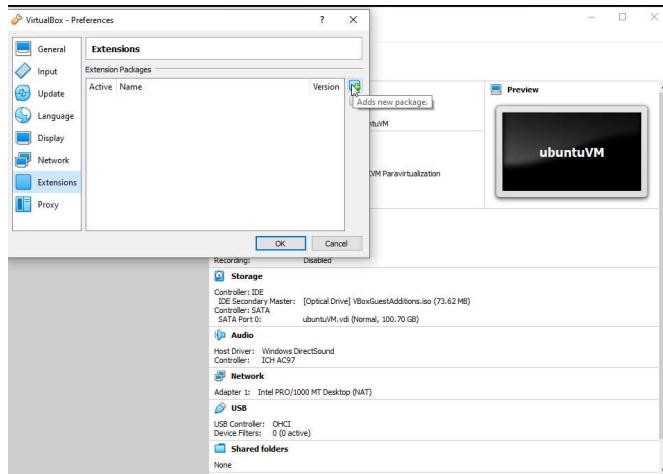
Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

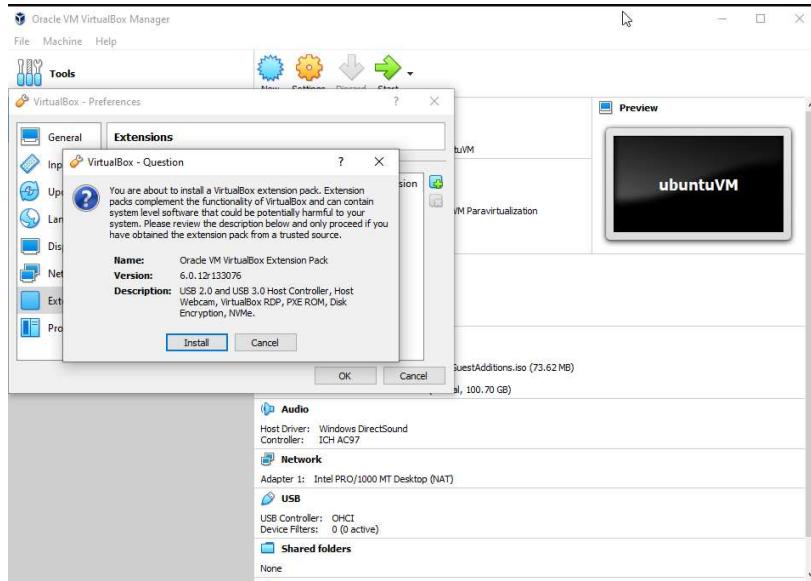
VirtualBox 6.1.2 Oracle VM VirtualBox Extension Pack

- [All supported platforms](#)

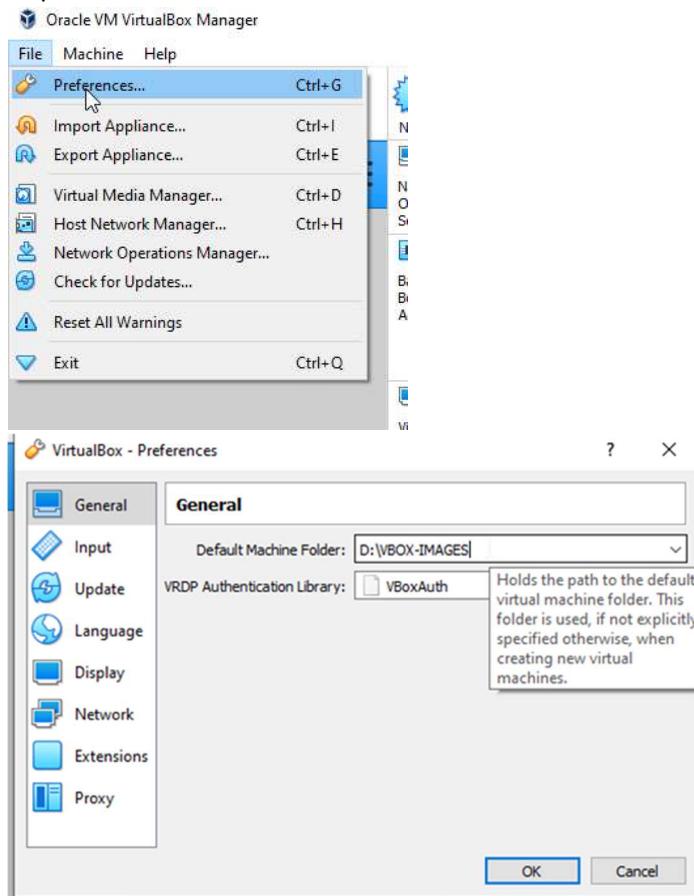
Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See [this chapter from the User Manual](#) for an introduction to this Extension Pack. The Extension Pack binaries are released under the [VirtualBox Personal Use and Evaluation License \(PUEL\)](#). Please install the same version extension pack as your installed version of VirtualBox.

- c. Add extension pack to Virtual Box (this will vary slightly depending on your version) – on windows go to File > Preferences > Extensions and click on add extension

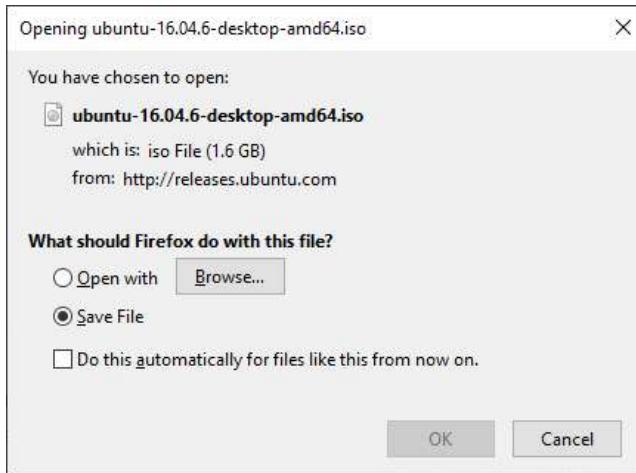
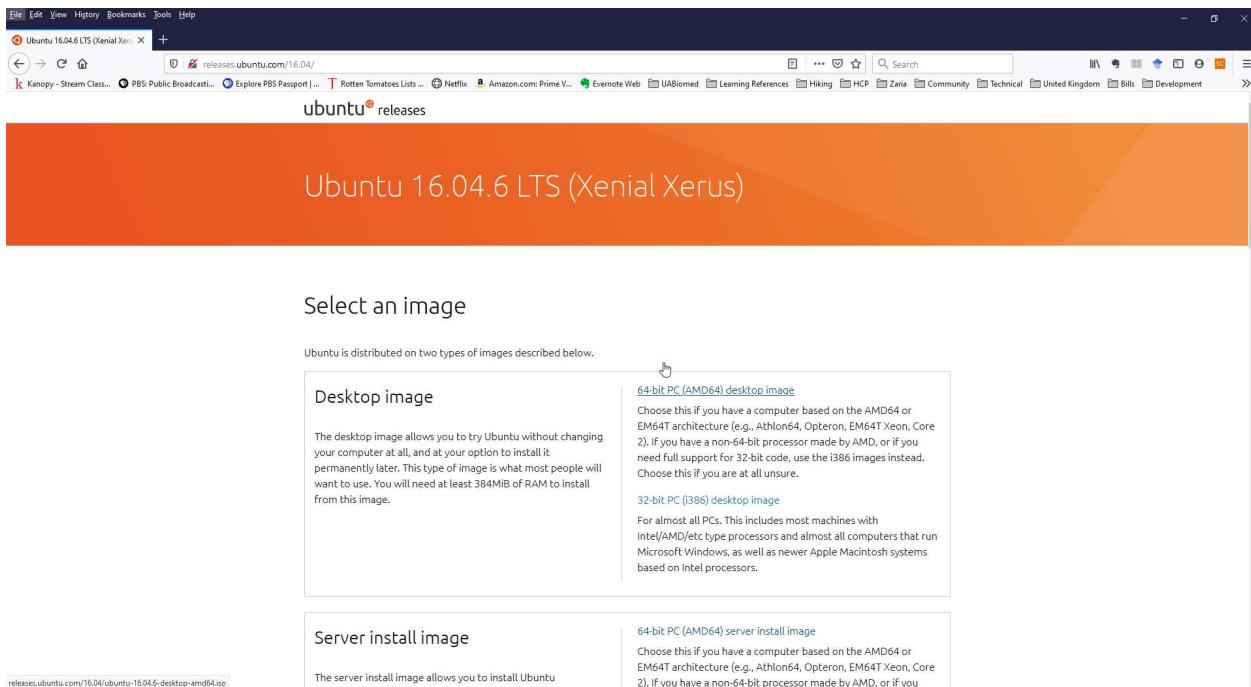




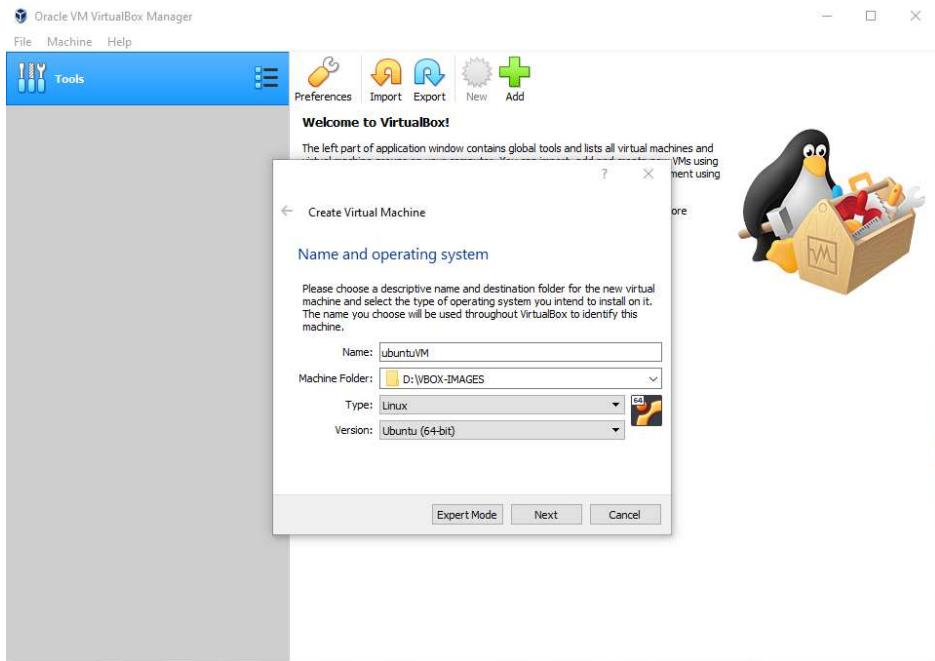
- d. Set Preference for where Virtual Box images are stored – choose location with as much GB room as possible



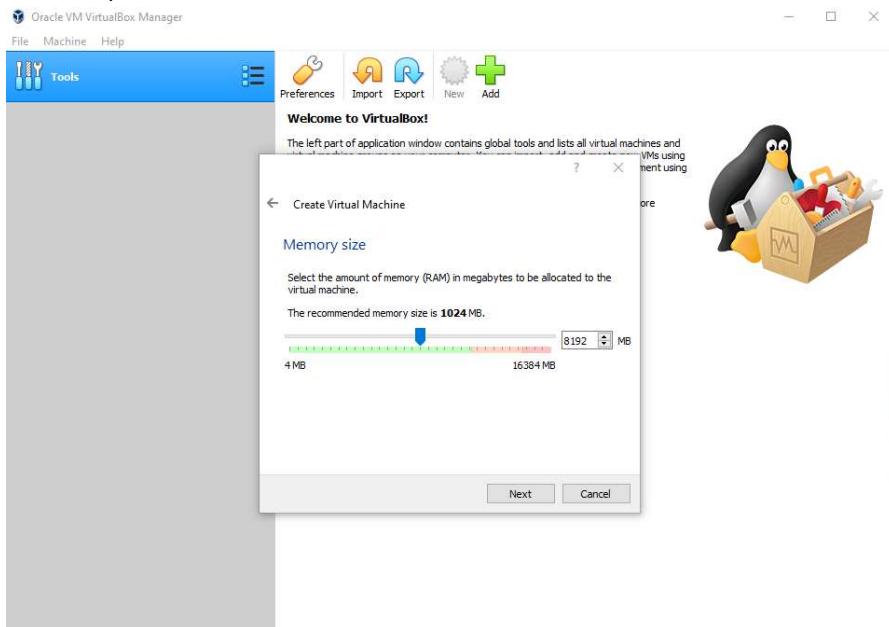
- e. Download Ubuntu Desktop Iso from <http://releases.ubuntu.com/16.04/>



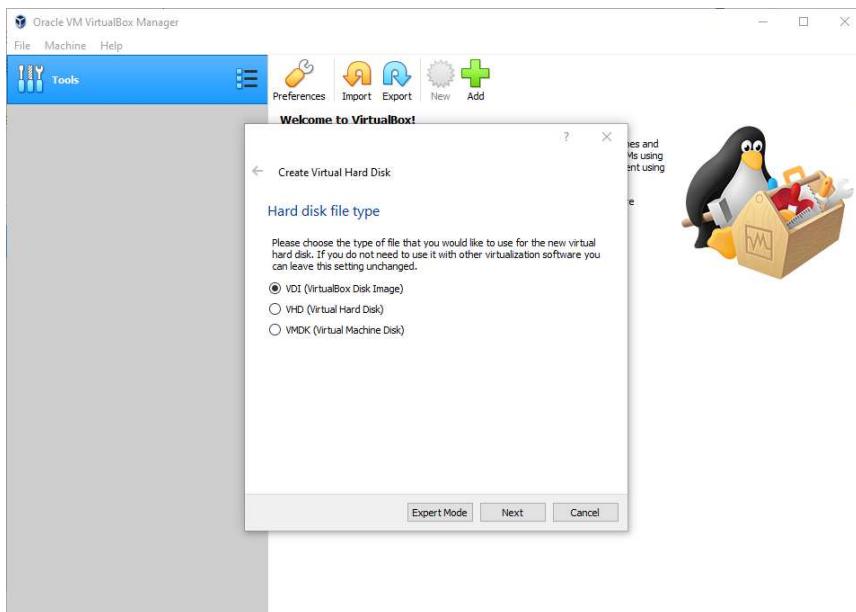
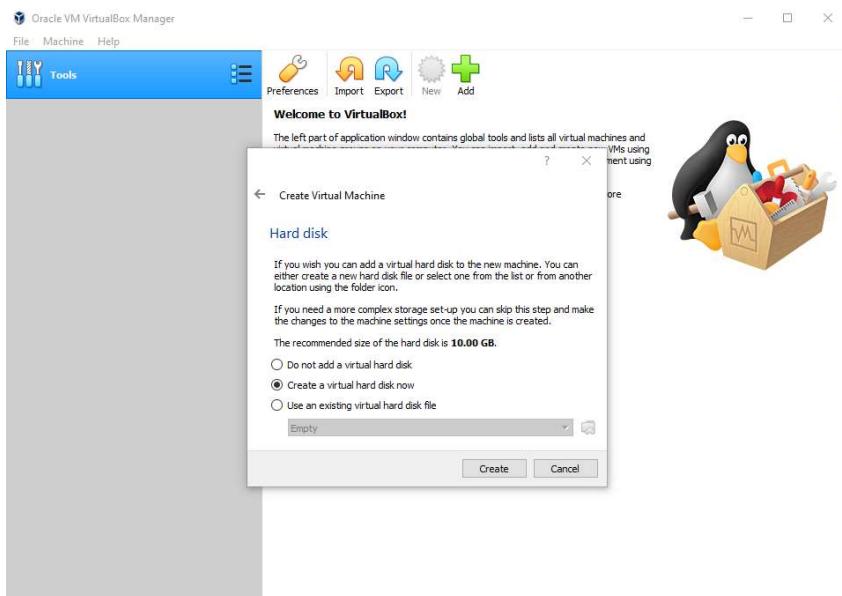
f. Start creation of Ubuntu VM by clicking **Machine > Add – select Ubuntu (64 bit)**

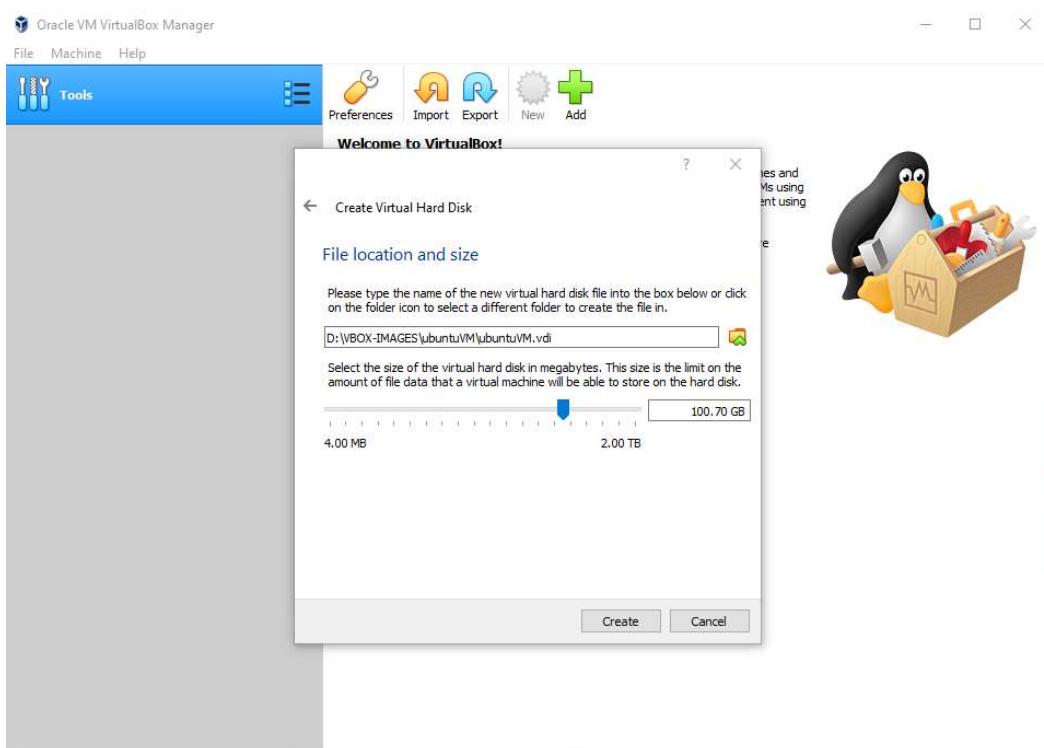
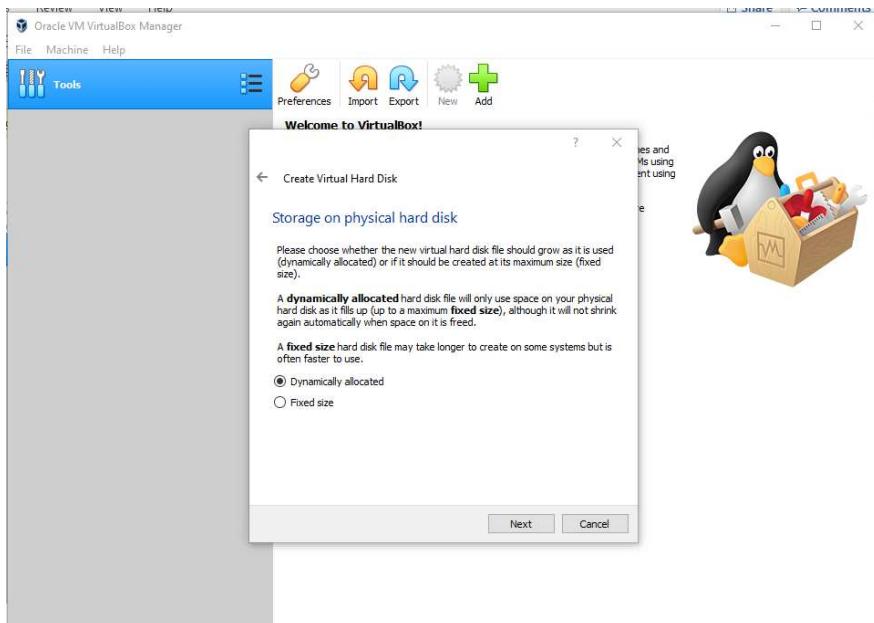


- g. Set memory (the more memory the faster your VM will run at the expense of your host machine)

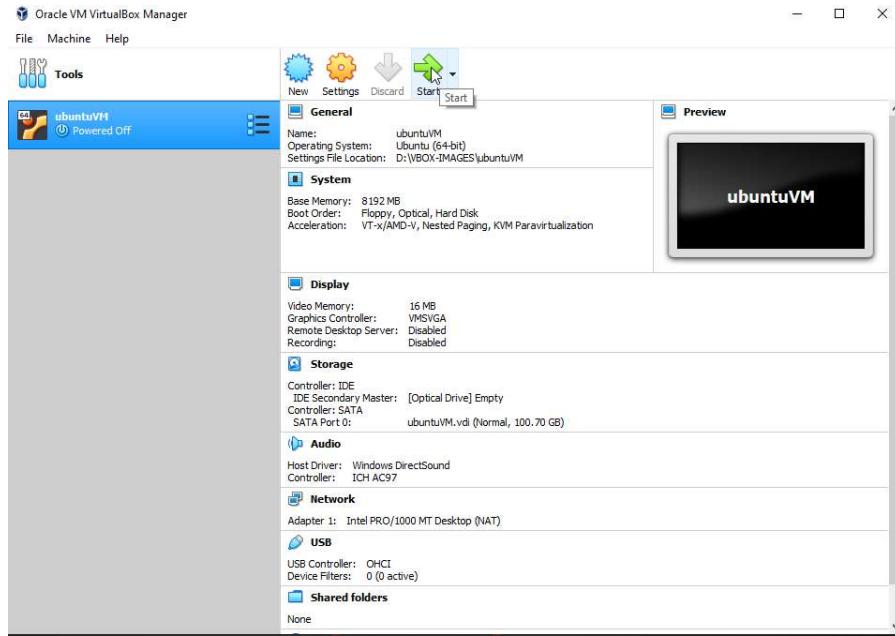


- h. Create hard disk – 100GB is created here

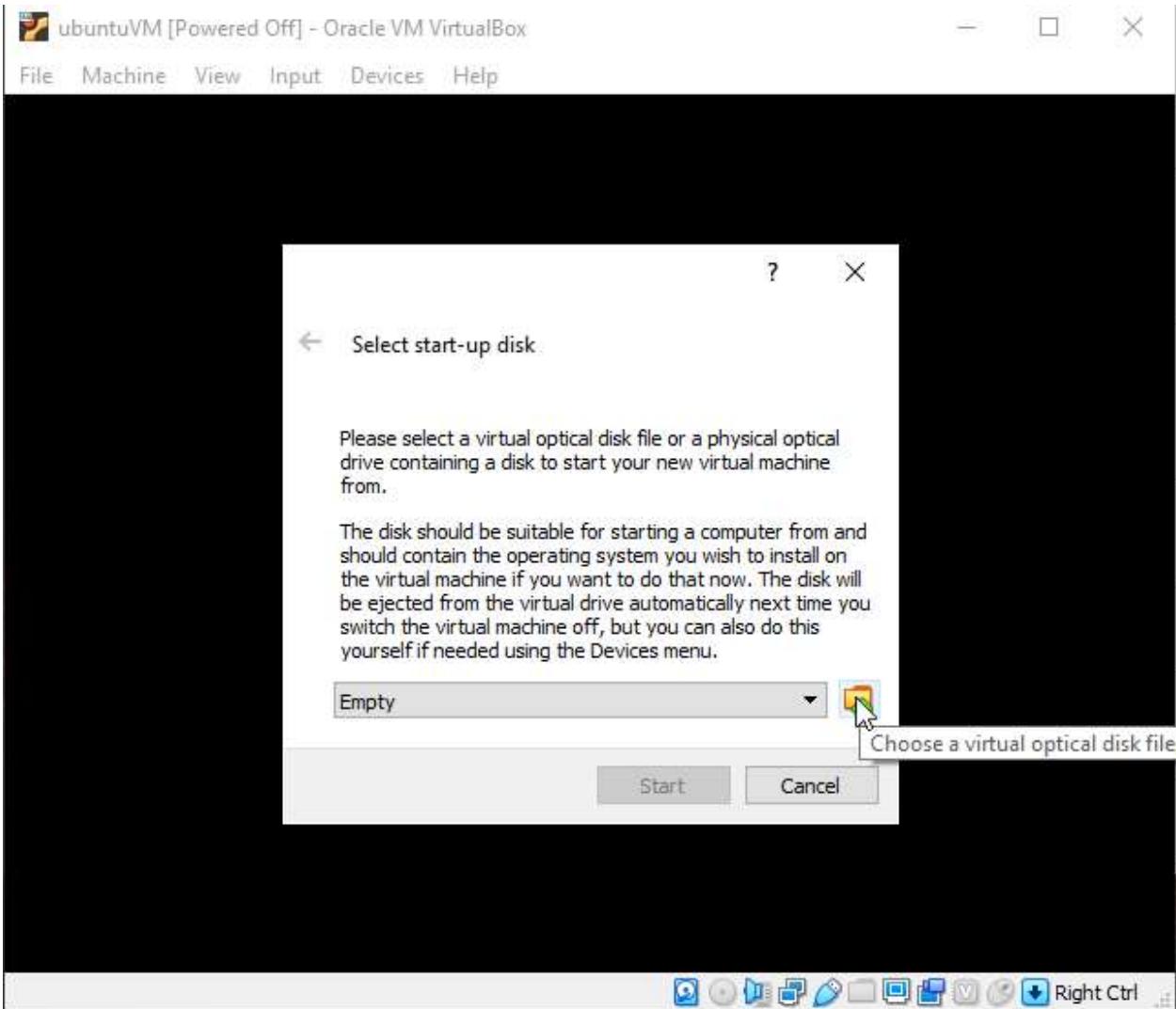


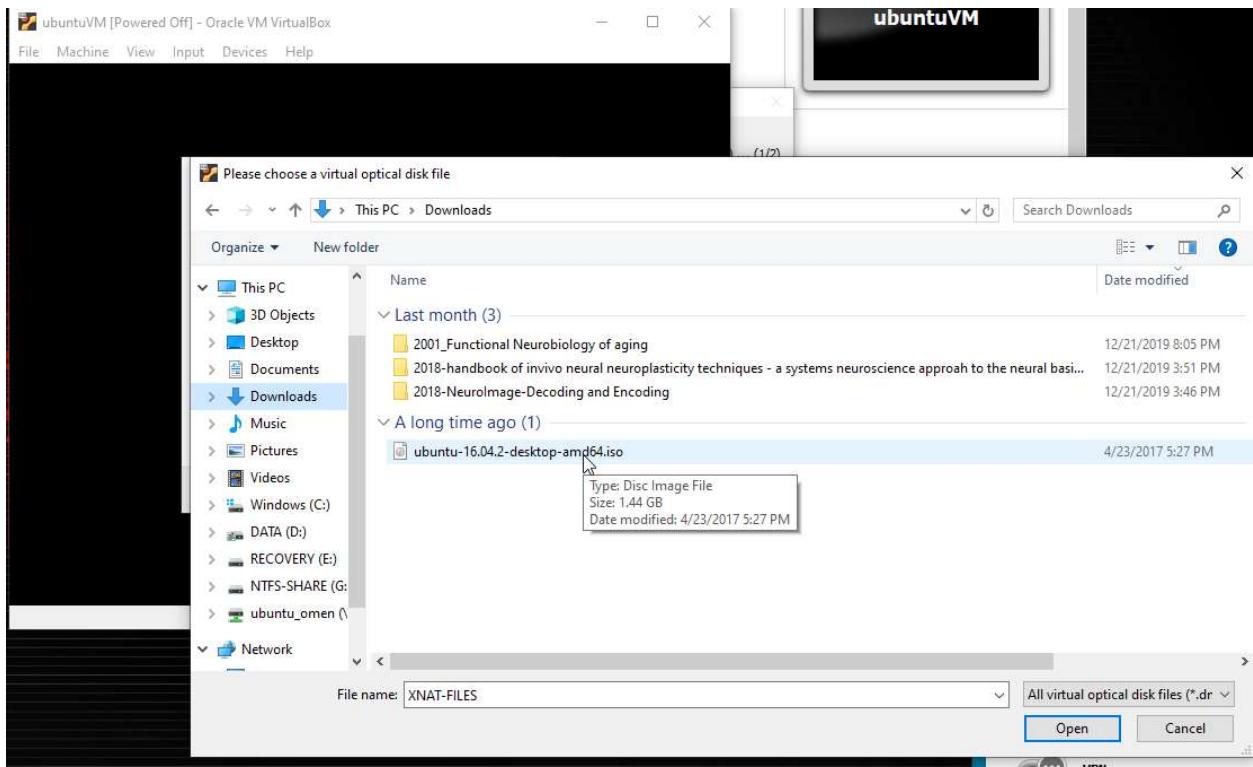


- i. Now that basic VM is created, its time to install Ubuntu – click **Start** -> to start the VM

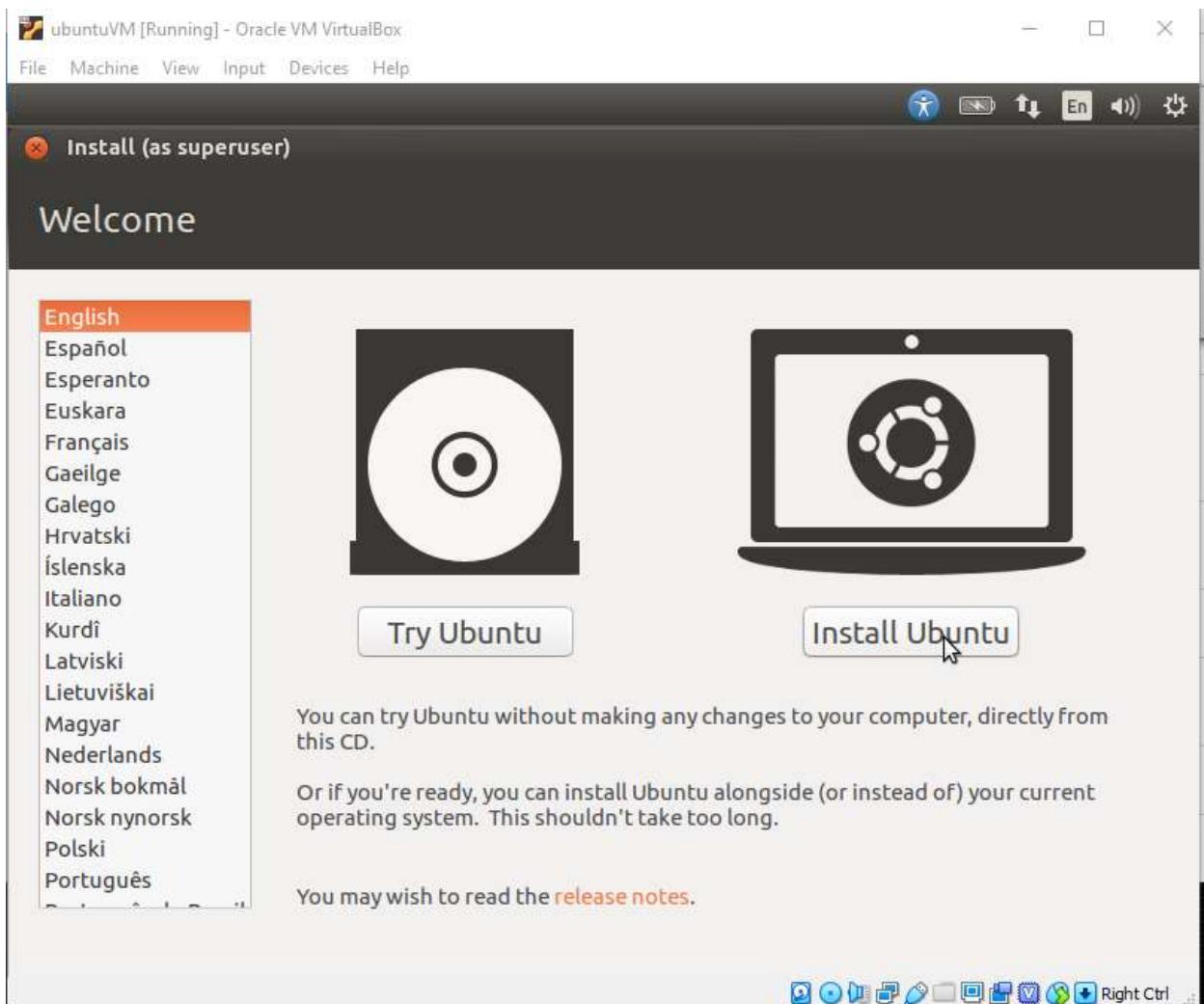


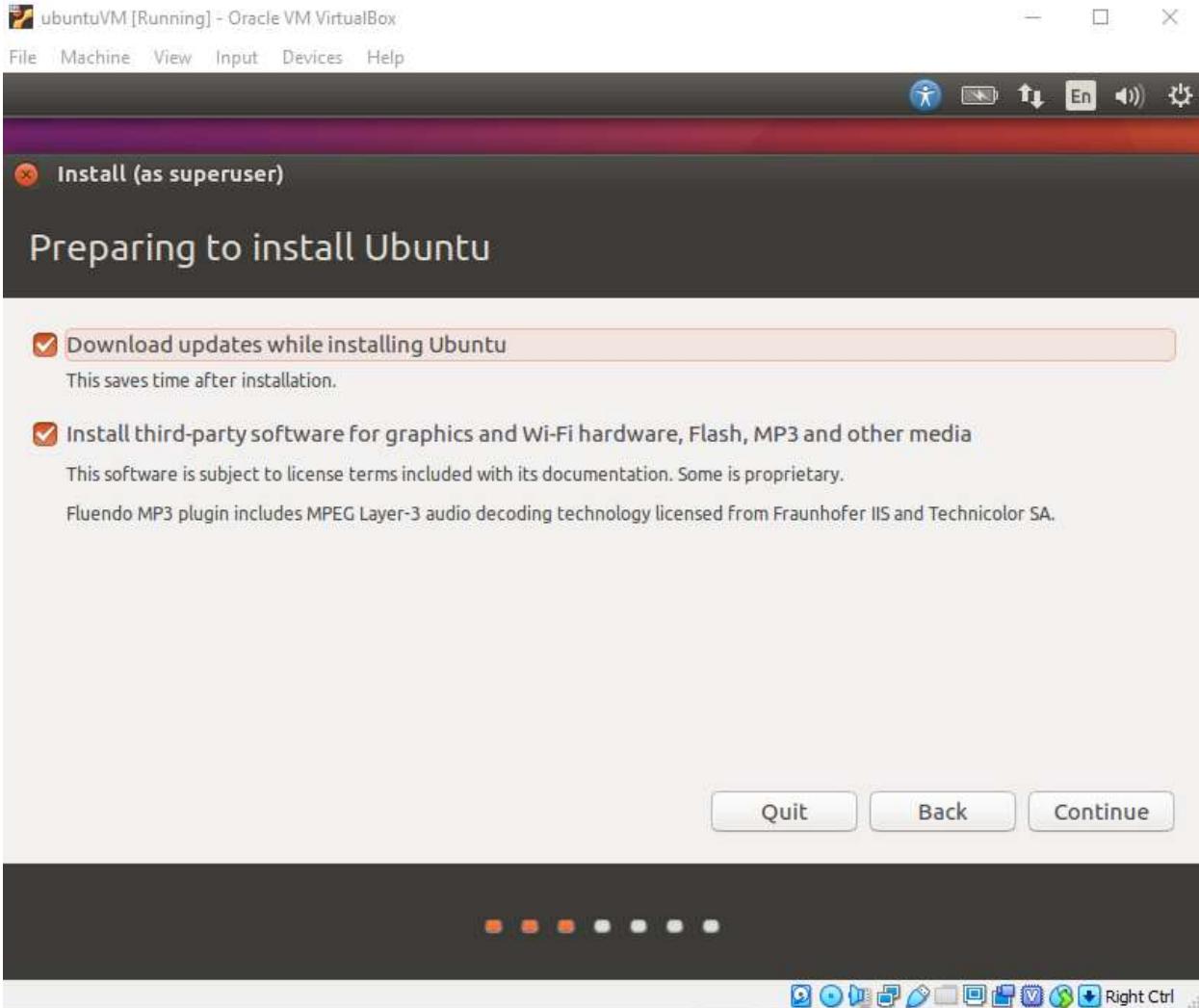
- j. Select the Ubuntu ISO and follow through steps in images below

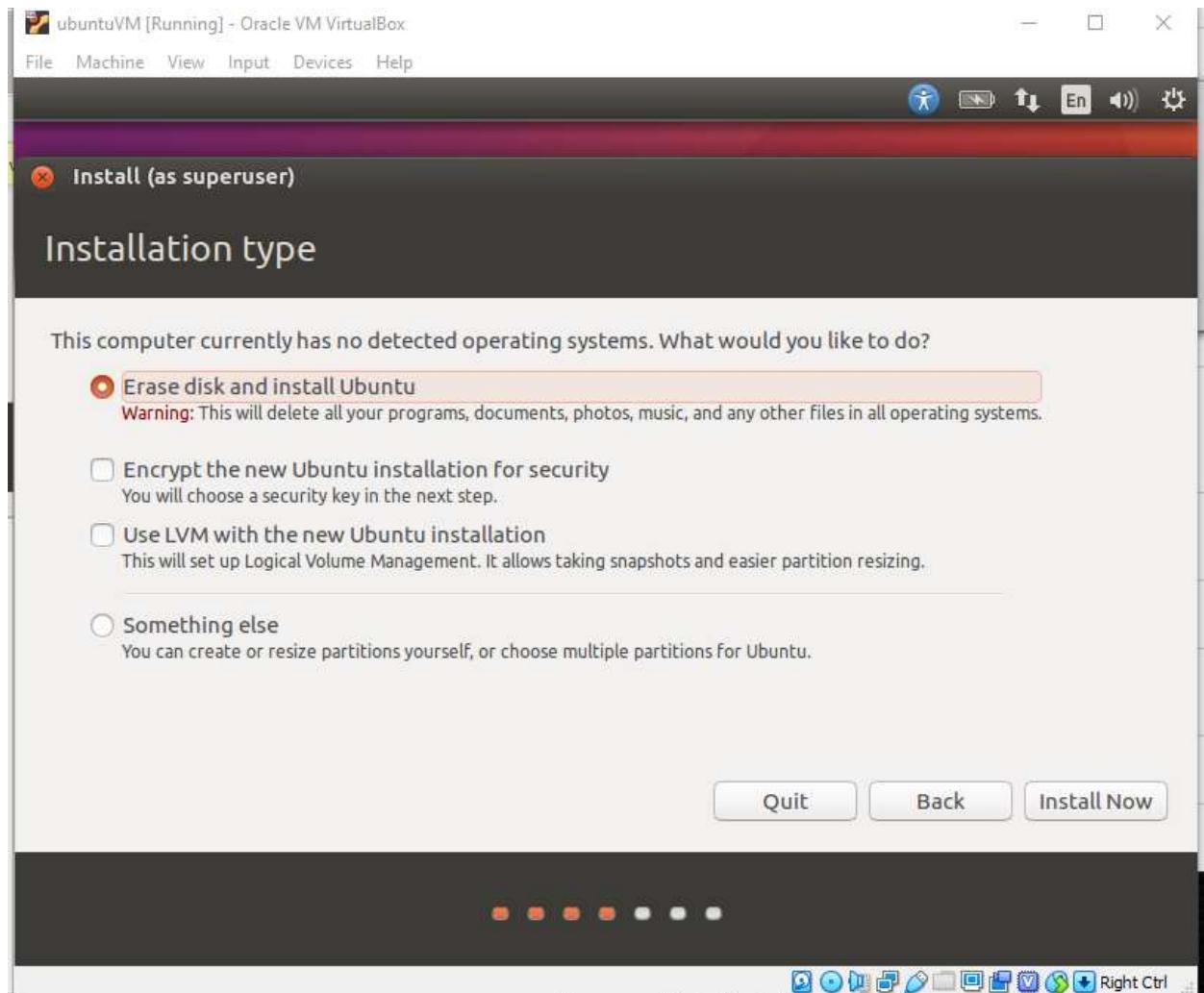


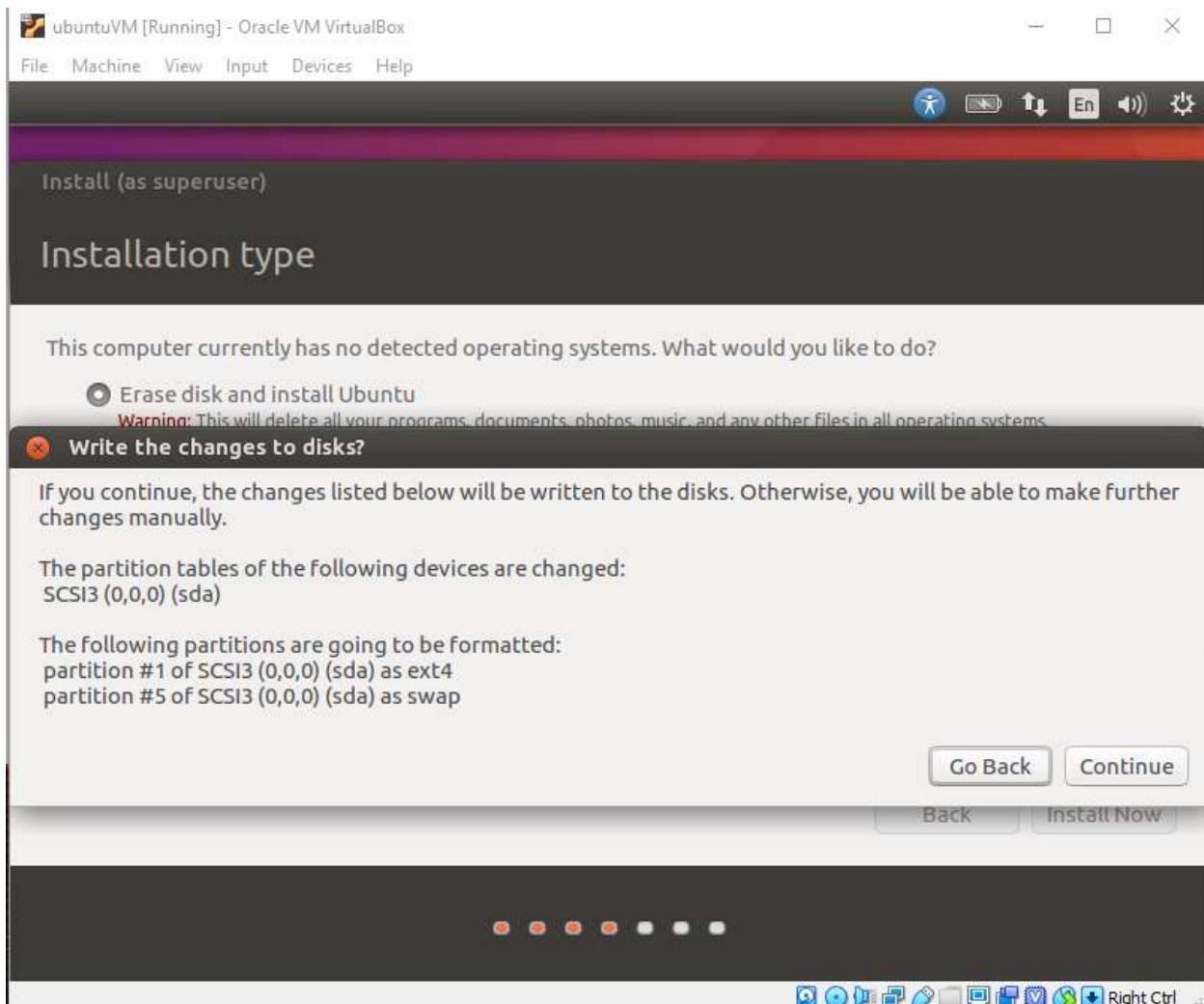


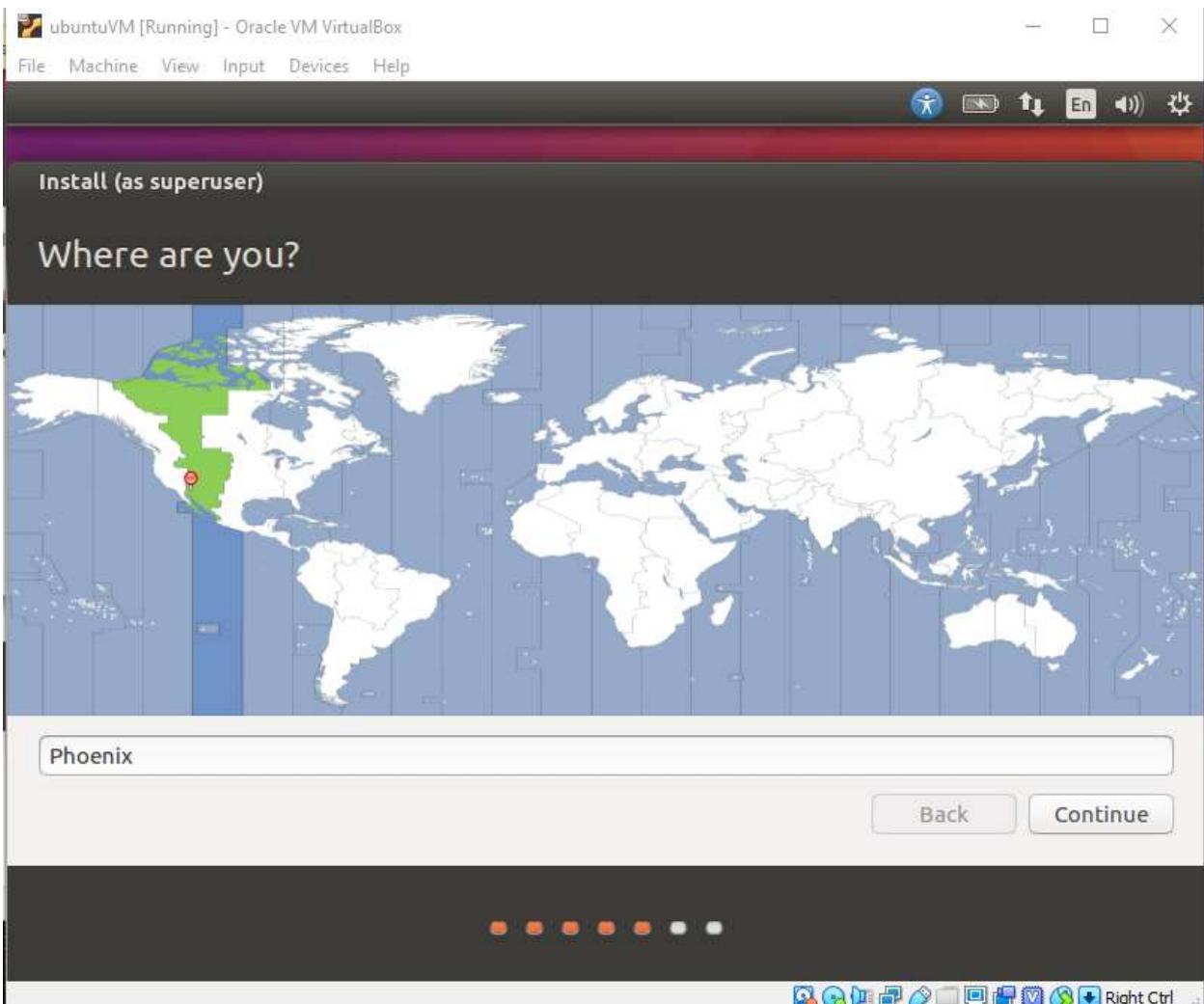
Select and click start

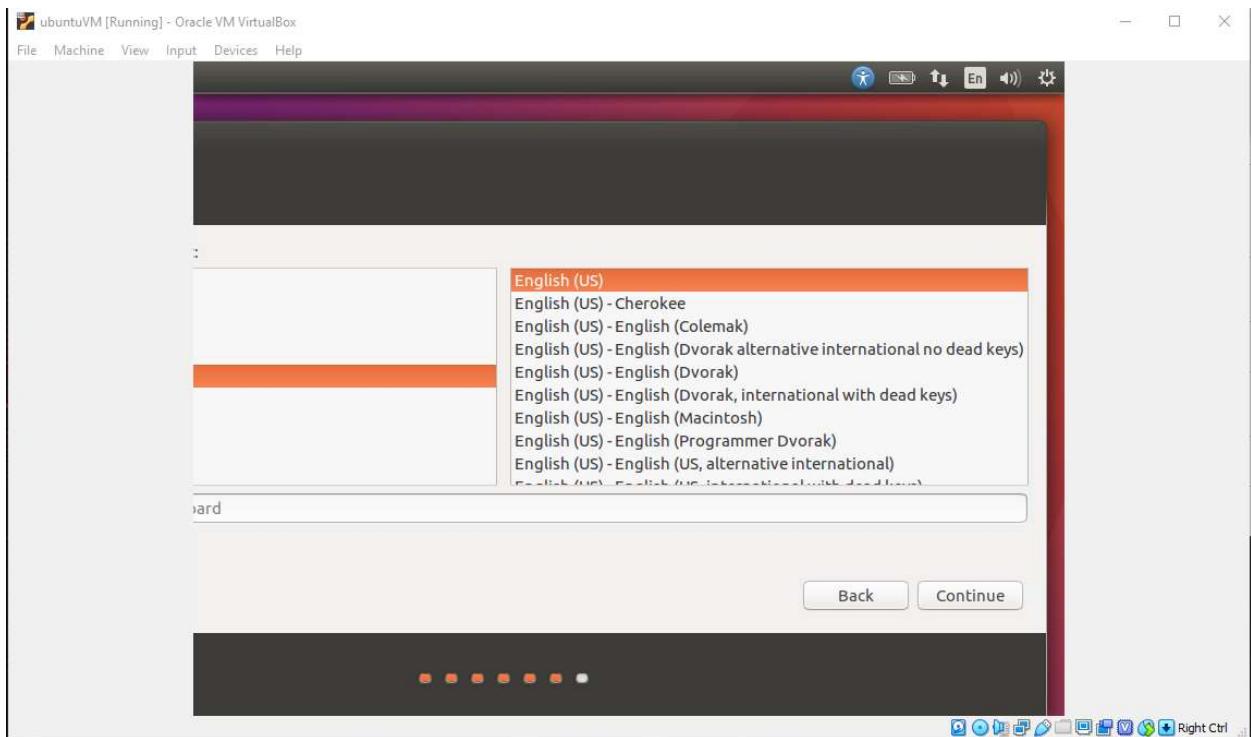






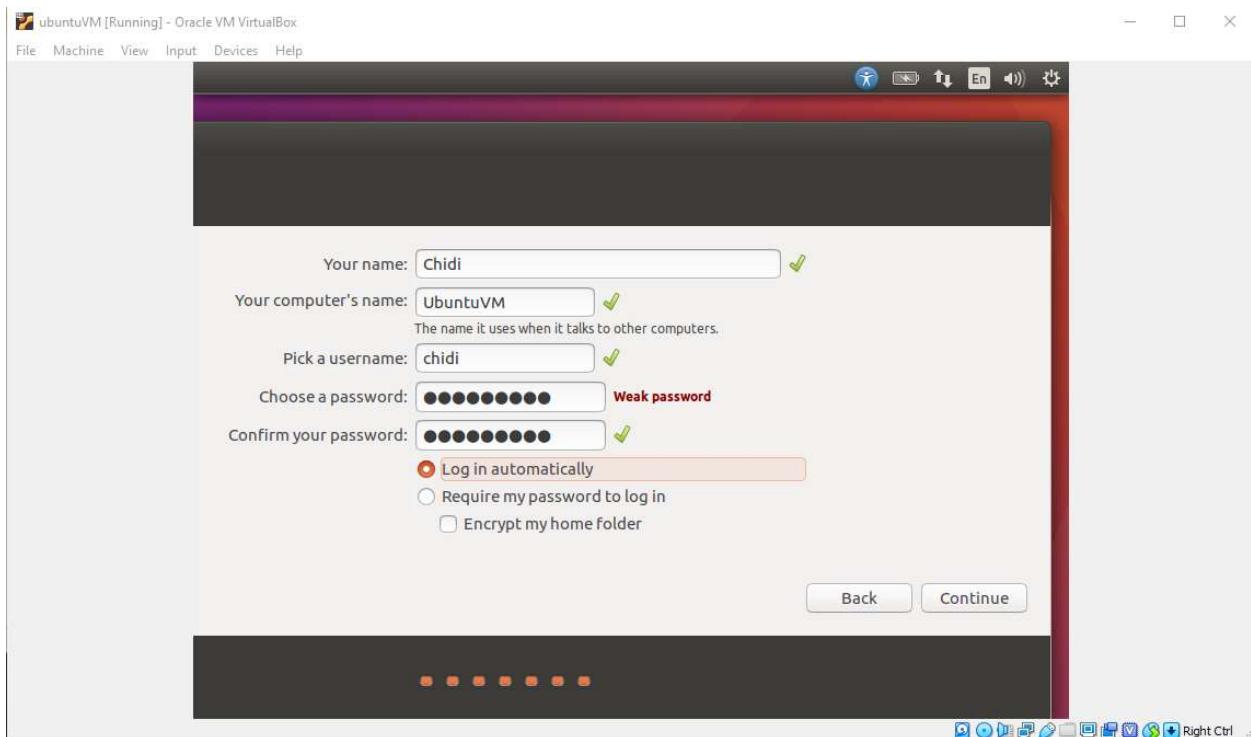


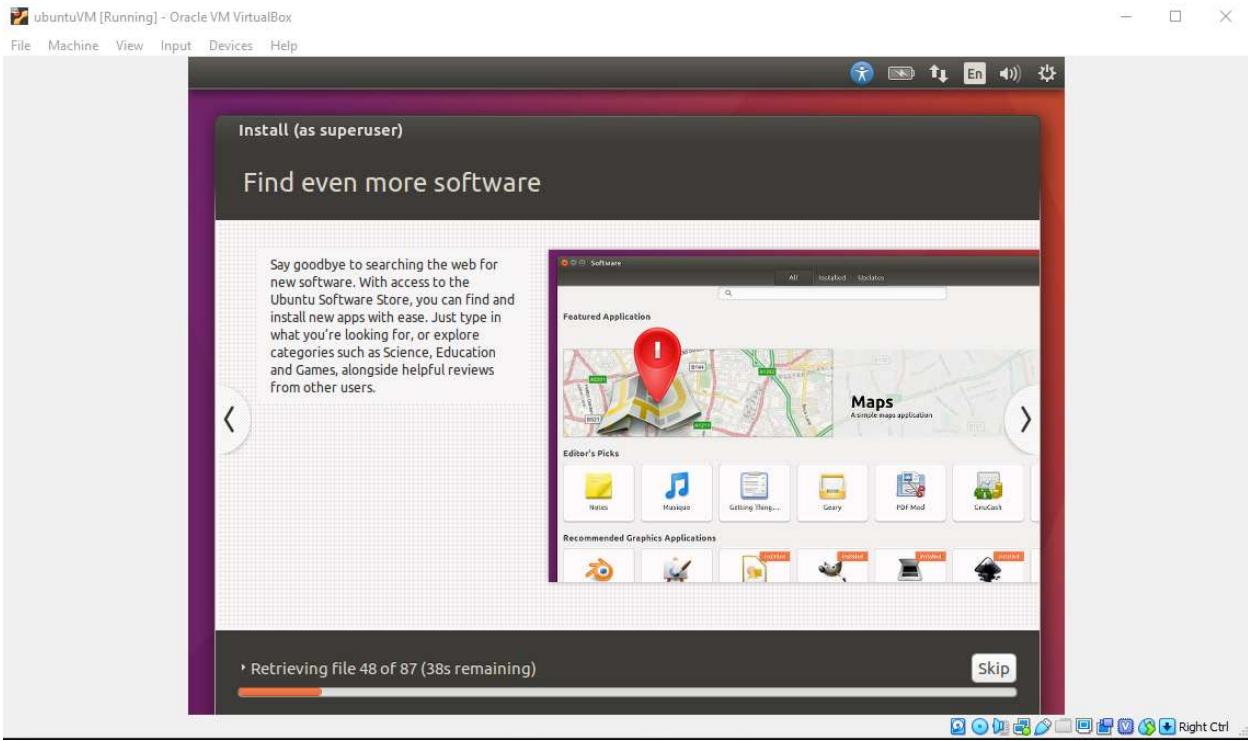


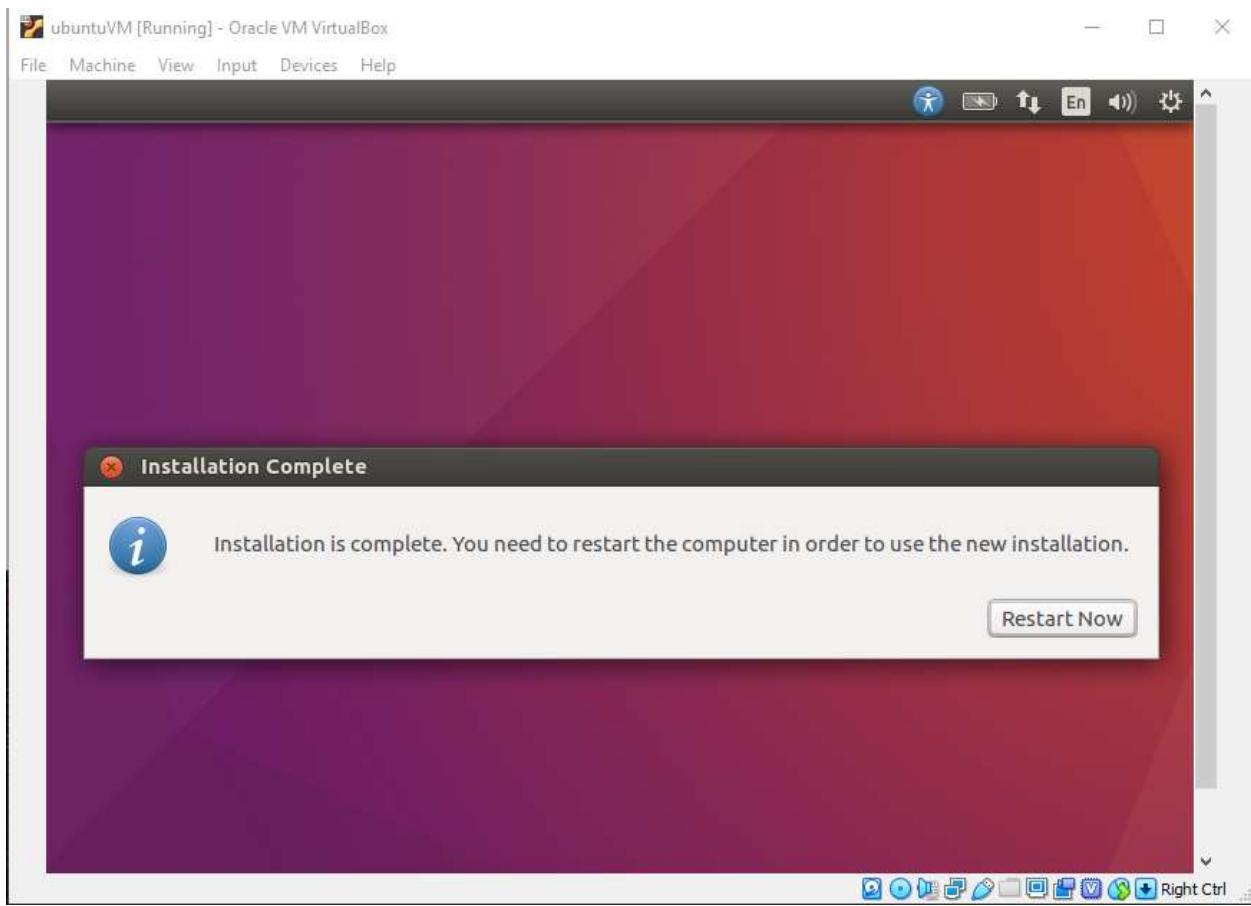


You may need to physically grab the window and move to the left to see “continue”

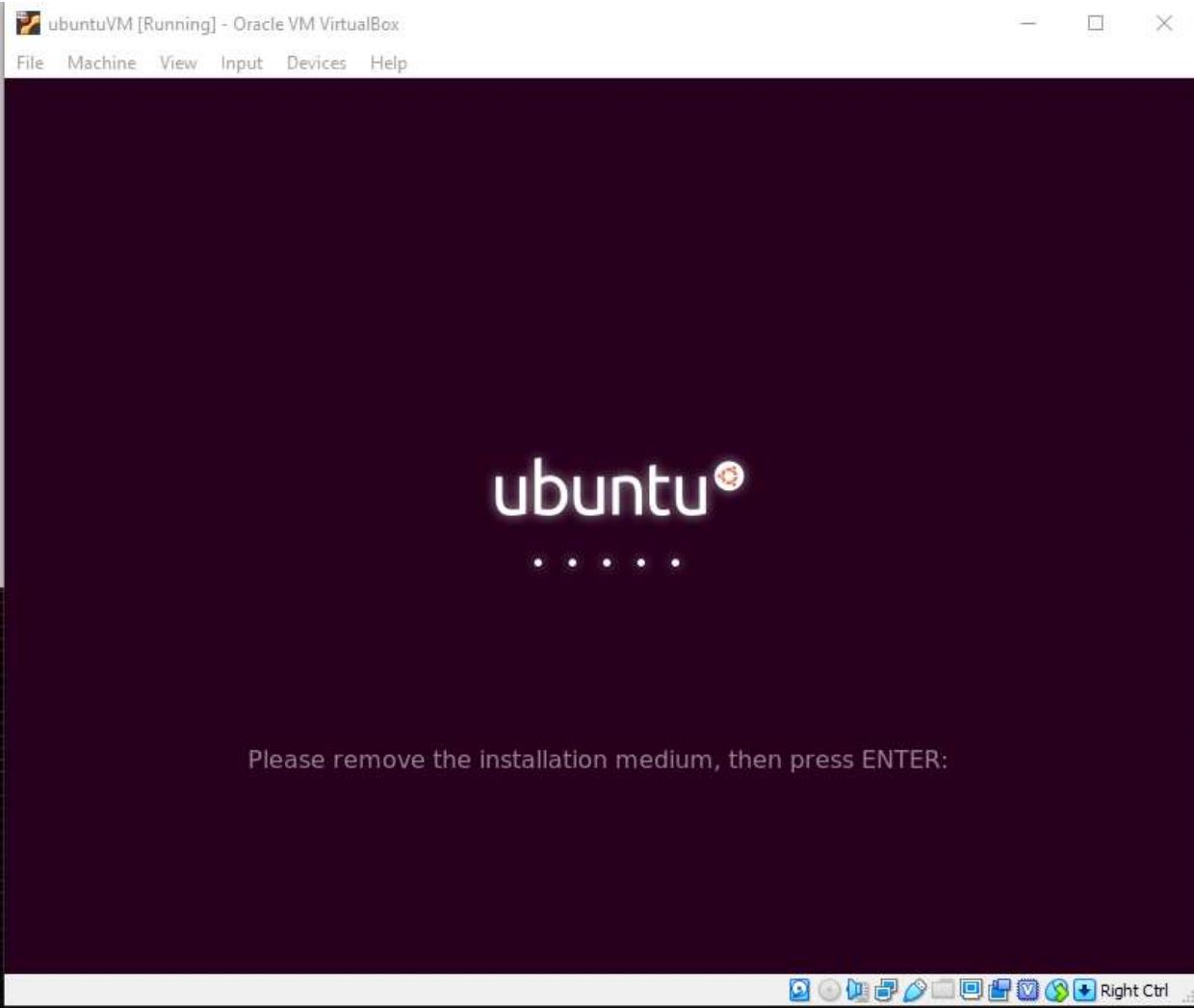
Provide your user name and credentials



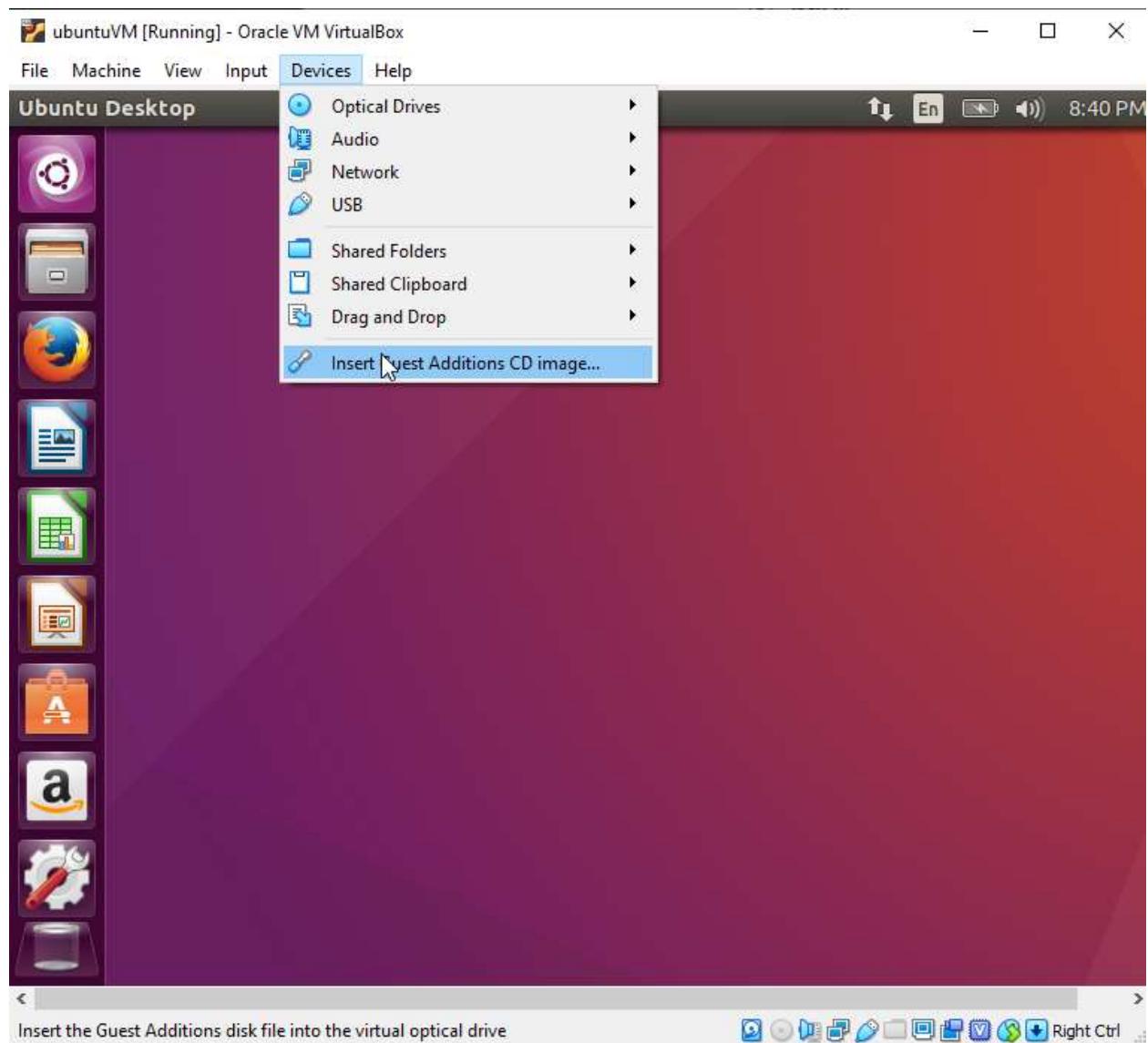




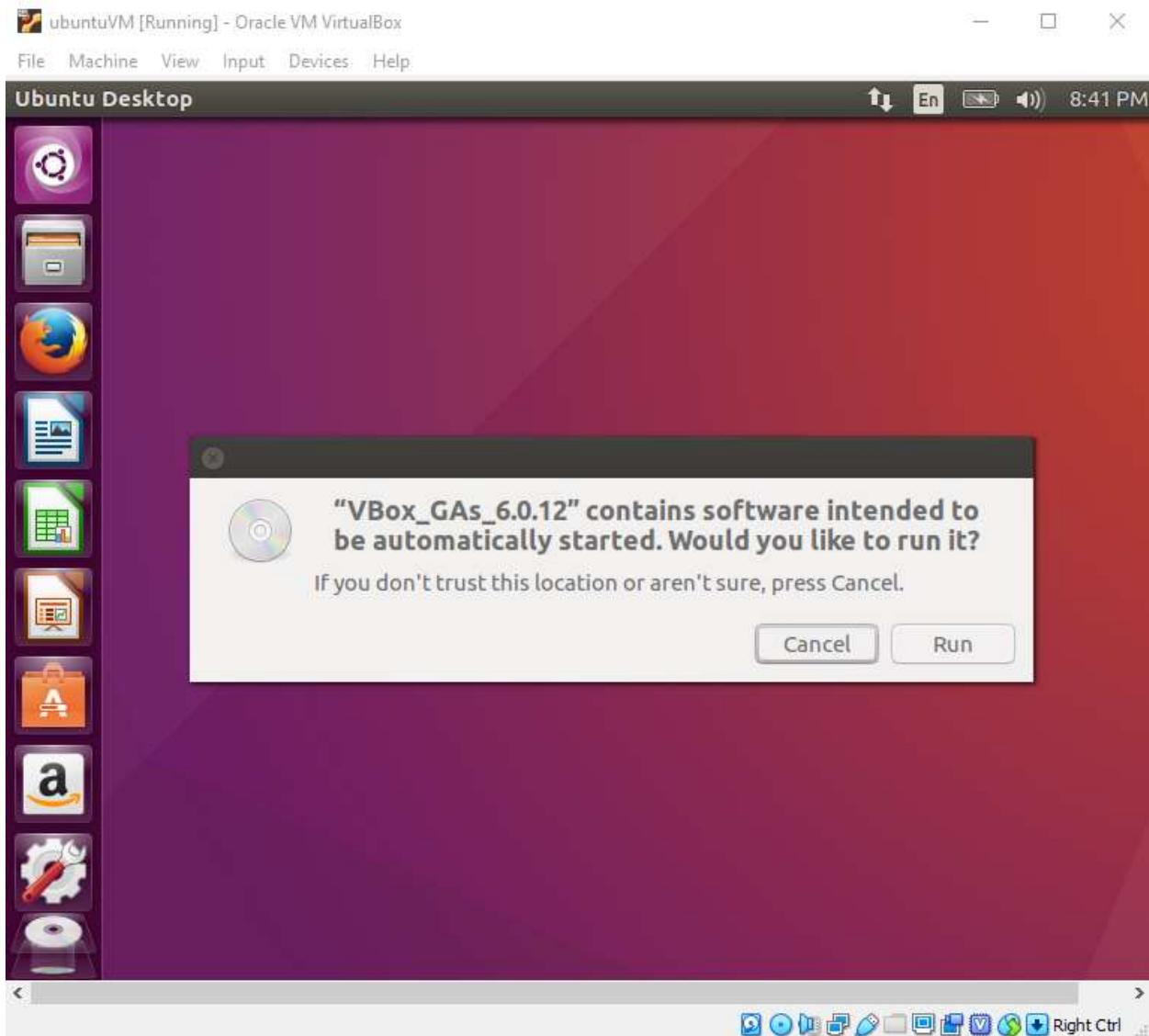
Click Restart Now



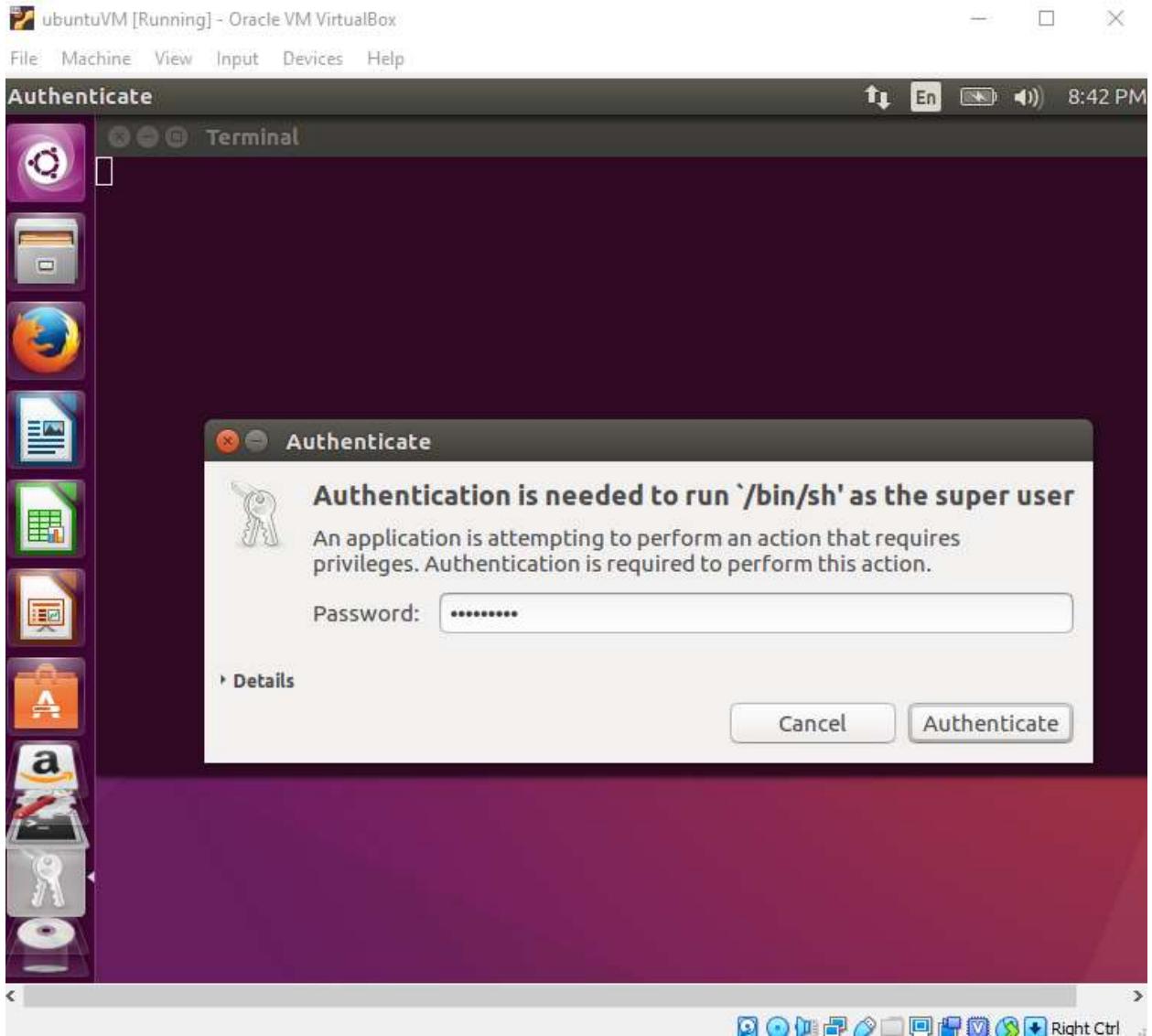
Press Enter



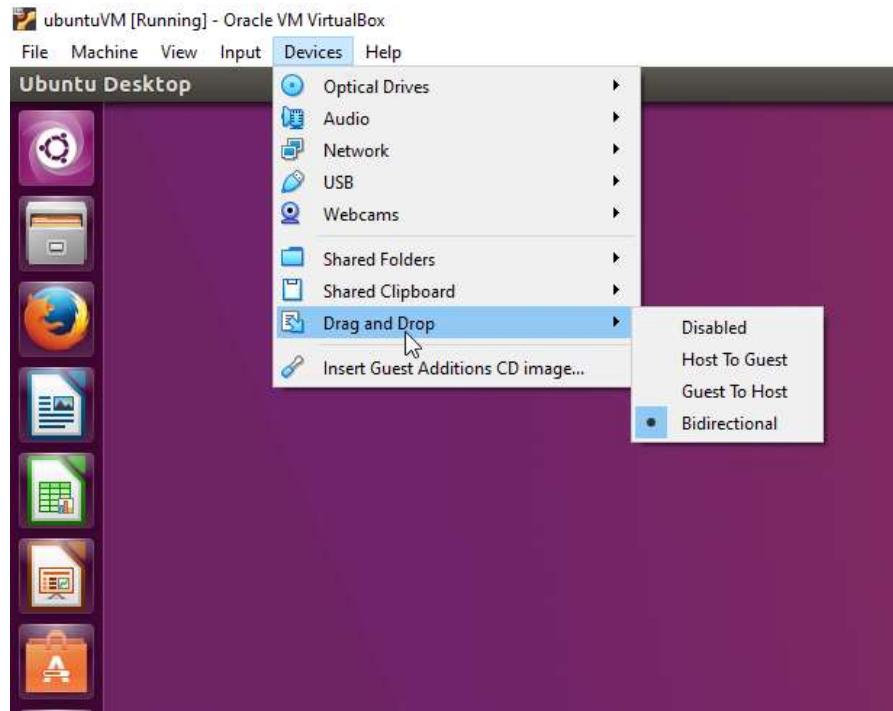
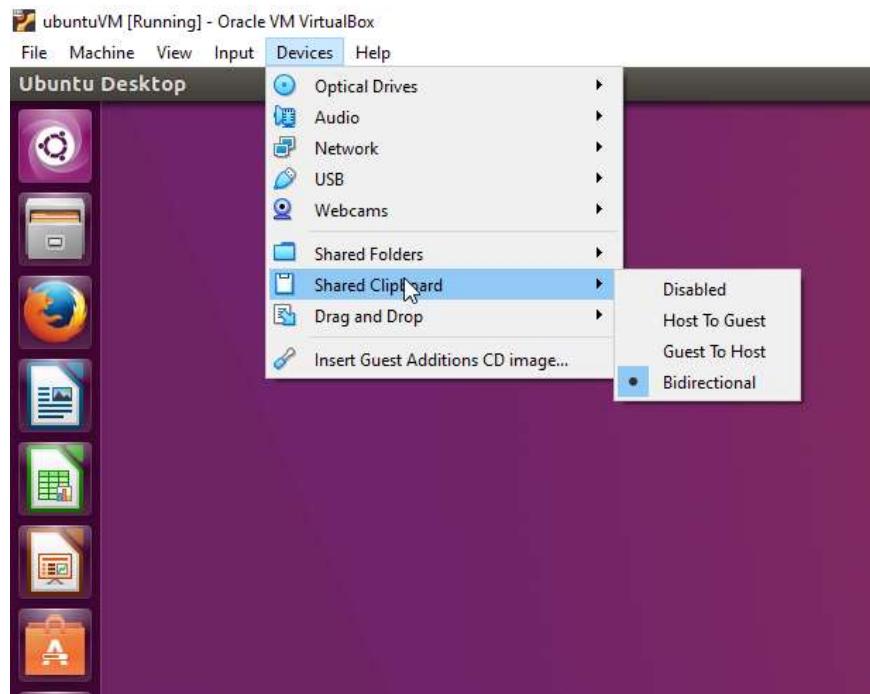
Insert the Guest Additions CD-image



Click Run



Select shared Clipboard and Drag and Drop



Ubuntu Desktop

↑ En 8:45 PM

About This Computer

Ubuntu Help...

System Settings...

Lock Ctrl+Alt+L

Guest Session

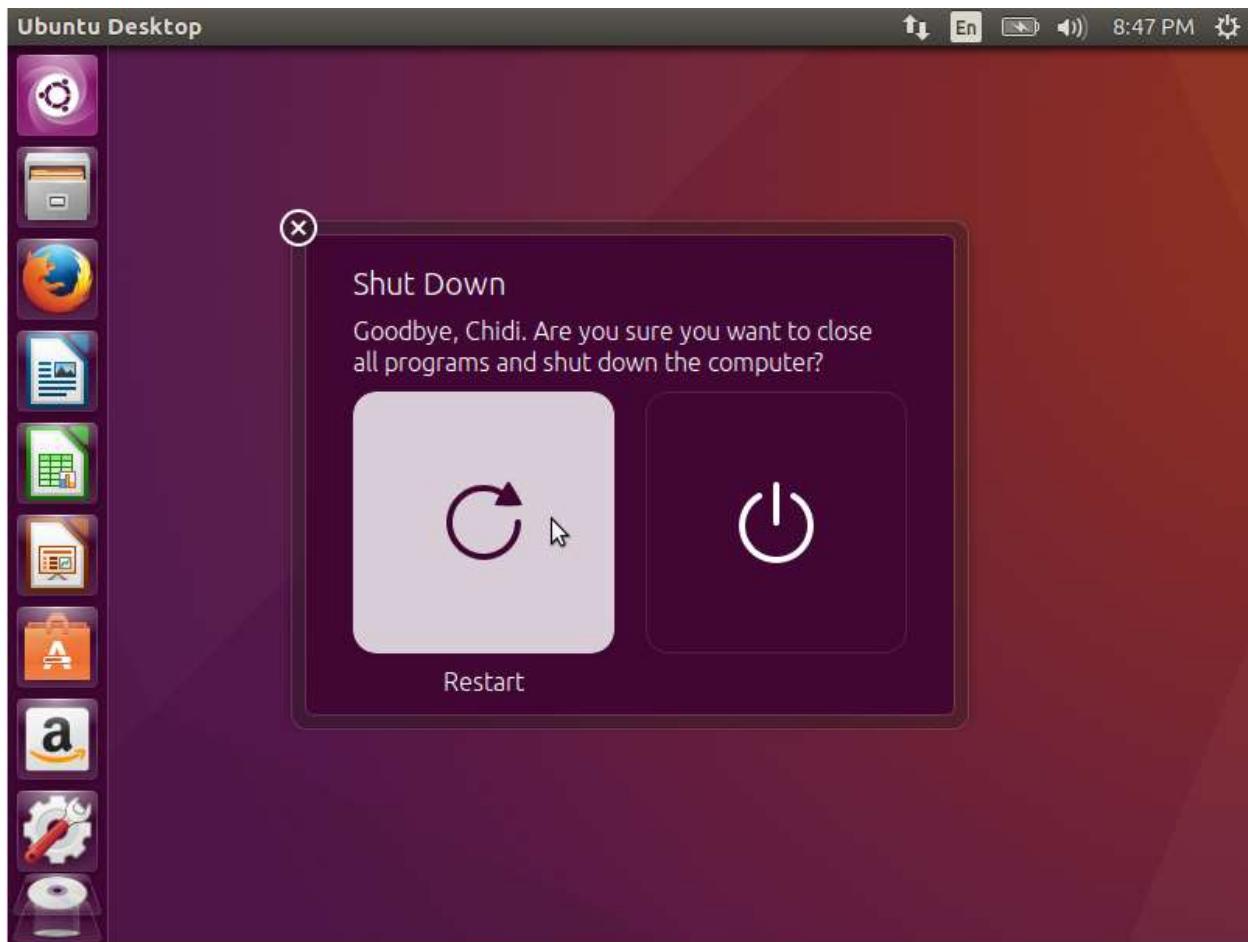
• Chidi

Log Out...

Suspend

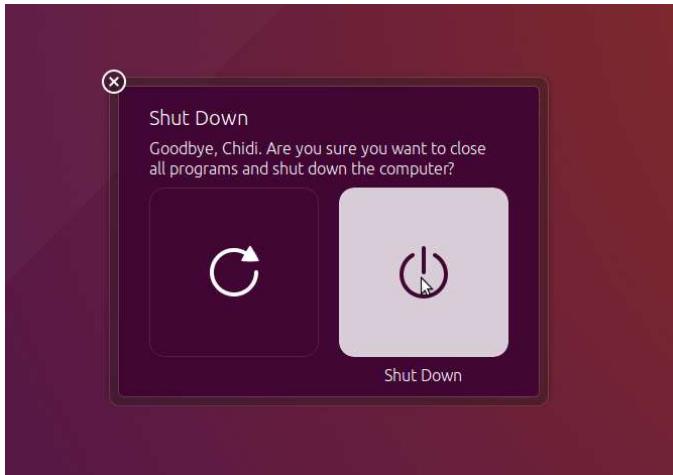
Shut Down...



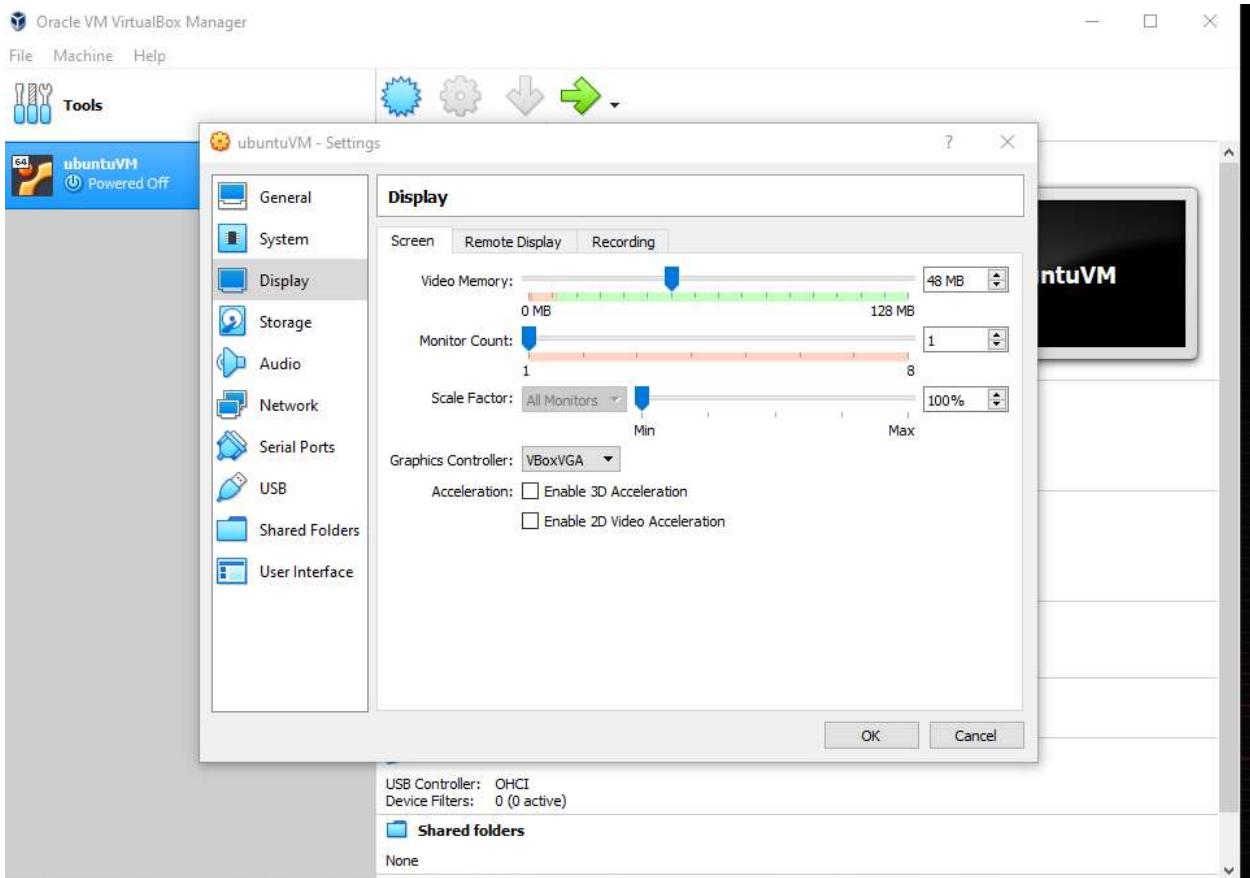


Shut down and restart to allow those changes

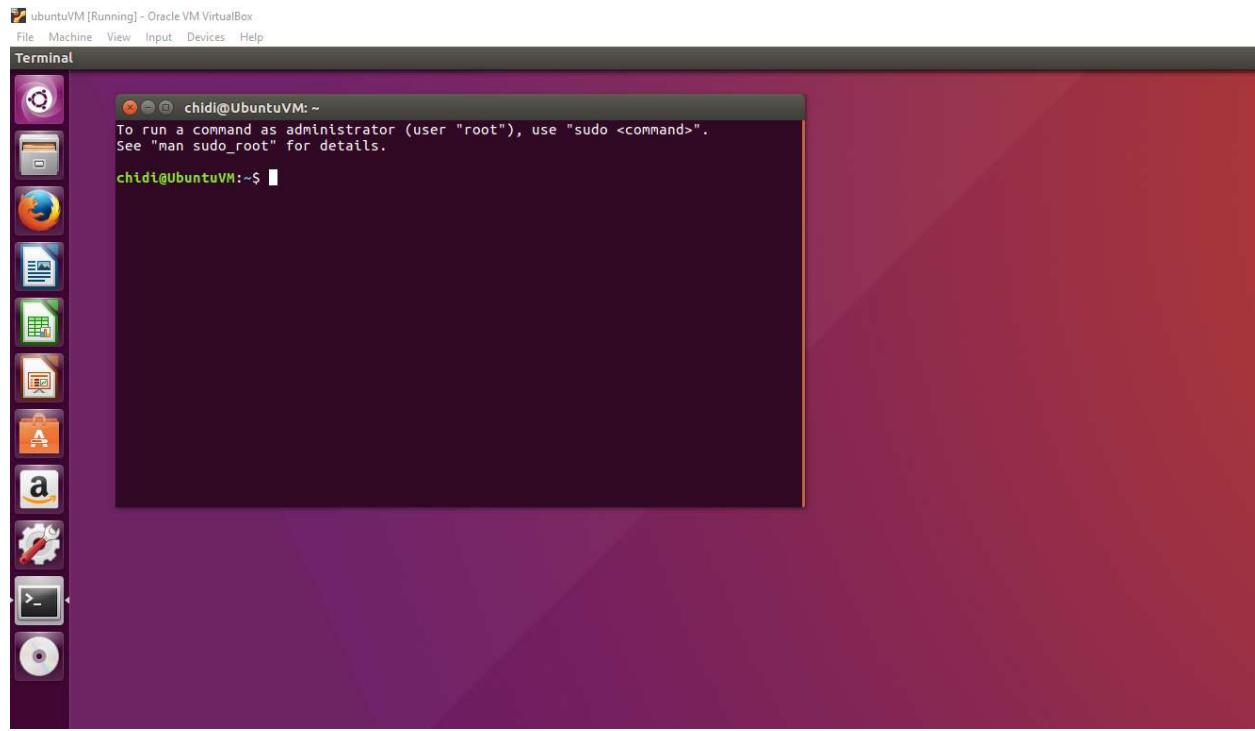
You may find that you cannot see the whole VM - if that is the case you may need to change your Video Card settings. Shutdown the Ubuntu VM and in settings change the Display memory and the Graphics Controller. You can change other settings of your VM this way (e.g. memory etc,..)



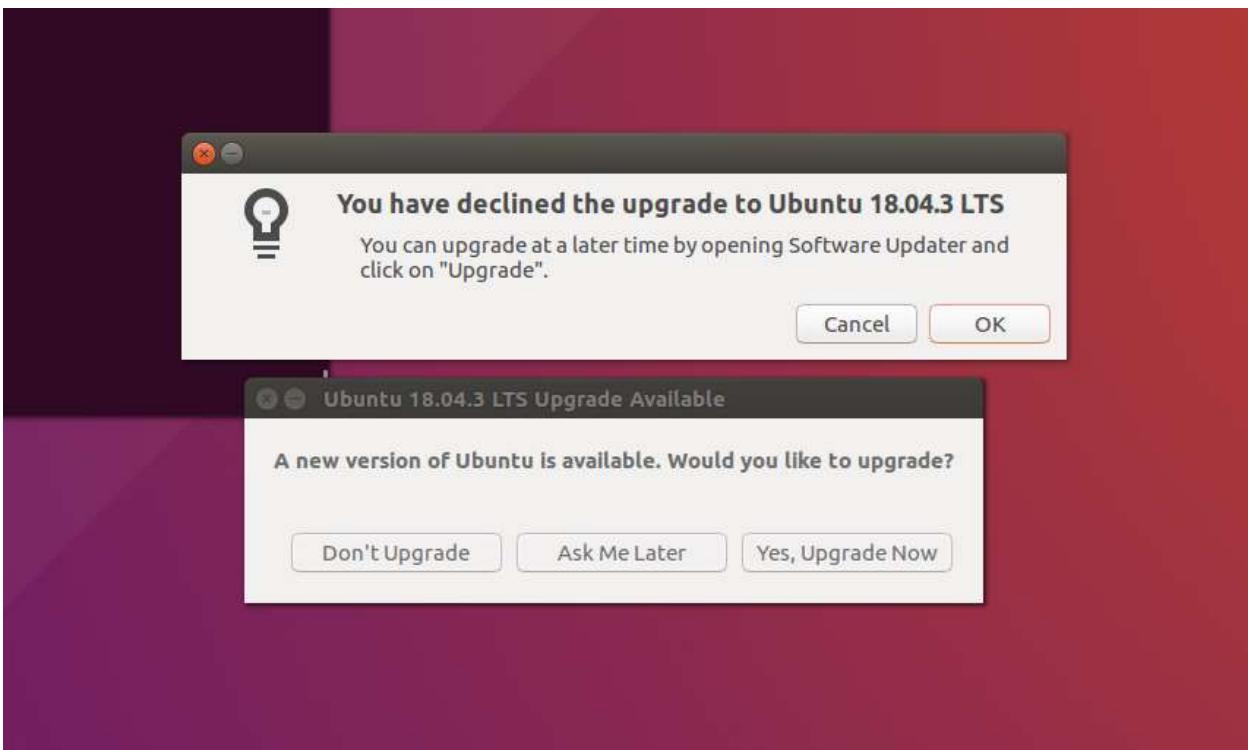
I found that VboxVGA helped with OpenGL 3.0 applications in Ubuntu!



2. Install Singularity on the Ubuntu Box



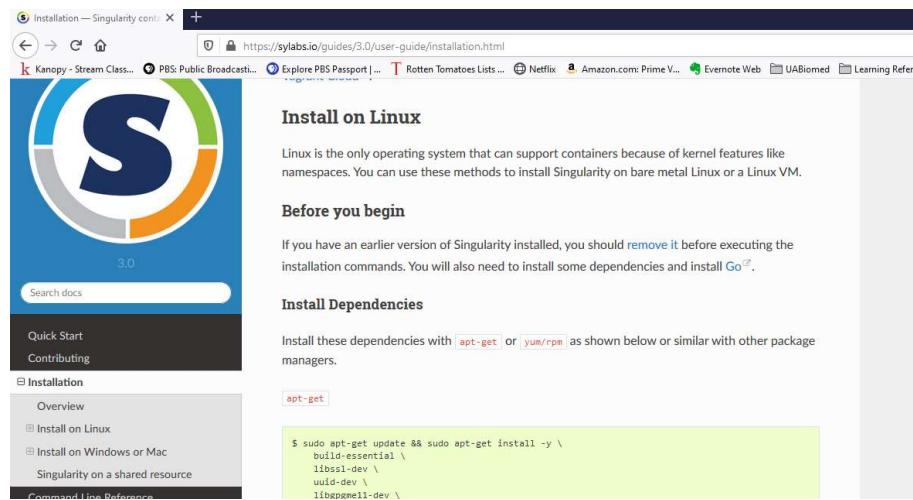
Open a terminal using CTRL-ALT-t



If you see the message above during the course of the installation to upgrade to Ubuntu 18.04 then click “Don’t Upgrade” and click “OK”

The instructions we will be following are referenced at the link below – but you won’t need them as I have replicated all the steps further below:

<https://sylabs.io/guides/3.0/user-guide/installation.html>

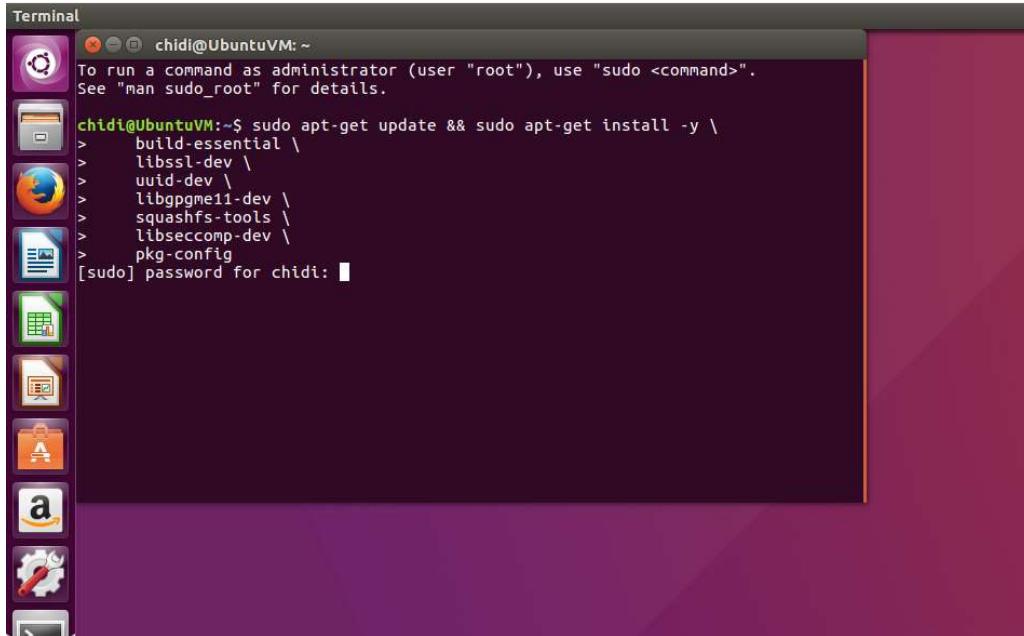


A screenshot of a web browser displaying the "Installation" section of the Singularity 3.0 user guide. The page features a large "Install on Linux" heading with a sub-section "Before you begin" containing instructions for removing previous versions and installing dependencies. A code block shows the apt-get command for dependency installation. The left sidebar includes links for "Quick Start", "Contributing", and sections on "Installation", "Overview", "Install on Linux", "Install on Windows or Mac", and "Singularity on a shared resource".

```
$ sudo apt-get update && sudo apt-get install -y \
    build-essential \
    libssl-dev \
    uuid-dev \
    libgpgme11-dev \
```

a. Copy and paste the commands below to install pre-reqs:

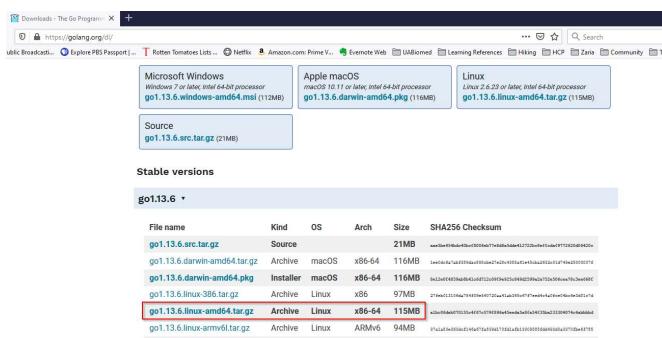
```
sudo apt-get update && sudo apt-get install -y \
build-essential \
libssl-dev \
uuid-dev \
libgpgme11-dev \
squashfs-tools \
libseccomp-dev \
pkg-config
```

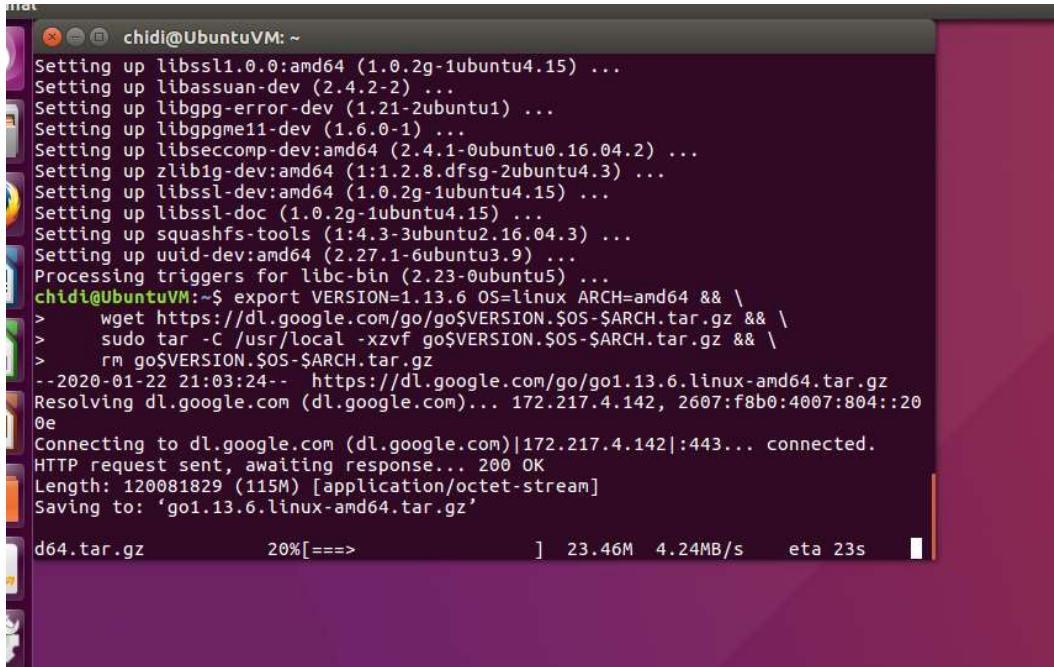


b. Copy and paste the commands below to install **Go version 1.13.6** which is the latest version of go:

```
export VERSION=1.13.6 OS=linux ARCH=amd64 && \
wget https://dl.google.com/go/go$VERSION.$OS-$ARCH.tar.gz && \
sudo tar -C /usr/local -xzvf go$VERSION.$OS-$ARCH.tar.gz && \
rm go$VERSION.$OS-$ARCH.tar.gz
```

Versions of go can be seen here - <https://golang.org/dl/>



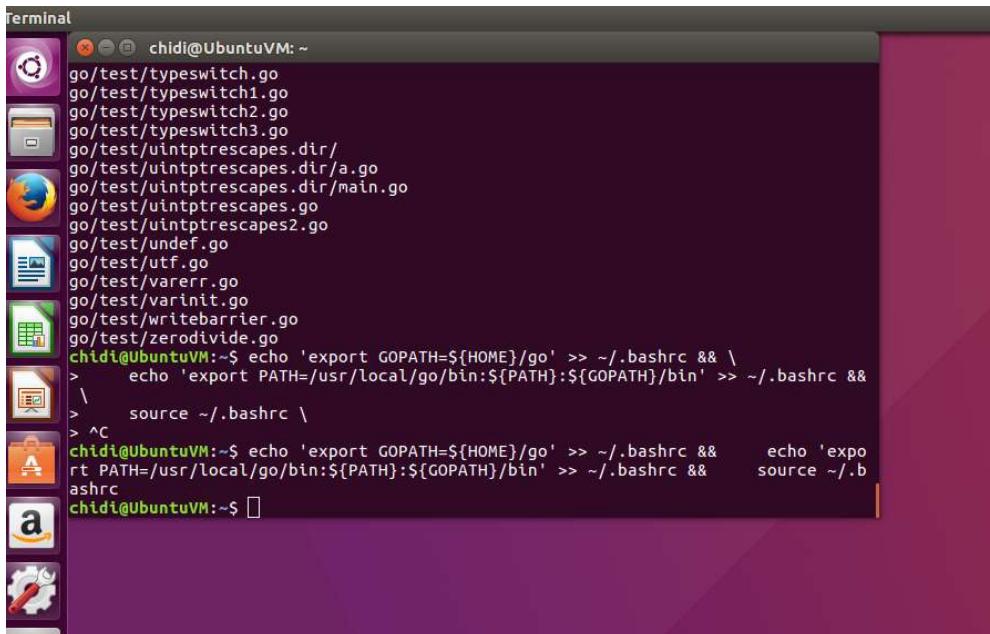


```
chidi@UbuntuVM: ~
Setting up libssl1.0.0:amd64 (1.0.2g-1ubuntu4.15) ...
Setting up libassuan-dev (2.4.2-2) ...
Setting up libpgpg-error-dev (1.21-2ubuntu1) ...
Setting up libpgpgme11-dev (1.6.0-1) ...
Setting up libseccomp-dev:amd64 (2.4.1-0ubuntu0.16.04.2) ...
Setting up zlib1g-dev:amd64 (1:1.2.8.dfsg-2ubuntu4.3) ...
Setting up libssl-dev:amd64 (1.0.2g-1ubuntu4.15) ...
Setting up libssl-doc (1.0.2g-1ubuntu4.15) ...
Setting up squashfs-tools (1:4.3-3ubuntu2.16.04.3) ...
Setting up uuid-dev:amd64 (2.27.1-6ubuntu3.9) ...
Processing triggers for libc-bin (2.23-0ubuntu5) ...
chidi@UbuntuVM:~$ export VERSION=1.13.6 OS=linux ARCH=amd64 && \
>     wget https://dl.google.com/go/go$VERSION.$OS-$ARCH.tar.gz && \
>     sudo tar -C /usr/local -xvf go$VERSION.$OS-$ARCH.tar.gz && \
>     rm go$VERSION.$OS-$ARCH.tar.gz
--2020-01-22 21:03:24--  https://dl.google.com/go/go1.13.6.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 172.217.4.142, 2607:f8b0:4007:804::20
Oe
Connecting to dl.google.com (dl.google.com)|172.217.4.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 120081829 (115M) [application/octet-stream]
Saving to: 'go1.13.6.linux-amd64.tar.gz'

d64.tar.gz      20%[==>           ] 23.46M  4.24MB/s   eta 23s
```

c. Set up go environment

```
echo 'export GOPATH=${HOME}/go' >> ~/.bashrc && \
    echo 'export PATH=/usr/local/go/bin:${PATH}: ${GOPATH}/bin' >> ~/.bashrc
&& \
    source ~/.bashrc
```



```
Terminal
chidi@UbuntuVM: ~
go/test/typeswitch.go
go/test/typeswitch1.go
go/test/typeswitch2.go
go/test/typeswitch3.go
go/test/uintptrescapes.dir/
go/test/uintptrescapes.dir/a.go
go/test/uintptrescapes.dir/main.go
go/test/uintptrescapes.go
go/test/uintptrescapes2.go
go/test/undef.go
go/test/utf.go
go/test/varerr.go
go/test/varinit.go
go/test/writebarrier.go
go/test/zerodivide.go
chidi@UbuntuVM:~$ echo 'export GOPATH=${HOME}/go' >> ~/.bashrc && \
>     echo 'export PATH=/usr/local/go/bin:${PATH}: ${GOPATH}/bin' >> ~/.bashrc &&
>     source ~/.bashrc \
> ^C
chidi@UbuntuVM:~$ echo 'export GOPATH=${HOME}/go' >> ~/.bashrc &&     echo 'expo
rt PATH=/usr/local/go/bin:${PATH}: ${GOPATH}/bin' >> ~/.bashrc &&     source ~/.b
ashrc
chidi@UbuntuVM:~$
```

d. Decide what version of singularity to install by checking the HPC

The screenshot shows a web browser interface with the URL <https://ood.hpc.arizona.edu/pun/sys/dashboard/b>. The browser tabs include "Installation — Singularity cont...", "Downloads - The Go Program...", and "My Interactive Sessions". The main content area displays the "My Interactive Sessions" dashboard for the Ood HPC system at Arizona. A tooltip for "Ocelote Shell Access" is visible, stating "Started or terminated" with a hand cursor icon. Below the tooltip, a reminder message says "Reminder - Upcoming Ocelote Shell Access and Maintenance" and lists "HPC Downtime scheduled on 2022-01-24".

Home / My Interactive Sessions

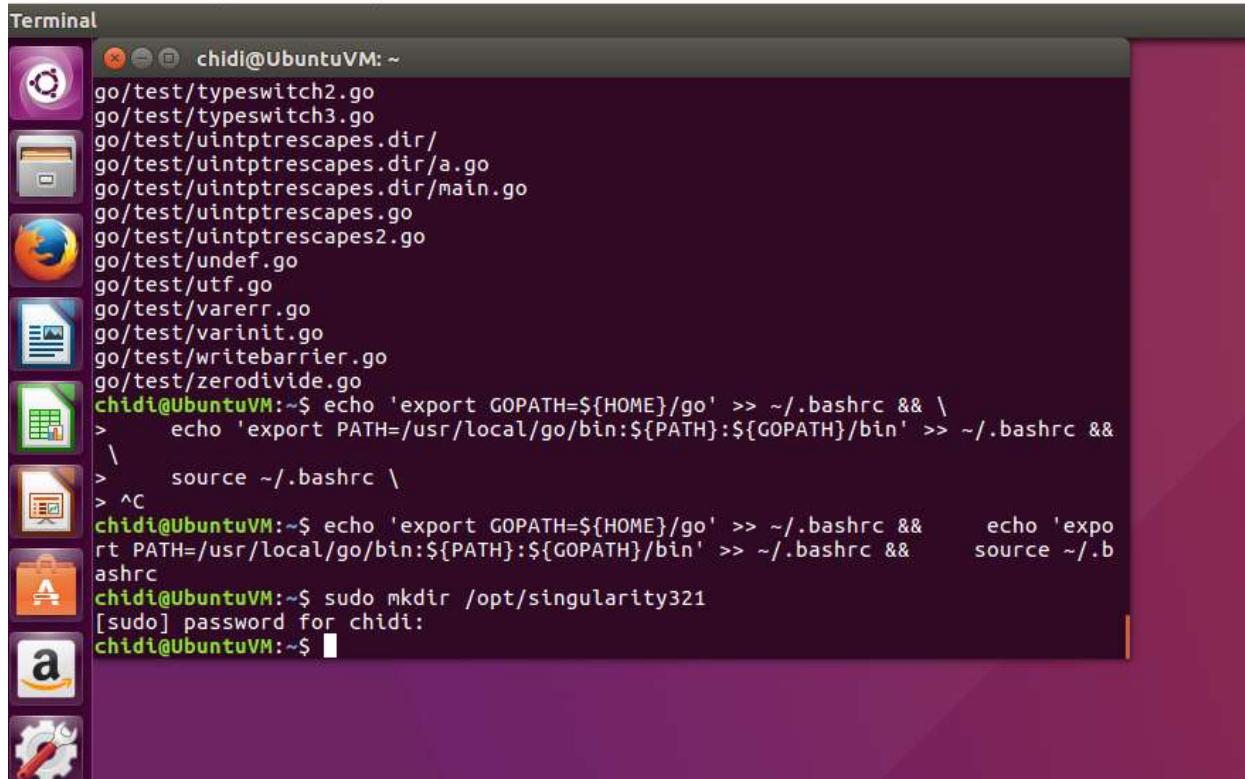
Last login: Tue Jan 21 06:53:28 2020 from ood.hpc.arizona.edu

```
[chidiugonna@login2 ~]$ module avail singularity
singularity/2/2.6.1 singularity/3/3.2 singularity/3/3.2.1 singularity/3/3.4.2
[chidiugonna@login2 ~]$
```

We will go ahead and install 2 versions of singularity - 3.4.2 and 3.2.1

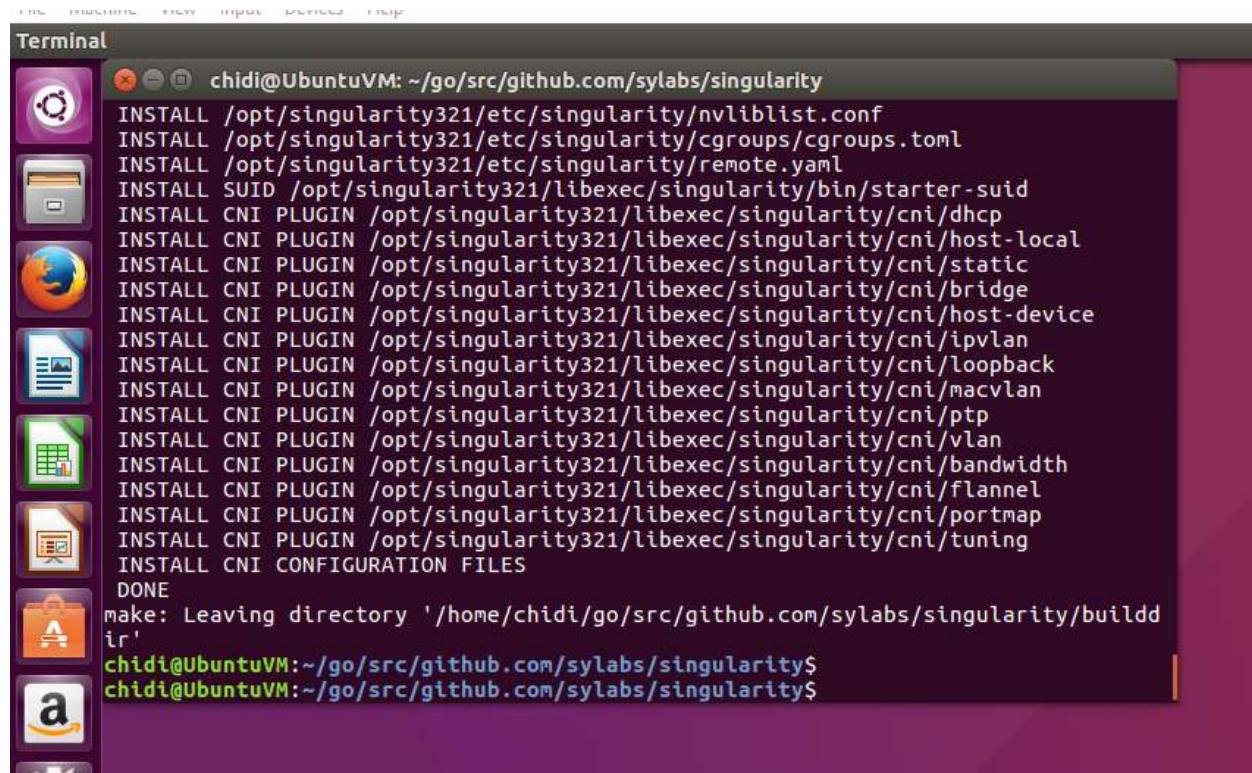
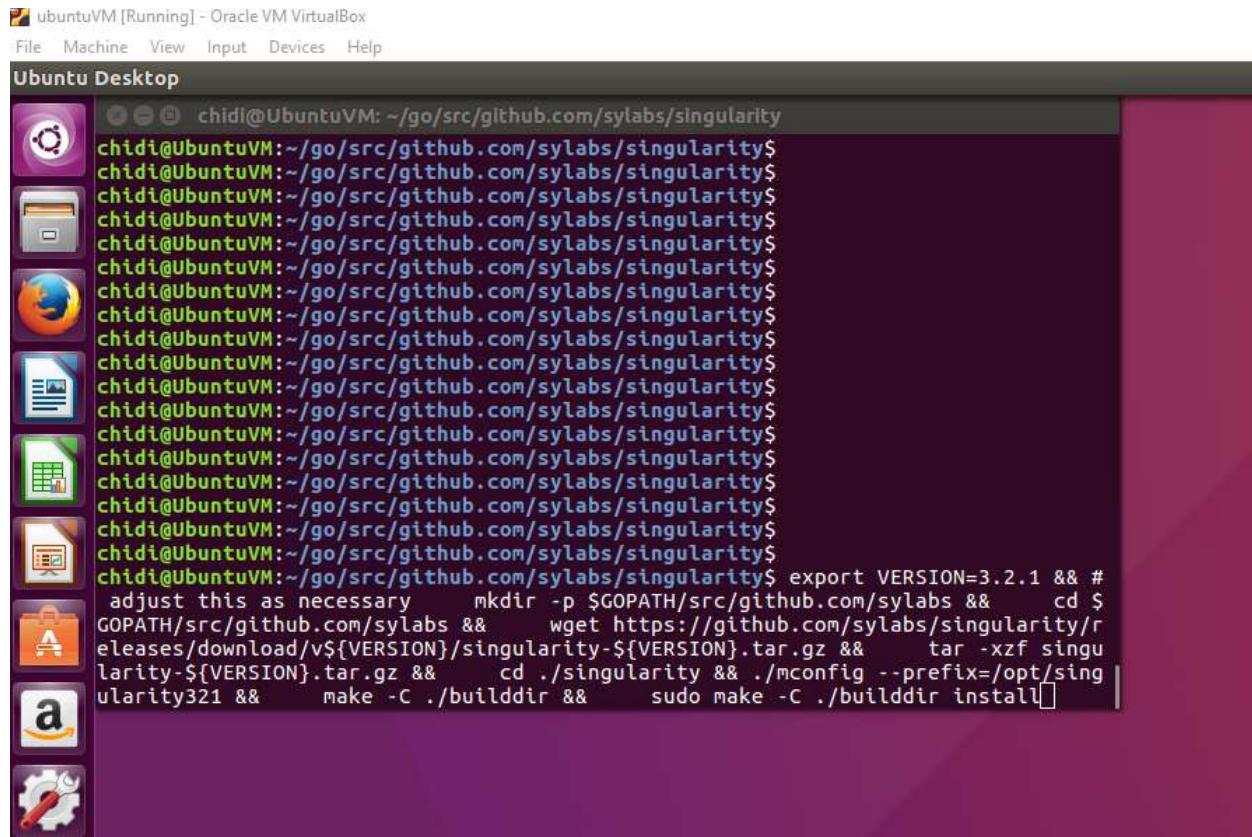
e. Create 2 directories for the installed singularity

```
sudo mkdir /opt/singularity321
sudo mkdir /opt/singularity342
```



f. Install singularity 3.2.1

```
export VERSION=3.2.1 && # adjust this as necessary \
mkdir -p $GOPATH/src/github.com/sylabs && \
cd $GOPATH/src/github.com/sylabs && \
wget
https://github.com/sylabs/singularity/releases/download/v${VERSION}/singularity-$VERSION.tar.gz && \
tar -xzf singularity-$VERSION.tar.gz && \
cd ./singularity && \
./mconfig --prefix=/opt/singularity321 && \
make -C ./builddir && \
sudo make -C ./builddir install
```



g. Install singularity 3.4.2

To install singularity 3.4.2 you first need to install cryptsetup-bin

```
sudo apt-get install cryptsetup-bin
```

also delete the old go singularity folder

```
rm -R $GOPATH/src/github.com/sylabs/singularity
```

after that you can run code below:

```
export VERSION=3.4.2 && # adjust this as necessary \
    mkdir -p $GOPATH/src/github.com/sylabs && \
    cd $GOPATH/src/github.com/sylabs && \
    wget
https://github.com/sylabs/singularity/releases/download/v${VERSION}/singularity-${VERSION}.tar.gz && \
    tar -xzf singularity-${VERSION}.tar.gz && \
    cd ./singularity && \
./mconfig --prefix=/opt/singularity342 && \
    make -C ./builddir && \
    sudo make -C ./builddir install
```

h. Edit the ~/.bashrc file

gedit ~/.bashrc

And add

```
export PATH=/opt/singularity321/bin:$PATH
```

```
export PATH=/opt/singularity342/bin:$PATH
```

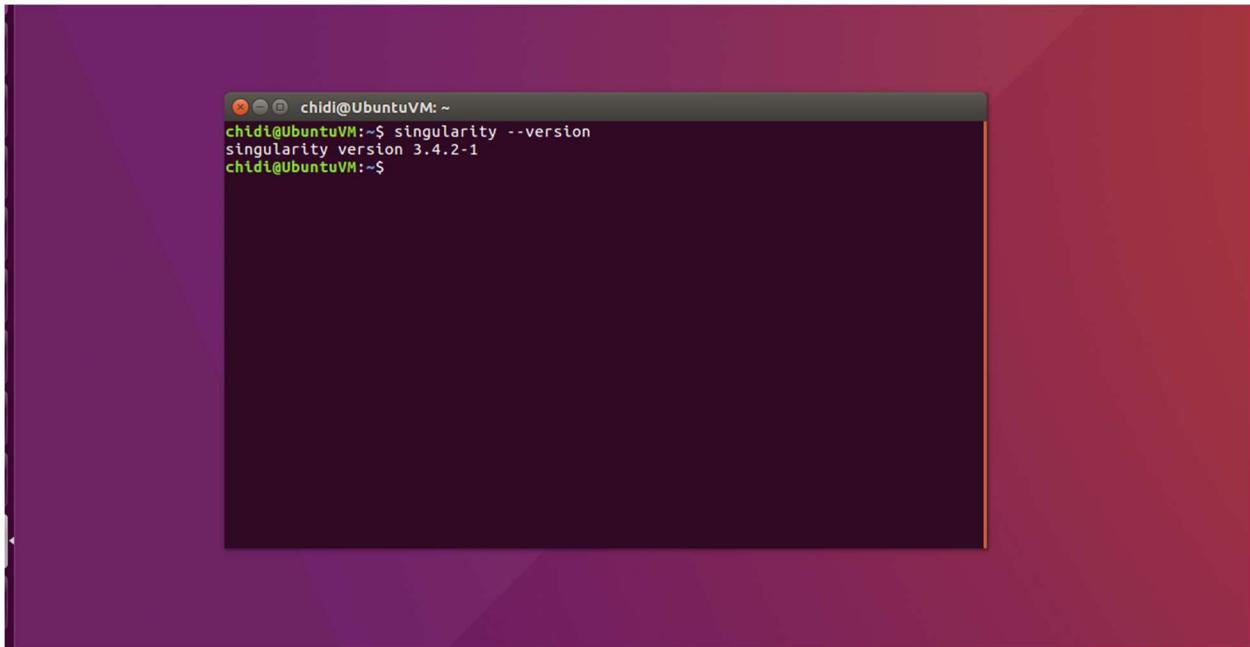
place a # in front of the version you want to disable for now – in the example below I have disabled singularity 321

```
venv$ vendor/github.com/gorilla/websocket/conn_read.go:11:10: (*Conn).Read redeclared
in this block
      previous declaration at ..../vendor/github.com/gorilla/websocket/conn.go:3
6
efile:160: recipe for target 'singularity' failed
e: *** [singularity] Error 2
e: Leaving directory '/home/chidi/go/src/github.com/sylabs/singularity/buildd
di@UbuntuVM:~/go/src/github.com/sylabs/singularity$ gedit ~/.bashrc
# sources /etc/bash_completion
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ] ||
     . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export GOPATH=${HOME}/go
export PATH=/usr/local/go/bin:${PATH}: ${GOPATH}/bin
#export PATH=/opt/singularity321/bin:$PATH
export PATH=/opt/singularity342/bin:$PATH
```

- i. Test that singularity version is installed by opening a new terminal ctrl-alt-t and running

```
singularity --version
```

Singularity version 3.4.2 installed and running



- j. Make singularity sudo-able

Because we have installed singularity to /opt , the sudo command cannot run singularity because this folder is not in the super-user's \$PATH. There are two approaches to addressing this which can be read about here <https://stackoverflow.com/questions/12996397/command-not-found-when-using-sudo>

Alias for Sudo

The most elegant solution is to create an alias for sudo that sources the current user's path as the default path for the super user. So the line below can be entered into the .bashrc file

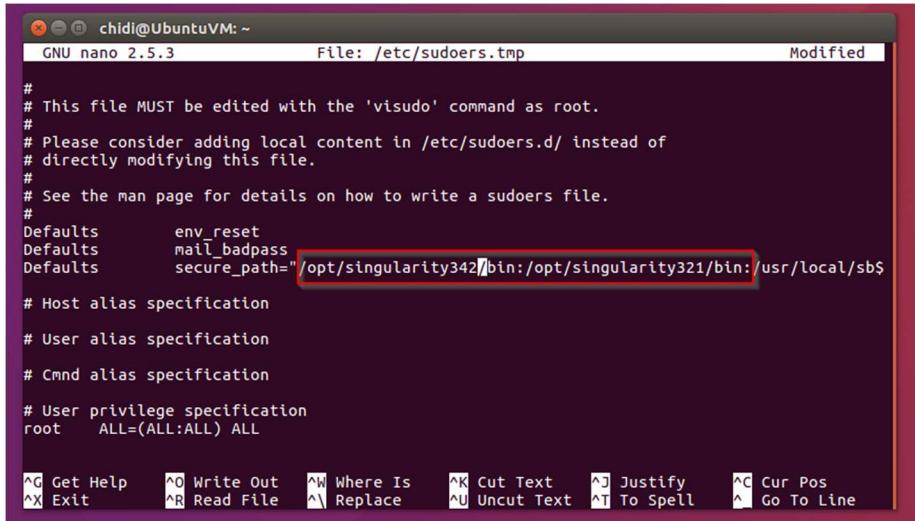
```
alias sudo='sudo -E env "PATH=$PATH"'
```

Visudo

An alternative approach is shown below but this will have to be edited every time the default singularity is changed as the first path in the list takes precedence.

```
sudo visudo
```

add /opt/singularity321/bin and /opt/singularity342/bin to secure_paths – note that whichever path is defined first takes precedence. So in the example below version 342 will run as sudo regardless of whether the user has it active or not in the .bashrc file.



```
chidi@UbuntuVM: ~
GNU nano 2.5.3          File: /etc/sudoers.tmp          Modified

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/opt/singularity342/bin:/opt/singularity321/bin:/usr/local/sbin$"

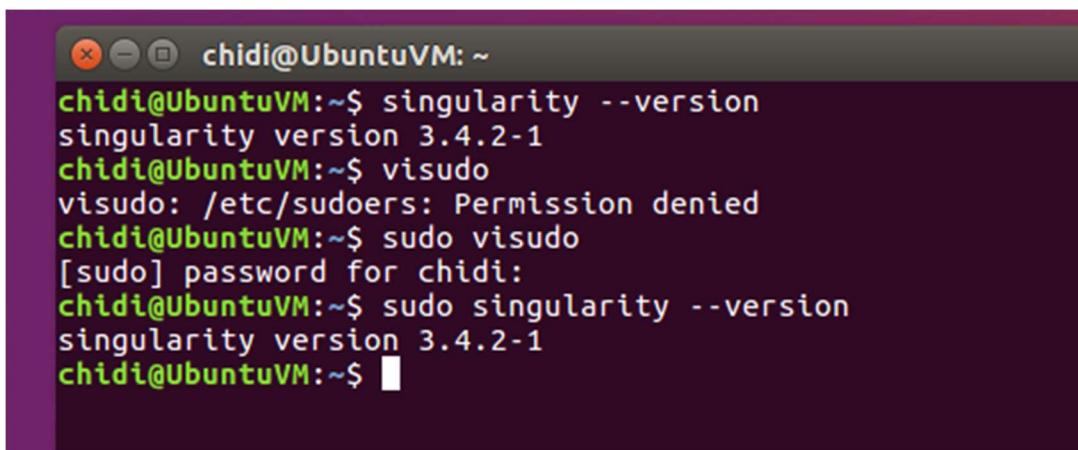
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL

^Q Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit     ^R Read File  ^Y Replace  ^U Uncut Text  ^T To Spell  ^G Go To Line
```

Press CTRL-X and then Y+Enter and then Enter to save

Test that singularity runs with sudo

```
sudo singularity --version
```



The screenshot shows a terminal window with a dark background and light-colored text. The window title is "chidi@UbuntuVM: ~". The user has run several commands:

```
chidi@UbuntuVM:~$ singularity --version
singularity version 3.4.2-1
chidi@UbuntuVM:~$ visudo
visudo: /etc/sudoers: Permission denied
chidi@UbuntuVM:~$ sudo visudo
[sudo] password for chidi:
chidi@UbuntuVM:~$ sudo singularity --version
singularity version 3.4.2-1
chidi@UbuntuVM:~$
```

3. Build your first Container from a Definitions file

We will build the lolcow.sif container as described here https://sylabs.io/guides/3.0/user-guide/build_a_container.html

We will work in a separate folder at \$HOME/containers – so go ahead and create it and cd into it as follows

```
mkdir ~/containers
```

```
cd ~/containers
```

Create and open a file called singularity.def

```
gedit singularity.def
```

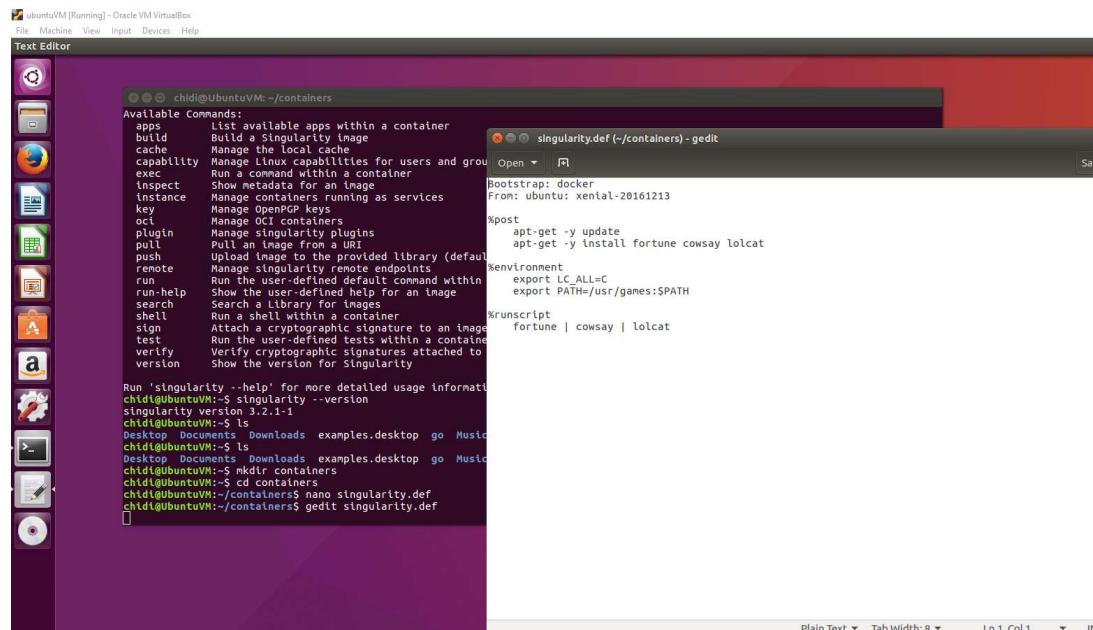
Copy the contents below into the file **singularity.def**

```
Bootstrap: docker
From: ubuntu:xenial-20161213

%post
    apt-get -y update
    apt-get -y install fortune cowsay lolcat

%environment
    export LC_ALL=C
    export PATH=/usr/games:$PATH

%runscript
    fortune | cowsay | lolcat
```



Notice the four important sections in the def file:

The header tells you what docker image you want to use as the foundation of your image.

```
Bootstrap: docker
From: ubuntu:xenial-20161213
```

%post session provides instructions to run during the build

```
%post
apt-get -y update
apt-get -y install fortune cowsay lolcat
```

%environment provides the variables to set up in the container at run time (and also when shelling into the container)

```
%environment
export LC_ALL=C
export PATH=/usr/games:$PATH
```

%runscript are commands to run when image starts

```
%runscript
fortune | cowsay | lolcat
```

more information about definition file is here https://sylabs.io/guides/3.0/user-guide/definition_files.html

1. Build the container lolcow.sif

```
sudo singularity build lolcow.sif singularity.def
```

```
chidi@UbuntuVM:~/containers$ nano singularity.def
chidi@UbuntuVM:~/containers$ sudo singularity build lolcow.sif singularity.def
INFO: Starting build...
Getting image source signatures
Skipping fetch of repeat blob sha256:b3e1c725a85f0953e81815b7c7aabfad9ebfd90af53f99248981282b8045d787
Skipping fetch of repeat blob sha256:4daad8bdde31f5969d6a44800e993a911bed25853b2f89417b07ce7f0ac67714
Skipping fetch of repeat blob sha256:63fe8c0068a811103c702452985368fc99014f83b5f0151cf631644f13f383b2
Skipping fetch of repeat blob sha256:a470713c436f2350e8a9f1c1e6c823290f331254c6c21d5657056499d8575608
Skipping fetch of repeat blob sha256:bd842a2105a862f62ba4ad726531029be49046c290307e9c07a688c89f4606e6
Copying config sha256:d743f0aba036a57af4dcbb1ac4bf6c472c2f22216a752404baee5e996b37ee1c3
2.62 KiB / 2.62 KiB [=====] 0s
Writing manifest to image destination
Storing signatures
INFO: Running post scriptlet
+ apt-get -y update
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:4 http://archive.ubuntu.com/ubuntu xenial/main Sources [1103 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial/restricted Sources [5179 B]
Get:6 http://archive.ubuntu.com/ubuntu xenial/universe Sources [9802 kB]
29% [6 Sources 5183 kB/9802 kB 53%]
```

456 kB/s 46s

```
Processing triggers for systemd (229-4ubuntu12) ...
Processing triggers for ca-certificates (20170717~16.04.2) ...
Updating certificates in /etc/ssl/certs...
148 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
INFO: Adding environment to container
INFO: Adding runscript
INFO: Creating SIF file...
INFO: Build complete: lolcow.sif
chidi@UbuntuVM:~/containers$
```

2. Test the singularity container using the **run** command

```
singularity run lolcow.sif
```

```
Singularity: command not found
chidi@UbuntuVM:~/containers$ singularity run lolcow.sif
/ There is a great discovery still to be \
| made in Literature: that of paying \
| literary men by the quantity they do \
\ NOT write.
```



```
chidi@UbuntuVM:~/containers$
```

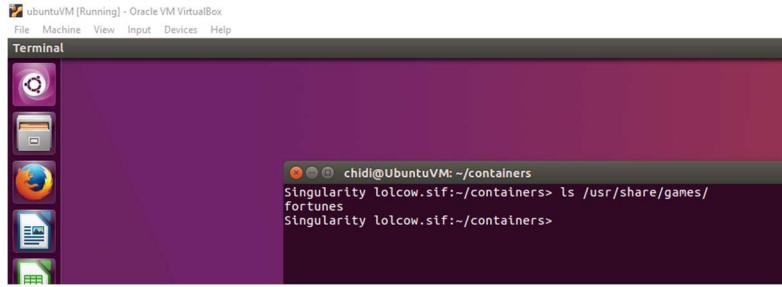
You can also shell into a singularity container. This is almost like “logging” into your singularity container. When you shell into a container you are running within that container almost like being within a virtual environment.

```
Singularity shell lolcow.sif
```

This puts you into the shell of the container and you can explore it like you would a separate virtual machine.

List the games and see where the fortune app has been installed

```
ls /usr/share/games
```



Run the applications directly in the shell

```
fortune  
  
fortune | lolcat  
  
fortune | cowsay | lolcat
```

```
chidi@UbuntuVM:~/containers$ singularity shell lolcow.sif  
Singularity lolcow.sif:~/containers> lolsay  
bash: lolsay: command not found  
Singularity lolcow.sif:~/containers> fortune  
Stay the curse.  
Singularity lolcow.sif:~/containers> lolcat  
^CSingularity lolcow.sif:~/containers> fortune | lolcat  
You will be awarded a medal for disregarding safety in saving someone.  
Singularity lolcow.sif:~/containers> fortune | cowsay  
  
/ You may be infinitely smaller than some \  
| things, but you're infinitely larger |  
\ than others.  
-----  
 \ \ ^__^  
  (oo)\_____) )\/\/  
   ||----w |  
   ||     ||  
Singularity lolcow.sif:~/containers> fortune | cowsay | lolcat  
/ You are dishonest, but never to the \  
| point of hurting a friend. |  
-----  
 \ \ ^__^  
  (oo)\_____) )\/\/  
   ||----w |  
   ||     ||  
Singularity lolcow.sif:~/containers>
```

4. Build the same container using a sandbox

Sometimes it is more convenient to build your container in an exploratory manner – by shelling into it and then adding modules and functionality as needed. Then converting it to a singularity file when you are done. This is also useful when it is difficult to automate the build procedure in a definition file.

1. First build an empty sandbox based on ubuntu which will form the foundation for our image

```
sudo singularity build lolcow.sandbox docker://ubuntu:xenial-20161213
```

```
chidi@UbuntuVM:~/containers$ 
chidi@UbuntuVM:~/containers$ 
chidi@UbuntuVM:~/containers$ sudo singularity build --sandbox lolcow.sandbox docker://ubuntu:xenial-20161213
INFO: Starting build...
Getting image source signatures
Skipping fetch of repeat blob sha256:b3e1c725a85f0953e81815b7c7aabfad9ebfd90af53f99248981282b8045d787
Skipping fetch of repeat blob sha256:4daad8bbde31f5969da44800e993a911bed25853b2f89417b07ce7f0ac67714
Skipping fetch of repeat blob sha256:63fe8c0068a811103c702452985368fc99014f83b5f0151cf631644f13f383b2
Skipping fetch of repeat blob sha256:4a70713c436f2350e8a9f1c1e6c823290f331254c6c21d5657056499d8575608
Copying config sha256:d743f0aba036a57af4dc1ac4bf6c472czf22216a752404baee5e996b37ee1c3
  2.62 KiB / 2.62 KiB [=====] 0s
Writing manifest to image destination
Storing signatures
INFO: Creating sandbox directory...
INFO: Build complete: lolcow.sandbox
chidi@UbuntuVM:~/containers$
```

2. Now we will perform the steps in the %post section directly by shelling into the sandbox using the --writable keyword

```
sudo singularity shell --writable lolcow.sandbox
```

once within the shell then run commands in %post section

```
apt-get -y update
apt-get -y install fortune cowsay lolcat
```

```
Singularity lolcow.sandbox:~> apt-get update
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:4 http://archive.ubuntu.com/ubuntu xenial/main Sources [1103 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial/restricted Sources [5179 B]
Get:6 http://archive.ubuntu.com/ubuntu xenial/universe Sources [9802 kB]
Get:7 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages [1558 kB]
Get:8 http://archive.ubuntu.com/ubuntu xenial/restricted amd64 Packages [14.1 kB]
Get:9 http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [9827 kB]
Get:10 http://archive.ubuntu.com/ubuntu xenial-updates/main Sources [435 kB]
Get:11 http://archive.ubuntu.com/ubuntu xenial-updates/restricted Sources [2696 B]
Get:12 http://archive.ubuntu.com/ubuntu xenial-updates/universe Sources [333 kB]
Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1408 kB]
Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [13.1 kB]
Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [1008 kB]
89% [15 Packages 656 kB/1008 kB 65%]
```

357 kB/s 6s

3. We will also add an editor nano which we need to update the environment and runscript manually

```
apt-get install nano
```

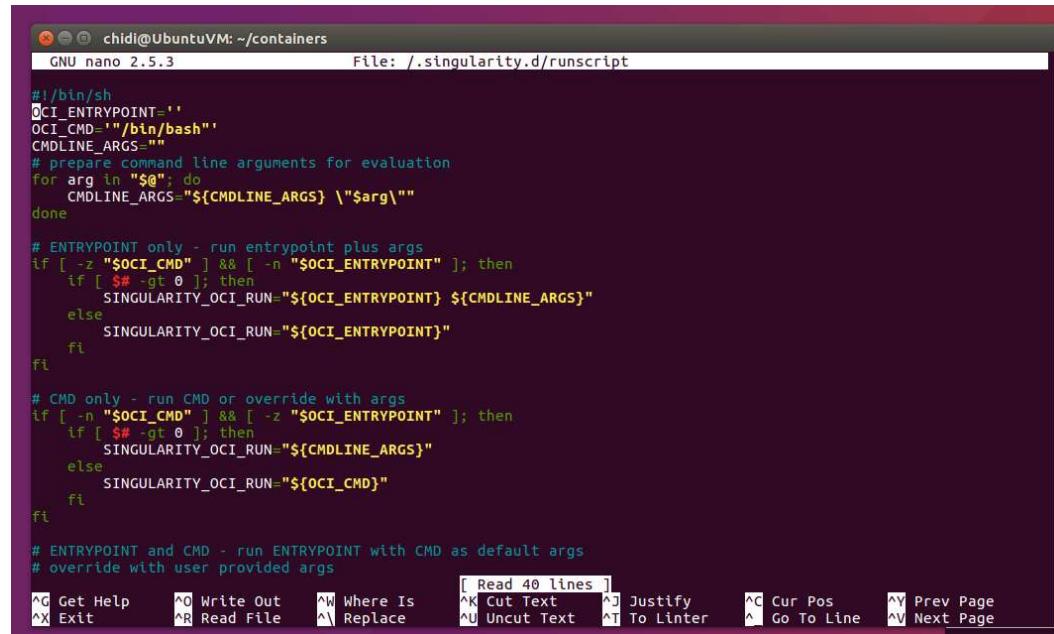
```
Singularity lolcow.sandbox:~>
Singularity lolcow.sandbox:~> apt-get install nano
```

4. Now edit the runscript using nano

```
nano ./singularity.d/runscript
```

```
Singularity lolcow.sandbox:~> ls ./singularity.d
Singularity actions env labels.json libs runscript startscript
Singularity lolcow.sandbox:~> nano ./singularity.d/runscript
```

5. Delete all the lines in the file except for #!/bin/bash – in nano you can delete 1 line at a time by placing the cursor on the line and pressing CTRL-k



```
chidi@UbuntuVM: ~/containers
GNU nano 2.5.3                               File: ./singularity.d/runscript

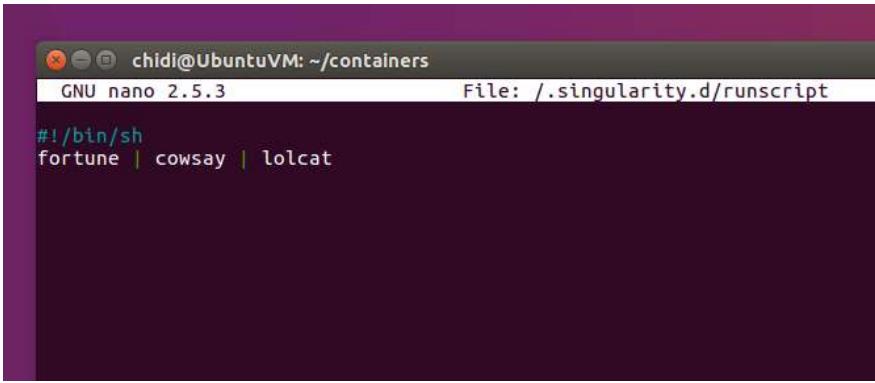
#!/bin/sh
OCI_ENTRYPOINT=''
OCI_CMD='"/bin/bash"'
CMDLINE_ARGS=""
# prepare command line arguments for evaluation
for arg in "$@"; do
    CMDLINE_ARGS="${CMDLINE_ARGS} \"$arg\""
done

# ENTRYPOINT only - run entrypoint plus args
if [ -z "$OCI_CMD" ] && [ -n "$OCI_ENTRYPOINT" ]; then
    if [ $# -gt 0 ]; then
        SINGULARITY_OCI_RUN="${OCI_ENTRYPOINT} ${CMDLINE_ARGS}"
    else
        SINGULARITY_OCI_RUN="${OCI_ENTRYPOINT}"
    fi
fi

# CMD only - run CMD or override with args
if [ -n "$OCI_CMD" ] && [ -z "$OCI_ENTRYPOINT" ]; then
    if [ $# -gt 0 ]; then
        SINGULARITY_OCI_RUN="${CMDLINE_ARGS}"
    else
        SINGULARITY_OCI_RUN="${OCI_CMD}"
    fi
fi

# ENTRYPOINT and CMD - run ENTRYPOINT with CMD as default args
# override with user provided args
[ Read 40 lines ]  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page
^q Get Help  ^Q Write Out  ^W Where Is  ^U Uncut Text  ^T To Linter  ^G Go To Line  ^V Next Page
^x Exit  ^R Read File  ^\ Replace
```

6. Add fortune | cowsay | lolcat to runscript



chidi@UbuntuVM: ~/containers
GNU nano 2.5.3 File: ./singularity.d/runscript

```
#!/bin/sh
fortune | cowsay | lolcat
```

7. Press CTRL-x, y and ENTER to save

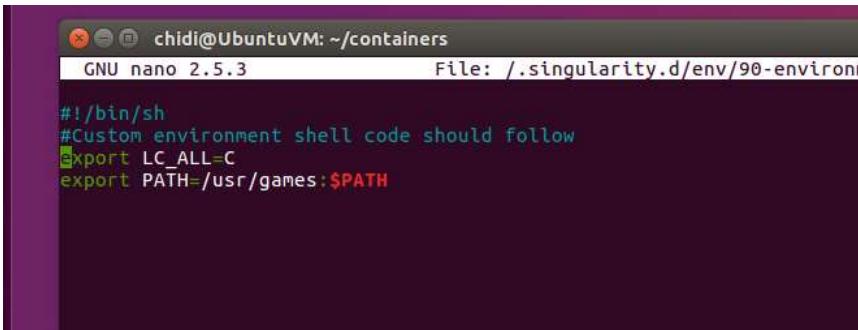


Singularity: /etc/singularity.d/singularity.tcl: /etc/singularity.d/singularity.tcl: 01-base.sh 90-environment.sh 95-apps.sh 99-runtimevars.sh
10-docker2singularity.sh 94-appsbase.sh 99-base.sh
Singularity lolcow.sandbox:~> nano ./singularity.d/env/90-environment.sh

8. Edit the 90-environment.sh file with the environmental variables in the %environment section

```
nano ./singularity.d/env/90-environment.sh
```

```
export LC_ALL=C
export PATH=/usr/games:$PATH
```



chidi@UbuntuVM: ~/containers
GNU nano 2.5.3 File: ./singularity.d/env/90-environment.sh

```
#!/bin/sh
#Custom environment shell code should follow
export LC_ALL=C
export PATH=/usr/games:$PATH
```

Save file using CTRL-x, y and ENTER

9. Exit the container

```
10-docker2singularity.sh 94-appsbase.sh    99-base.sh  
Singularity lolcow.sandbox:> nano /.singularity.d/env/9  
Singularity lolcow.sandbox:> exit
```

10. Now test your sandbox to see that it runs like the first container that you built

```
Singularity run lolcow.sandbox
```

```
singularity: command not found  
chidi@UbuntuVM:~/containers$ singularity run lolcow.sandbox  
< You will never know hunger. >  
-----  
 \  ^__^ )\ \  )\/\ /\\  
  ||----w ||----w |  
 chidi@UbuntuVM:~/containers$
```

11. Convert your Sandbox to Sif file for easy portability to the hpc

```
sudo singularity build lolcow2.sif lolcow.sandbox
```

```
chidi@UbuntuVM:~/containers$ sudo singularity build lolcow2.sif lolcow.sandbox  
[sudo] password for chidi:  
INFO: Starting build...  
INFO: Creating SIF file...
```

```
INFO: Build complete: lolcow2.sif  
chidi@UbuntuVM:~/containers$ ls  
lolcow2.sif  lolcow.sandbox  lolcow.sif  singularity.def  
chidi@UbuntuVM:~/containers$
```

12. Test that this new sif file works too

```
Singularity run lolcow2.sif
```

```
chidi@UbuntuVM:~/containers$ singularity run lolcow2.sif
/ There is no hunting like the hunting of \
| man, and those who have hunted armed \
| men long enough and liked it, never \
| care for anything else thereafter.
| \
| -- Ernest Hemingway
|
\  ^__^
 \  ooo\
   (__)\_____
    ||----w |
     ||     |
chidi@UbuntuVM:~/containers$
```

5. Copy sif file to HPC and test it on the HPC

1. Use scp to transfer sif file to HPC – format is scp [source] [destination]

In our example source is the file lolcow2.sif

And destination is [yournetid]@sftp.hpc.arizona.edu:/path/to/local/directory

```
scp lolcow2.sif chidiugonna@sftp.hpc.arizona.edu:/extra/chidiugonna
```

```
-rw-rw-r-- 1 chidi chidi 231 Jan 22 22:39 singularity.def
chidi@UbuntuVM:~/containers$ scp lolcow2.sif chidiugonna@sftp.hpc.arizona.edu:/extra/chidiugonna
The authenticity of host 'sftp.hpc.arizona.edu (206.207.252.16)' can't be established.
ECDSA key fingerprint is SHA256:0D17jN0+8xmMc8BVveqe9FijlNZsMQxv8swXsfR6etg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'sftp.hpc.arizona.edu,206.207.252.16' (ECDSA) to the list of known hosts.
Password:
Duo two-factor login for chidiugonna

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-3532
2. Phone call to XXX-XXX-3532
3. SMS passcodes to XXX-XXX-3532

Passcode or option (1-3): 2
lolcow2.sif
100% 97MB 16.2MB/s 00:06
chidi@UbuntuVM:~/containers$
```

2. Log on to HPC (either with the console or desktop app) and navigate to location of sif file and run the following in a terminal.

```
module load singularity
```

```
Singularity run lolcow2.sif
```

```
Last login: Tue Jan 21 13:55:23 2020 from ood.hpc.arizona.edu
[chidiugonna@login3 ~]$ cd /extra/chidiugonna
[chidiugonna@login3 chidiugonna]$ ls
BMW      SIMNIBS  SingularityImages_3.2.1  fmriSignal  nklab-config-post.sh~  o
PIPELINE  SIMNIBS3 TMS                  lolcow2.sif  nklab-tortoise-v0.1.sif  r
[chidiugonna@login3 chidiugonna]$ module load singularity
[chidiugonna@login3 chidiugonna]$ singularity run lolcow2.sif

/ O, it is excellent To have a giant's \
| strength; but it is tyrannous To use it \
| like a giant.

| -- Shakespeare, "Measure for Measure",
\ II, 2 \
-----
 \ ^__^
  (oo)\_____
   (__)\       )\/\
    ||----w |
     ||     |

[chidiugonna@login3 chidiugonna]$ 
```

6. Install Docker

Reference <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

1. Uninstall old versions of docker

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

2. Install using repository

```
sudo apt-get update
```

```
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

3. Verify that fingerprint matches last 8 chars of 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

```
sudo apt-key fingerprint 0EBFCD88
```

4. Set up stable repository

```
sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
```

5. Update and install latest version of docker

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

6. Verify that docker installed correctly

```
sudo docker run hello-world
```

7. Add user to docker group

```
sudo usermod -aG docker your-user
```

