How to build your virtual machine

# Get Linux Mint

<https://linuxmint.com/edition.php?id=249>

download Mint 18.3 “Sylvia” – Xfce (64 bit)

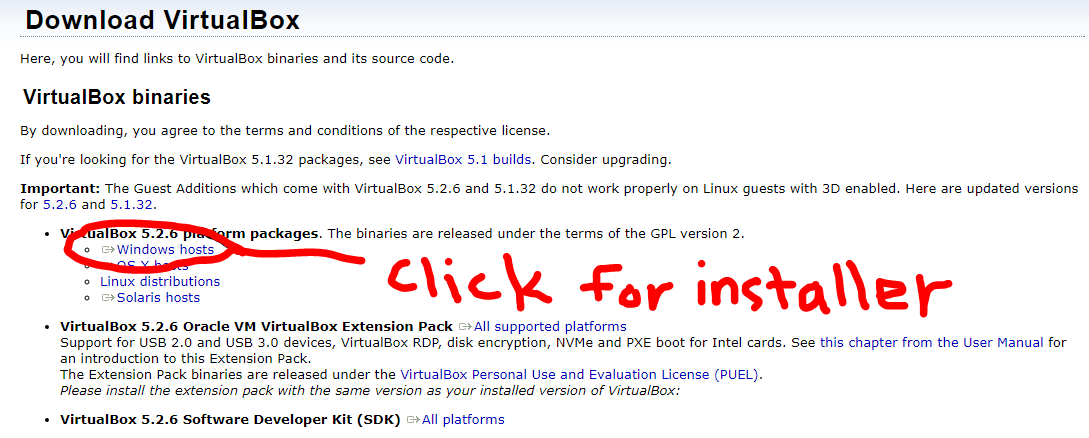
sha256 verify (optional, see <https://linuxmint.com/verify.php> )

a78be201c766133d35a39b0555e00503f5fe47ff2bb17d6e27953397f44dcf09 \*linuxmint-18.3-xfce-64bit.iso

# Get VirtualBox

Oracle VM VirtualBox 5.2.6

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



# Setup VirtualBox

launch oracle virtual box

click new to create a new Virtual Machine

name: Django\_Server

select: type = linux, version= Ubuntu(64)

select memory = 2048MB

leave “Create a virtual hard disk now”

click next;

first select Hard disk Type; I use VHD (I’ve had problems with VDI, but many people leave it as the default)

next select file location (I put my virtual machines in <user>/documents/virtual\_machines/ )

you probably will never visit this file location

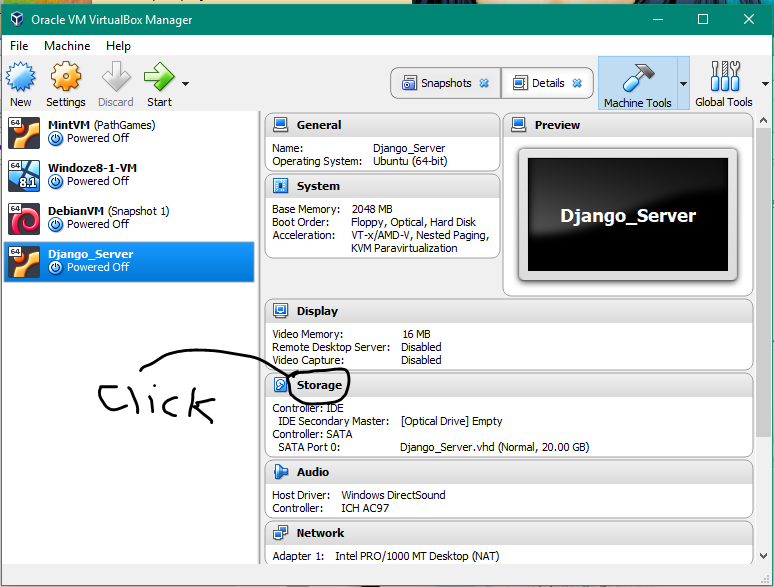
leave dynamically allocated selected

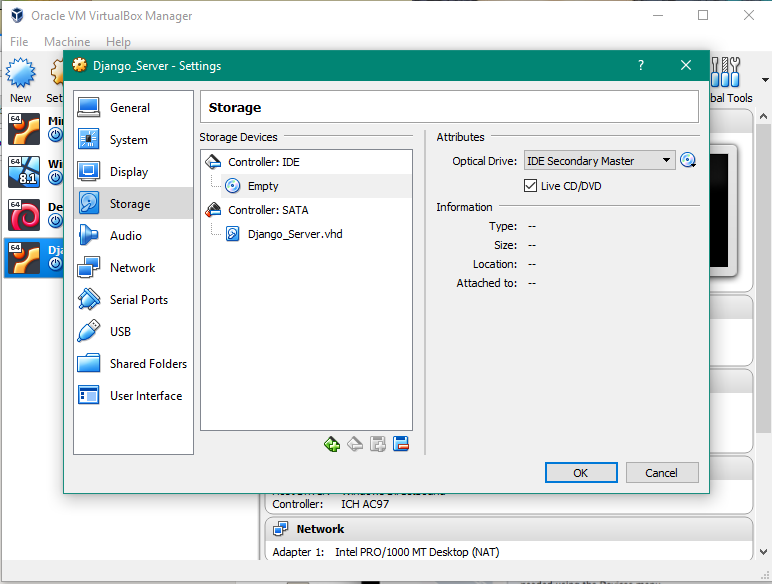
raise file size to 20.0 GB

\*on the network tab it defaults to **NAT**. This allows your VM out to the internet but incoming connections cannot route to it. This is fine, but understand you will only be able to visit your Django website from the browser inside the virtual machine. If you want your other home computers to be able to view the robot waitress change this setting to **bridged**. Later, make note of the IP address your router gives the VM. You will still be firewalled by the router but have access from home.

\* you should be fine with the display default of 16MB as your host machine will be the one actually doing the work of rendering the desktop.

Next we are going to set the VM to boot to the Linux ISO we downloaded earlier





click on Empty underneath controller: IDE in the center window labeled Storage Devices

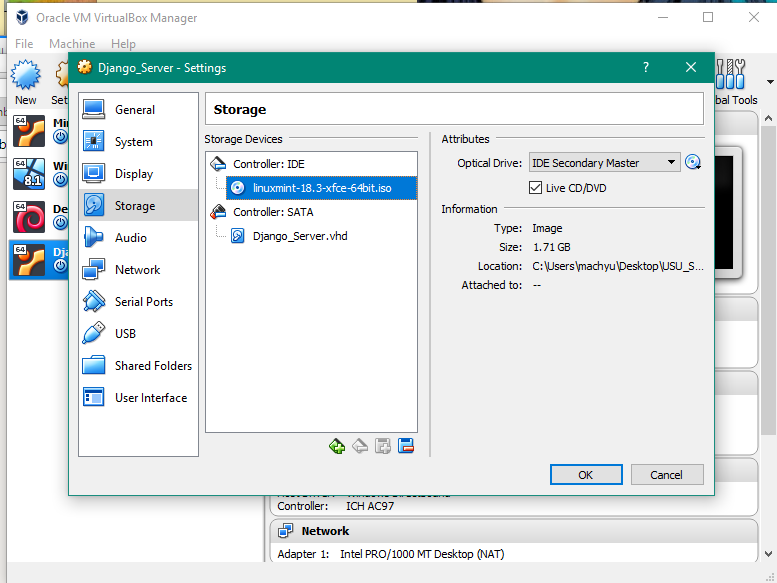
then mark the box “live CD/DVD”

then click the CD picture in the Attributes section

choose “choose virtual optical disk file”

navigate to where you saved the mint 18.3 download

select that file



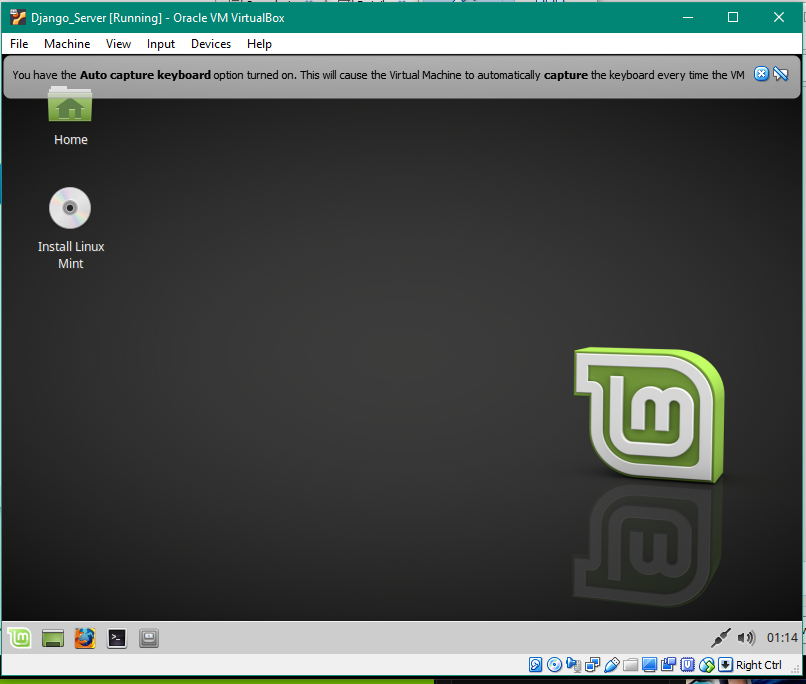
press ok to save all the changes

# Install and set up Mint

press start;

[now you are booting into the mint virtual machine]

in about 30 seconds you should land on the mint desktop



you can ‘x’ the capture warnings.

then click on the Install Linux Mint

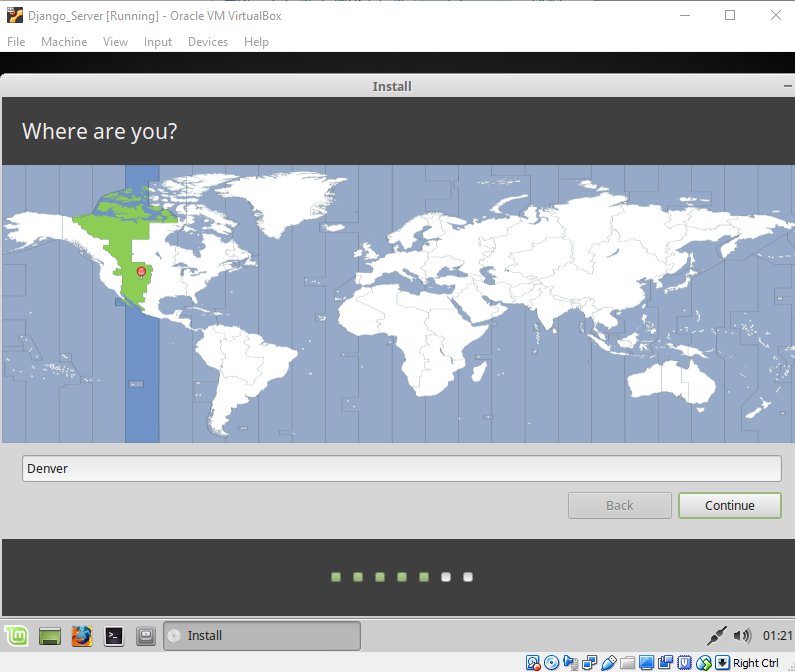
[don’t worry, it will install in the 20GB space you created for it and not over you file system]

answers to install questions;

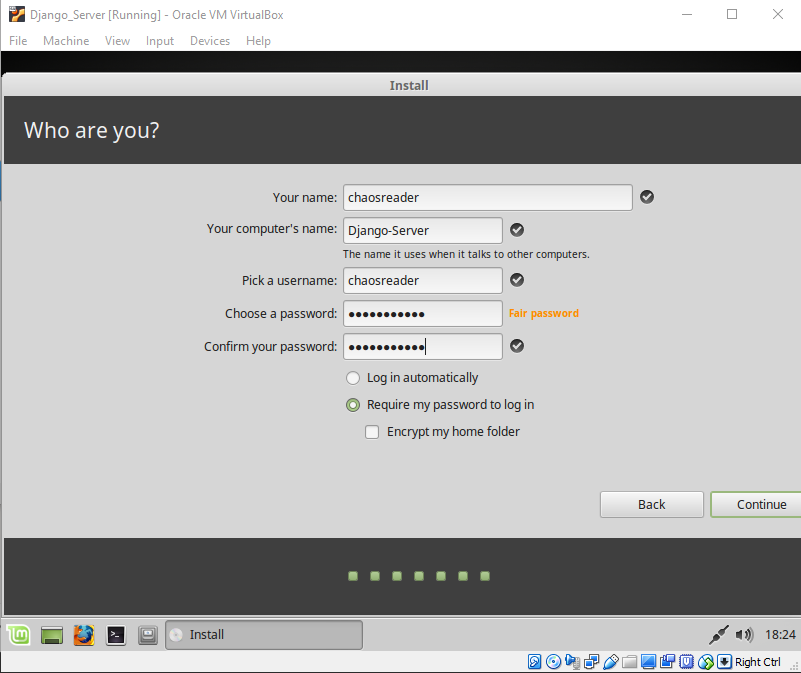
1. English
2. check to include third party software.
3. Erase disk and install Linux Mint

[you do not need to select Encryption or LVM \*that’s to create snapshots on metal installs, virtual box will do our snap shots.]

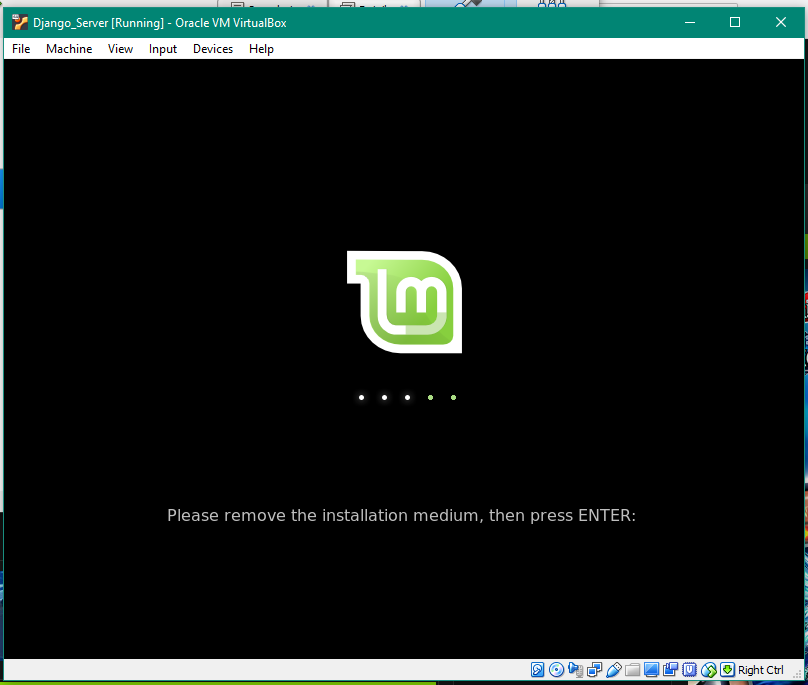
1. accept the partitioning scheme by clicking continue



1. select Denver if not done already
2. English(US) and English(US) for keyboard. click continue
3. set up passwords (it’s just a demo machine so write down username and PW if you think you might forget)

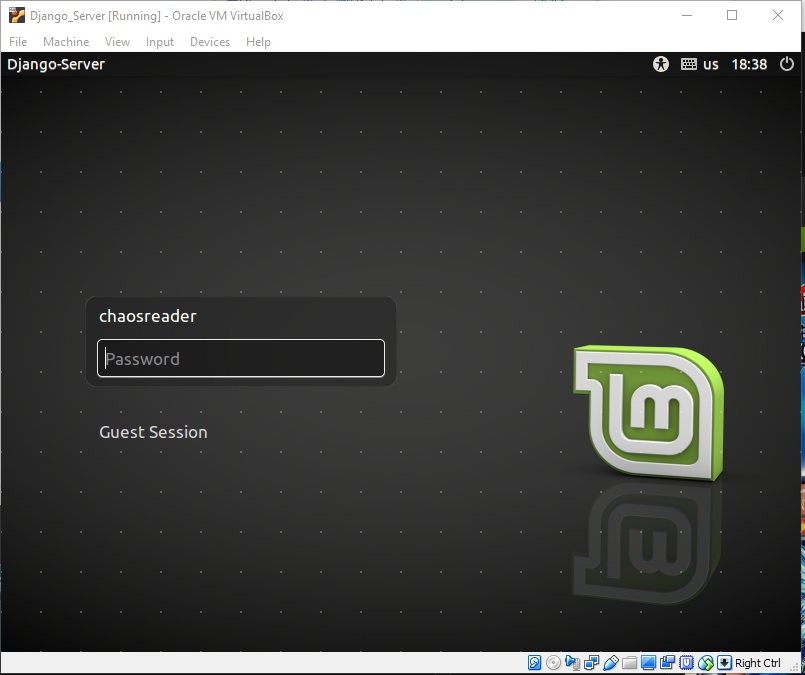


it will finish installing and ask you to restart



click devices -> Optical drives -> remove disk from virtual drive then click on mint again (optional)

press enter when it tells you to.

congratulation you have built a virtual machine. 

# Setup Linux Mint for Development

enter the following commands from inside the VM using the terminal (<https://www.howtogeek.com/howto/22283/four-ways-to-get-instant-access-to-a-terminal-in-linux/>)

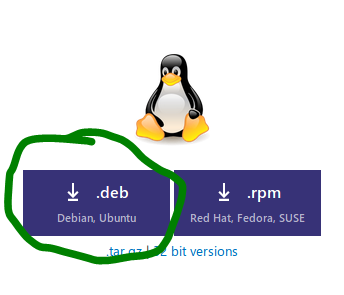
$ sudo apt update -y

$ sudo apt upgrade -y

$ sudo apt install -y python3-pip git

inside mint open Firefox and go to:

<https://code.visualstudio.com/Download>



save the file. It should end up in your download folder

you can now close Firefox.

$ mkdir builds

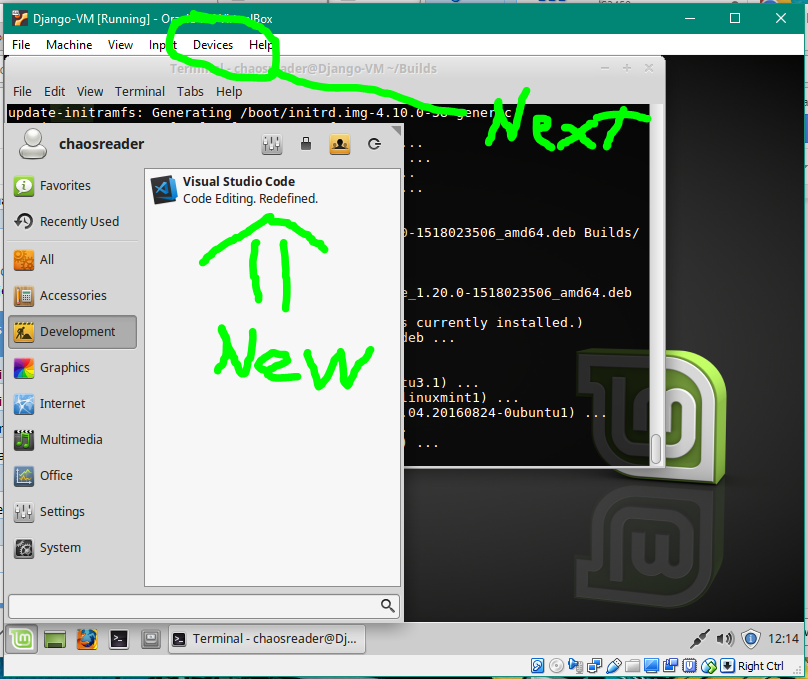
$ mv Downloads/code<tab complete> builds/

$ cd builds

$ sudo dpkg -i code<tab complete>

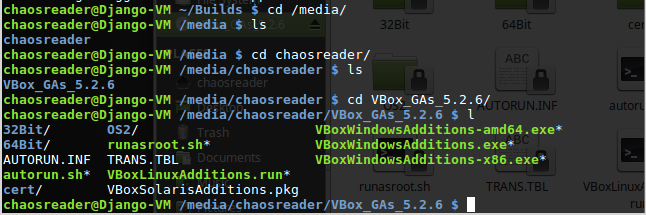
these instructions assume you start in your home folder and are relative to that.

you should have a new program installed the hard way! To see it click the leaf button (it’s like the start button in Windows) and hover over development



next click devices and select “insert guest additions CD”

in a second it should pop a notice that guest additions CD was mounted



navigate to /media/<username>/VBox\_<tab>/

run

$ sudo ./VBoxLinuxGuestAdditions.run

$ sync

close everything and reboot the VM

when you reboot, you should be able to get a full screen view of mint. I had to toggle back and forth twice to get it to start going full screen.

$ sudo –H pip3 install Django==2.0.2

NOTE, if the above doesn’t work try

$ sudo pip3 install Django==2.0.2

\* ignore the pip 8.1.1 warning

Shutdown the VM and make a snapshot. Mine is titled “Project-Start”

on virtual box manager -> machine tools drop down -> snap shots -> then click take (camera with a + icon)

# Connect to GitHub

Decide where you want your development files to live. Good locations include in your home directory and on the desktop. The below instructions assume you want it in your home directory

Open a terminal window and ensure you are in the home directory

$ cd ~

## Download the project into a directory with a name of your choice

For this example, we will put the files for the project into a directory called Server in the home folder

$ mkdir Server

$ cd Server

$ git init

$ git remote add origin <https://github.com/Abeham1/cs3450_groupProject.git>

$ git fetch origin

## Let GIT do all the work

$ git clone https://github.com/Abeham1/cs3450\_groupProject.git

# Test Server

Navigate to the manage.py folder as this is the Django control file for the project. Assuming you went the Git Clone method above then

$ cd cs3450\_groupProject/Django/cs3450\_group\_project/workspace/

$ python manage.py runserver

Open Firefox (in mint) and go to where the server is running as noted in the terminal. Make sure to use the port number after the colon. In other words, 127.0.0.1 won’t work but 127.0.0.1:8000 will.

You should see the homepage of our project or the default Django screen.

# GIT Workflow

As this is a collaborative project we should never be working directly in the master branch. Before you start working on any new pages or code make a branch.

$ git checkout -b new\_branch\_name

Commit early and commit often to your local branch. Good practice is to not do a git push until the code is working (even if not finished it should build / display).

Another best practice with your git commits is to use multi line commits

Short description of feature being worked on

<blank line>

Description of what is in this commit. Don’t put notes to others or TODO’s here as they will be lost too easily.

The top line of the comment will show up as the commit note in GitHub with a … after the note which has all of the rest of the commit details.

When a feature is finished we can merge that feature into Master for everyone to use

$ git checkout master

$ git pull

$ git merge new\_branch\_name

$ git push

# Old Directions

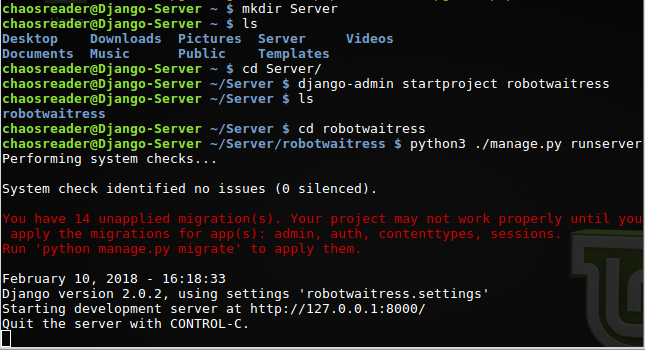
~ $ mkdir Server

~ $ cd Server

~/Server $ django-admin startproject robotwaitress

~/Server $ cd robotwaitress

~/Server/robotwaitress $ python3 ./manage.py runserver



open Firefox (in mint) and go to where the server is running. make sure to use the port number after the colon. In other words, 127.0.0.1 won’t work but 127.0.0.1:8000 will

