

### Step 1 - Setup Ubuntu

- Open VirtualBox
- Click “New”, give it a name (phi, for instance). Follow the screenshot:

The screenshot shows the 'Name and operating system' section with the following fields:

- Name:
- Machine Folder:
- Type:  (dropdown menu)
- Version:  (dropdown menu)

The 'Memory size' section shows a slider from 4 MB to 8192 MB, with a value of 1024 MB selected.

The 'Hard disk' section has three radio buttons:

- ☐ Do not add a virtual hard disk
- ☒ Create a virtual hard disk now
- ☐ Use an existing virtual hard disk file

Below the radio buttons is a dropdown menu showing 'debian64.vdi (Normal, 8.00 GB)'.

At the bottom right are three buttons: 'Guided Mode', 'Create', and 'Cancel'.

- Click “Create” for both screenshots above and below:

The screenshot shows the 'File location' section with a text box containing 'C:\Users\Ken\VirtualBox VMs\phi\phi.vdi'.

The 'File size' section shows a slider from 4.00 MB to 2.00 TB, with a value of 10.00 GB selected.

The 'Hard disk file type' section has six radio buttons:

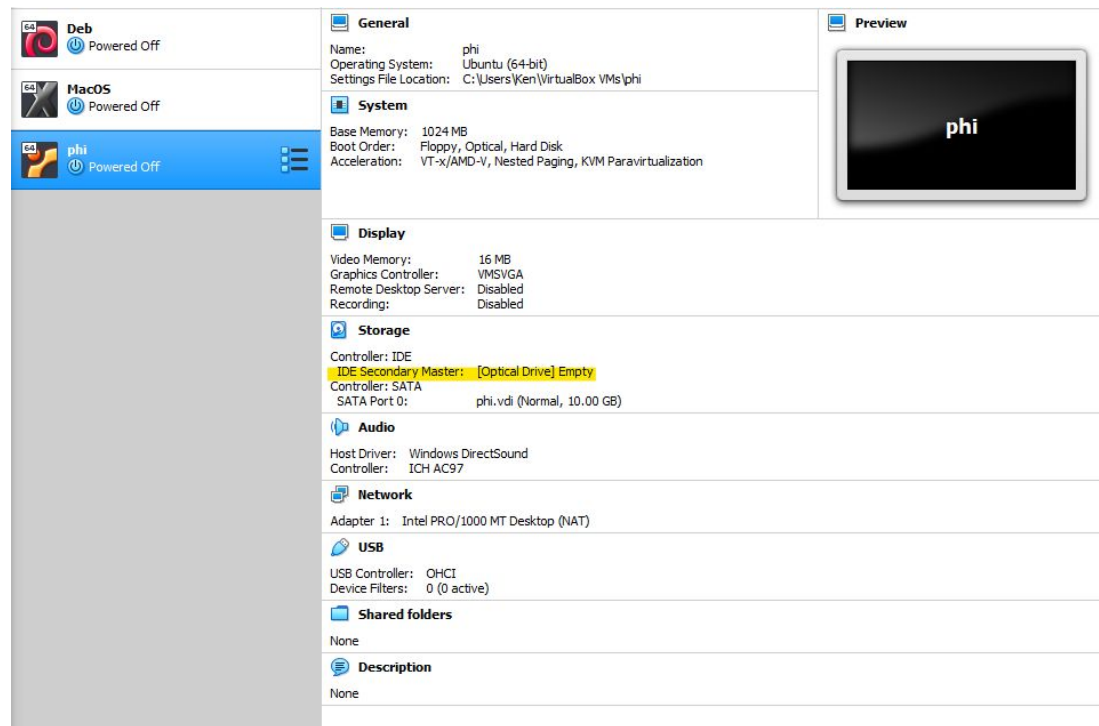
- ☒ VDI (VirtualBox Disk Image)
- ☐ VHD (Virtual Hard Disk)
- ☐ VMDK (Virtual Machine Disk)
- ☐ HDD (Parallels Hard Disk)
- ☐ QCOW (QEMU Copy-On-Write)
- ☐ QED (QEMU enhanced disk)

The 'Storage on physical hard disk' section has three radio buttons and one checkbox:

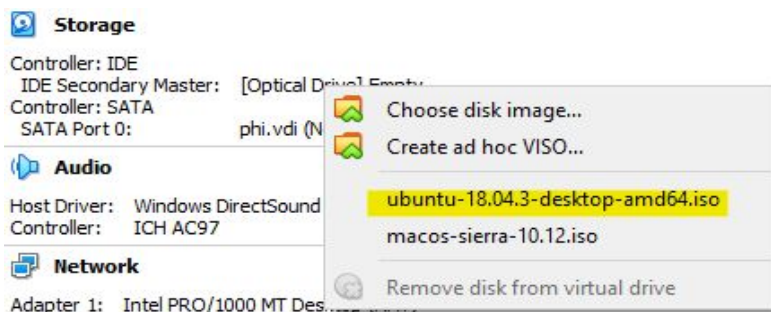
- ☒ Dynamically allocated
- ☐ Fixed size
- ☐ Split into files of less than 2GB

At the bottom right are three buttons: 'Guided Mode', 'Create', and 'Cancel'.

- Click the highlighted portion:



- Select the iso file that you just downloaded from [here](#):



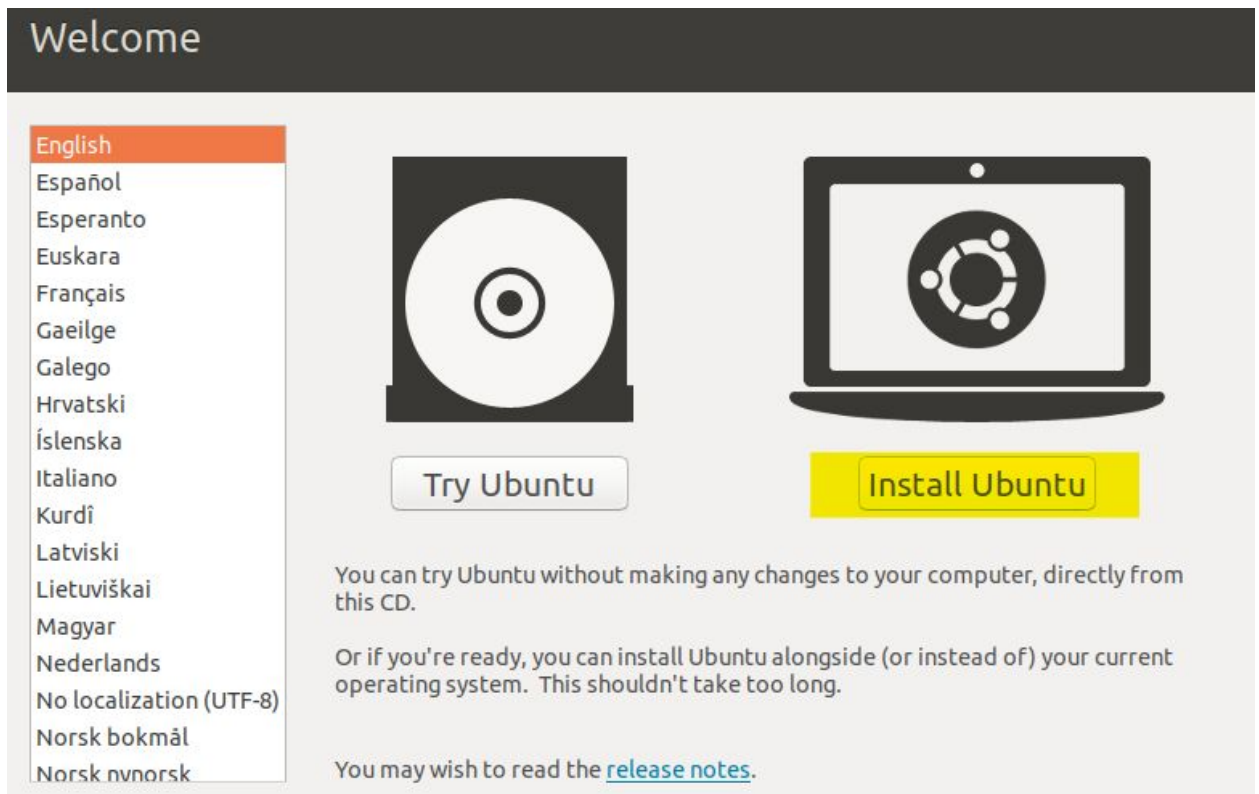
- Then, click “Settings”



- In the “System” section, go to the “Motherboard” tab and set Base Memory to 2048MB.
- Then, the “Display” section. In the “Screen” tab, set Video Memory to 128MB or the highest you could do
- Click “Ok”
- Then, click “Start”



- Follow through the instructions of the Ubuntu installation:



- If there is any unresponsiveness, wait for ~2 mins. If the problem persists, power off the machine and try again.
- Then, select your language of choice and "Continue".

- Make sure you have the following choice in the next section:

## Updates and other software

**What apps would you like to install to start with?**

☐ Normal installation  
Web browser, utilities, office software, games, and media players.

☒ Minimal installation  
Web browser and basic utilities.

**Other options**

☒ Download updates while installing Ubuntu  
This saves time after installation.

☐ Install third-party software for graphics and Wi-Fi hardware and additional media formats  
This software is subject to license terms included with its documentation. Some is proprietary.

Quit Back Continue

- Click “Continue” and in the next section, select “Erase disk and install Ubuntu”. Follow through.
- Customize your timezone and your user account accordingly. Try to keep it simple.

## Who are you?

Your name:  ✓

Your computer's name:  ✓  
The name it uses when it talks to other computers.

Pick a username:  ✓

Choose a password:  Short password

Confirm your password:  ✓

☐ Log in automatically  
☒ Require my password to log in

Back Continue

- Let it install.
- Restart VM by clicking “Restart Now”, then “Power Off” the VM.
- Click “Start” in the VB Dashboard to reboot Ubuntu.
- Your Ubuntu is ready for the next step!

## Step 2 - [Setup](#) Python 3, virtual environment, and Git

- Open terminal and type the following: (agree on everything asked)  
`$ sudo apt update`
- You should get something similar to this:

```
phi@phi:~$ sudo apt update
[sudo] password for phi:
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
123 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

- Proceed by typing:  
`$ sudo apt -y upgrade`
- Let the terminal do its work.
- Then, type:  
`$ python3 -V`
- You should have an output similar to this  
“Python 3.X.X”, where X represents a number. I had 3.6.8.
- Then type:  
`$ sudo apt install -y python3-pip`
- Followed by:  
`$ sudo apt install build-essential libssl-dev libffi-dev python3-dev`
- Now, set up our virtual environment. Type the following in terminal:  
`$ sudo apt install -y python3-venv`
- Install Git by typing:  
`$ sudo apt-get update`  
`$ sudo apt-get install git`



### Step 3 - Installing TensorFlow (TF)

- Follow the instructions mentioned in **Step 2** of this [guide](#).
- While following the instructions, you may encounter something as such:

```
>>> sess = tf.Session()
2019-09-29 21:44:40.112807: I tensorflow/core/platform/cpu_feature_guard.cc:142
] Your CPU supports instructions that this TensorFlow binary was not compiled t
o use: AVX2
2019-09-29 21:44:40.119632: I tensorflow/core/platform/profile_utils/cpu_utils.
cc:94] CPU Frequency: 2711995000 Hz
2019-09-29 21:44:40.125421: I tensorflow/compiler/xla/service/service.cc:168] X
LA service 0x4ccee10 executing computations on platform Host. Devices:
2019-09-29 21:44:40.125490: I tensorflow/compiler/xla/service/service.cc:175]
StreamExecutor device (0): <undefined>, <undefined>
>>> █
```

- This isn't something to stress about. You can choose to ignore it or [not](#).
- Now, proceed to **Step 3** of the link mentioned at the beginning of our third step.

### Step 4.1 - Trying things out!

- Suppose you've finished all the steps in the guide above, you should have tried running your `classify_image.py` for at least once.
- In fact, this is as far as we could get during our meeting because your president is retarded.
- Anyways, TO TRY DIFFERENT IMAGES, you must be in the `~/tf-demo/models/tutorials/image/imagenet` directory.
- Then, do  

```
$ python classify_image.py --image_file  
/home/user_name/Downloads/name_of_image
```
- `user_name` refers to the name you gave to your computer. In my case, it was phi.
- `name_of_image` refers to the name of the image you just downloaded/saved. Be sure to include its extension (jpg/jpeg/png) as well.
- Your path may be different than mine, but when the article mentioned "Absolute Path", it means the directory of your image.
- You can identify the directory of your image by navigating to the directory you saved the image with the help of the following commands:

```
$ cd name_of_directory      <- Takes you "into" the specified directory.
$ cd..                     <- Takes you "out" by one directory.
$ cd                       <- Takes you "out" all the way.
$ pwd                      <- Prints the path of your directory so that you can
                           type it down in the
```

`~/tf-demo/models/tutorials/image/imagenet`

directory.

---

Shoutout to Brian Yu and Matthew Jiang for pointing out that we should try including `image_file` and `--` as part of our argument. In fact, we did, but we were so close to getting the syntax right. At least its solved now :)

---

#### Step 4.2 - Try different images!

- Here we have a picture of a dog



- If I let the program to classify it, it would result in something like this:

```
(tensorflow-dev) phi@phi:~/tf-demo/models/tutorials/image/imagenet$ python classify_image.py --image_file /home/phi/Downloads/dog.jpeg
```

```
|  
|  
|  
V
```

```
Siberian husky (score = 0.48829)  
Eskimo dog, husky (score = 0.47239)  
malamute, malemute, Alaskan malamute (score = 0.00567)  
dogsled, dog sled, dog sleigh (score = 0.00198)  
white wolf, Arctic wolf, Canis lupus tundrarum (score = 0.00043)  
(tensorflow-dev) phi@phi:~/tf-demo/models/tutorials/image/imagenet$
```

Pretty accurate isn't it?

- What if I try something like this:



```
(tensorflow-dev) phi@phi:~/tf-demo/models/tutorials/image/imagenet$ python classify_image.py --image_file /home/phi/Downloads/dogncat.jpeg
```

|  
|  
|  
V

```
Siberian husky (score = 0.55026)  
Eskimo dog, husky (score = 0.33505)  
dogsled, dog sled, dog sleigh (score = 0.00504)  
malamute, malemute, Alaskan malamute (score = 0.00237)  
eggnog (score = 0.00119)  
(tensorflow-dev) phi@phi:~/tf-demo/models/tutorials/image/imagenet$ python classify_image.py --image_file /home/phi/Downloads/dogncat.jpeg
```

- Not as much anymore, isn't it? I mean, where's the cat?
- This is just one of many examples that demonstrated the limitations of computer vision, but that is not to say that it will be like this forever.
- We can train our program to be specialized in certain objects by supplying data sets that solely has the images of the targeted object. For instance, if I were to train my image classifier solely for the purpose of identifying cats, I am certain that it can tell that there is a cat, but not a dog. Thus, to have it to identify both, I will have to supply images of both animals to increase its accuracy, which sounds easy but its not.



Here's another example that shows the level of specialization of our image classifier:



```
(tensorflow-dev) phi@phi:~/tf-demo/models/tutorials/image/imagenet$ python classify_image.py --image_file /home/phi/Downloads/doggocream.jpeg
```

|  
|  
|  
V

```
Walker hound, Walker foxhound (score = 0.39345)  
bluetick (score = 0.13793)  
EntleBucher (score = 0.11709)  
English foxhound (score = 0.08456)  
Appenzeller (score = 0.06078)  
(tensorflow-dev) phi@phi:~/tf-demo/models/tutorials/image/imagenet$
```

- It is evident that our classifier was not trained to identify ice cream other than dogs.
- Do try it out with other images and see what you get!

