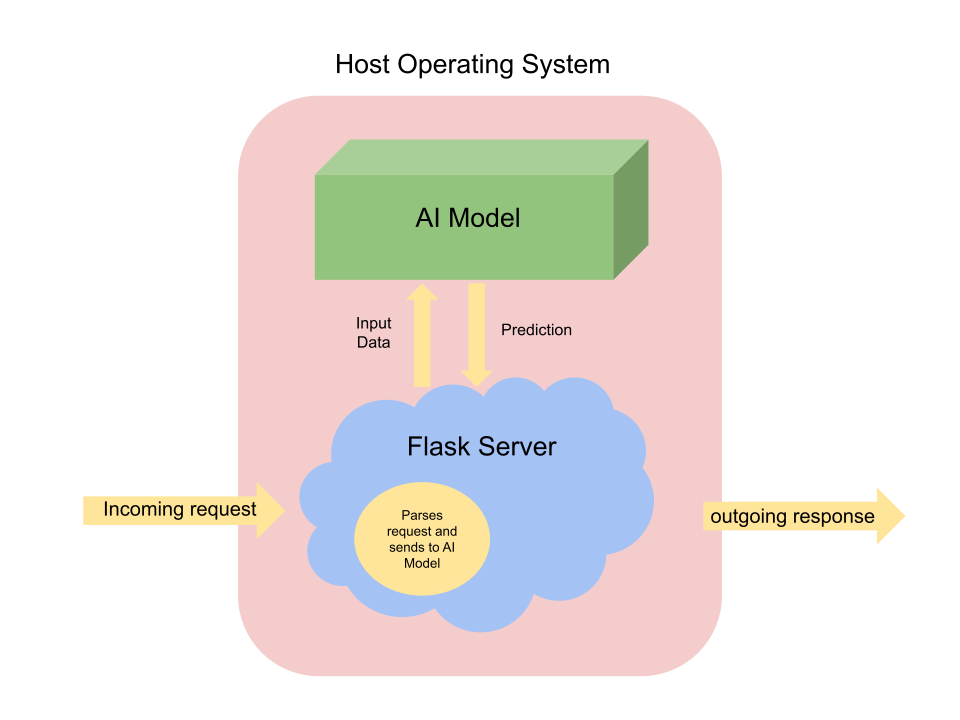
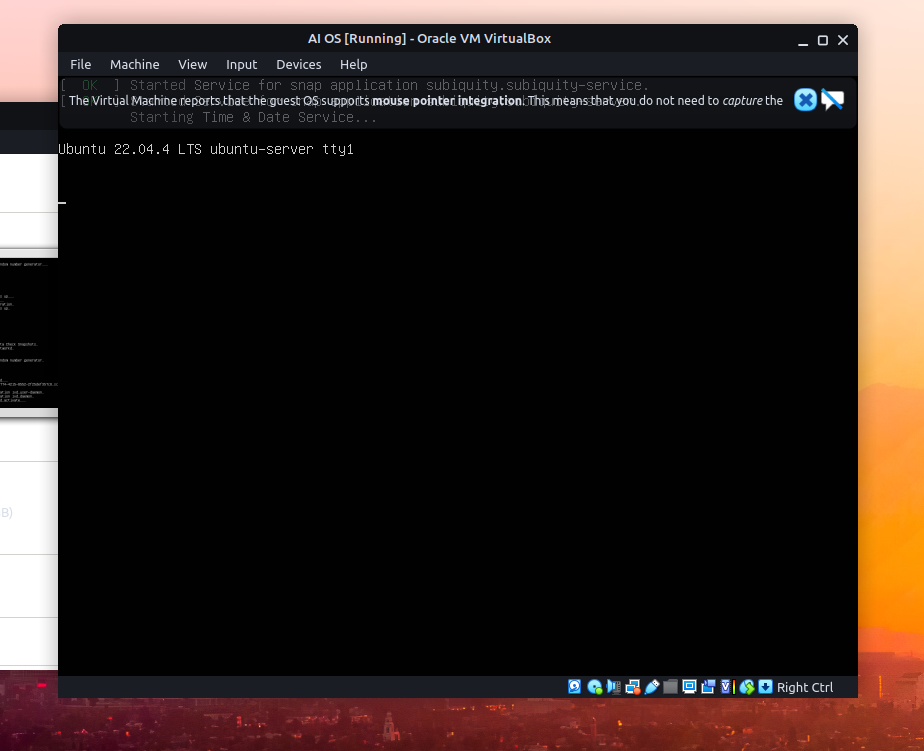
Final Project Specifications EGR226A, for the group of Michael E Newbold, and Garret Washburn.

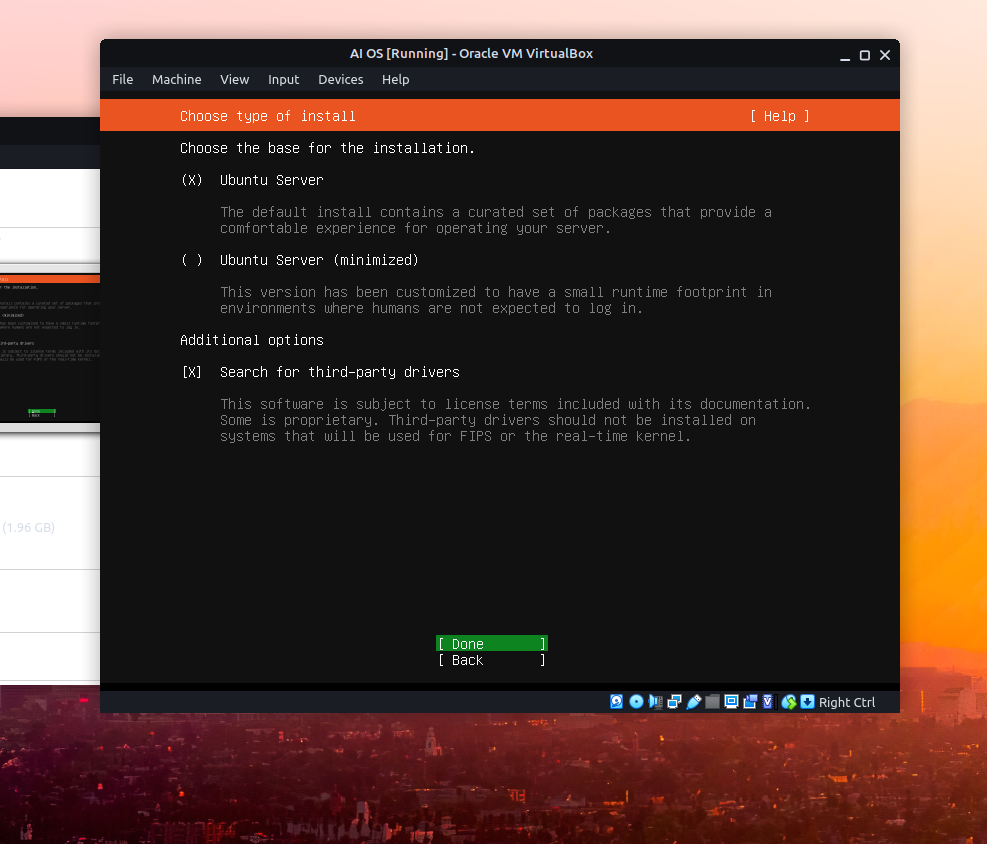
AI is a fast-evolving domain whose dynamic nature necessitates the use of constructively-purposeful infrastructure to host AI models, which means that the need for resourceful and purpose-built infrastructures to support AI model hosting has never been greater. Our project intends to create an OS that is specifically designed for AI model hosting. We have based this work on Ubuntu Server 22.04, whose core functionalities we have improved by adding key elements and optimizations needed for smooth deployment of AI models. In addition, it had the latest Python version as well as inclusive drivers for peripherals like microphones and speakers, which made it best-suited for processing sound in artificial intelligence applications (AI). This project is based off of the following image, as described in the midterm project.



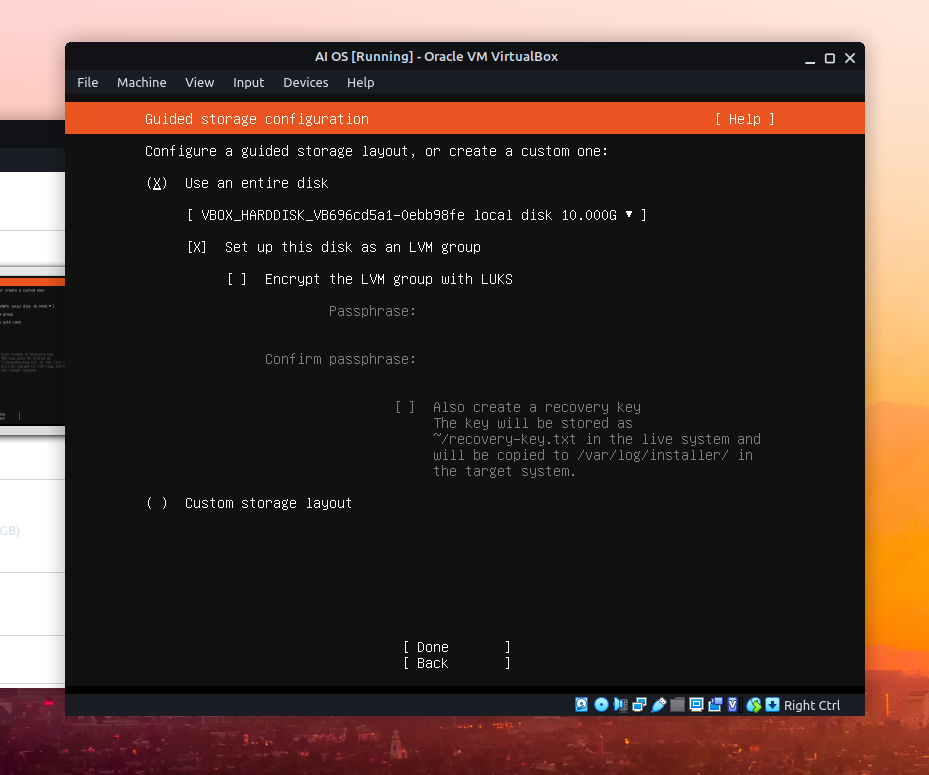
The image serves as a diagrammatic representation of our project's architecture and flow by showing how parts work together to build up our customized operating system’s components with their own unique properties or functions. By doing so, we seek to aid in enhancing AI infrastructure thereby enabling easier and more effective ways of deploying and managing AI models.

**Installing Ubuntu Server**

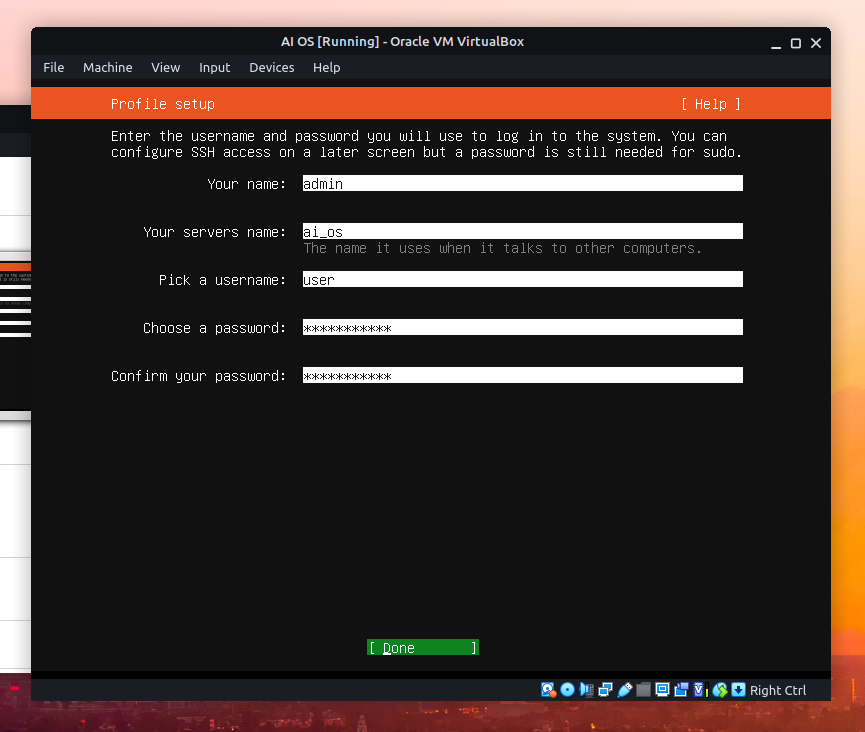




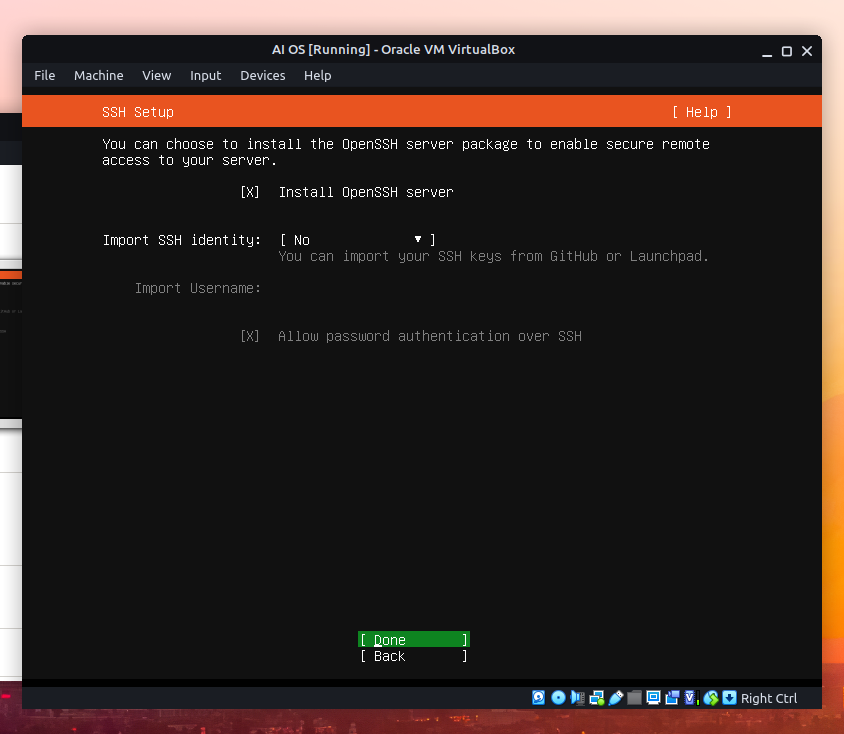
We went ahead and chose to install the base version of Ubuntu Server, as it is already quite minimized, as well as search for third-party drivers as this OS needs to be able to run on a wide array of hardware.



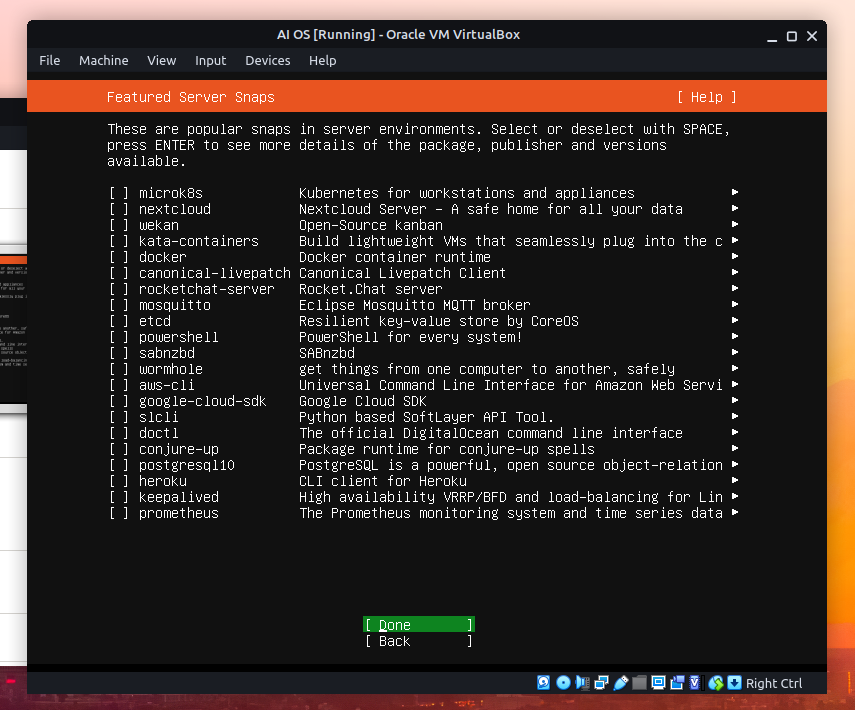
We didn’t really see a need for encryption.



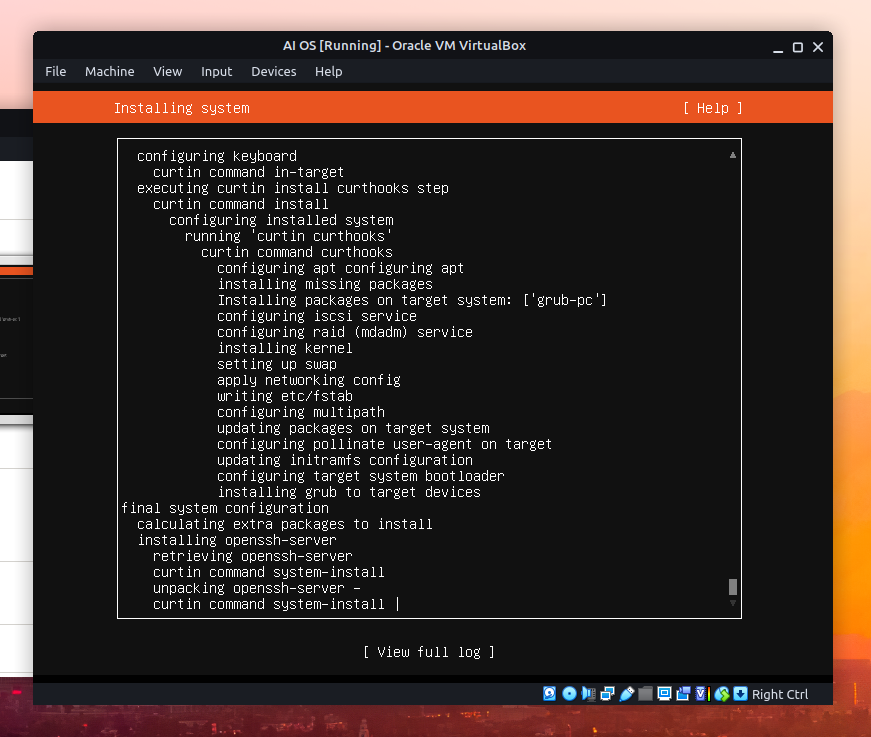
Setting up the profile information, we decided to keep it simple but to use the ai\_os name for the server.



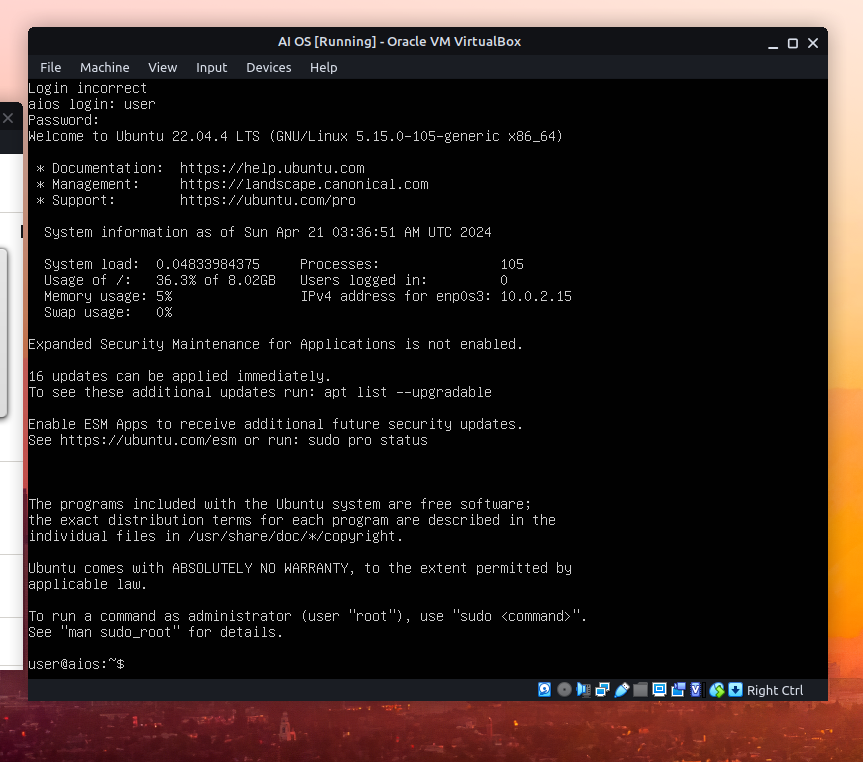
We DID decide to use ssh, as this server will not have a desktop environment and it is likely that the users will use this operating system without a monitor as that is part of it’s intended use.



We didn’t install any snaps, as a matter of fact, we plan to come back to snapd and cut it out, as it is unnecessary for this OS.



After all of the configuration, we gave it plenty of time to compile.



After everything was set up, it was time to begin the process of configuring the default Ubuntu Server OS to become our AI\_os operating System.

**Configuring AI\_OS**

I wanted to check and see if Python was already installed, so I ran

‘’’

Sudo apt list –installed | grep python

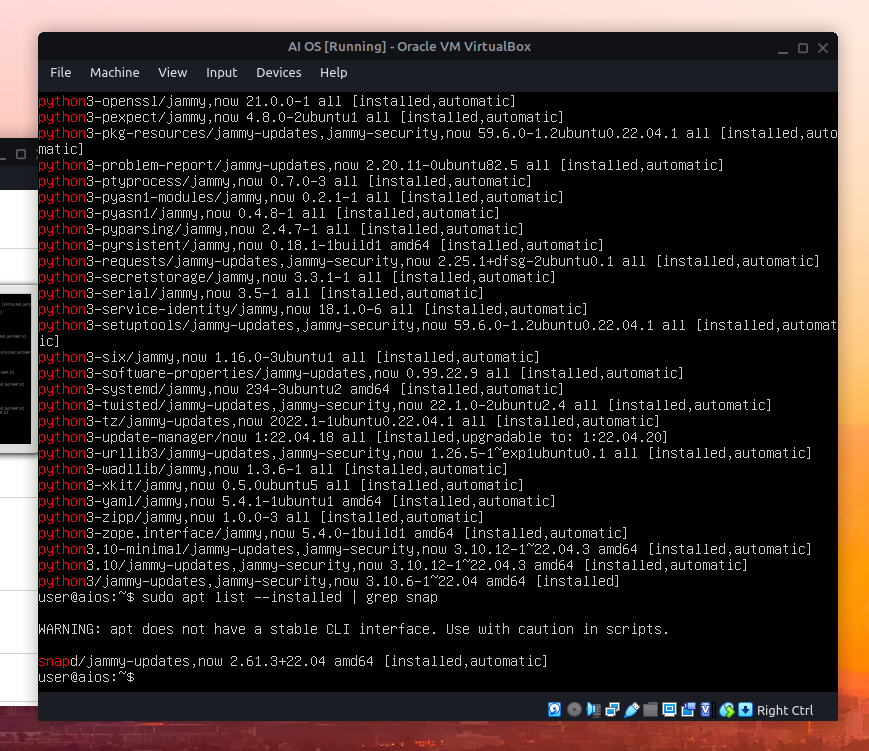
‘’’

Which confirmed that Python was installed. However, I wanted to start chopping out some of the bulky unnecessary software. I started with the package manager I have the most personal beef with, snapd. I chopped it out after searching for it using:

‘’’

Sudo apt list –installed | grep snapd

‘’’



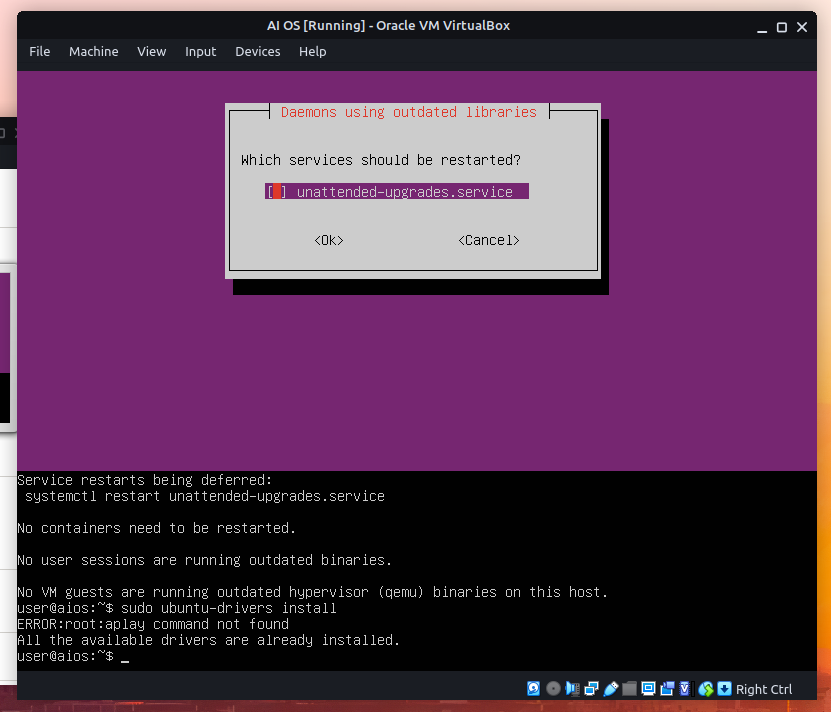
To make sure that the OS was compatible with most hardware, I went ahead and ran:

‘’’

sudo ubuntu-drivers install

‘’’

Which installed.



Additionally, I also went ahead and installed net-tools.

From there, all that was left to do was to install all of the necessary python packages for working with AI models. The baseline packages I went ahead and included were Tensorflow, keras, langchain, langchain\_core, numpy, pandas, pillow, scikit-learn, and most importantly Flask as that is the Library python will use for hosting the servers… it’s kind of important as the whole project is an operating system that hosts AI servers!

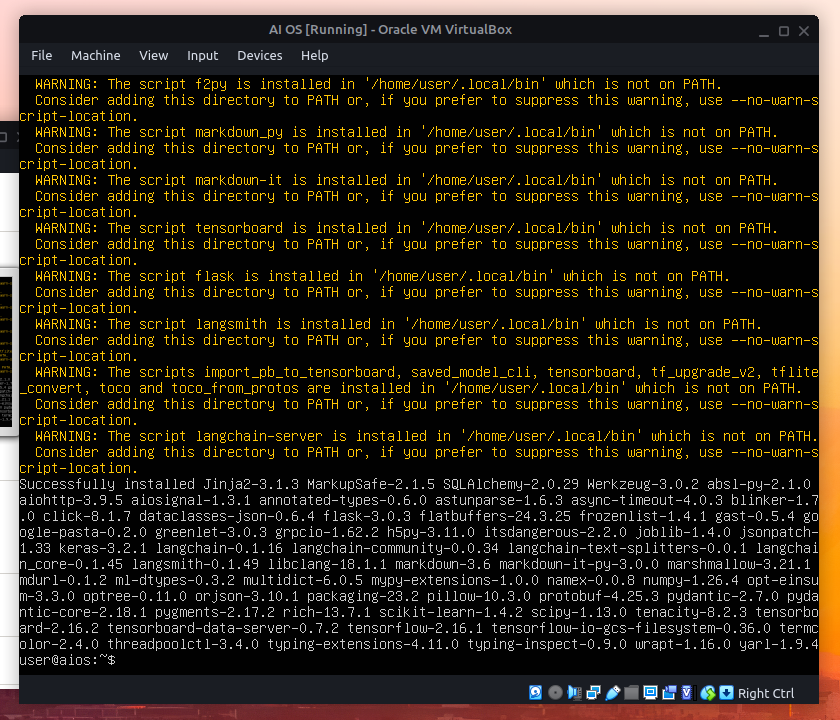
Here is the pip install command I used:

‘’’

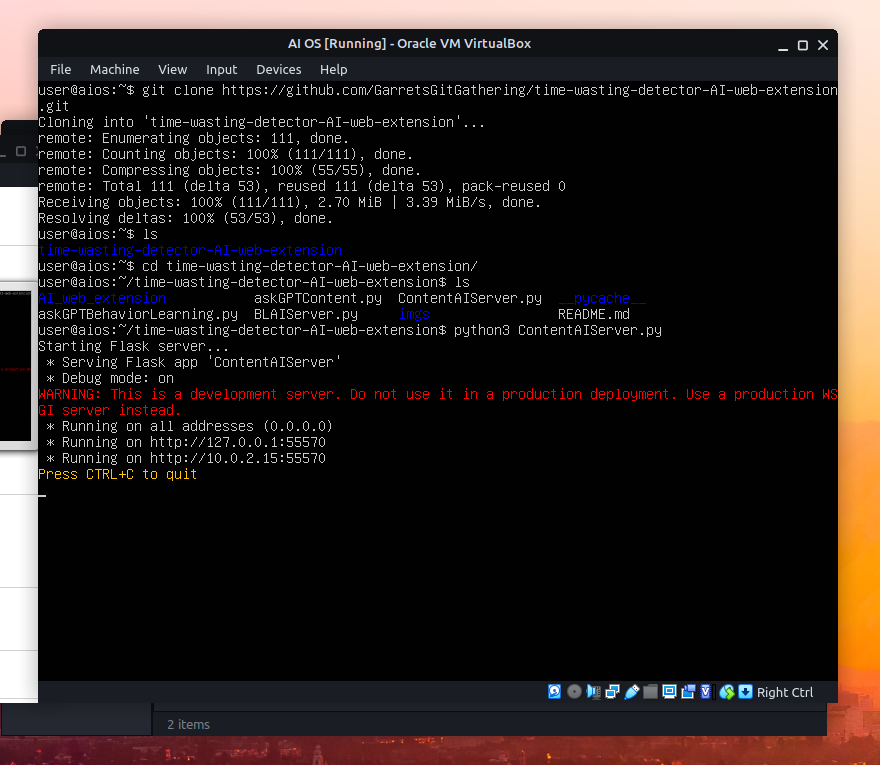
Pip3 install tensorflow keras langchain langchain\_core langchain\_openai google-generativeai numpy pillow scikit-learn flask

‘’’

This took… quite a while to install, ~8 mins.



From there, all that was left to do was test it out tossing an AI model on it and running a server. We are going to use a server for another project Garret has developed, which uses the langchain library.



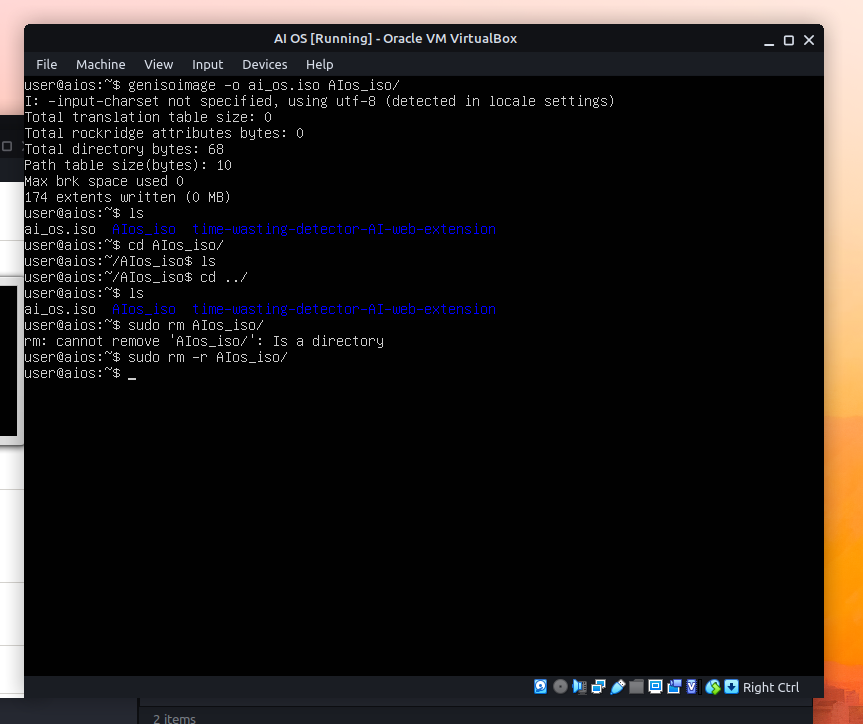
As you can see, the Operating System works great. From there all there is to do was to make a copy of the iso image, so it can be shared and moved around!

‘’’

Sudo apt install genisoimage

genisoimage -o output\_image.iso directory\_name

‘’’



That completes the project!