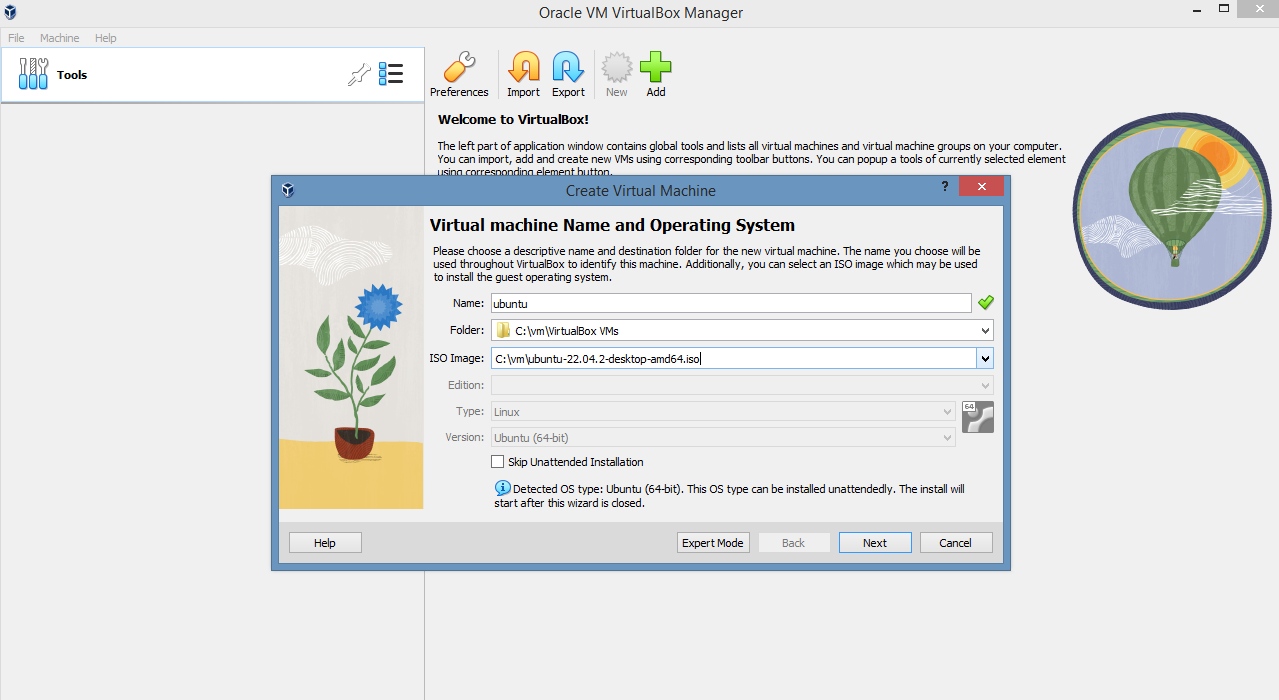
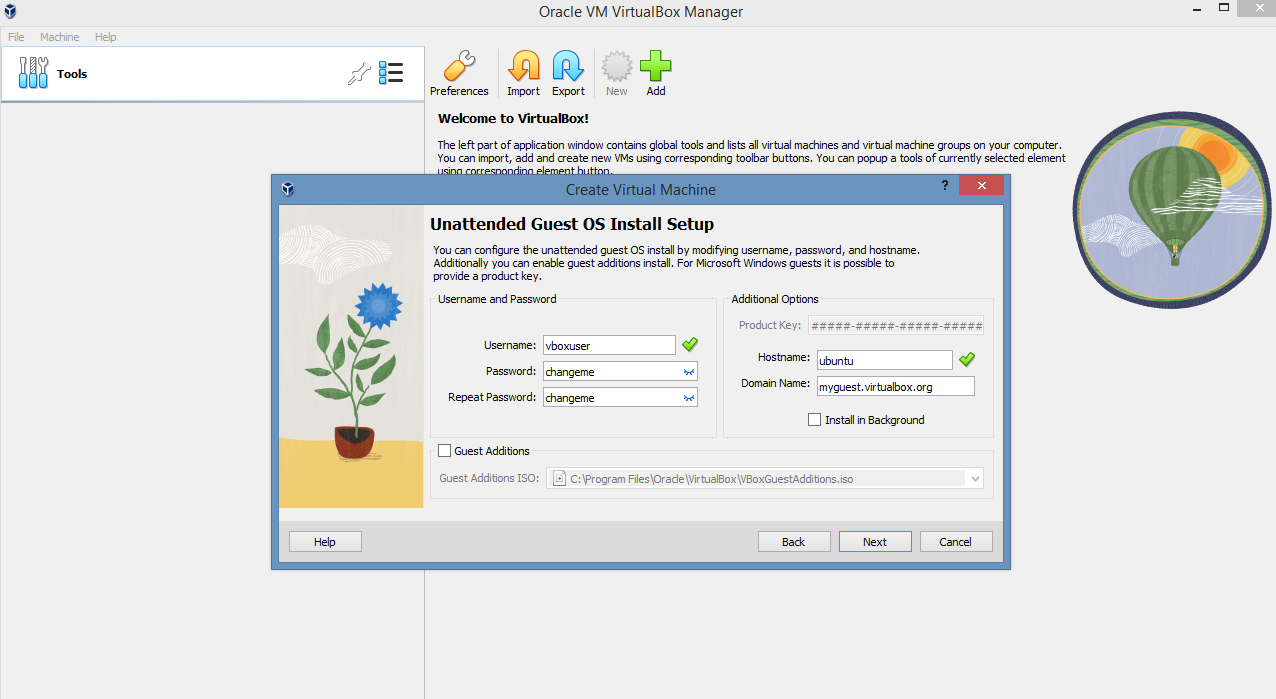
1. **Host a Ubuntu Virtual Machine using Oracle VM Virtual Box**

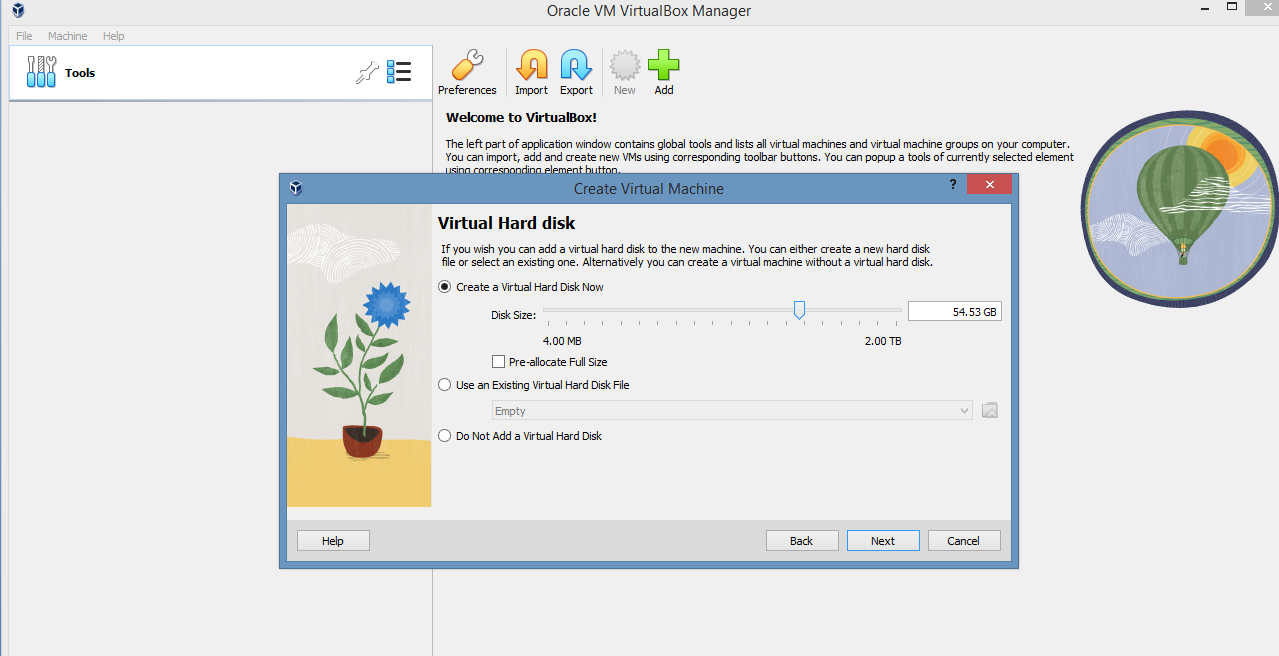
Install ubuntu OS inside a VM



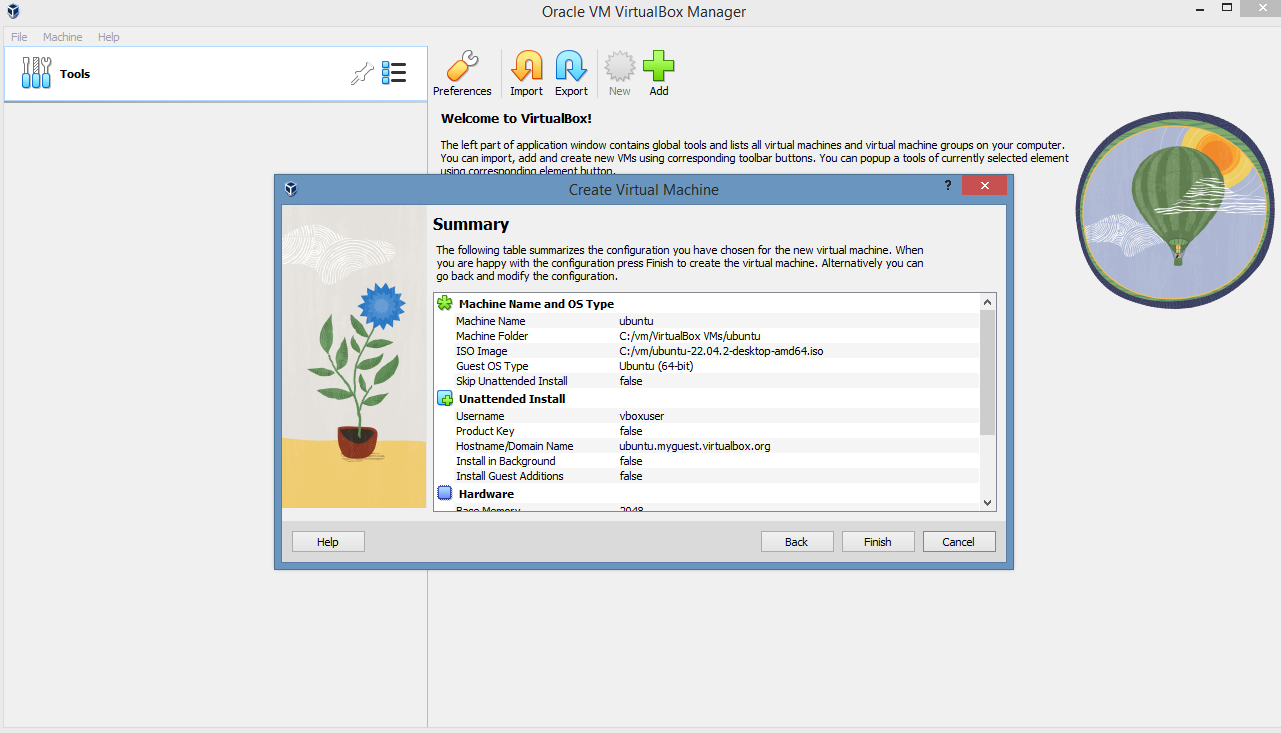
-



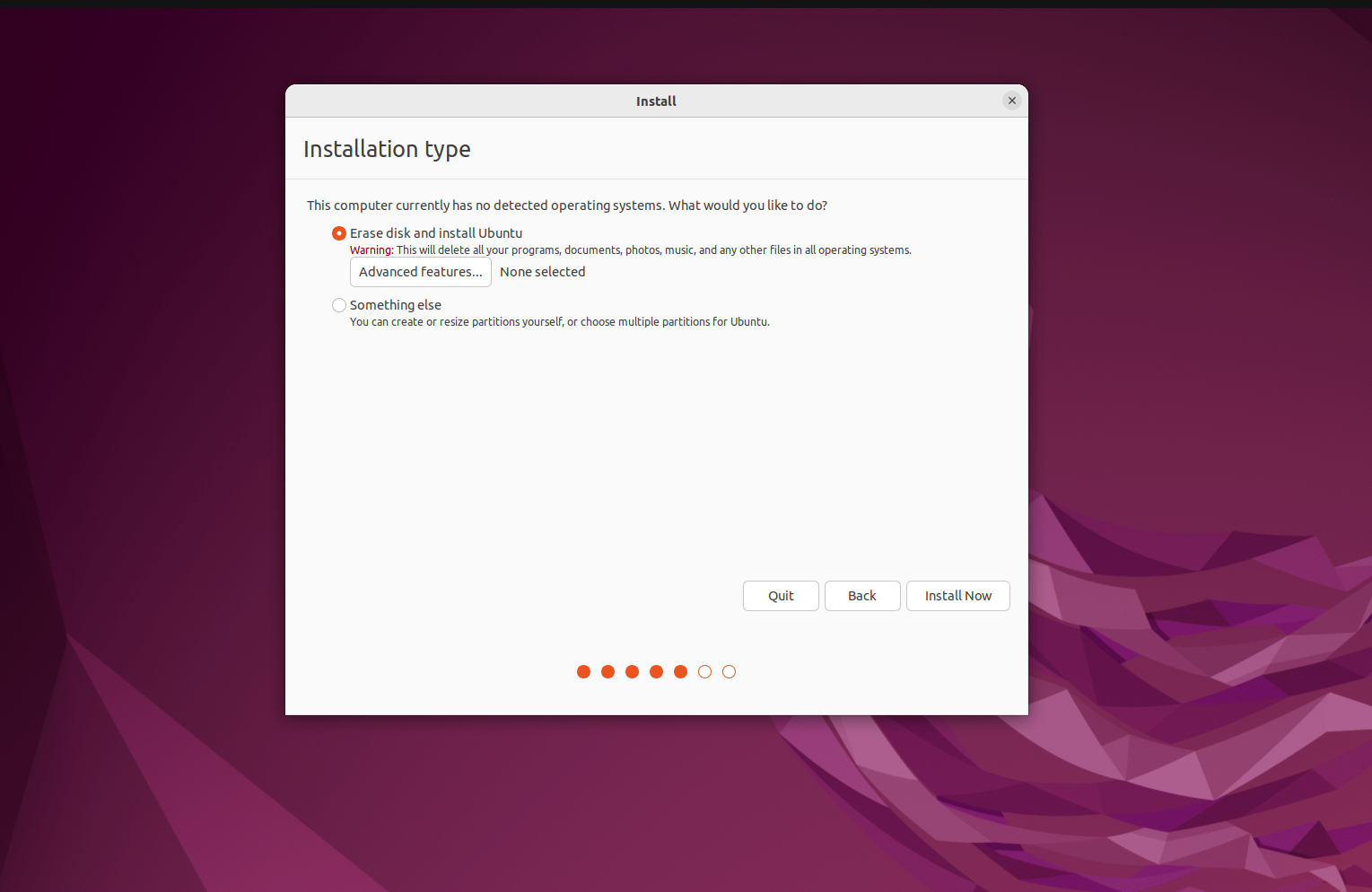
-

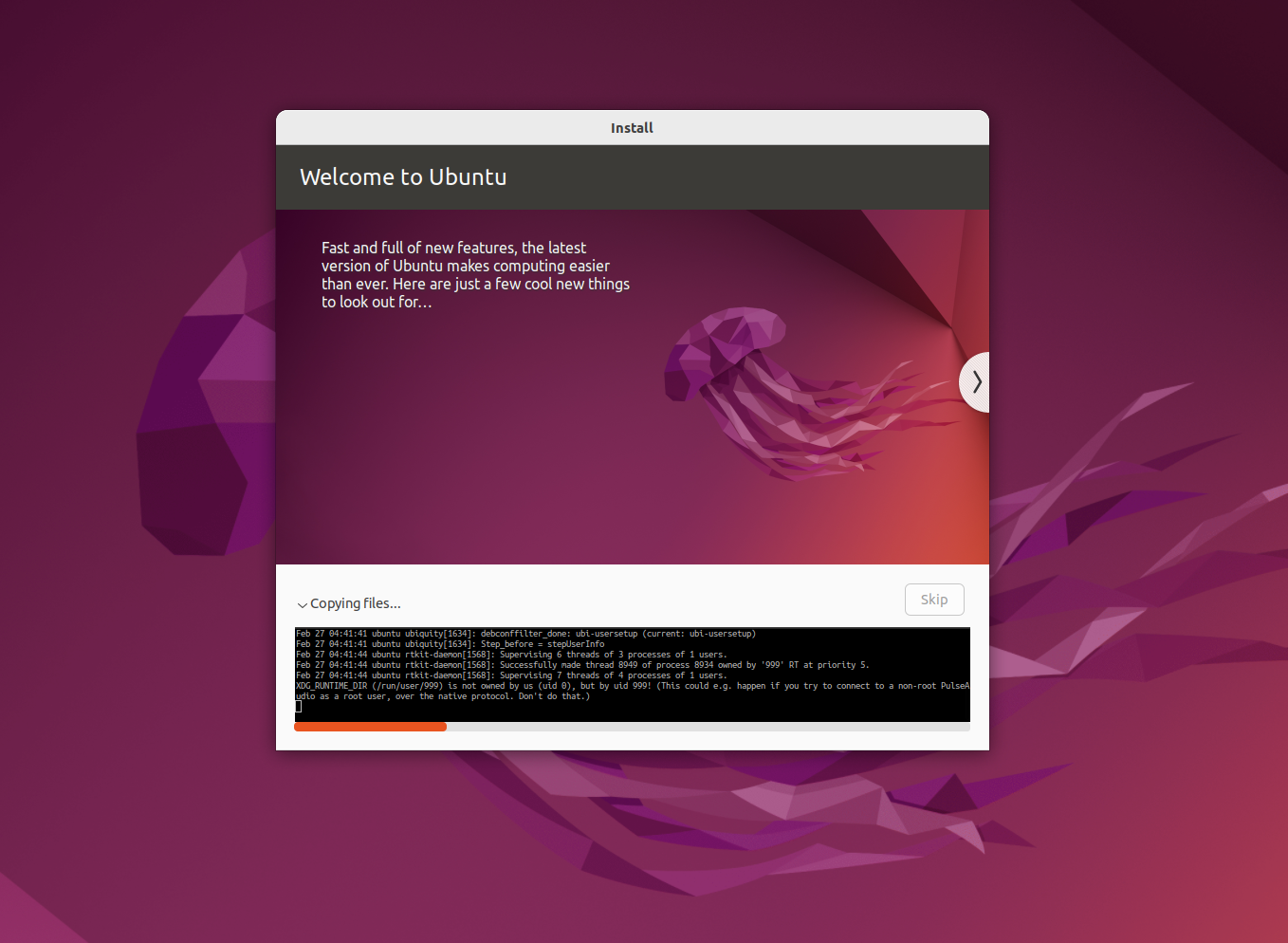


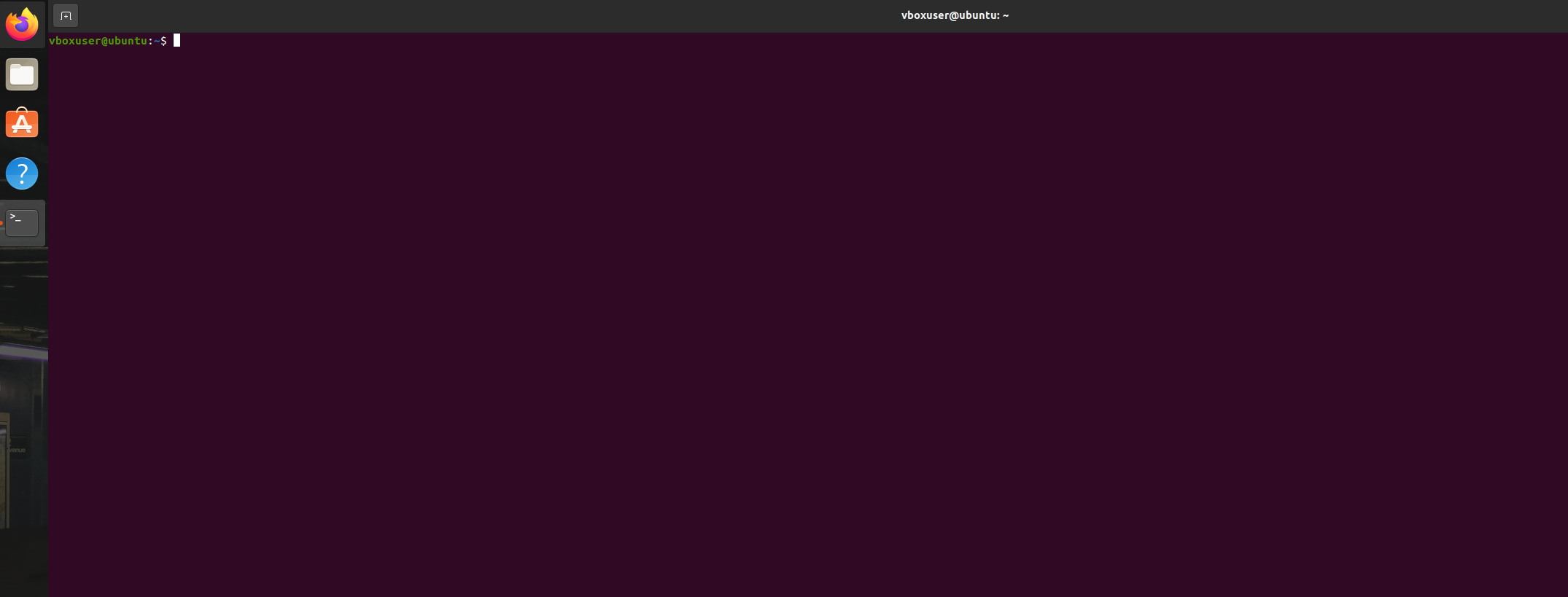
-



-

-

-

**2. Set up Visual Studio code on Ubuntu**

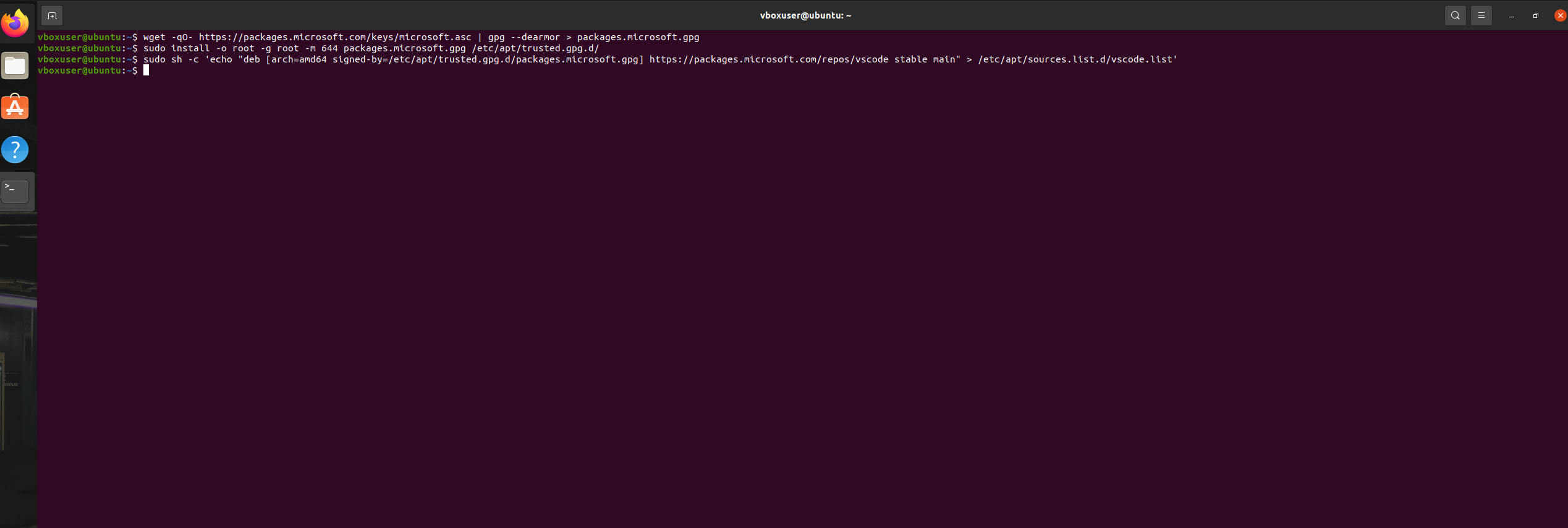
Add the Microsoft repository by running the following commands

curl -sSL https://packages.microsoft.com/keys/microsoft.asc | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/microsoft.gpg

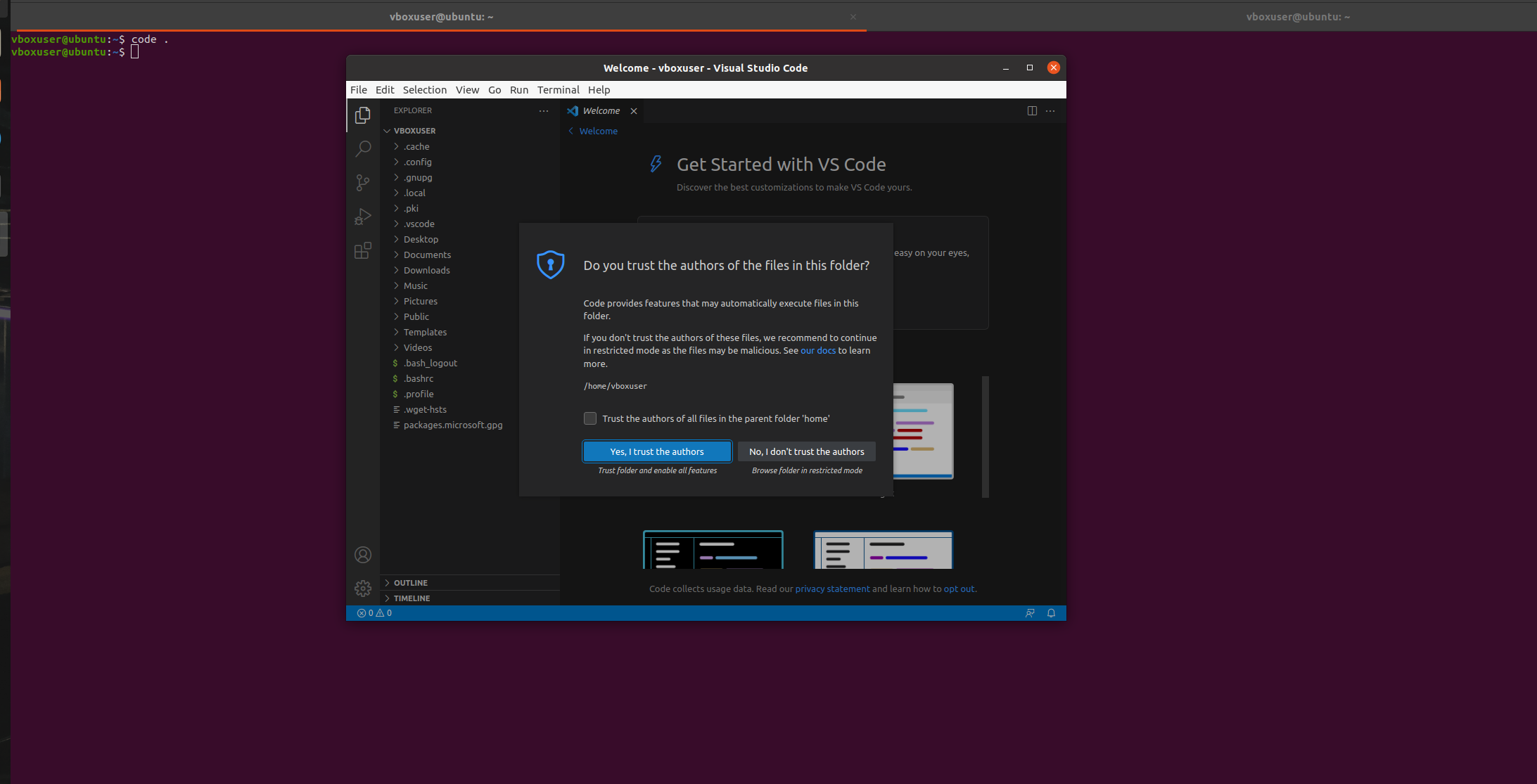
echo "deb [arch=amd64 signed-by=/etc/apt/trusted.gpg.d/microsoft.gpg] https://packages.microsoft.com/repos/vscode stable main" | sudo tee /etc/apt/sources.list.d/vscode.list

sudo apt update

sudo apt install -y code

Install Visual Studio Code by running the following command

Launch vscode



**3. Set up Python**

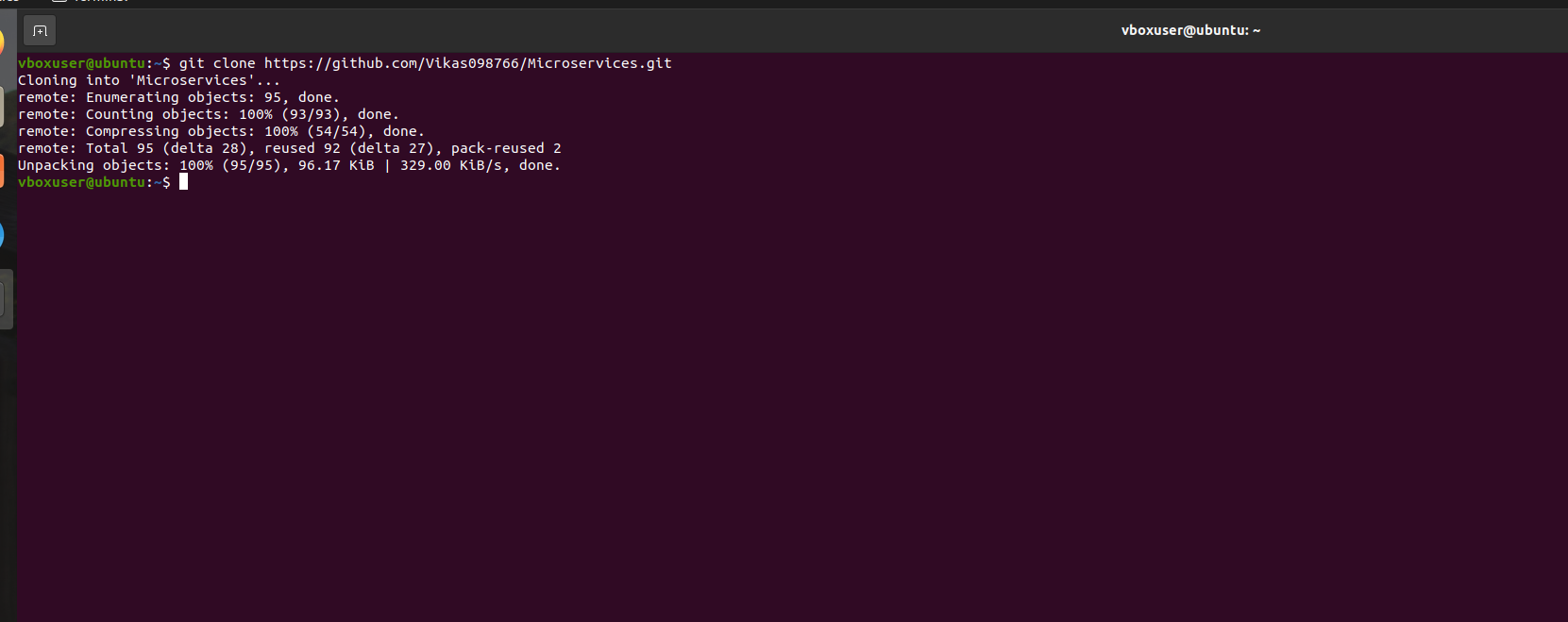
Install Python3 using below command

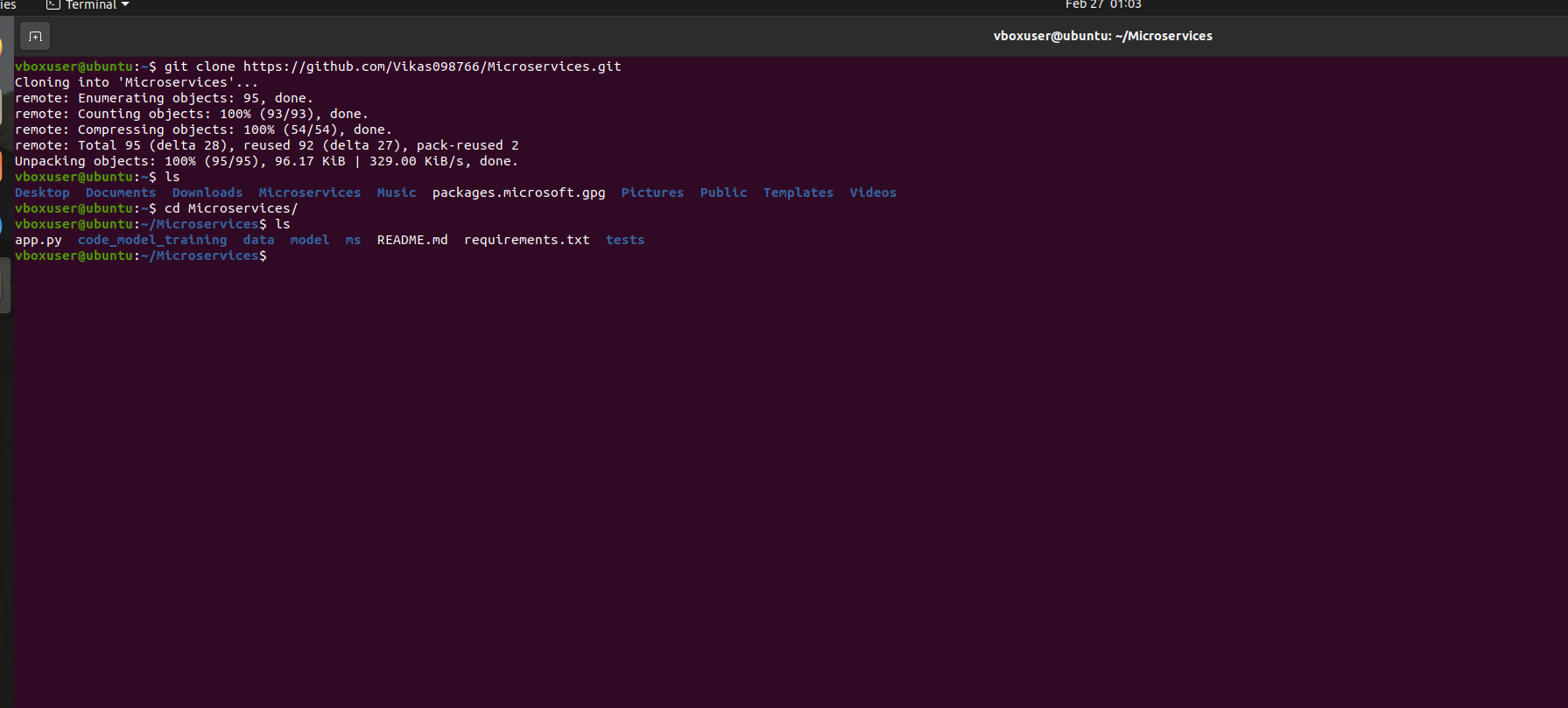
sudo apt install python3

Launch python

**4.Clone this Github repositoryhttps://github.com/Vikas098766/Microservices.git using command**

git clone https://github.com/Vikas098766/Microservices.git

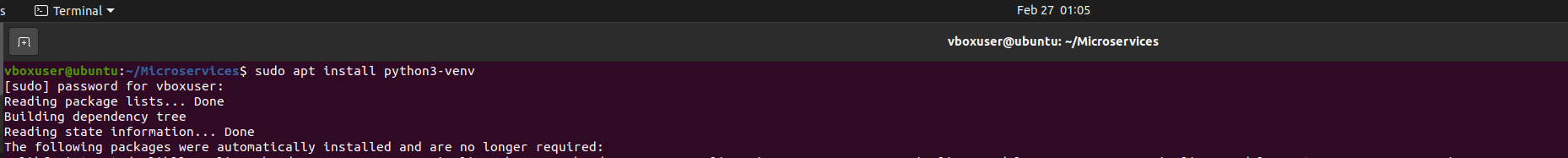
-

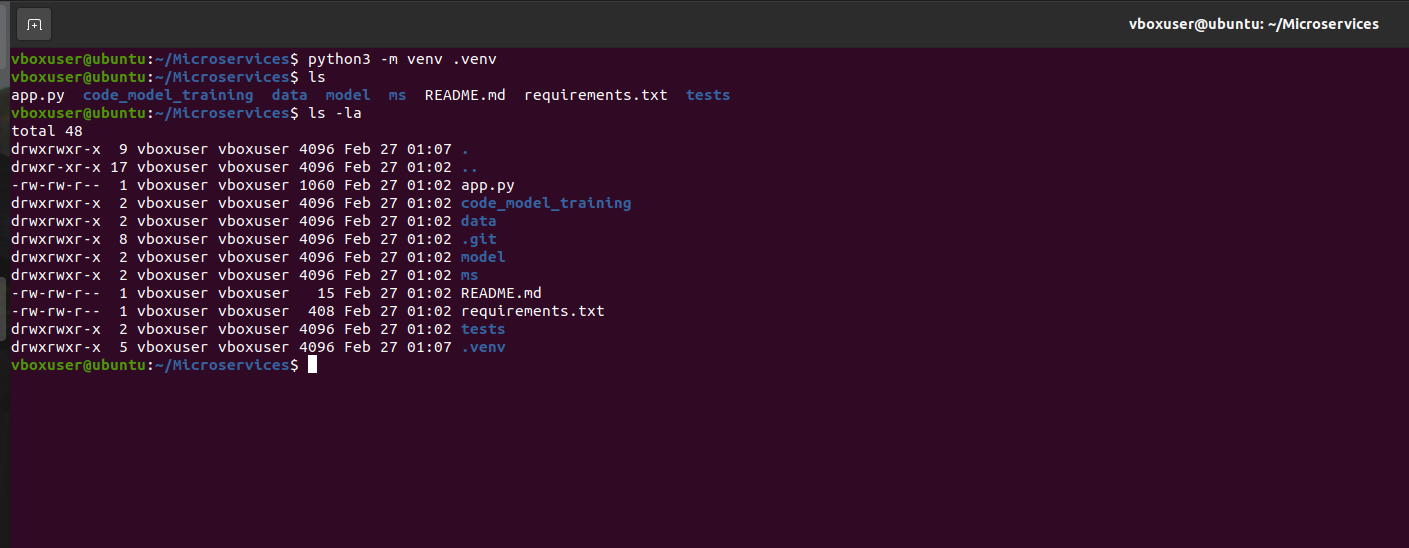
Create a Virtual Environment

Install the python3-venv using command and create a new virtual env

sudo apt install python3-venv

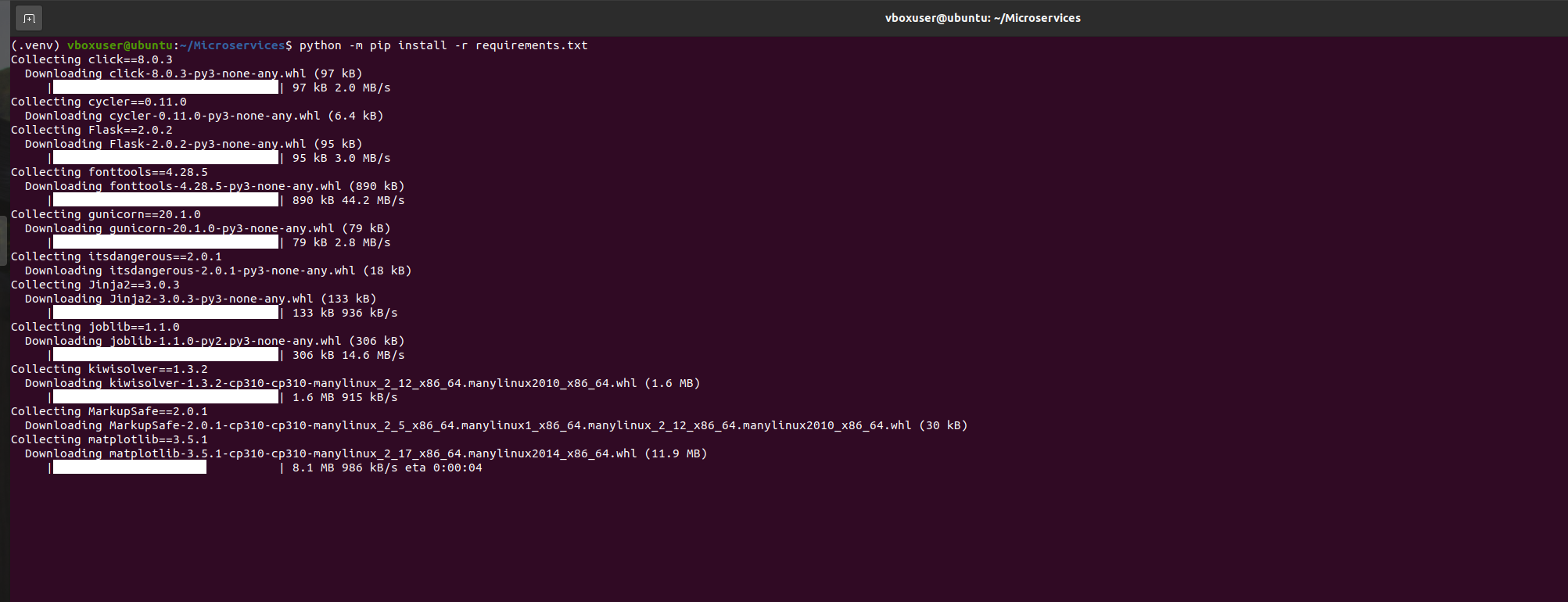
python3 -m venv .venv

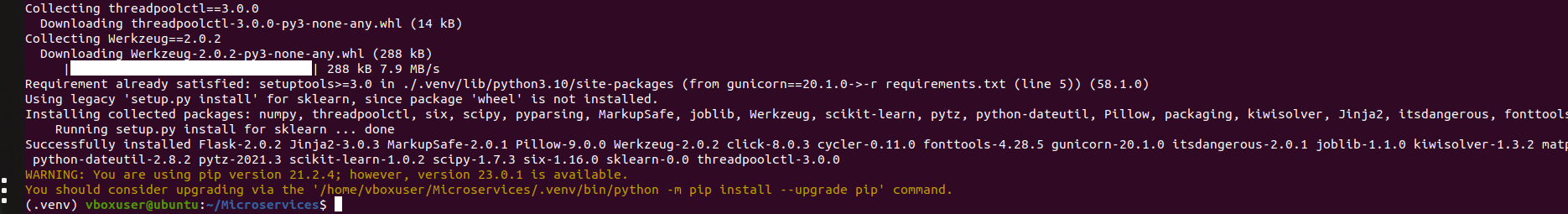
Create the virtual environment by running the following command

Activate the virtual environment

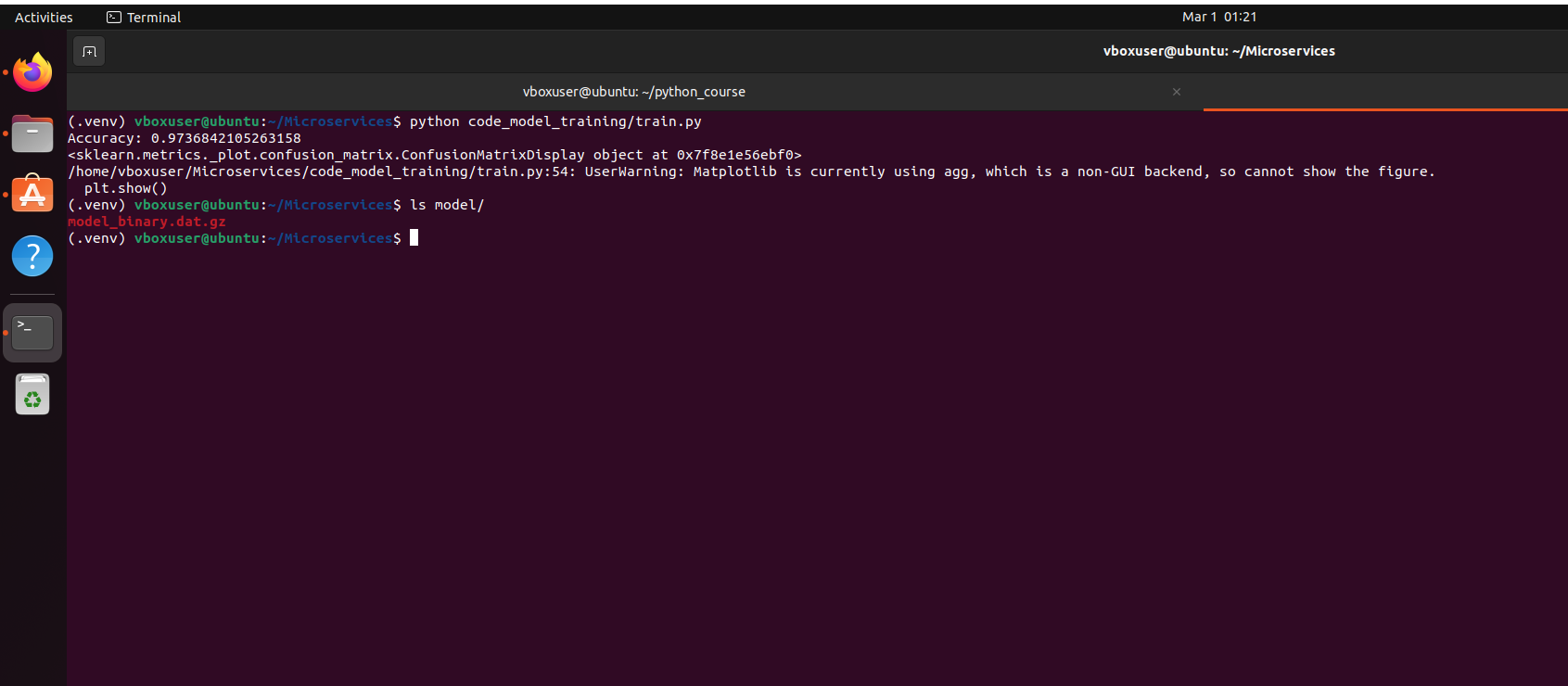
 **6.Install the dependencies from requirements.txt file.**

Python -m pip install -r requirements.txt

-

**7.Train and save the model**

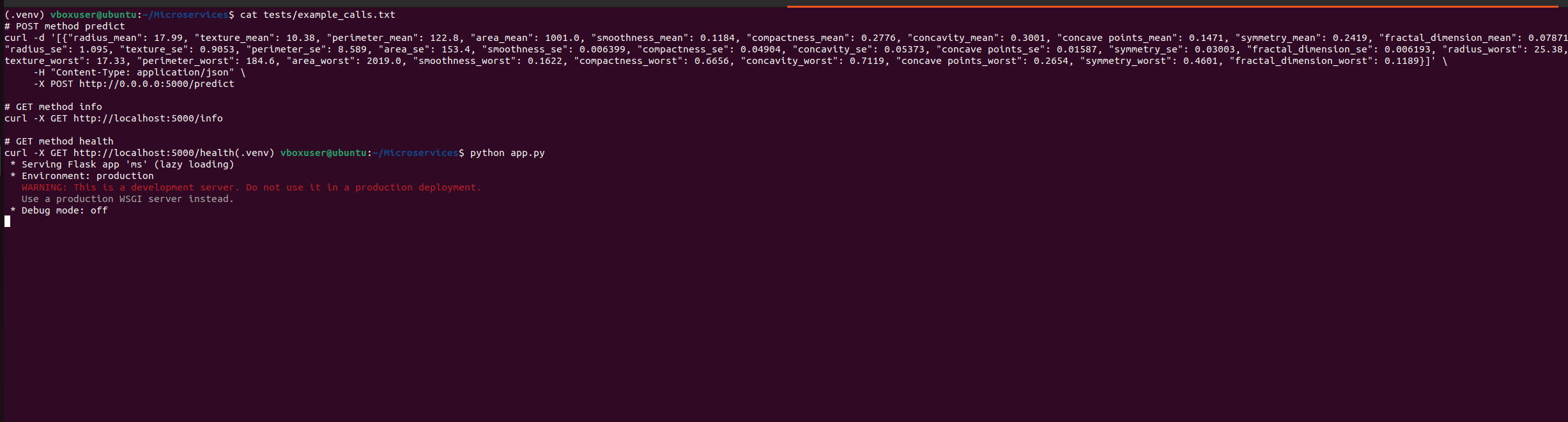
python code\_model+training/train.py



**8.Test the Flask web application**

Start the flask development server

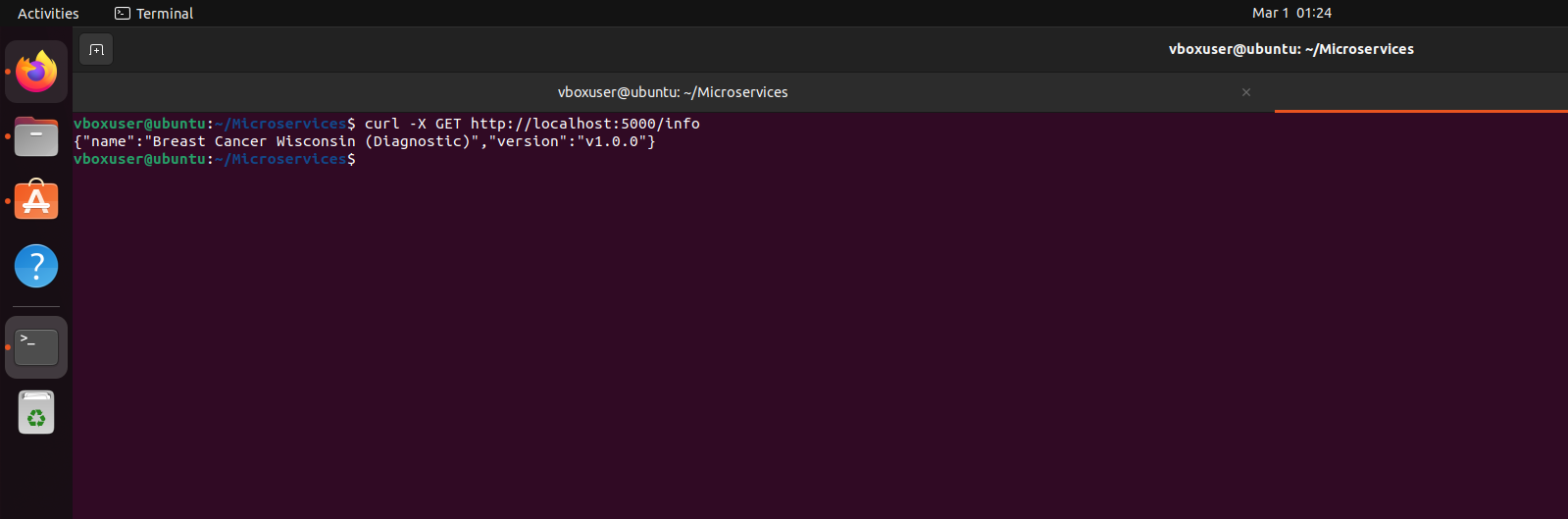
python app.py



**9.Test the application and make predictions using the example calls available in the folder/tests**

Get the app info

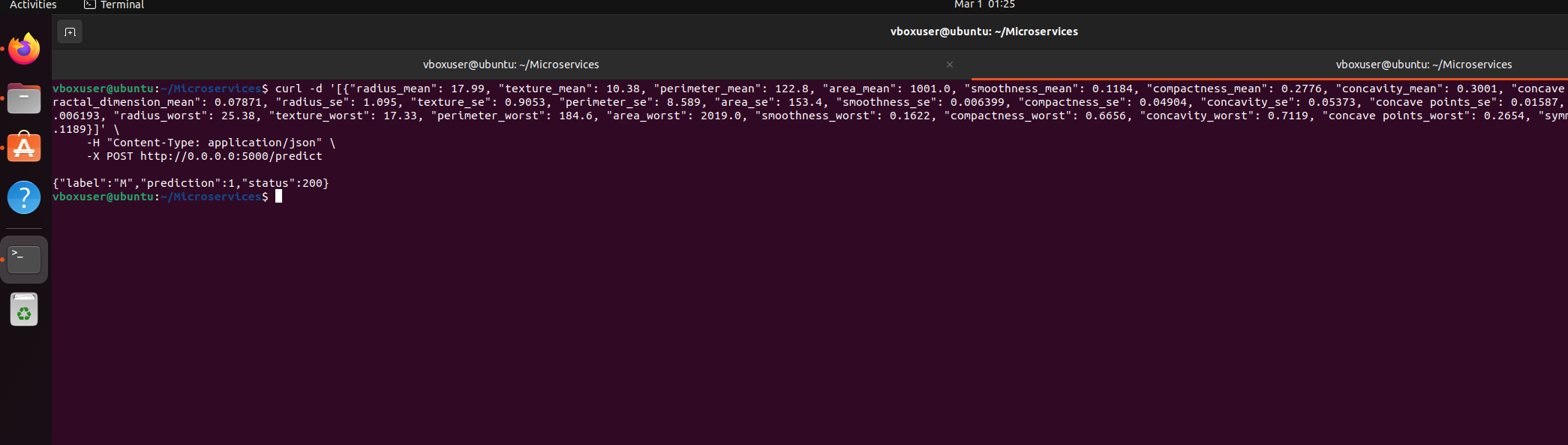
curl -X GET http://localhost:5000/info

Call the predict API

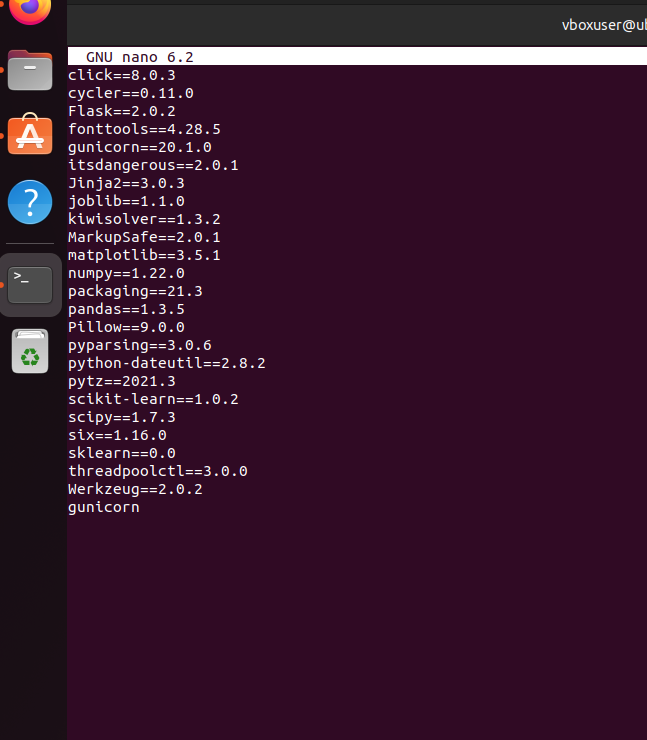
curl -d '[{"radius\_mean": 17.99, "texture\_mean": 10.38, "perimeter\_mean": 122.8, "area\_mean": 1001.0, "smoothness\_mean": 0.1184, "compactness\_mean": 0.2776, "concavity\_mean": 0.3001, "concave points\_mean": 0.1471, "symmetry\_mean": 0.2419, "fractal\_dimension\_mean": 0.07871, "radius\_se": 1.095, "texture\_se": 0.9053, "perimeter\_se": 8.589, "area\_se": 153.4, "smoothness\_se": 0.006399, "compactness\_se": 0.04904, "concavity\_se": 0.05373, "concave points\_se": 0.01587, "symmetry\_se": 0.03003, "fractal\_dimension\_se": 0.006193, "radius\_worst": 25.38, "texture\_worst": 17.33, "perimeter\_worst": 184.6, "area\_worst": 2019.0, "smoothness\_worst": 0.1622, "compactness\_worst": 0.6656, "concavity\_worst": 0.7119, "concave points\_worst": 0.2654, "symmetry\_worst": 0.4601, "fractal\_dimension\_worst": 0.1189}]' \

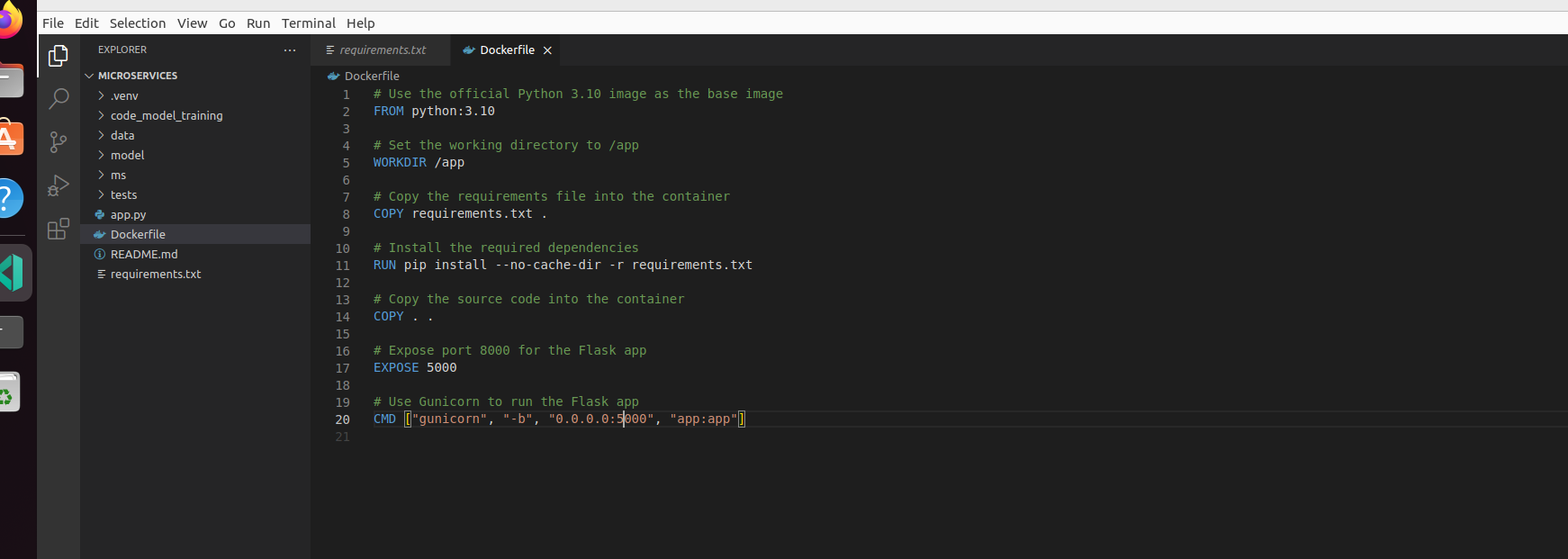
-H "Content-Type: application/json" \

-X POST http://0.0.0.0:5000/predict

**10. Create a docker image containing everything needed to run the application**.

Add gunicorn server dependency in requirements file

Create Dockerfile

Install Docker engine using below command

sudo apt-get update

sudo apt-get install \

ca-certificates \

curl \

gnupg \

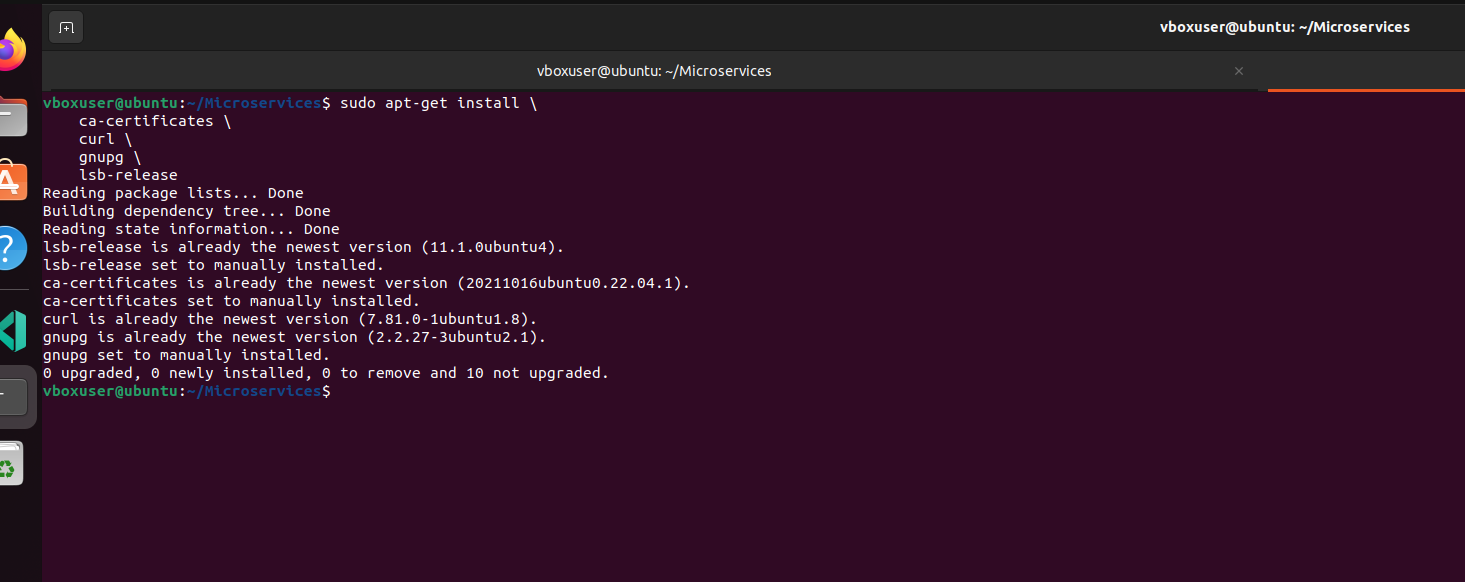
lsb-release

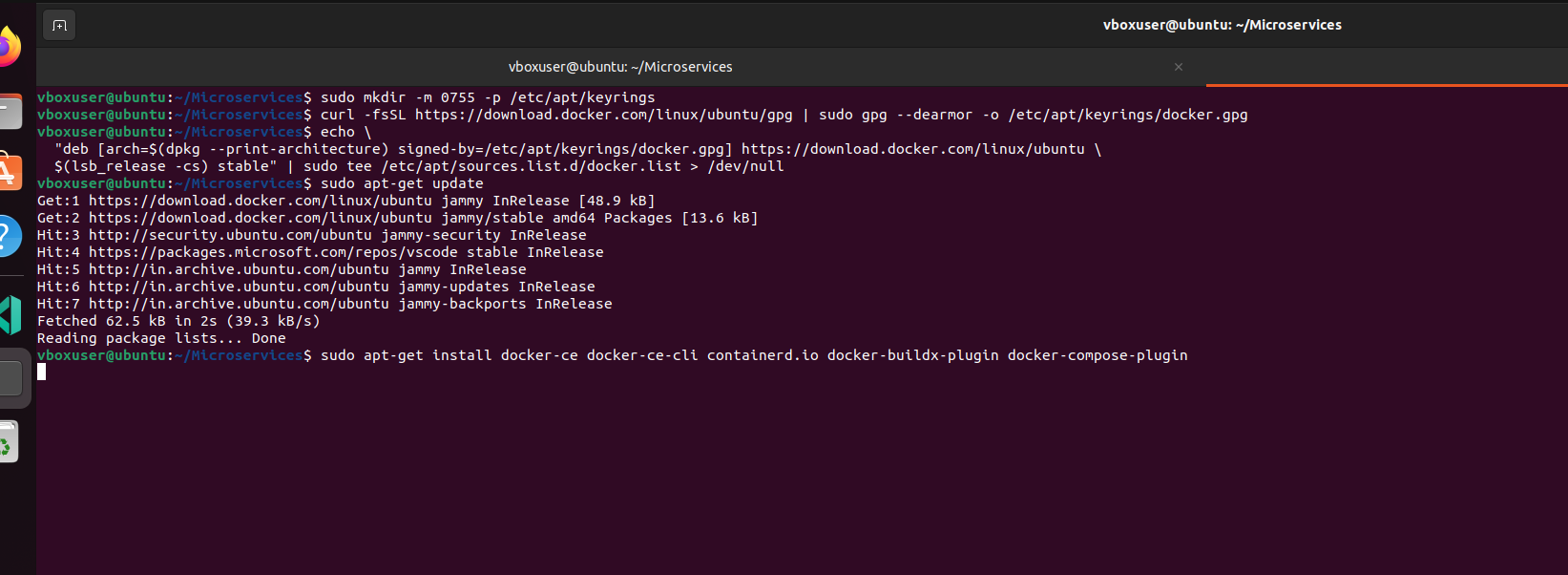
sudo mkdir -m 0755 -p /etc/apt/keyrings

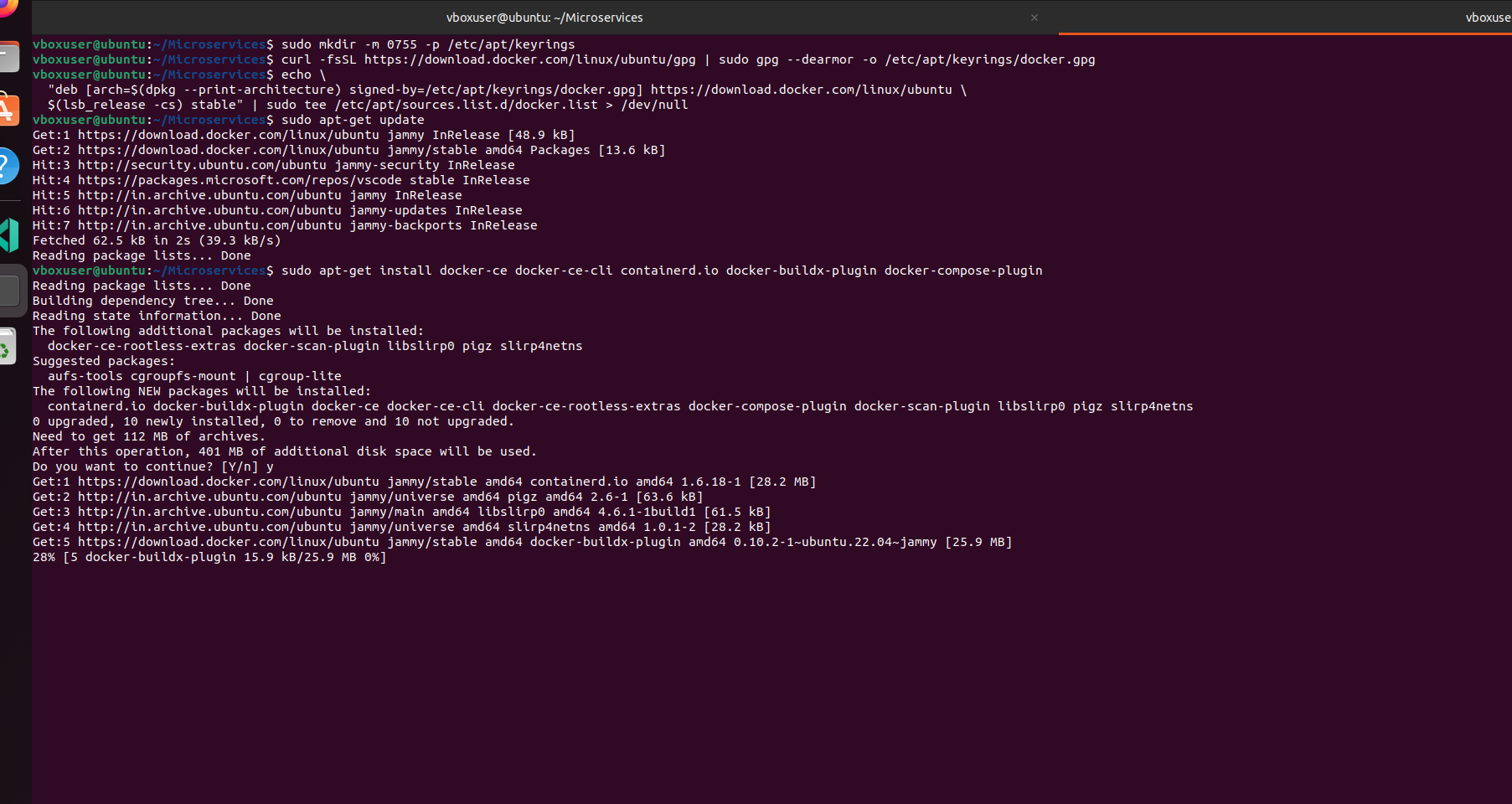
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo apt-get update

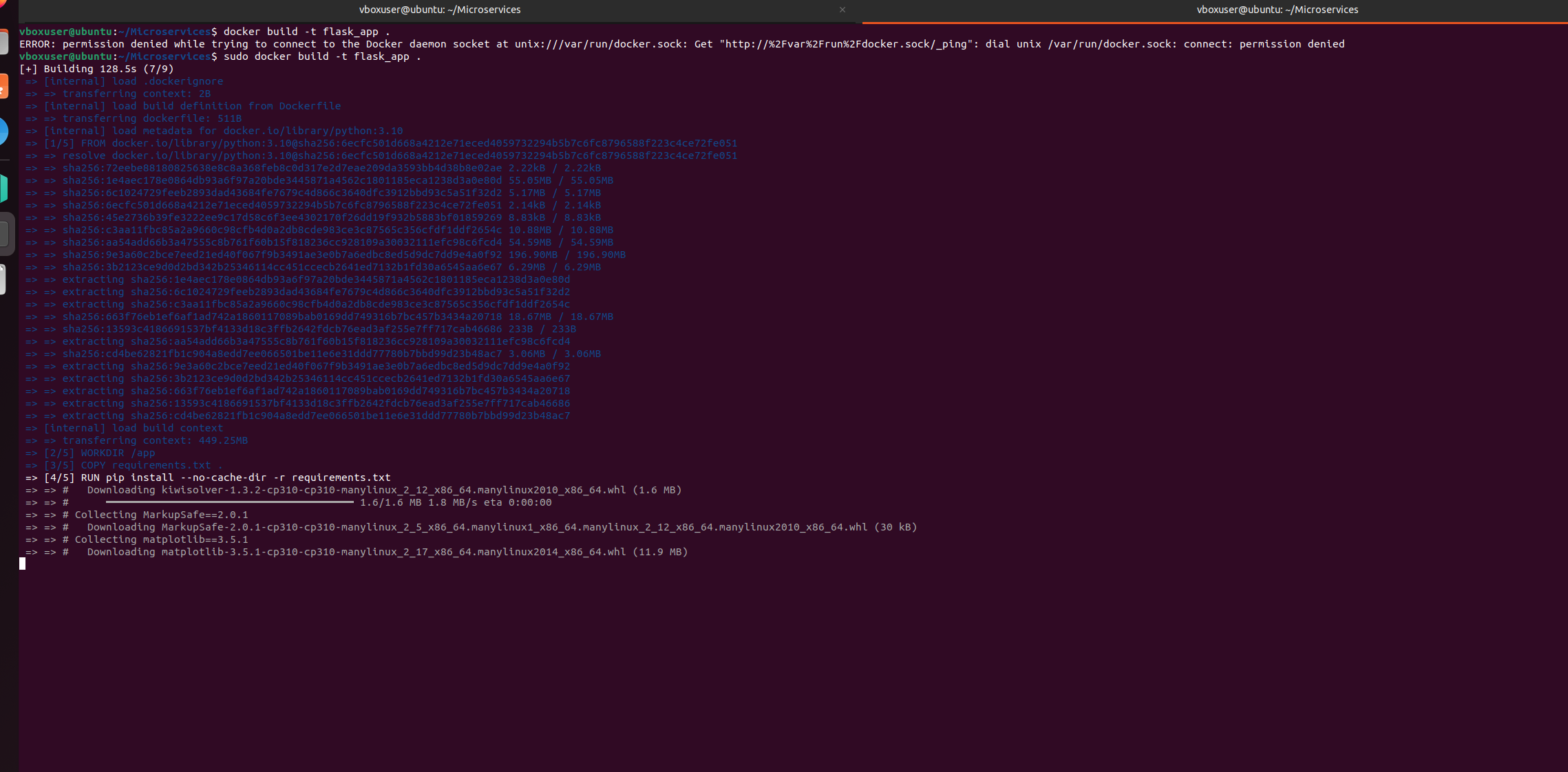
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

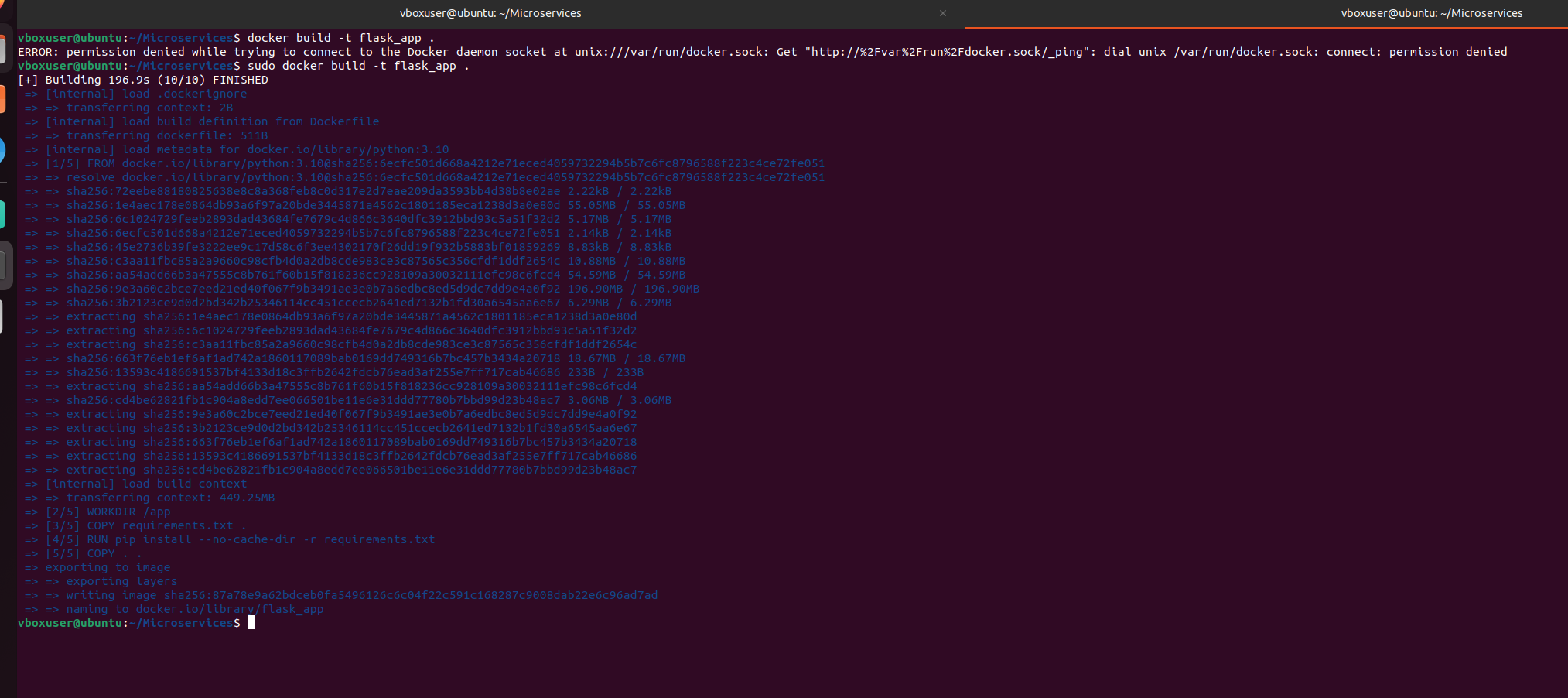
-

-

Build docker image using below command

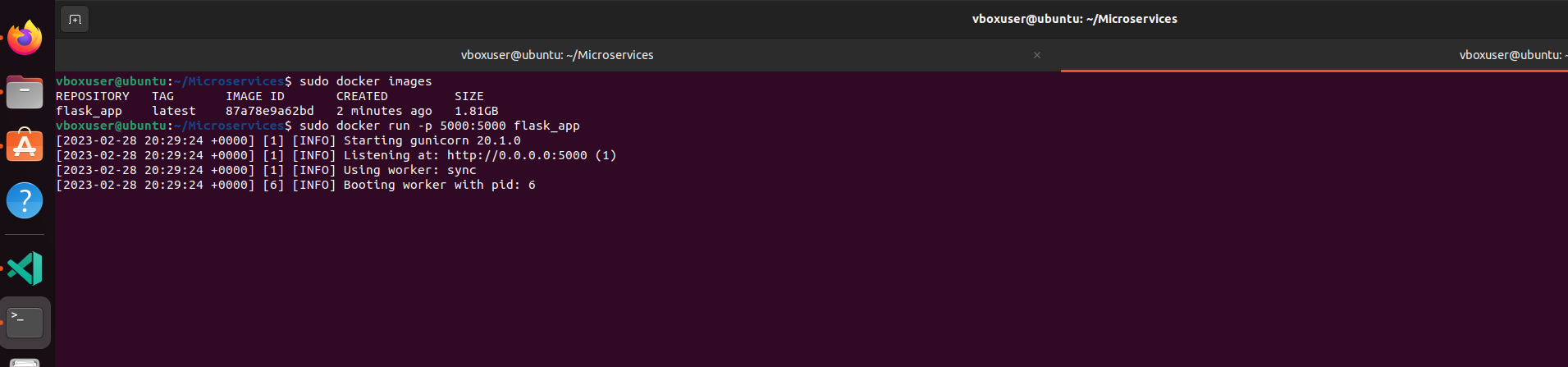
docker build -t flask\_app .

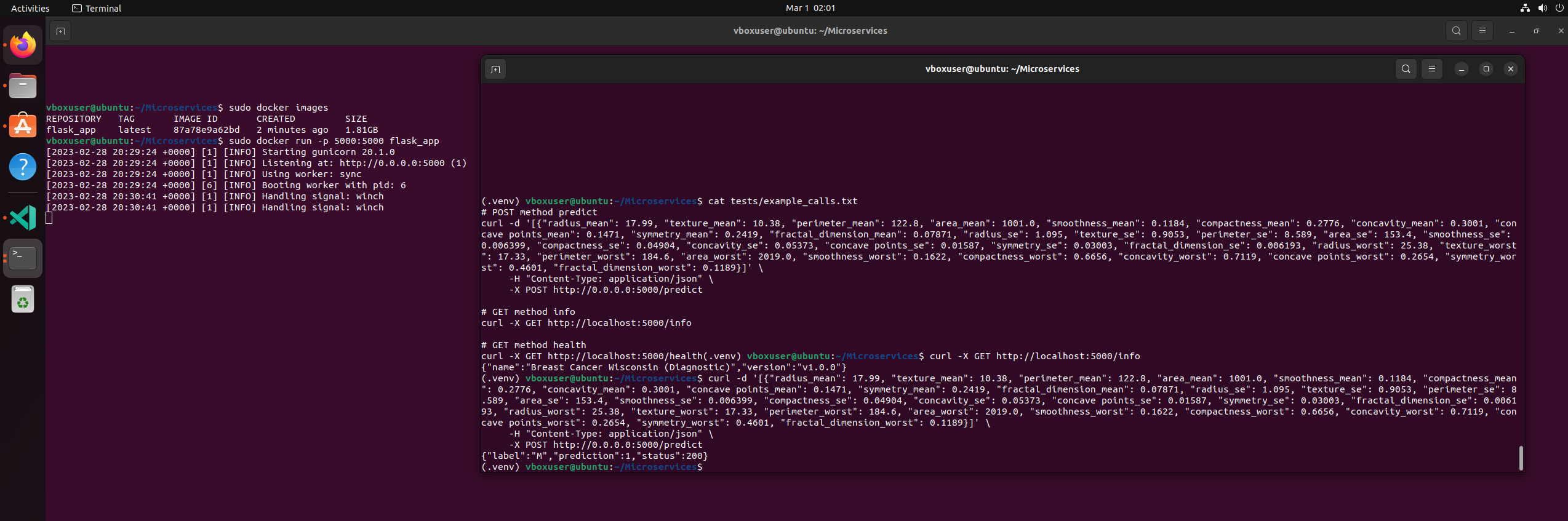
-



**11.Run the containerized application as a prediction service and test it locally by passing some example calls and get the prediction**

sudo docker run -p 5000:5000 flask\_app

Access the API

**References**

https://docs.docker.com/engine/install/ubuntu/

https://docs.docker.com/build/

https://flask.palletsprojects.com/en/2.2.x/