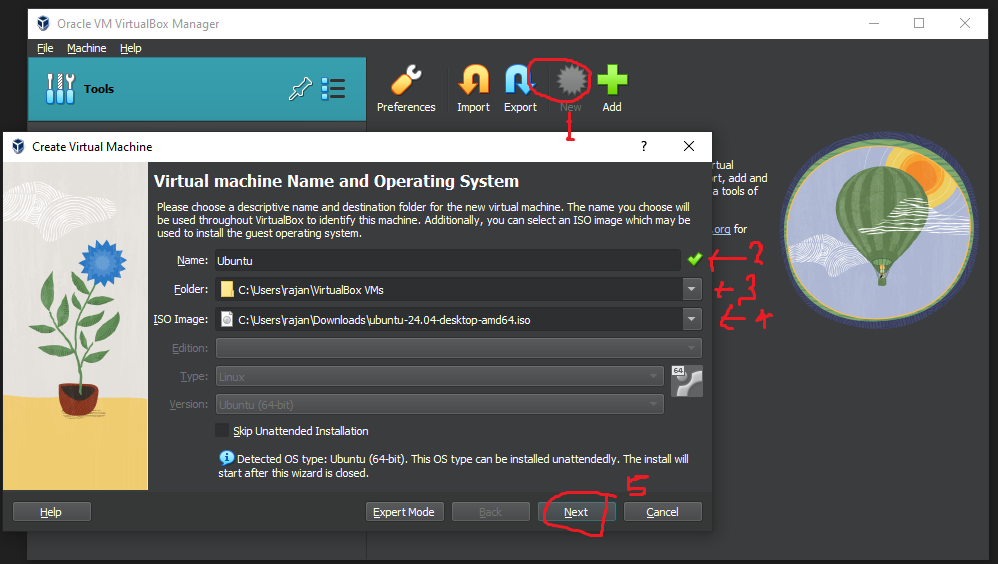
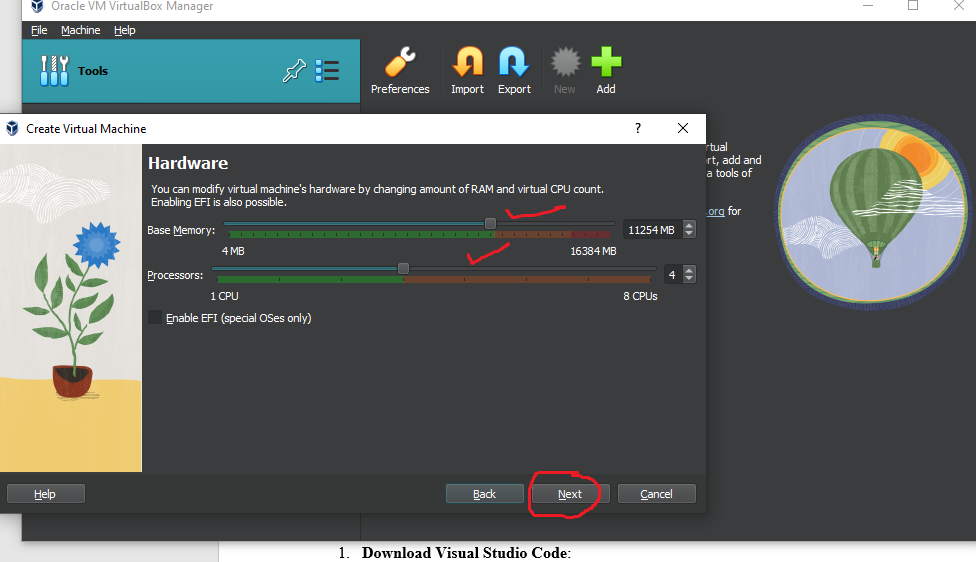
**Task 1: Host a Ubuntu Virtual Machine using Oracle VM VirtualBox**

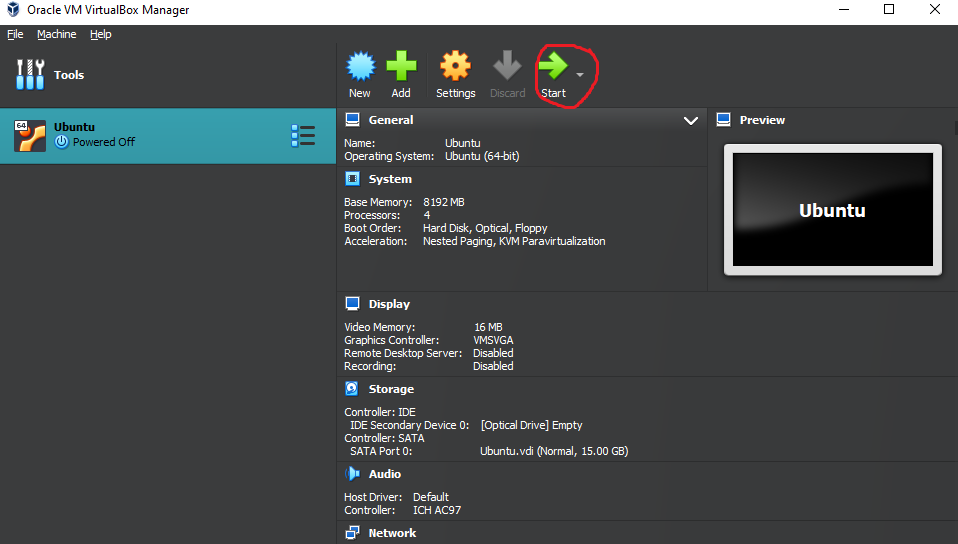
1. **Download Ubuntu ISO**:
   * Go to the [Ubuntu website](https://ubuntu.com/download/desktop) and download the Desktop version of Ubuntu as an ISO file.
2. **Install Oracle VM VirtualBox**:
   * Download and install [Oracle VM VirtualBox](https://www.virtualbox.org/).
3. **Create a New Virtual Machine**:
   * Open VirtualBox and click on "New" to create a new virtual machine.



* + Choose "Linux" as the type and "Ubuntu (64-bit)" as the version.
  + Allocate memory and create a virtual hard disk.



1. **Install Ubuntu**:
   * Select the newly created virtual machine and click "Start".

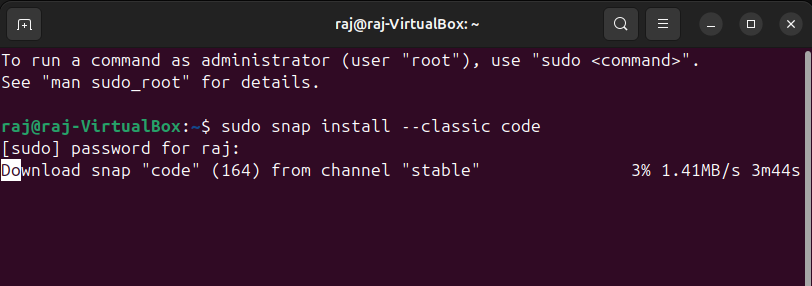


* + When prompted, select the Ubuntu ISO file you downloaded.
  + Follow the installation instructions to install Ubuntu on the virtual machine.

1. **Set Up Networking**:
   * Ensure the virtual machine is configured to use NAT or bridged networking so it can access the internet.

**Task 2: Set up Visual Studio Code on Ubuntu VM**

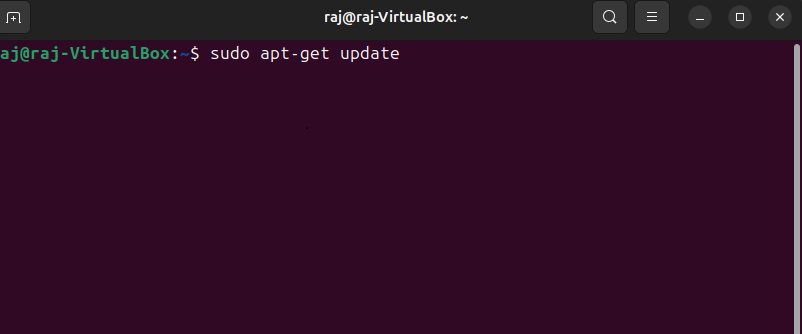
1. **Download Visual Studio Code**:
   * On Ubuntu, you can download Visual Studio Code from the official website or sudo snap install --classic code



1. **Install Visual Studio Code Extensions**
   * Open Visual Studio Code and install extensions for Python, Docker, etc., based on your requirements.

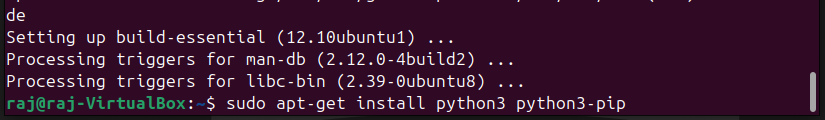
**Task 3: Set up Python**

1. **Install Python**:
   * Ubuntu typically comes with Python pre-installed. To install Python 3 and pip (Python package manager):



sudo apt update

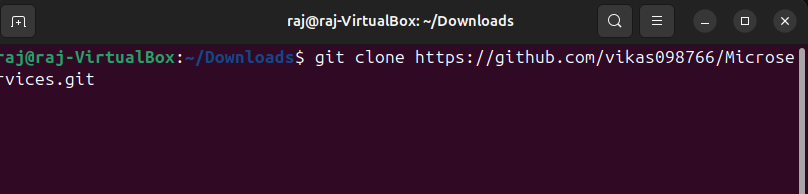
sudo apt install python3 python3-pip



**Task 4: Clone the GitHub Repository**

1. **Clone the Repository**:
   * Open a terminal in Ubuntu and clone the repository:

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



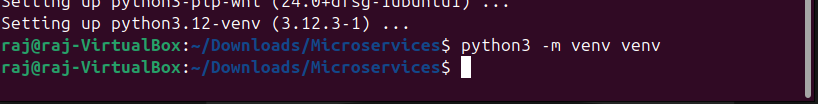
**Task 5: Create a Virtual Environment**

1. **Navigate to the Project Directory**:

cd Microservices

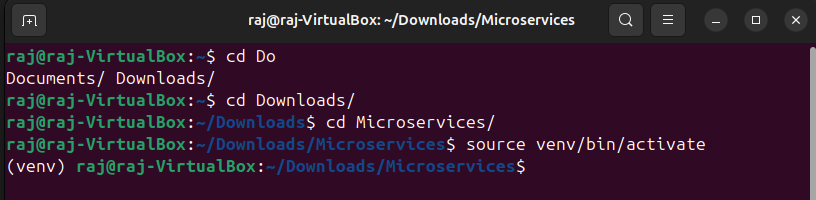
1. **Create a Virtual Environment**:

python3 -m venv venv



1. **Activate the Virtual Environment**:

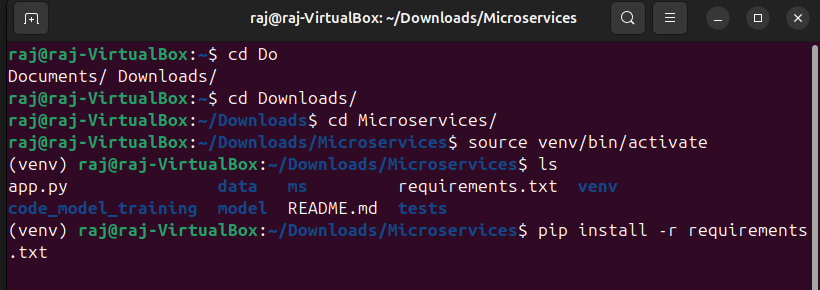
source venv/bin/activate



**Task 6: Install Dependencies from requirements.txt**

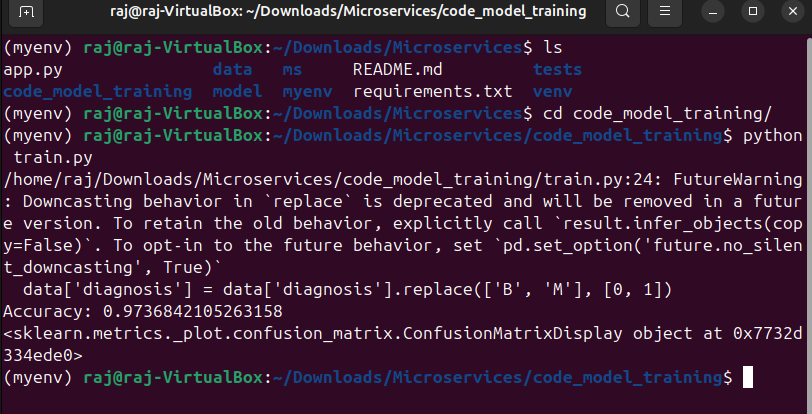
1. **Install Dependencies**:

pip install -r requirements.txt



**Task 7: Train and Save the Model**

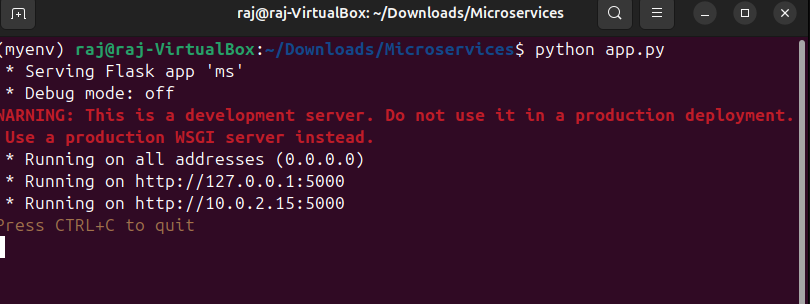
1. **Follow the instructions specific to the repository for training the model**.

****

**Task 8: Test the Flask Web Application**

1. **Run the Flask Application**:

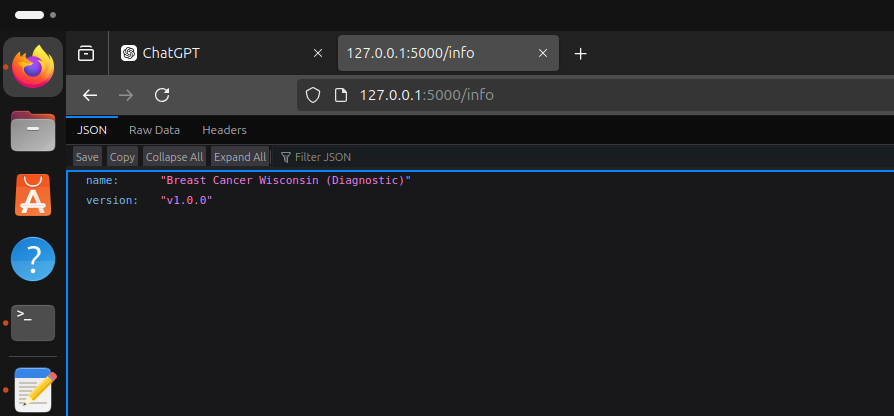
python app.py



1. **Access the Application**:
   * Open a web browser on your host machine and go to http://localhost:5000 to test the Flask app.

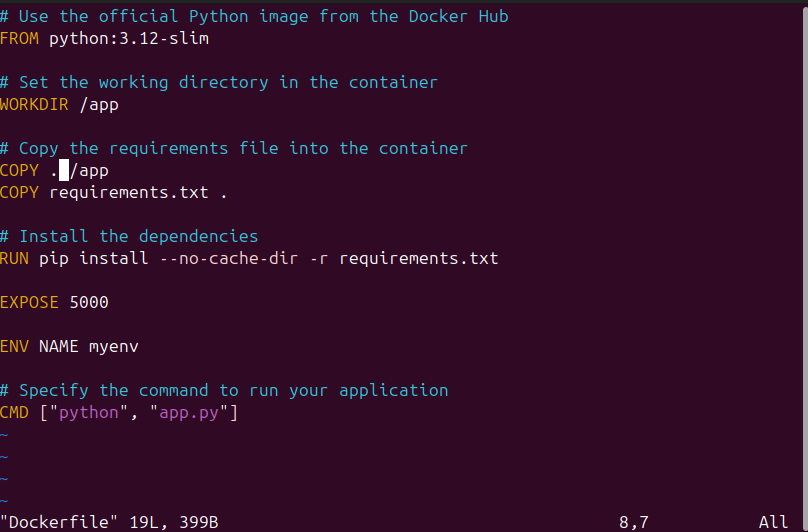
**Task 9: Test the Application and Make Predictions**

1. **Navigate to the tests folder** and run the example calls provided to test the predictions.

****

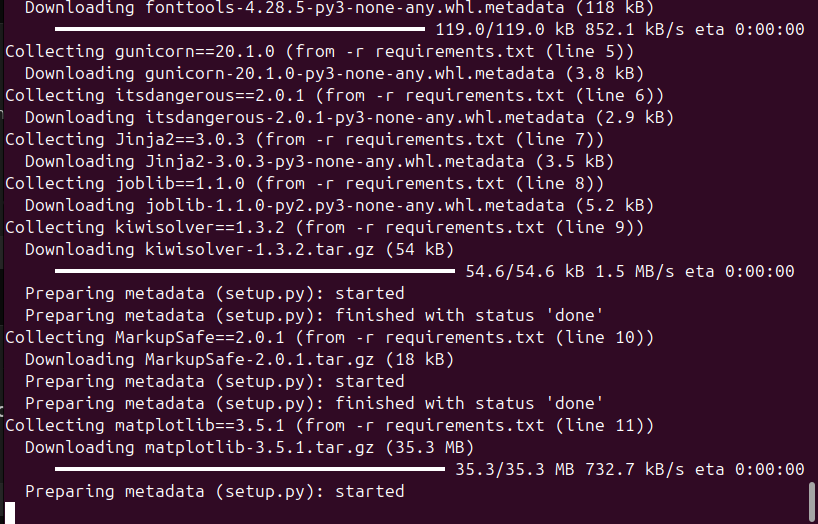
**Task 10: Create a Docker Image**

1. **Dockerfile**:
   * Create a Dockerfile in the project directory with instructions to build the Docker image.



1. **Build the Docker Image**:

docker build -t myapp .



**Task 11: Run the Containerized Application and Test Locally**

1. **Run the Docker Container**:

docker run -p 5000:5000 myapp

1. **Test the Application**:
   * Access http://localhost:5000 from your web browser to ensure the Flask app is running.
   * Use tools like curl or Postman to send example calls to the API running inside the Docker container.