Microservice Deployment on VirtualBox VMs (Parrot OS)

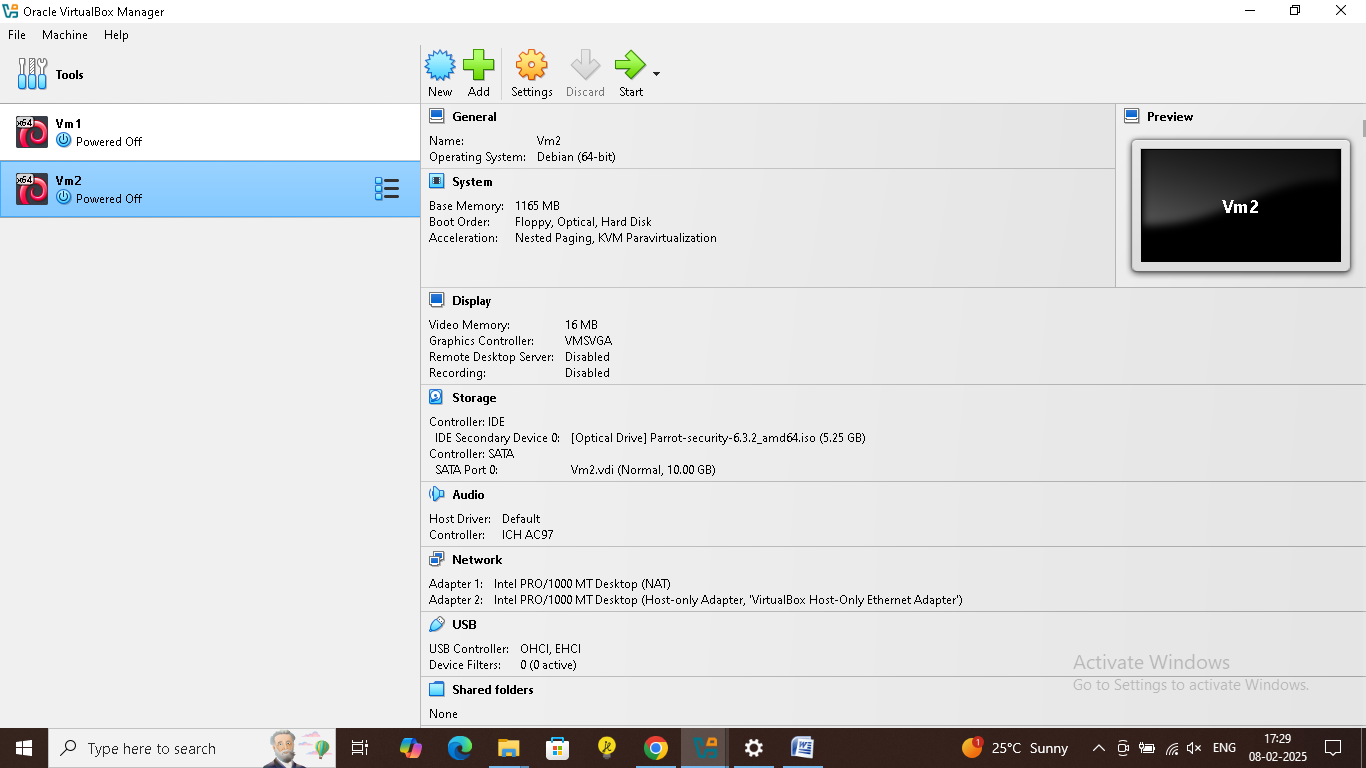
# 1. Objective

The objective of this project is to create and configure multiple Virtual Machines (VMs) using VirtualBox, establish a network between them, and deploy a simple microservice-based application across the connected VMs. The microservice will be a basic Flask API running on Parrot OS.

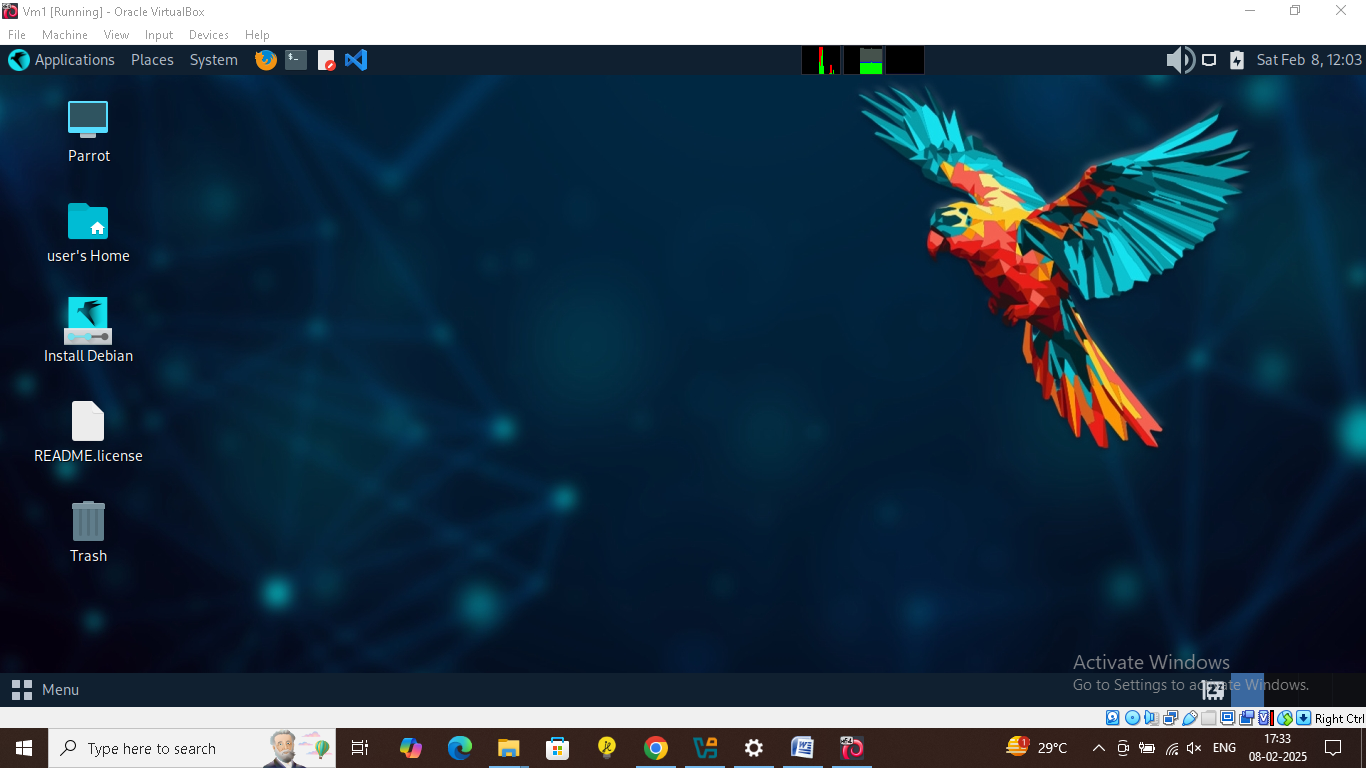
# 2. Step-by-Step Instructions for Implementation

## a. Installation of VirtualBox and Creation of VMs

1. Install VirtualBox:  
- Download and install VirtualBox from <https://www.virtualbox.org/>.

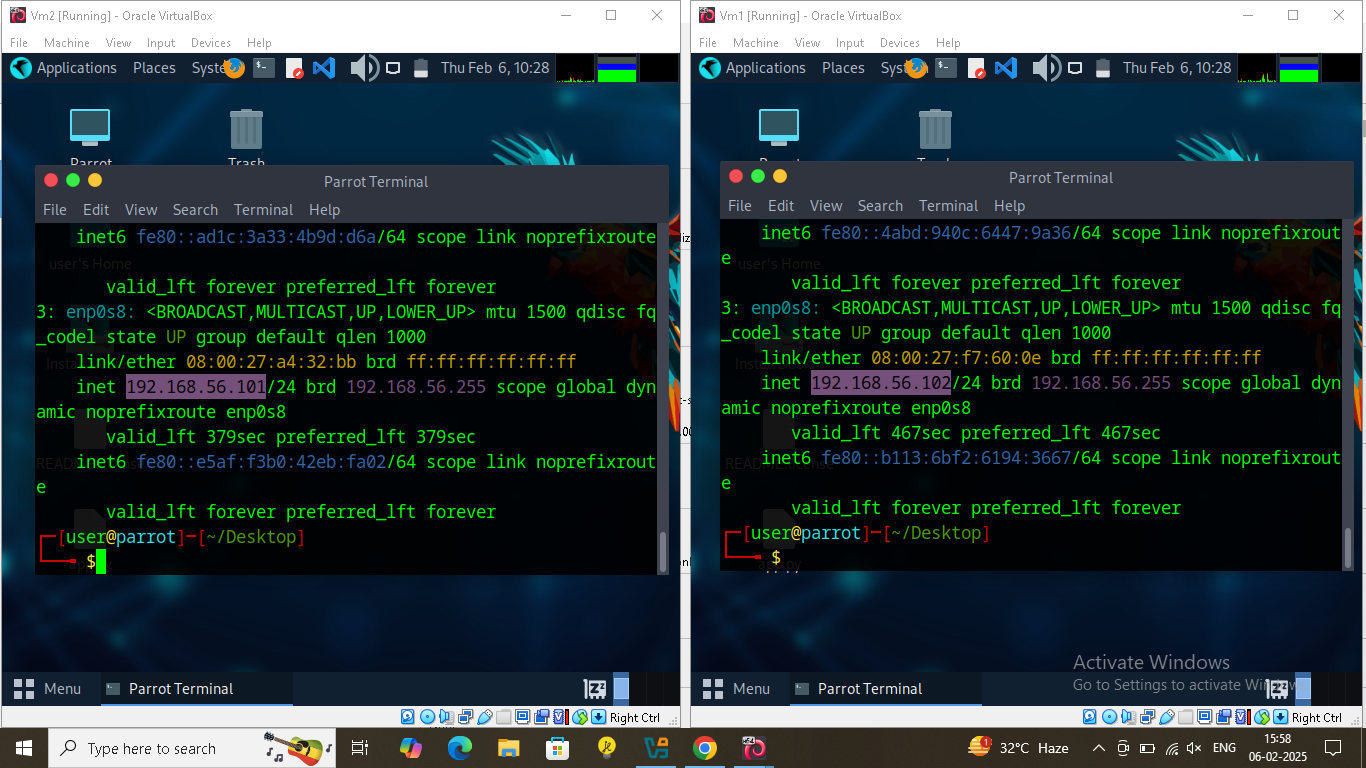


2. Download Parrot OS:  
- Download the Parrot OS ISO from <https://www.parrotsec.org/download/>.  
3. Create Virtual Machines:  
- Open VirtualBox and click New to create a VM.  
- Select Linux as the operating system and Ubuntu (64-bit) as the version (since Parrot OS is based on Debian).  
- Allocate memory and create a virtual hard disk.  
- Mount the Parrot OS ISO as the installation disk to install the operating system on the VM.  
- Repeat this step to create multiple VMs (e.g., VM1 for the Flask API server, VM2 for a recieving service).



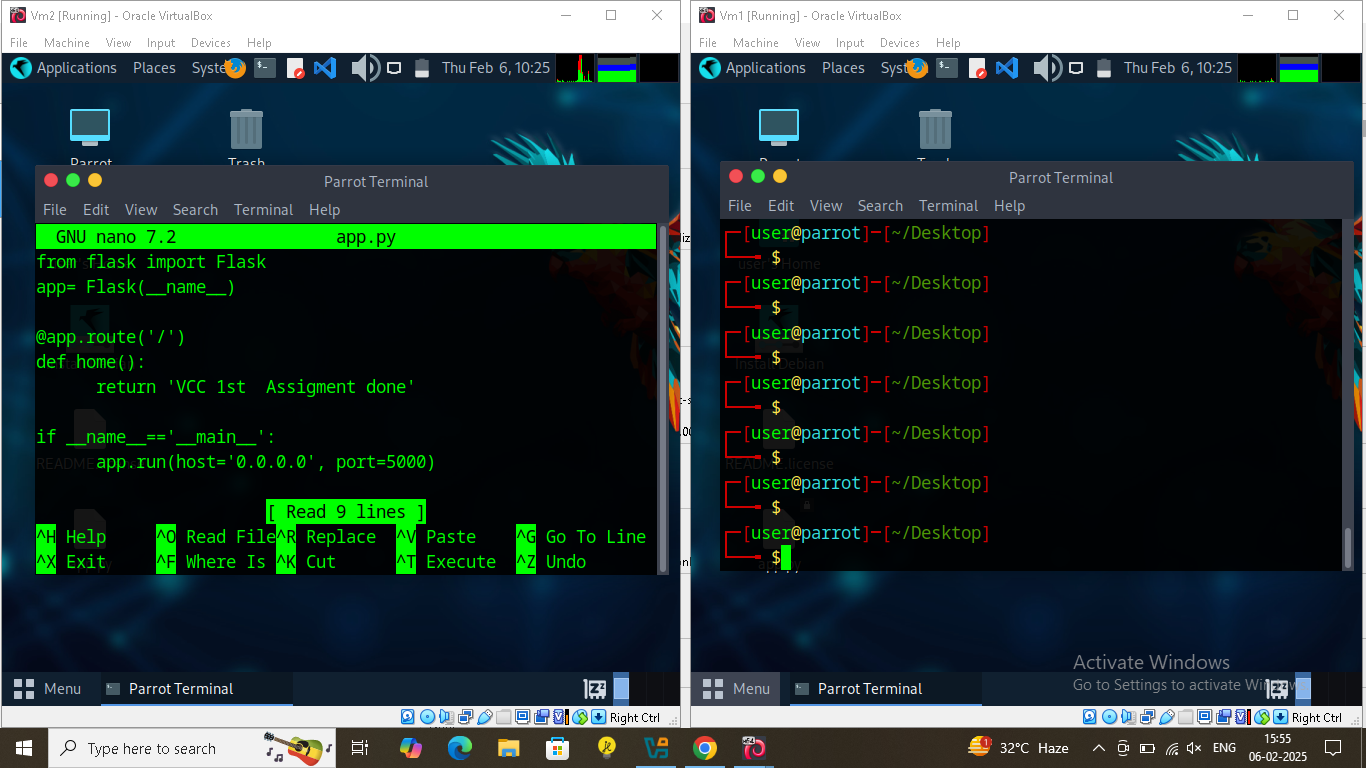
## b. Configuration of Network Settings to Connect the VMs

1. Network Configuration in VirtualBox:  
- In VirtualBox, for each VM, go to Settings > Network.  
- Set Attached to to Internal Network or Host-Only Adapter to ensure that the VMs can communicate with each other without external access.  
- Assign static IP addresses to the VMs:  
 - VM1: 192.168.56.101  
 - VM2: 192.168.56.102  
 - Verify the network connectivity between VMs using the ping command in the terminal of each VM.

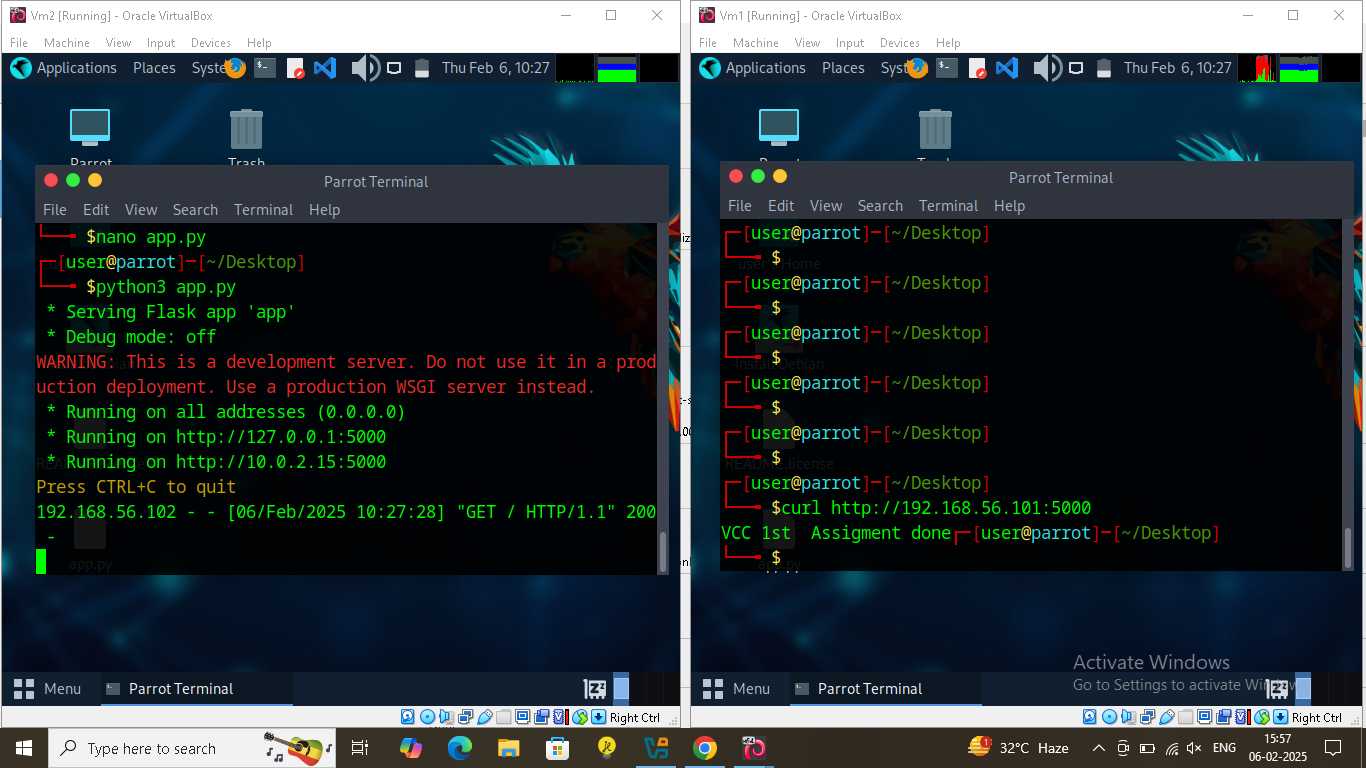


## c. Deployment of a Simple Microservice Application

1. Install Python and Flask on VM1:  
- On VM1 (the API server), open a terminal and install Python and pip:  
 sudo apt update  
 sudo apt install python3-pip  
- Install Flask using pip:  
 pip3 install flask  
**2. Create the Flask Application:**  
- Create a file called app.py and add the following code:  
 from flask import Flask  
 app = Flask(\_\_name\_\_)  
  
 @app.route('/')  
 def home():  
 return ‘VCC Assigment 1 Complted’  
  
 if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(host='0.0.0.0', port=6000)



3. Run the Flask Application:  
- In VM1’s terminal, run the app:  
 python3 app.py  
 The Flask app will now be running and accessible on http://192.168.56.101:6000.  
4. Access the Flask API from Other VMs (VM2):  
- On VM2 , open a terminal and use curl or a web browser to test the Flask API:  
 curl http://192.168.1.101:6000  
 The response should be: ‘VCC Assigment 1 Done



# 3. Architecture Design

Here’s a basic architecture diagram showing the communication between VMs:

VM1

192.168.56.101

(flask api server)

VM2

192.168.56.102  
  
(receiver)

# 4. Link to Source Code Repo

<https://github.com/Chirag0054/Microservice-Deployment-on-VirtualBox-VMs>