Virtualisation & Cloud Computing

Assignment # 01

- **By**
Mandar Bhalerao (M23CSA025)

GitHub URL :
https://github.com/m23csa025/VCC-Assignments/tree/VCC-Asgnmt-%2301

Google Drive URL :
https://drive.google.com/drive/folders/1PnbE63E62Yb32XuncrSQosxQcpS0dJqJ?usp=share_link

Question :

Objective:
Create and configure multiple Virtual Machines (VMs) using VirtualBox, establish a network between them, and deploy a microservice-based application across the connected VMs.

Deliverables:

1. Document Report:
   Step-by-Step Instructions for Implementation:
   Installation of VirtualBox and creation of multiple VMs.
   Configuration of network settings to connect the VMs.
   Deployment of a simple microservice application (e.g., a RESTful API or a Node.js-based service) across the VMs.

2. Architecture Design:
   Diagram showing the connection of VMs and their roles in hosting the microservice application.

3. Link to Source Code Repo:
   GitHub or similar repository containing the source code of the microservice application and deployment configuration.

4. **Link to Recorded Video Demo:**
   A video demonstrating the creation, configuration, and hosting process with a voice-over explanation.

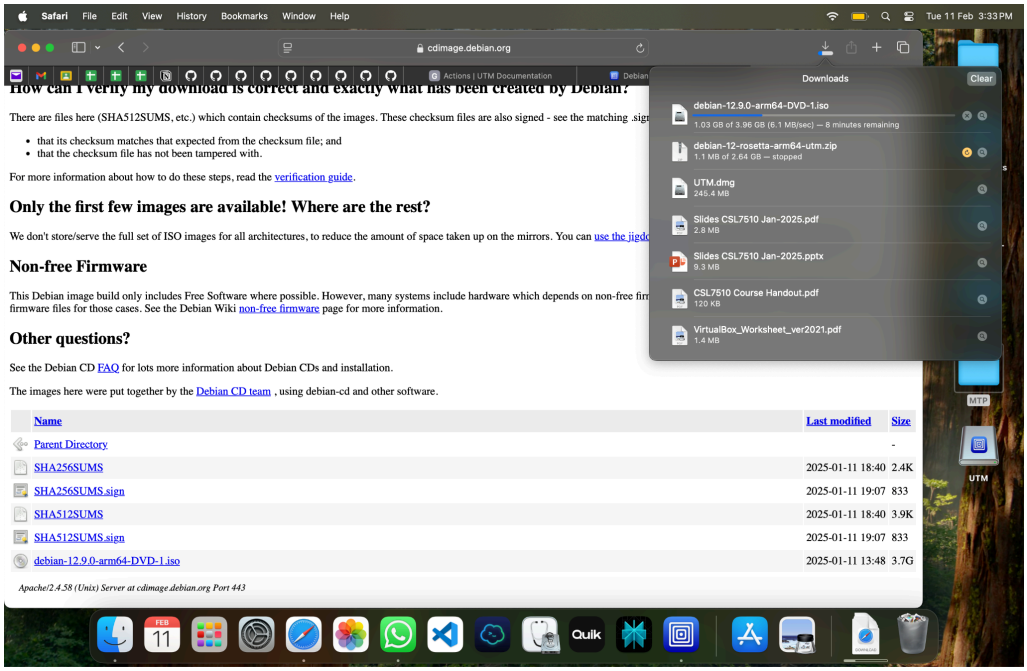
Solution :

1. My laptop configuration : Macbook Air (M2) 8 GB 512 GB
2. Virtualbox is not fully compatible with macbook.
3. Search for alternative which is available for free, in app store or is reliable.
4. Possible options : 1. **Universal Turing Machine (UTM)**
   - App Store (Paid version)
   - Website (free version) https://mac.getutm.app/
   - Reliable

   2. **Parallels Desktop**
   - App store (free)
5. Chose UTM based on reviews and star ratings.

6. Downloaded UTM and Installed in Macbook.
7. **Choice of VM Environment**
   a. ARM version
   b. Choice between Ubuntu/ Linux/Windows/MacOS and various other versions available on UTM
   c. I chose Debian 12 which is a version of Linux for ease of operation on macbook

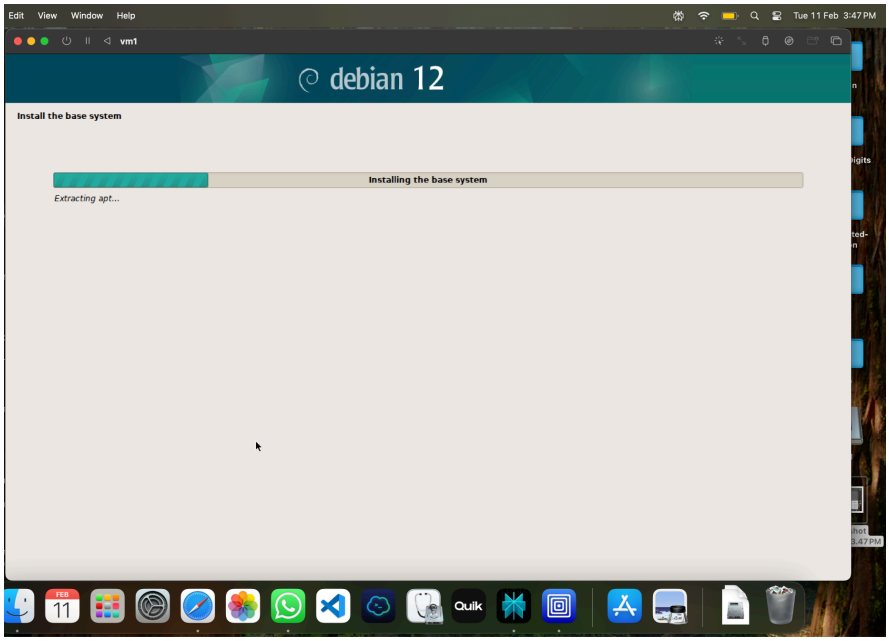
8. Download Debian 12 from website http://cdimage.debian.org/



9. Open UTM. Install and configure **vm1**.

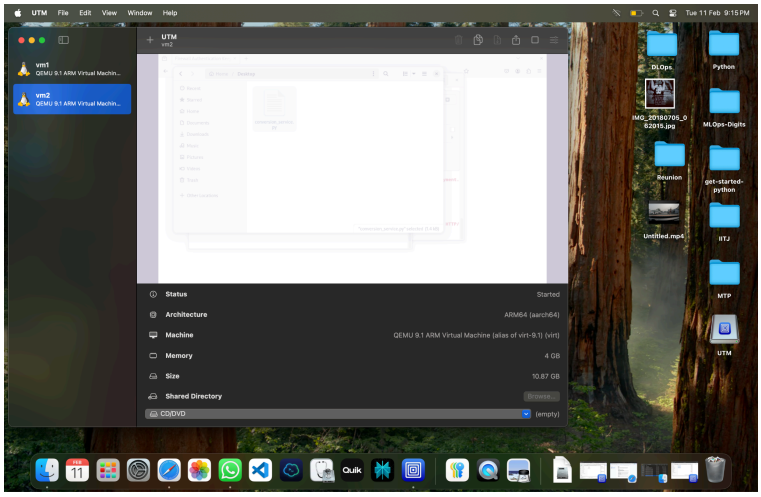
RAM - 4 GB
HDD - 20 GB
Debian Desktop Environment
GNOME

User : mandar



10. Install and configure **vm2.**

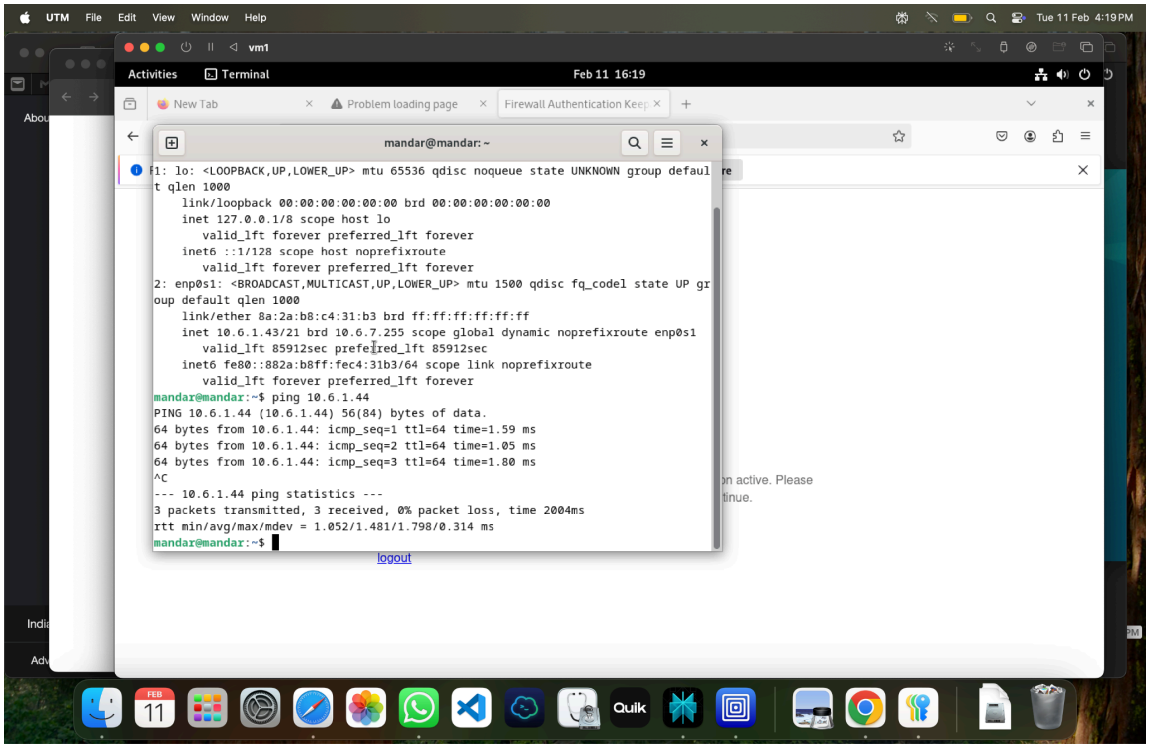
RAM - 4 GB
HDD - 20 GB
Debian Desktop Environment
GNOME
User : mandar1

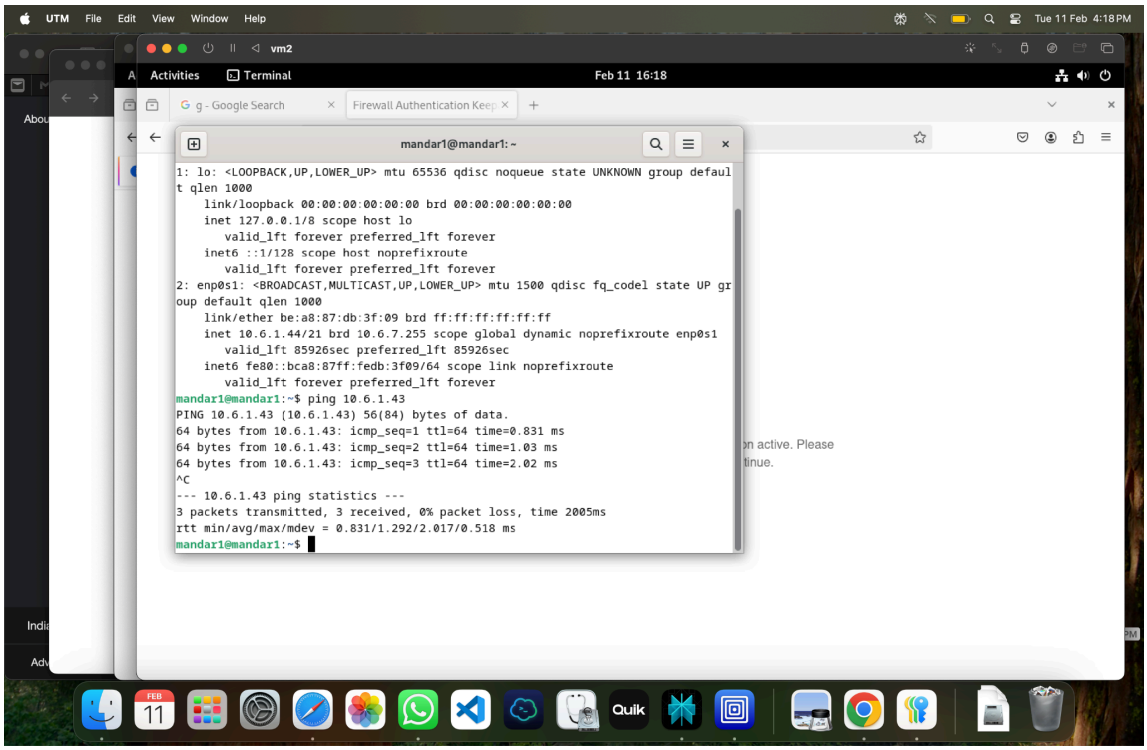


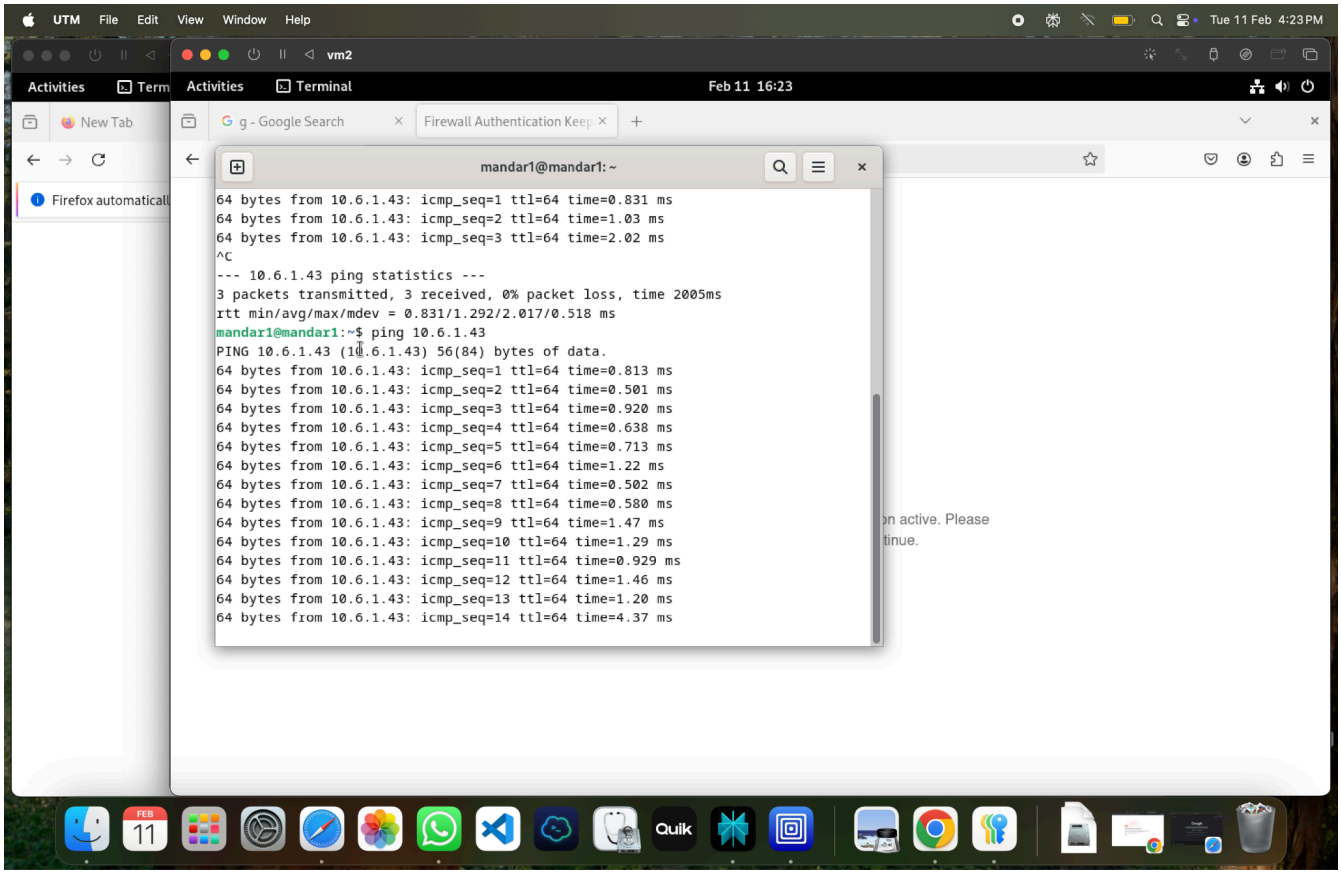
11. Configure networking in vm1 & vm2.
    a. Disable wifi
    b. Connect to wired LAN/Network
    c. Configure / Auto Configure IP address.
    d. IP for vm1 : 10.6.1.43
    e. IP for vm2 : 10.6.1.44

12. Run ping micro service.

    a. On vm1 : ping 10.6.1.44
    b. On vm2 : ping 10.6.1.43
    c. Successful ping on both VMs.
    d. Screen shots enclosed.


UTM File Edit View Window Help
Tue 11 Feb 4:19 PM
vm1
Activities
Terminal
Feb 11 16:19
New Tab
Problem loading page
Firewall Authentication Keep
mandar@mandar: ~
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
link/ether 8a:2a:b8:c4:31:b3 brd ff:ff:ff:ff:ff:ff
inet 10.6.1.43/21 brd 10.6.7.255 scope global dynamic noprefixroute enp0s1
valid_lft 85912sec preferred_lft 85912sec
inet6 fe80::882a:b8ff:fec4:31b3/64 scope link noprefixroute
valid_lft forever preferred_lft forever
mandar@mandar:~$ ping 10.6.1.44
PING 10.6.1.44 (10.6.1.44) 56(84) bytes of data.
64 bytes from 10.6.1.44: icmp_seq=1 ttl=64 time=1.59 ms
64 bytes from 10.6.1.44: icmp_seq=2 ttl=64 time=1.05 ms
64 bytes from 10.6.1.44: icmp_seq=3 ttl=64 time=1.80 ms
^C
--- 10.6.1.44 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.052/1.481/1.798/0.314 ms
mandar@mandar:~$
logout


UTM File Edit View Window Help
Tue 11 Feb 4:18 PM
vm2
Activities
Terminal
Feb 11 16:18
g - Google Search
Firewall Authentication Keep
mandar1@mandar1: ~
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
link/ether be:a8:87:db:3f:09 brd ff:ff:ff:ff:ff:ff
inet 10.6.1.44/21 brd 10.6.7.255 scope global dynamic noprefixroute enp0s1
valid_lft 85926sec preferred_lft 85926sec
inet6 fe80::bca8:87ff:fedb:3f09/64 scope link noprefixroute
valid_lft forever preferred_lft forever
mandar1@mandar1:~$ ping 10.6.1.43
PING 10.6.1.43 (10.6.1.43) 56(84) bytes of data.
64 bytes from 10.6.1.43: icmp_seq=1 ttl=64 time=0.831 ms
64 bytes from 10.6.1.43: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 10.6.1.43: icmp_seq=3 ttl=64 time=2.02 ms
^C
--- 10.6.1.43 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.831/1.292/2.017/0.518 ms
mandar1@mandar1:~$


UTM File Edit View Window Help
Tue 11 Feb 4:23 PM
vm2
Activities Terminal
Feb 11 16:23
mandar1@mandar1: ~
64 bytes from 10.6.1.43: icmp_seq=1 ttl=64 time=0.831 ms
64 bytes from 10.6.1.43: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 10.6.1.43: icmp_seq=3 ttl=64 time=2.02 ms
^C
--- 10.6.1.43 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.831/1.292/2.017/0.518 ms
mandar1@mandar1:~$ ping 10.6.1.43
PING 10.6.1.43 (10.6.1.43) 56(84) bytes of data.
64 bytes from 10.6.1.43: icmp_seq=1 ttl=64 time=0.813 ms
64 bytes from 10.6.1.43: icmp_seq=2 ttl=64 time=0.501 ms
64 bytes from 10.6.1.43: icmp_seq=3 ttl=64 time=0.920 ms
64 bytes from 10.6.1.43: icmp_seq=4 ttl=64 time=0.638 ms
64 bytes from 10.6.1.43: icmp_seq=5 ttl=64 time=0.713 ms
64 bytes from 10.6.1.43: icmp_seq=6 ttl=64 time=1.22 ms
64 bytes from 10.6.1.43: icmp_seq=7 ttl=64 time=0.502 ms
64 bytes from 10.6.1.43: icmp_seq=8 ttl=64 time=0.580 ms
64 bytes from 10.6.1.43: icmp_seq=9 ttl=64 time=1.47 ms
64 bytes from 10.6.1.43: icmp_seq=10 ttl=64 time=1.29 ms
64 bytes from 10.6.1.43: icmp_seq=11 ttl=64 time=0.929 ms
64 bytes from 10.6.1.43: icmp_seq=12 ttl=64 time=1.46 ms
64 bytes from 10.6.1.43: icmp_seq=13 ttl=64 time=1.20 ms
64 bytes from 10.6.1.43: icmp_seq=14 ttl=64 time=4.37 ms
New Tab
g - Google Search
Firewall Authentication Keep
Firefox automaticall
n active. Please
tinue.
Quik

13. **Currency Conversion Service**.

This service consists of two microservices:

a. Exchange Service (VM1) – Provides exchange rates for different currencies and allows updating rates.

Functionality

The Exchange Service manages currency exchange rates. It has two key Endpoints:

- **GET** `/rate/<currency>`: Retrieves the exchange rate for a given currency.
- **POST** `/rate`: Updates the exchange rate for a specified currency.

Maintains an in-memory dictionary for exchange rates (`exchange_rates`).

Handles both retrieval and update of exchange rates.

Returns appropriate HTTP status codes and error messages.

b. Conversion Service (VM2) – Uses the Exchange Service to convert an amount from one currency to another.

Functionality
The Conversion Service converts a given amount from one currency to another using the Exchange Service for rate information. It has one main endpoint:

- **POST** `/convert`:
  Converts an amount from one currency to another.

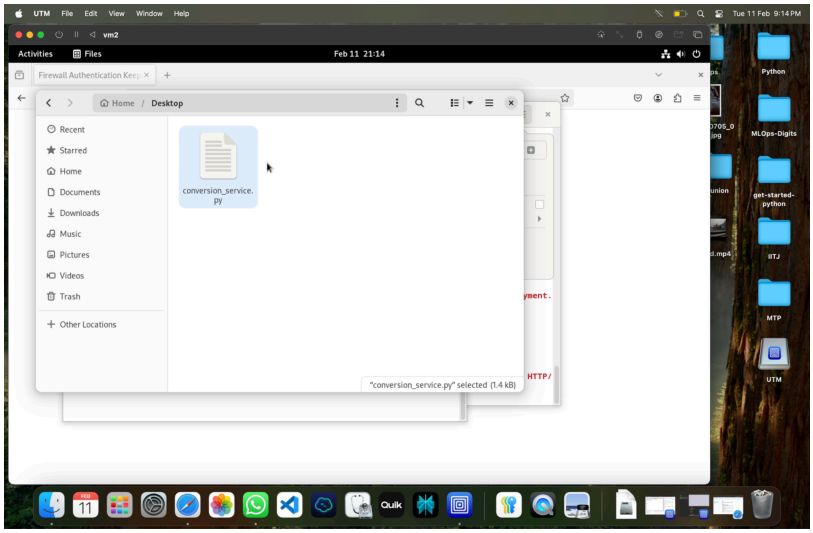
Sends GET requests to the Exchange Service for both source and target Currencies.

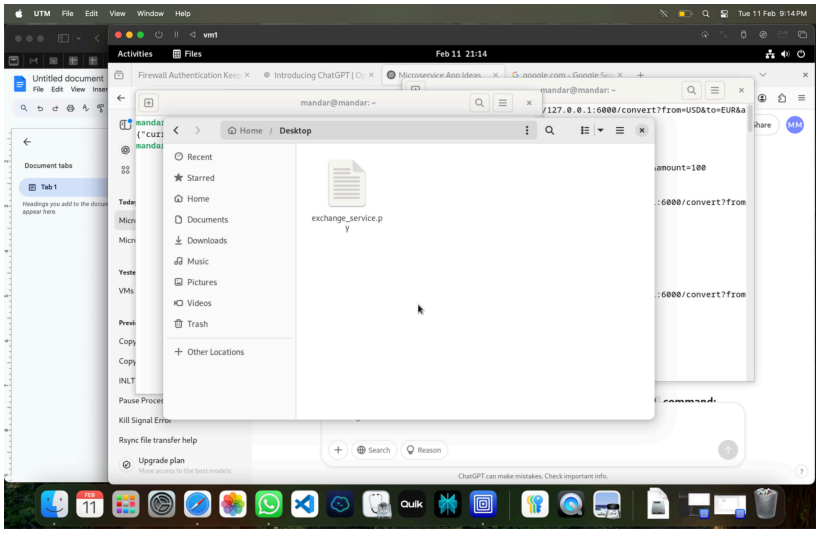
Calculates the converted amount based on the retrieved rates.

Handles errors such as missing parameters or invalid currencies.

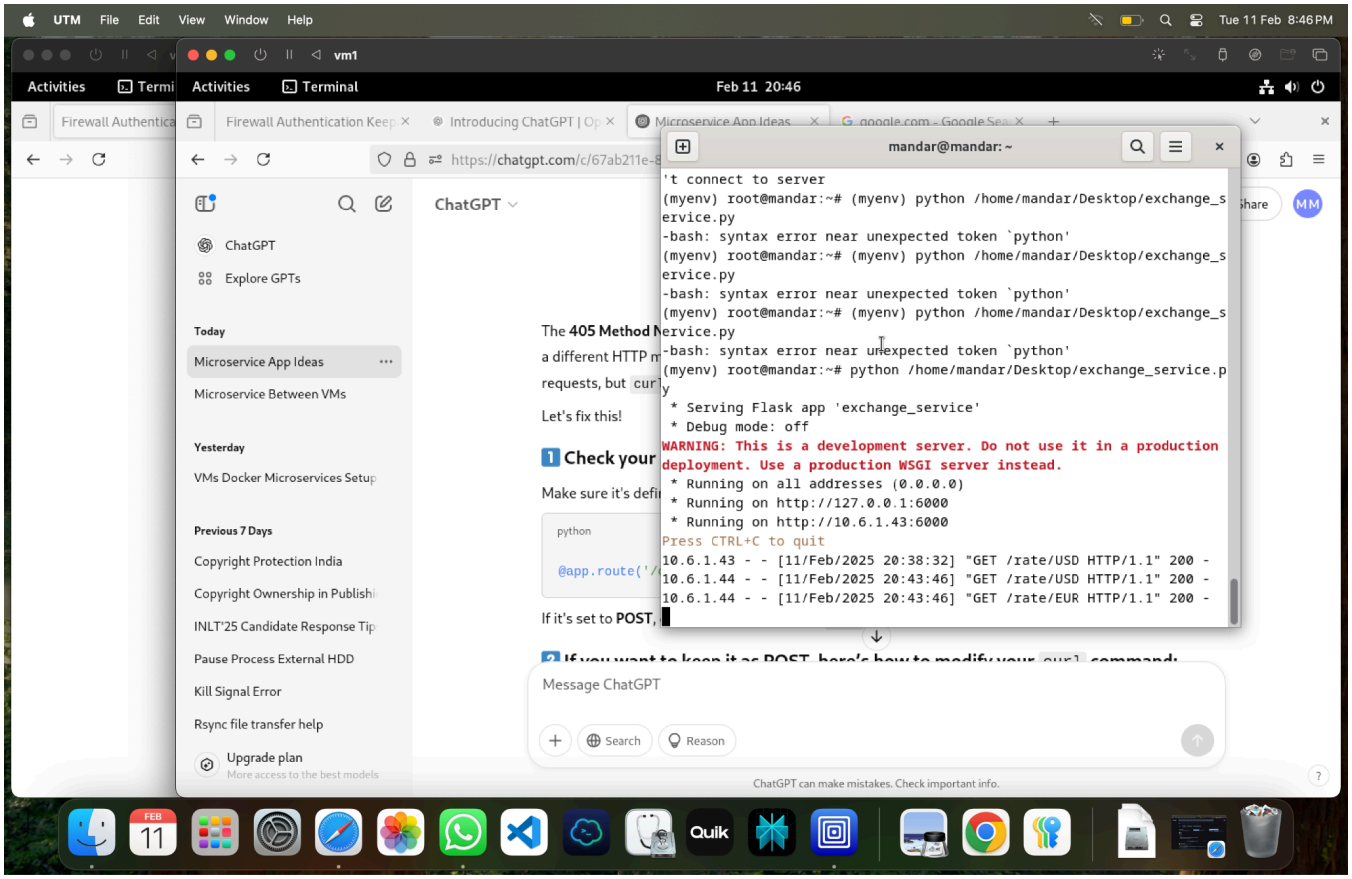
c. Install relevant python and Flask.

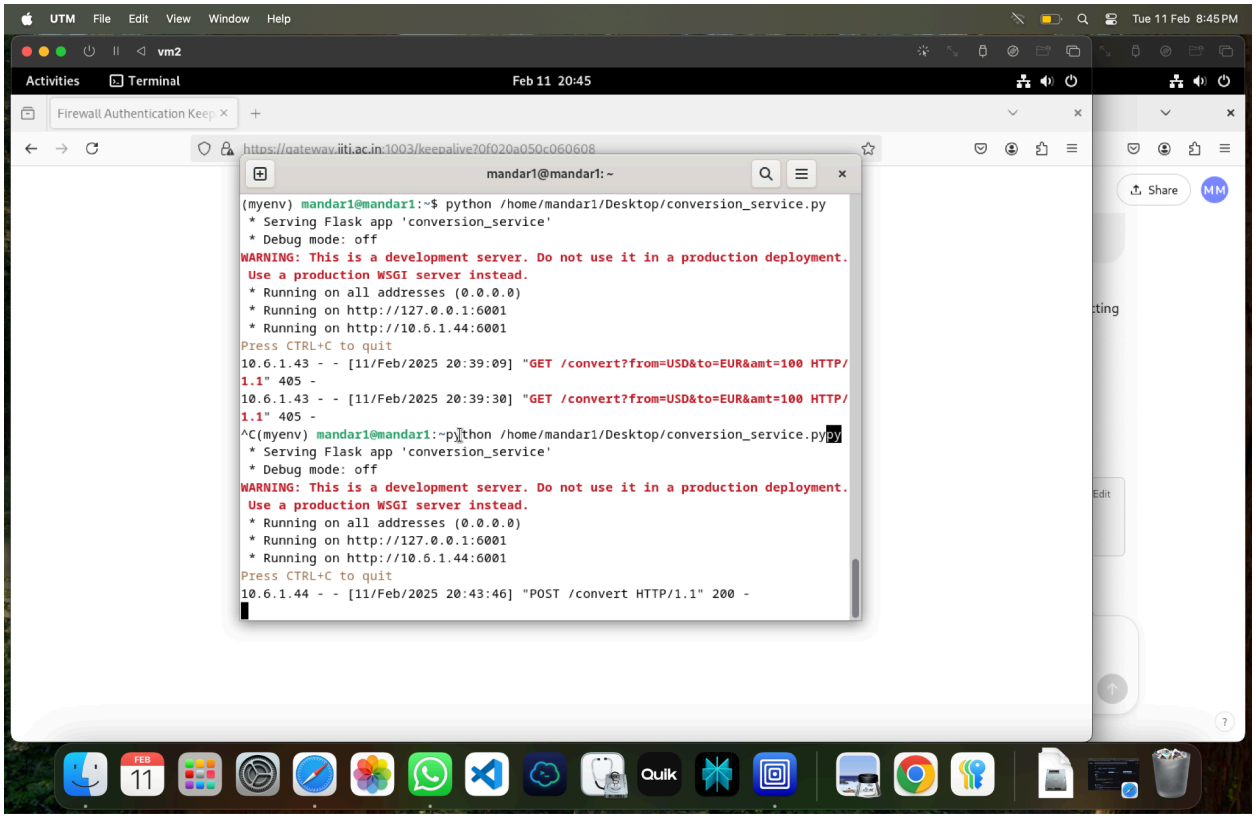
d. Save exchange.py (vm1) and conversion.py (vm2) on desktop of respective VMs.





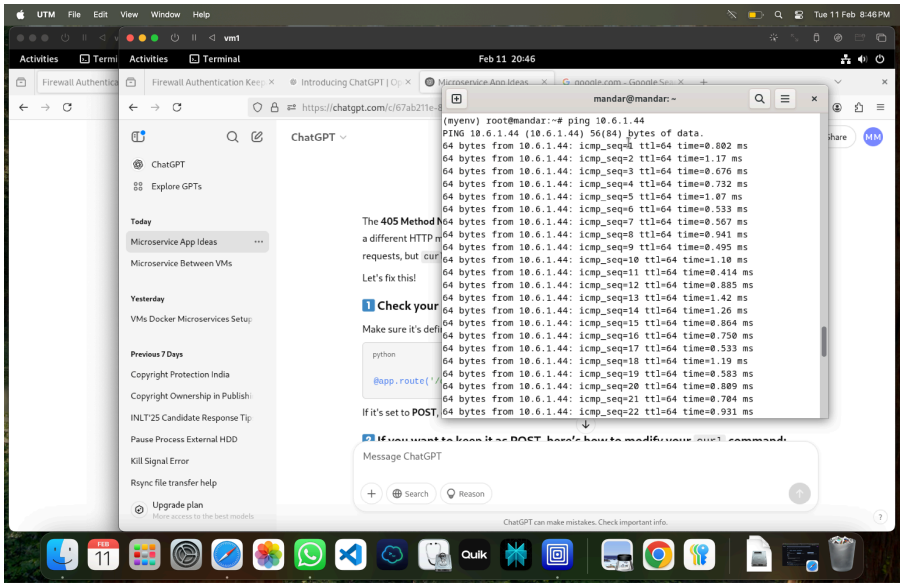
e. Activate and run both microservices.

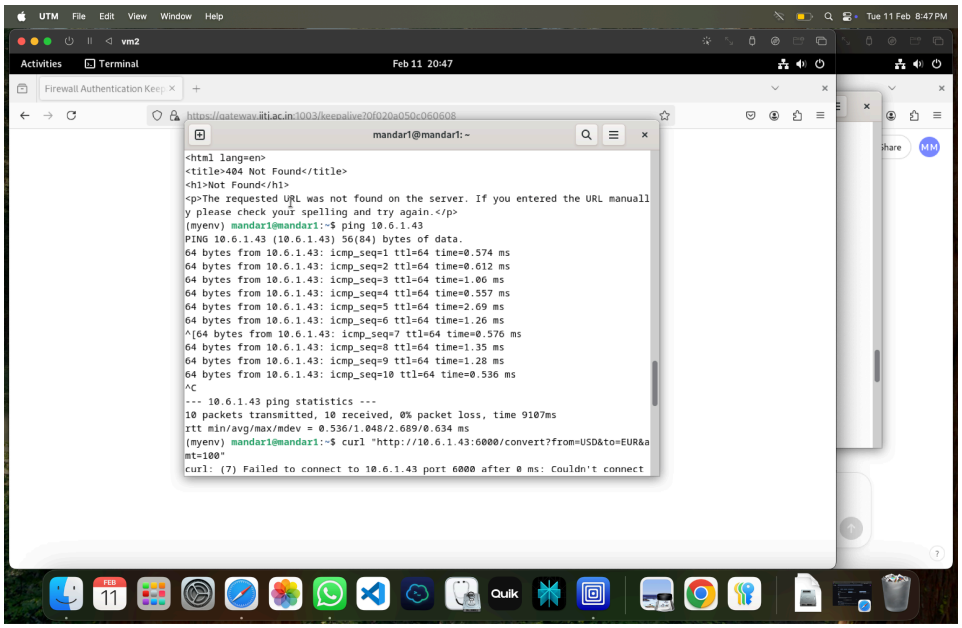




f. Testing and Validation :

Check for connection between two VMs using ping again .



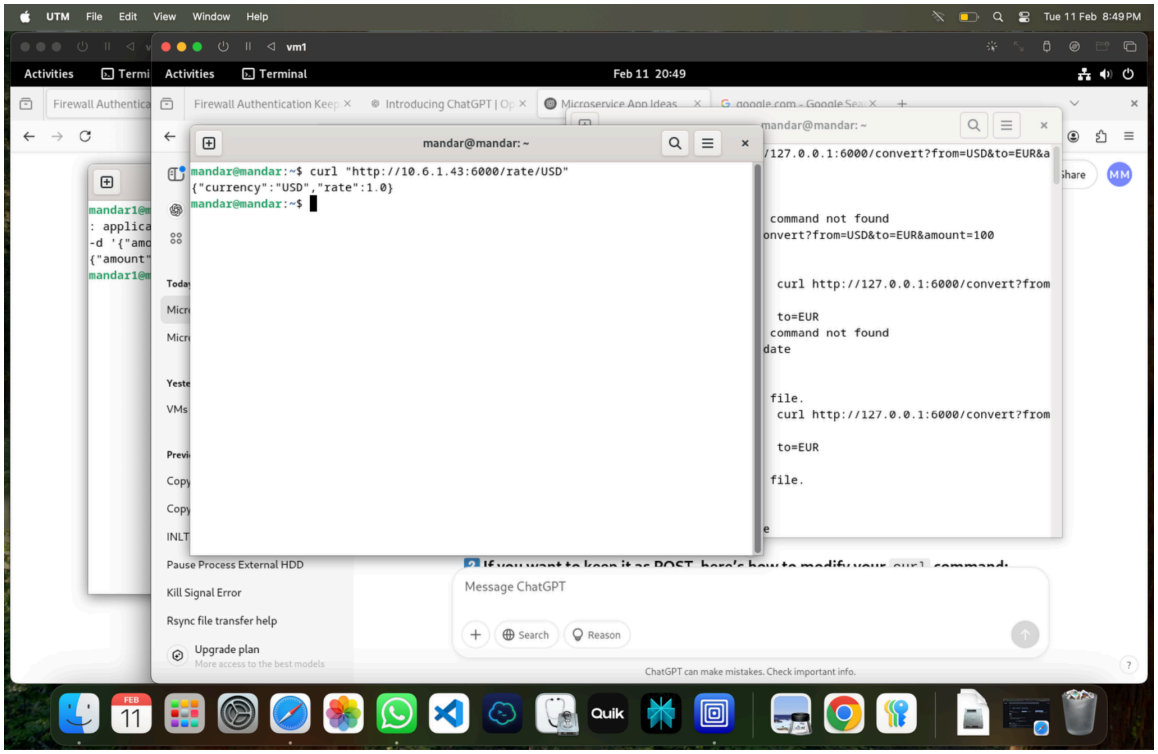


g. Check functionality of exchange service on vm1.

curl "http://10.6.1.43:6000/rate/USD"

Expected Response :

{"currency":"USD","rate":1.0}

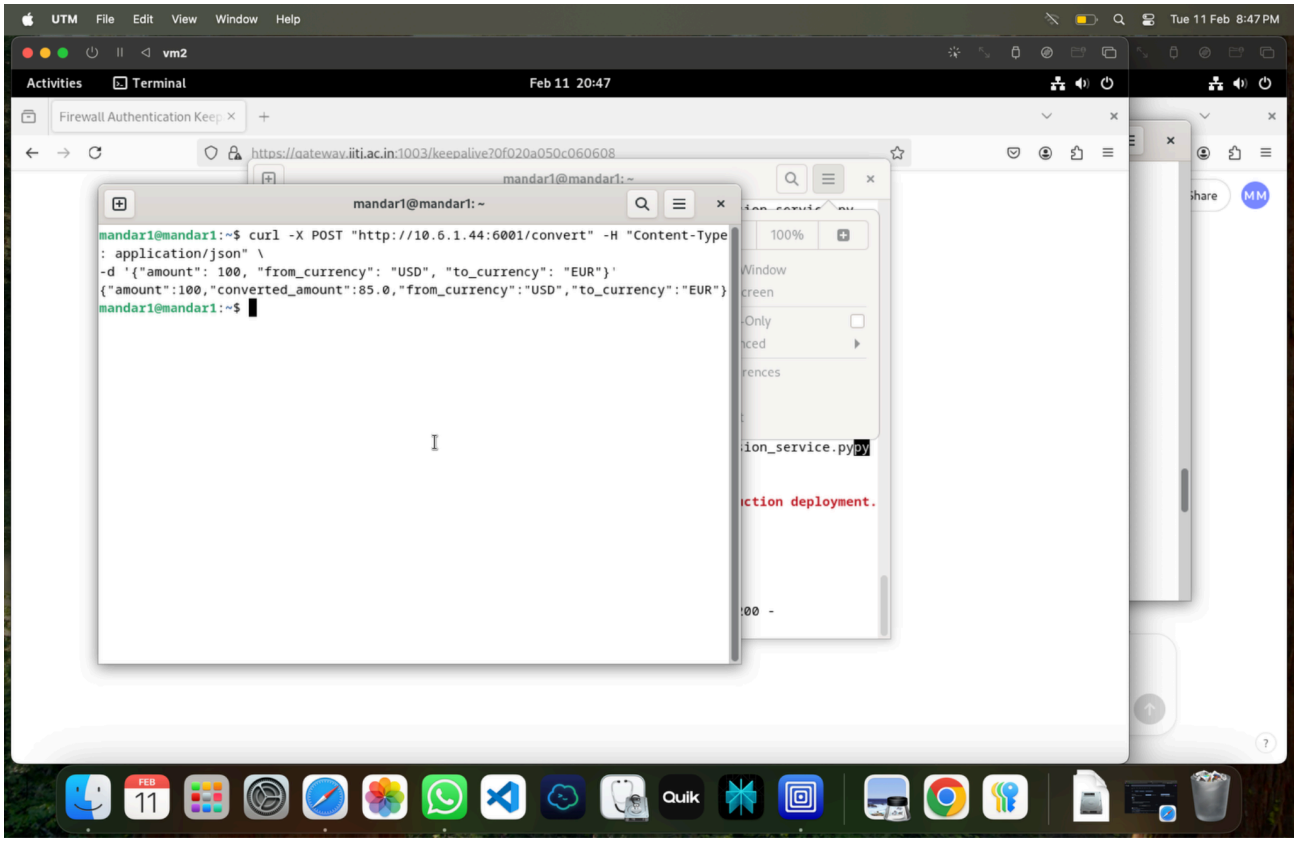


h. Check functionality of conversion service on vm2.

curl -X POST "http://10.6.1.44:6001/convert" -H "Content-Type: application/json" \
-d '{"amount": 100, "from_currency": "USD", "to_currency": "EUR"}'

Expected Response :

{
  "from_currency": "USD",
  "to_currency": "EUR",
  "amount": 100,
  "converted_amount": 85.0
}



Conclusion :

Currency conversion micro services successfully implemented on vm1 & vm2.