**Weather Microservice Application**

# Indian Institute of Technology, Jodhpur

## 

## ****Name:**** Swatantra Yadav

## ****Roll Number:** G24AI1065**

## ****Gmail:** g24ai1065@iitj.ac.in**

## ****Subject:** Virtualization and Cloud Computing**

## ****Assignment:** Assignment 1**

## ****Professor:** Sumit Kalra**

# Overview

This project involves creating and configuring multiple Virtual Machines (VMs) using VirtualBox, establishing a network between them, and deploying a weather microservice application across the connected VMs. The microservice provides real-time weather temperature for a given city using the WeatherAPI.com API.

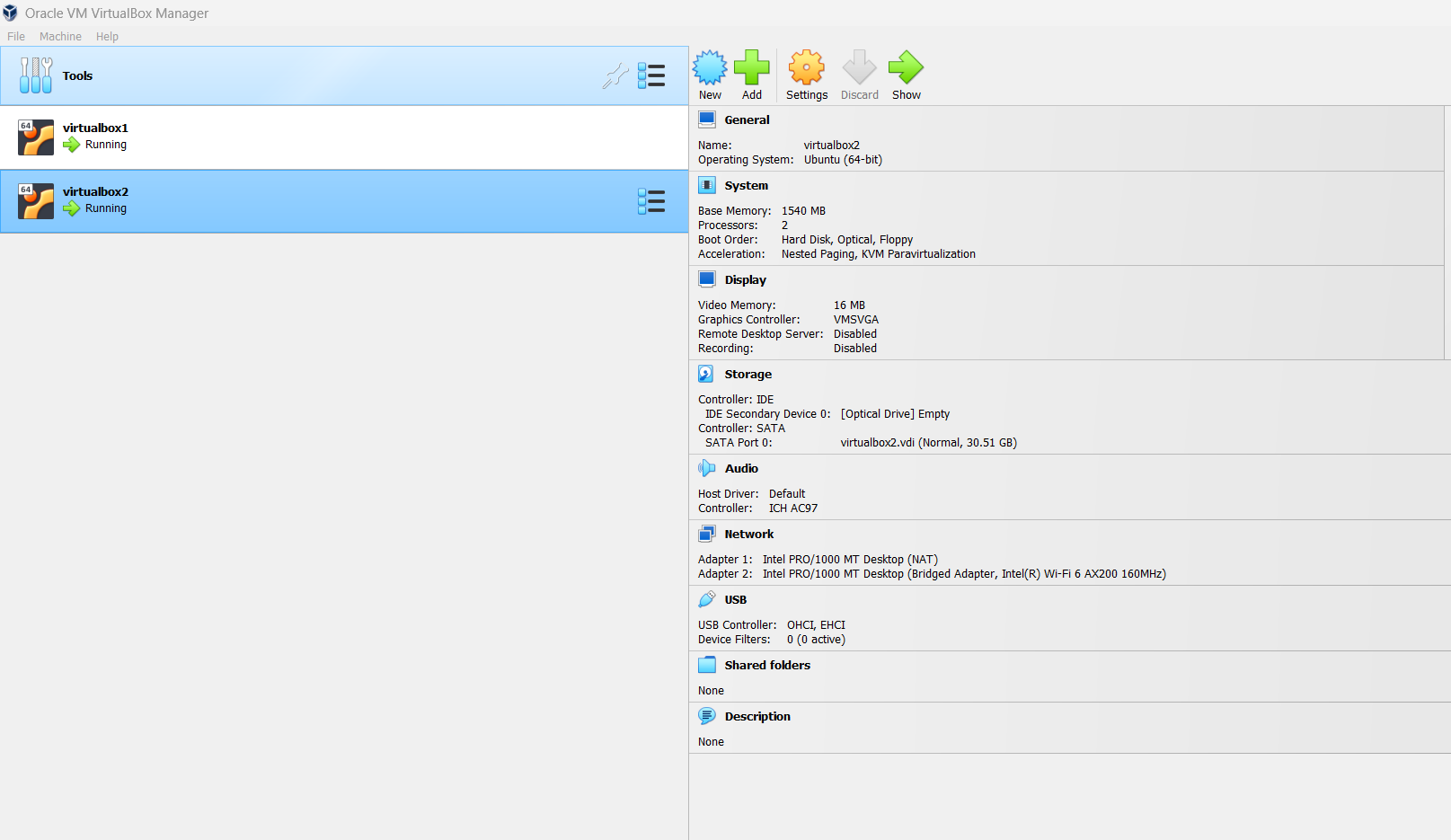
**Setup Instructions**

## ****VirtualBox Machine Creation****

Downloaded and installed VirtualBox from the official website: [VirtualBox](https://www.virtualbox.org/).

Installed **Ubuntu 22.04 LTS** on both VMs.

* Created a new virtual machine with the following specifications:
  + **Name:** Ubuntu
  + **RAM:** 1.5 GB (On Both VM)
  + **Virtual Hard Disk:** 30GB (On Both VM)



## ****2. Establishing Network Connection****

 Configured the network adapter in VirtualBox settings:

* Selected **"Bridged Adapter"** to allow the VM to use the same network as the host machine.
* Alternatively, used **"NAT"** with port forwarding for internet access and local connectivity.
* Using **Shared Folders**, created a shared network between the virtual machine and the host machine.

## ****3. Installing and Configuring SSH****

After obtaining a unique IP, updated and installed OpenSSH Server using:

* + **sudo apt update**
  + **sudo apt install openssh-server**

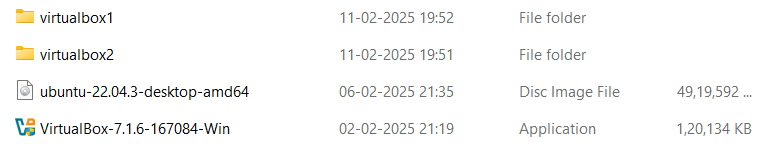
Then, enabled, started, and verified the SSH service:

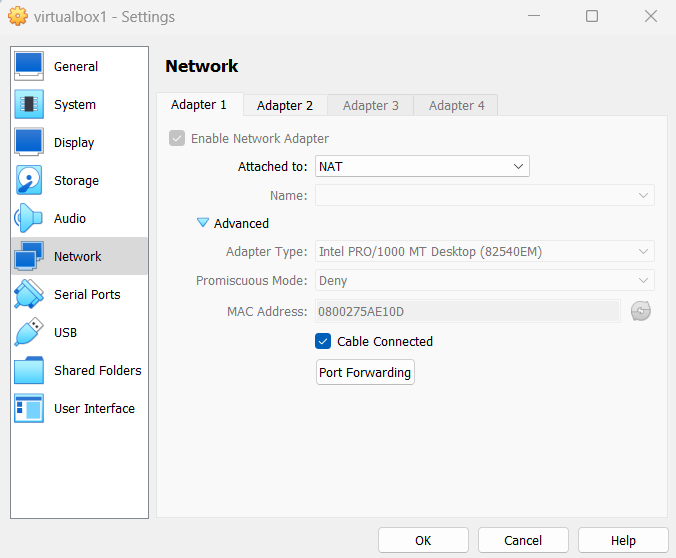
* + **sudosystemctl enable ssh**
  + **sudosystemctl start ssh**
  + **sudosystemctl status ssh**

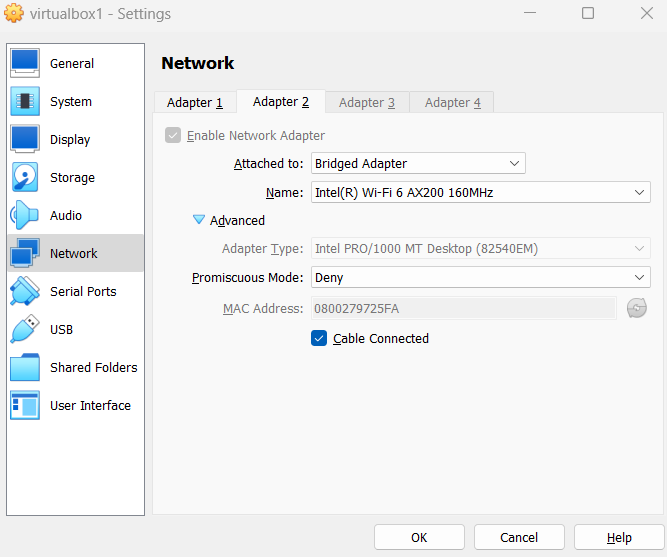
Repeated the same process for **VM2**. Verified network connectivity between VMs using:

ping 192.168.29.78

Successful packet exchange confirmed network establishment.







# Microservice Application Deployment

## ****1. Installing Dependencies****

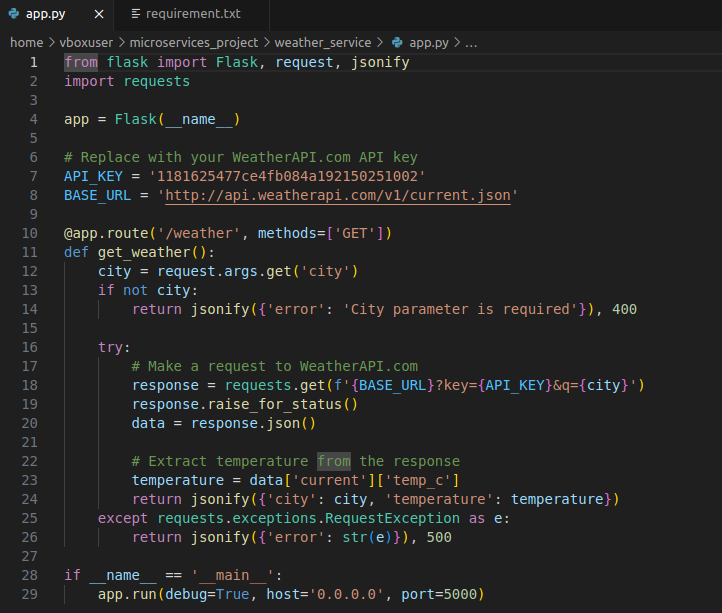
Installed necessary packages on **VM1**:

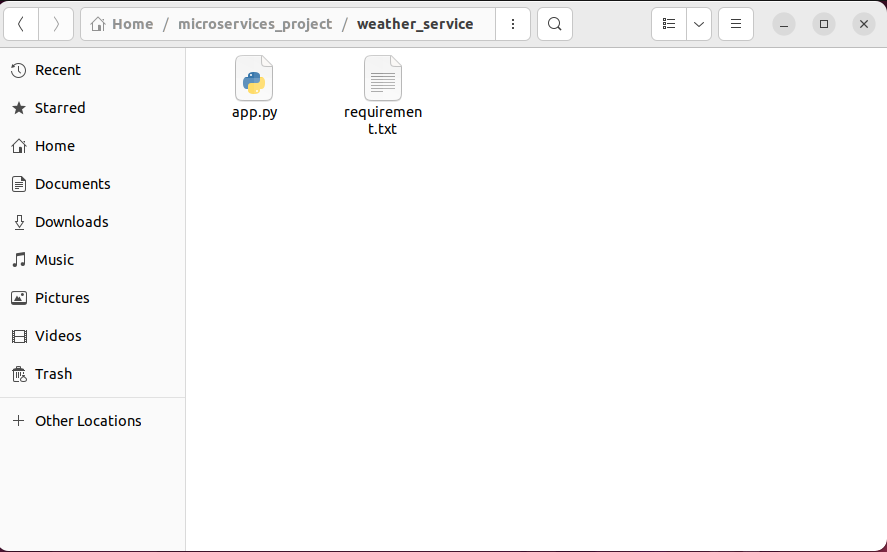
sudo apt install python3-pip

pip3 install flask requests

## ****2. Implementing the Microservice****

* Registered and obtained an **API key** from Weatherstack.com.
* Created a Python script **weather\_service.py** to fetch weather data:





* Ran this script on **VM1**: python3 weather\_service.py
* Tested the API: curl localhost:5000/weather/Delhi
* Successfully retrieved weather data for **Delhi**.

## ****3. Running the Microservice Across VMs****

* Opened **VM2** and connected to **VM1**. After connecting, accessed the microservice running on **VM1** from **VM2**: curl 192.168.29.78:5000/weather/London
* Successfully retrieved weather data from **VM1** to **VM2**.

**Architecture Design**

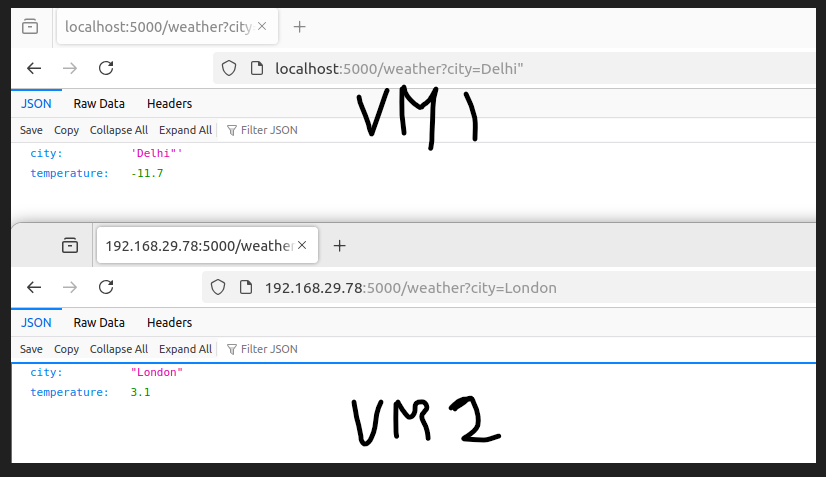
Below is the high-level architecture diagram for this setup:

+----------------------+ +----------------------+

| VM1 (Microservice) | <-----> | VM2 (Client) |

| IP: 192.168.29.235 | | IP: 192.168.29.78 |

| Running Flask App | | Accessing via SSH |



**GitHub Repository**

Successfully deployed the code to GitHub.

**GitHub Repository:** <https://github.com/ydv2027/weather-microservice>

Video :

**Conclusion**

This project successfully demonstrates the creation, configuration, and deployment of a weather microservice across multiple VMs using **VirtualBox**. The microservice is accessible from both VMs, and the setup is tested for functionality and performance.