

MICROSERVICES ON VIRTUALBOX

Report By Vasu Vishnoi

CONTENT TABLE

Introduction

Technologies used

Installing VMs

Creating Microservice

Communication to Service

Testing the services

Demo Video

Architecture

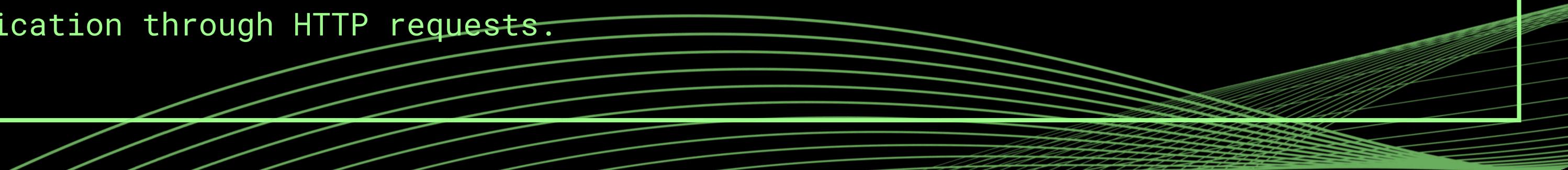
INTRODUCTION

What we will do?

In the contemporary landscape of software development, microservices architecture has become pivotal. This report provides a comprehensive guide on how to use Oracle VirtualBox to create multiple Virtual Machines (VMs) and host a microservice-based application.

How to do it?

The objective of this assignment is to set up two virtual machines (VMs) using VirtualBox, connect them via a NAT network, and deploy a simple Python-based API on each VM. The primary goal is to enable communication between the two VMs using the Python APIs, demonstrating inter-VM communication through HTTP requests.



ooo

TECHNOLOGIES USED:

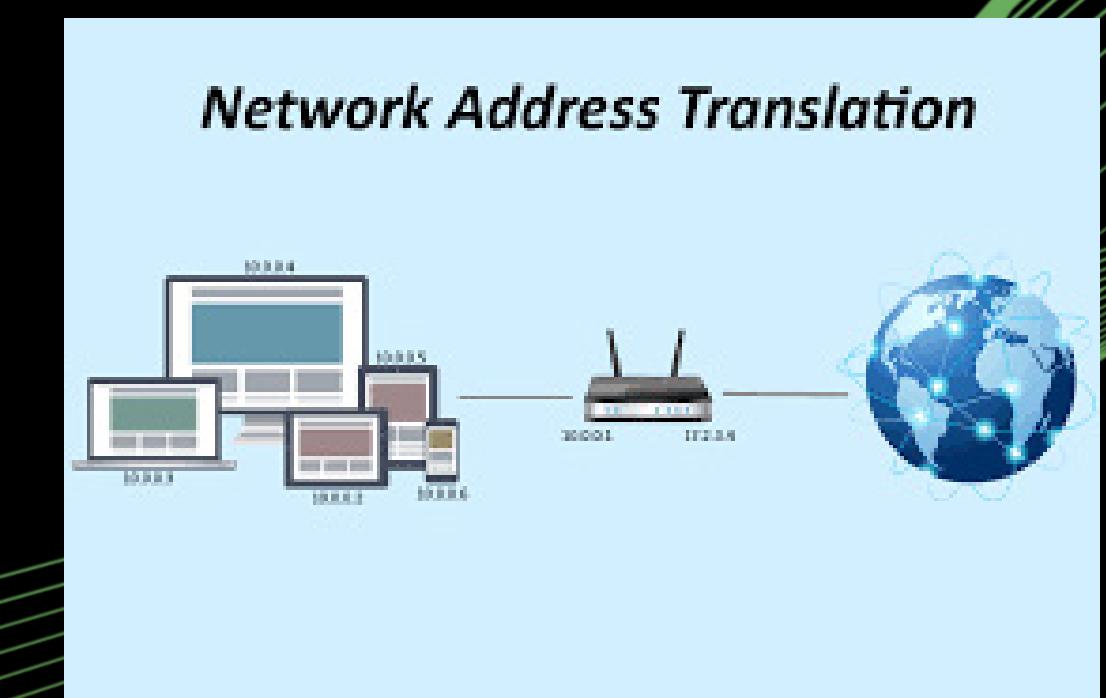
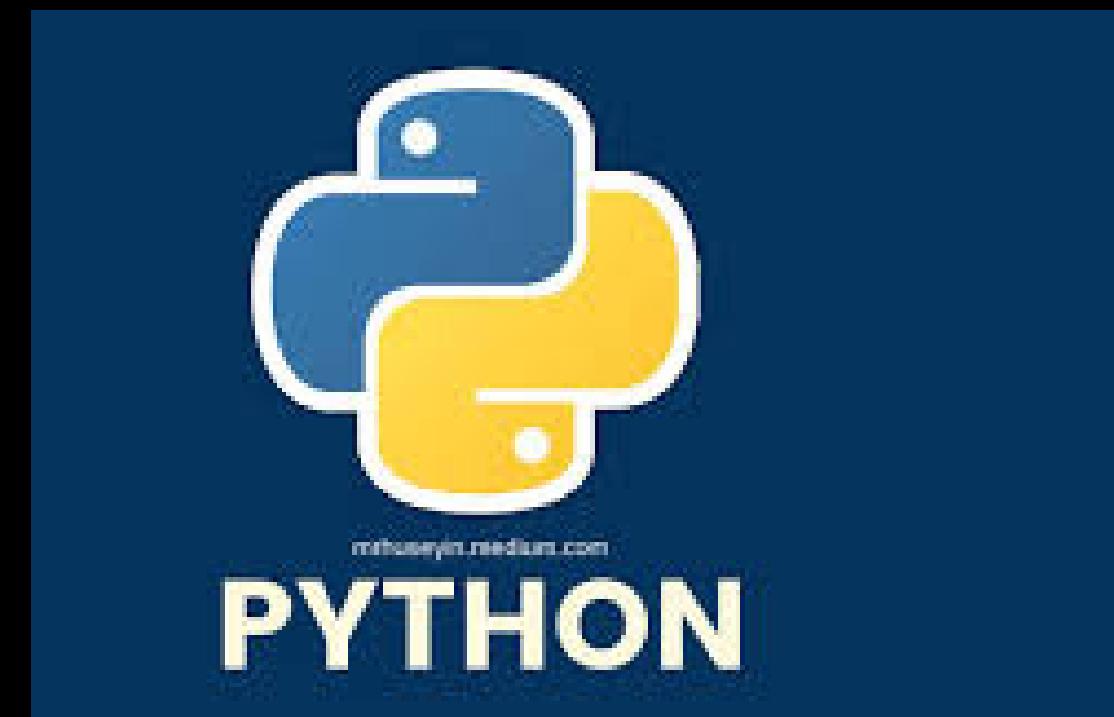
VIRTUALBOX: VIRTUALIZATION SOFTWARE TO CREATE AND MANAGE VIRTUAL MACHINES (VMS).

PYTHON: PROGRAMMING LANGUAGE USED TO DEVELOP THE API.

FLASK: LIGHTWEIGHT WEB FRAMEWORK FOR BUILDING THE PYTHON API.

HTTP: PROTOCOL USED FOR COMMUNICATION BETWEEN THE APIs ON THE VMS.

NAT NETWORK: NETWORK CONFIGURATION IN VIRTUALBOX TO ALLOW INTERNET ACCESS AND PORT FORWARDING FOR THE VMS



→ INSTALLING VMS

Download Oracle VirtualBox

Download UBUNTU iso

Click New on Oracle VirtualBox and assign

Memory and Processor to VM after selecting downloaded iso

Click submit, VM is created and OS installation begins.

Open settings and select Network. Inside it in Adapter 1 select network type as Nat Network. Before that you need to add a NAT network in main preferences of Oracle VirtualBox

CREATING MICROSERVICE

Install Python and Flask on Each VM:

SSH into each VM or open the terminal.

Update package list: `sudo apt update`

Install Python and pip: `sudo apt install python3 python3-pip`

Install Flask: `pip3 install Flask`

Create a Simple Python API:

Create a project folder: `mkdir my_microservice && cd my_microservice`

Create a file named `app.py`

Run the Python API on Each VM

Set Up Port Forwarding



COMMUNICATION TO SERVICE

Run the Python API on Each VM:

Navigate to the project folder and run: `python3 app.py`

The API will be accessible on port 5000.

Set Up Port Forwarding:

In VirtualBox, go to Settings -> Network -> Adapter 1 (NAT) -> Advanced -> Port Forwarding.

Add rules for each VM:

VM 1: Forward host port 5001 to guest port 5000.

VM 2: Forward host port 5002 to guest port 5000.

Test Communication Between VMs:

On VM 1, test accessing VM 2's API: `curl http://<host_machine_ip>:5002/api`

Ensure that you receive a response from VM 2.

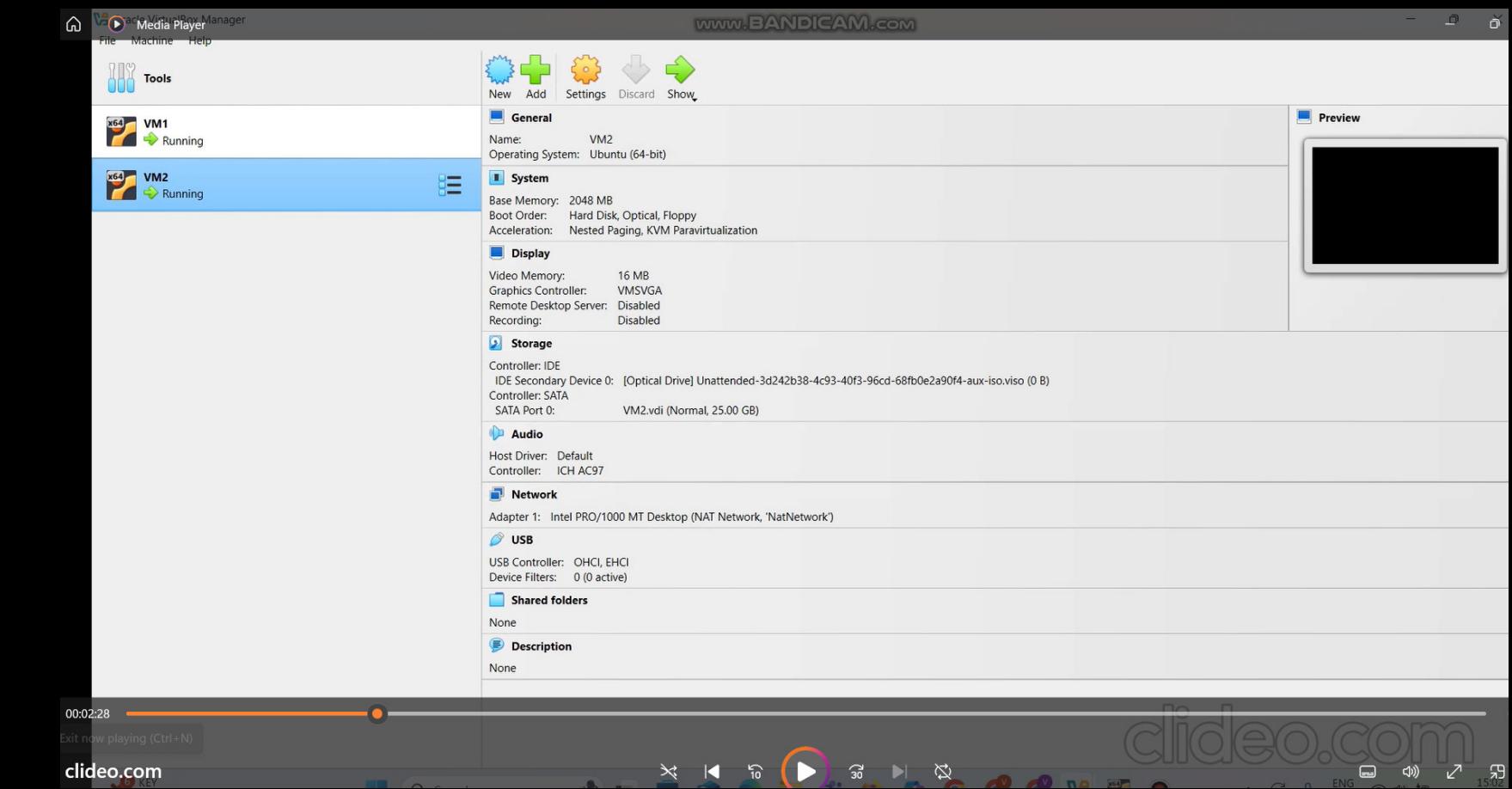
Test API on VM 1: On VM 1, use curl to access the API on VM 2:

`curl http://localhost:5002/api`

Test API on VM 2: On VM 2, use curl to access the API on VM 1:

`curl http://localhost:5001/api`

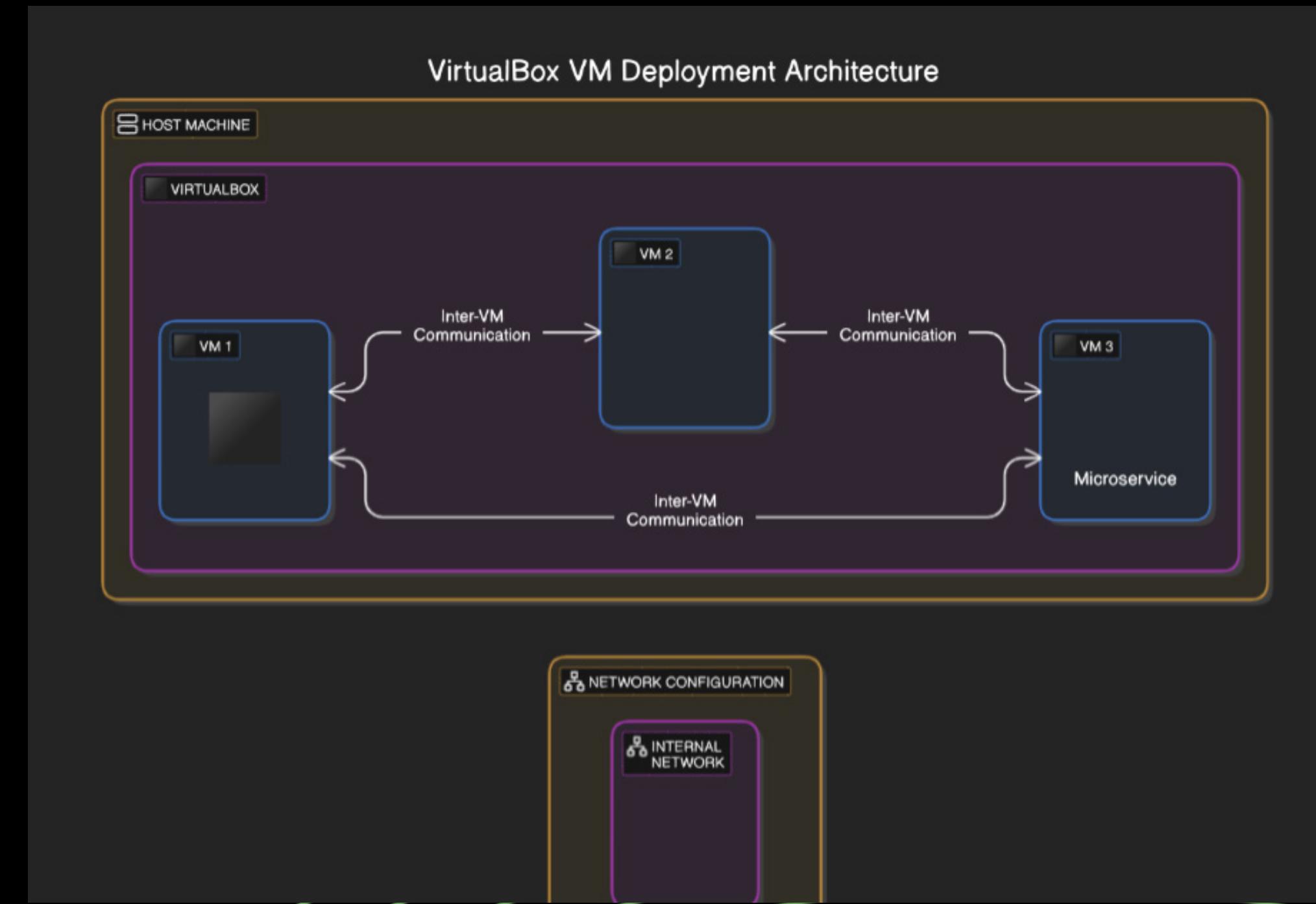
DEMO VIDEO



github link:

<https://github.com/Vasu1999-IIT/vcc-assignment-1>

ARCHITECTURE



○○○

03

TEST RESULT BROWSER

