

# Microservice-Based Application using VirtualBox

## 1. Introduction

### Objective

The objective of this project is to create multiple Ubuntu-based Virtual Machines (VMs) using VirtualBox, establish network connectivity between them, and deploy a microservice-based application across the VMs. The deployment consists of:

- A **Node.js microservice** running on **VM1** (Application Server)
- A **MySQL database** running on **VM2** (Database Server)
- An **Nginx reverse proxy** running on **VM3** (Load Balancer)

## 2. VirtualBox Installation and VM Setup

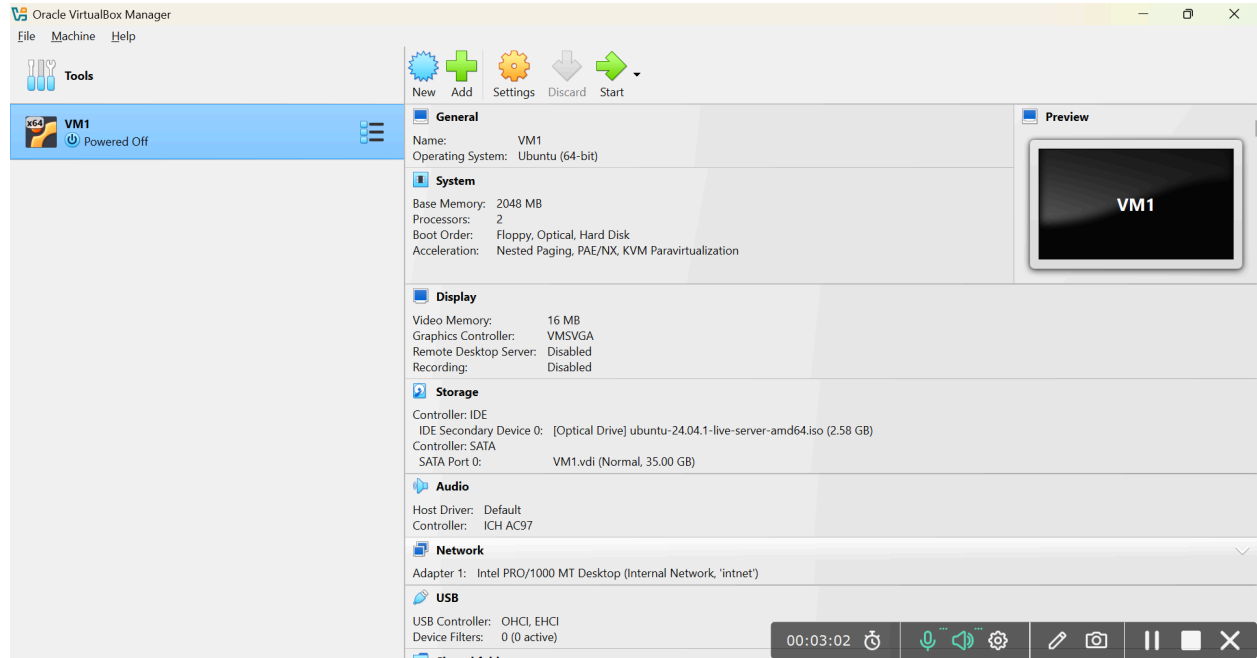
### 2.1 Installation of VirtualBox & Ubuntu ISO Image

1. Downloaded VirtualBox from <https://www.virtualbox.org/>
2. Installed the VirtualBox Extension Pack for additional functionality.
3. Downloaded the latest Ubuntu Server ISO from <https://ubuntu.com/download/server>

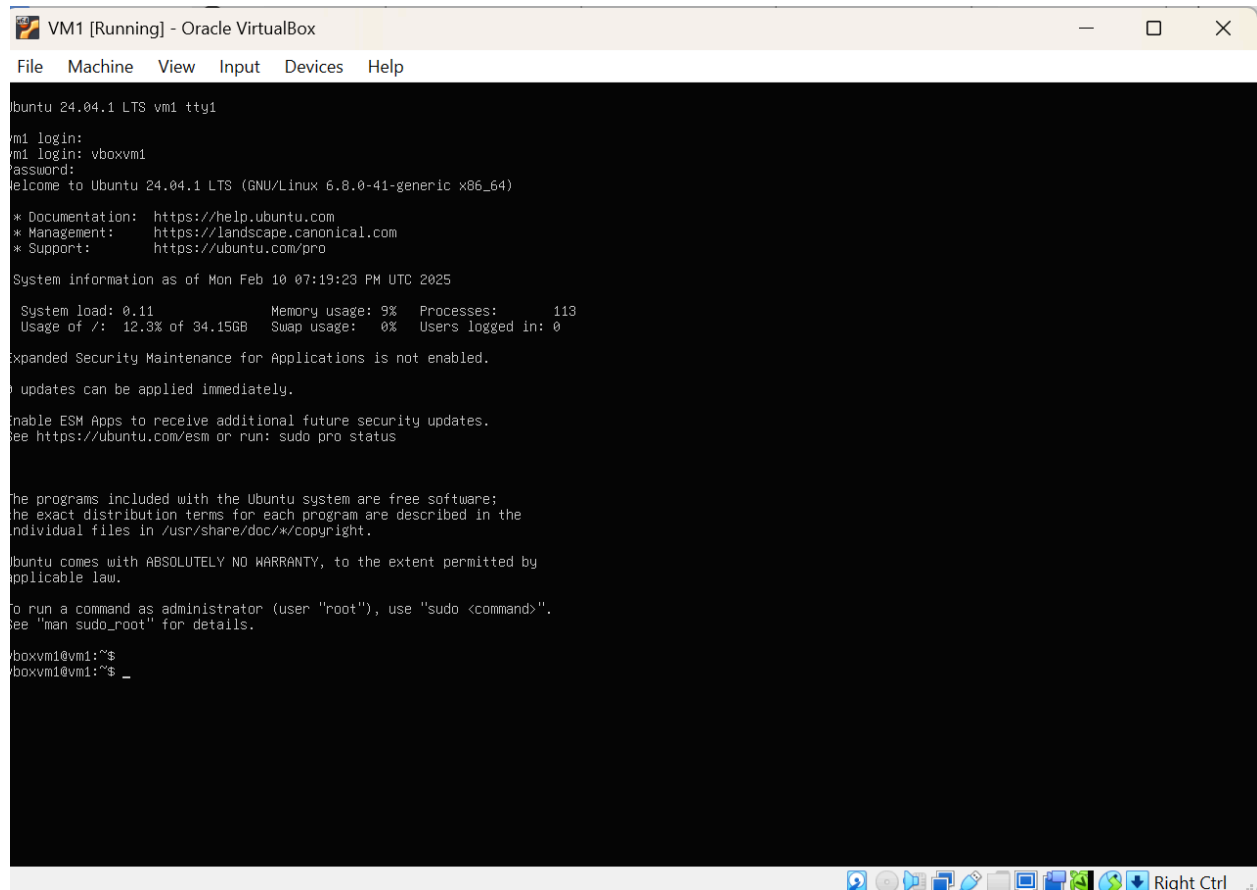
### 2.2 Creating Multiple Ubuntu VMs

1. Open VirtualBox and click **New**.
2. Set up VM1 with the following configurations:
  - OS: Linux (Ubuntu 64-bit)
  - RAM: 2GB (minimum)
  - Storage: 35GB (dynamically allocated)
  - Open **Settings > Network** for each VM.
  - Choose **Adapter 1 > Internal Network**.
  - Attach Ubuntu ISO and complete installation.

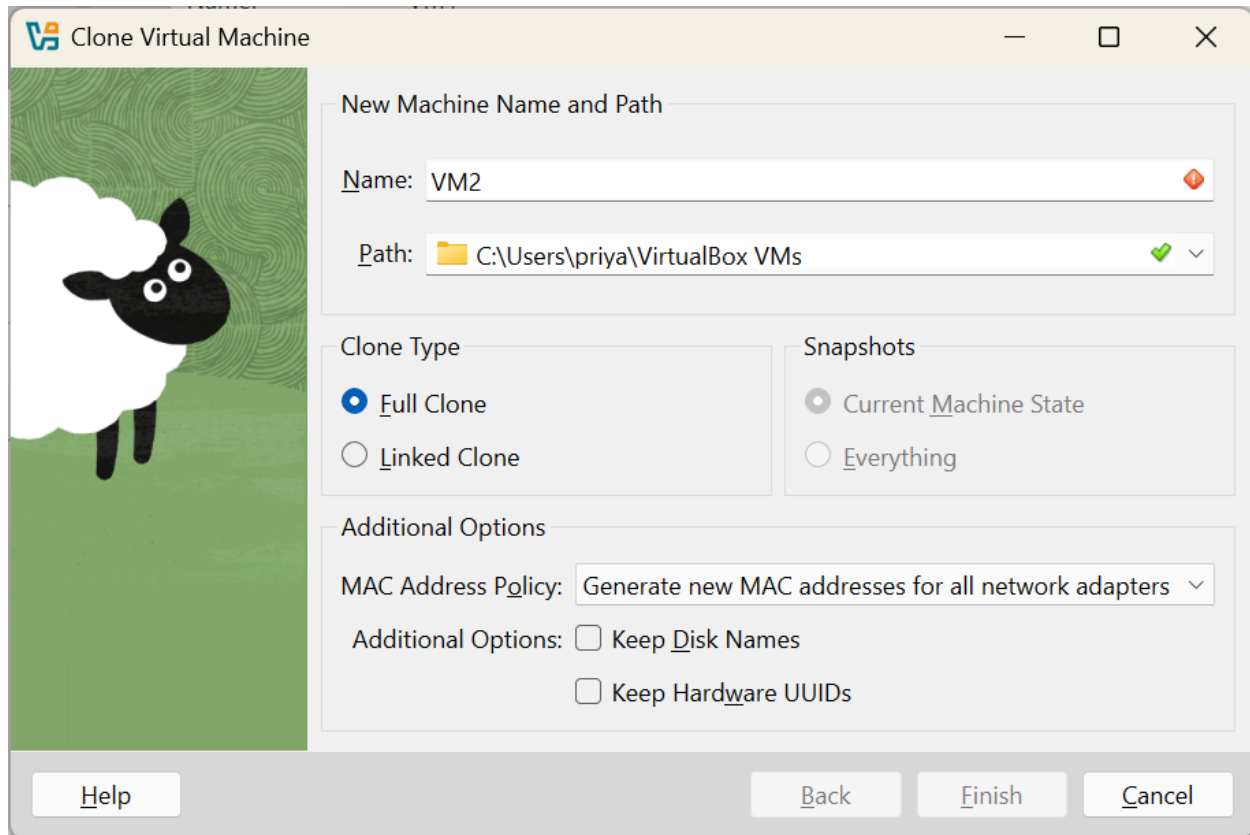
## VM1 Setup



## After all installations



3. Clone VM1 to create VM2 and VM3 (Full Clone method).



### 3. Networking Configuration

#### 3.1 Configuring Network

1. Assign static IPs to the VMs by modifying `/etc/netplan/00-installer-config.yaml`.

**VM1:**

```
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.101/24]
  version: 2
```

Repeat for VM2 (`192.168.1.102/24`) and VM3 (`192.168.1.103/24`).

VM	IP Address	Subnet Mask (CIDR)
VM1	<code>192.168.1.101/24</code>	255.255.255.0
VM2	<code>192.168.1.102/24</code>	255.255.255.0
VM3	<code>192.168.1.103/24</code>	255.255.255.0

Apply changes using:

```
sudo netplan apply
```

2. Test connectivity between VMs using:

```
ping 192.168.1.102 # From VM1 to VM2
ping 192.168.1.103 # From VM1 to VM3
```

VM2 [Running] - Oracle VirtualBox

File	Machine	View	Input	Devices	Help
64 bytes from 192.168.1.101:	icmp_seq=31	ttl=64	time=1.76 ms		
64 bytes from 192.168.1.101:	icmp_seq=32	ttl=64	time=2.92 ms		
64 bytes from 192.168.1.101:	icmp_seq=33	ttl=64	time=2.08 ms		
64 bytes from 192.168.1.101:	icmp_seq=34	ttl=64	time=1.71 ms		
64 bytes from 192.168.1.101:	icmp_seq=35	ttl=64	time=1.50 ms		
64 bytes from 192.168.1.101:	icmp_seq=36	ttl=64	time=0.982 ms		
64 bytes from 192.168.1.101:	icmp_seq=37	ttl=64	time=0.878 ms		
64 bytes from 192.168.1.101:	icmp_seq=38	ttl=64	time=1.87 ms		
64 bytes from 192.168.1.101:	icmp_seq=39	ttl=64	time=1.95 ms		
64 bytes from 192.168.1.101:	icmp_seq=40	ttl=64	time=2.19 ms		
64 bytes from 192.168.1.101:	icmp_seq=41	ttl=64	time=1.32 ms		
64 bytes from 192.168.1.101:	icmp_seq=42	ttl=64	time=1.86 ms		
64 bytes from 192.168.1.101:	icmp_seq=43	ttl=64	time=1.12 ms		
64 bytes from 192.168.1.101:	icmp_seq=44	ttl=64	time=3.16 ms		
64 bytes from 192.168.1.101:	icmp_seq=45	ttl=64	time=1.30 ms		
64 bytes from 192.168.1.101:	icmp_seq=46	ttl=64	time=1.41 ms		
64 bytes from 192.168.1.101:	icmp_seq=47	ttl=64	time=4.45 ms		
64 bytes from 192.168.1.101:	icmp_seq=48	ttl=64	time=1.91 ms		

VM1 [Running] - Oracle VirtualBox

File	Machine	View	Input	Devices	Help
64 bytes from 192.168.1.103:	icmp_seq=85	ttl=64	time=1.22 ms		
64 bytes from 192.168.1.103:	icmp_seq=86	ttl=64	time=1.87 ms		
64 bytes from 192.168.1.103:	icmp_seq=87	ttl=64	time=1.08 ms		
64 bytes from 192.168.1.103:	icmp_seq=88	ttl=64	time=0.684 ms		
64 bytes from 192.168.1.103:	icmp_seq=89	ttl=64	time=1.05 ms		
64 bytes from 192.168.1.103:	icmp_seq=90	ttl=64	time=3.86 ms		
64 bytes from 192.168.1.103:	icmp_seq=91	ttl=64	time=1.42 ms		
64 bytes from 192.168.1.103:	icmp_seq=92	ttl=64	time=0.599 ms		
64 bytes from 192.168.1.103:	icmp_seq=93	ttl=64	time=0.486 ms		
64 bytes from 192.168.1.103:	icmp_seq=94	ttl=64	time=1.91 ms		
64 bytes from 192.168.1.103:	icmp_seq=95	ttl=64	time=1.90 ms		
64 bytes from 192.168.1.103:	icmp_seq=96	ttl=64	time=1.67 ms		
64 bytes from 192.168.1.103:	icmp_seq=97	ttl=64	time=0.962 ms		
64 bytes from 192.168.1.103:	icmp_seq=98	ttl=64	time=3.04 ms		
64 bytes from 192.168.1.103:	icmp_seq=99	ttl=64	time=6.86 ms		
64 bytes from 192.168.1.103:	icmp_seq=100	ttl=64	time=3.06 ms		

## 4. Microservice Deployment

While installing dependencies , it failed due to Internet connection. So Switched back to **NAT** Adaptor and installed dependencies on each VM.

Few Common , for further usages

```
sudo apt install traceroute -y
sudo apt update --fix-missing
sudo apt install -f
sudo apt install ifupdown -y
sudo apt install isc-dhcp-client -y
sudo apt update && sudo apt upgrade -y
```

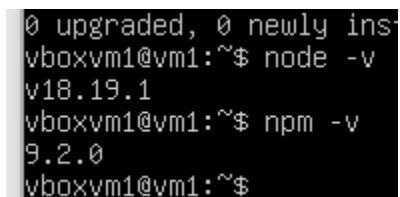
### 4.1 Deploying Node.js Microservice on VM1

1. Install Node.js and npm:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y nodejs npm git curl
```

2. Check Node.js version

```
node -v
npm -v
```



```
0 upgraded, 0 newly inst
vboxvm1@vm1:~$ node -v
v18.19.1
vboxvm1@vm1:~$ npm -v
9.2.0
vboxvm1@vm1:~$
```

3. Install Express.js  

```
npm install express cors
npm install -g pm2
```

4. Create a microservice:

```
mkdir microservice && cd microservice
npm init -y
npm install express mysql2 cors
```

```
npm install mysql2
```

3. Create `index.js`: Placed in the github.
4. Start the service:

```
node index.js
```

```
total 8
-rw-rw-r-- 1 vboxvm1 vboxvm1 227 Feb 11 09:55 package.json
-rwxrwx--- 1 vboxvm1 vboxvm1 1260 Feb 11 12:03 index.js
vboxvm1@vm1:~/microservices$ node index.js
Microservice running on port 3000
Connected to MySQL on VM2
```

## 4.2 Setting Up MySQL Database on VM2

1. Install MySQL:

```
sudo apt install mysql-client-core-8.0
sudo apt update && sudo apt install -y mysql-server
mysql --version
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
vboxvm1@vm1:/etc/netplan$ mysql --version
mysql Ver 8.0.41-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))
vboxvm1@vm1:/etc/netplan$ sudo apt install mysql-client-core-8.0
```

```
sudo systemctl start mysql
sudo systemctl enable mysql
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
vboxvm1@vm1:/etc/netplan$ mysql --version
mysql Ver 8.0.41-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))
vboxvm1@vm1:/etc/netplan$ sudo systemctl start mysql
vboxvm1@vm1:/etc/netplan$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
vboxvm1@vm1:/etc/netplan$
```

```
sudo systemctl status mysql
```

```

Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
vboxvm1@vm1:/etc/netplan$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-02-11 10:14:27 UTC; 3min 1s ago
     Main PID: 28755 (mysqld)
    Status: "Server is operational"
       Tasks: 37 (limit: 2276)
      Memory: 363.9M (peak: 378.2M)
         CPU: 3.764s
        CGroup: /system.slice/mysql.service
                └─28755 /usr/sbin/mysqld

Feb 11 10:14:25 vm1 systemd[1]: Starting mysql.service - MySQL Community Server...
Feb 11 10:14:27 vm1 systemd[1]: Started mysql.service - MySQL Community Server.
vboxvm1@vm1:/etc/netplan$

```

## 2. Create a database and table: using setup.sql

`sudo mysql`

```

mysql> CREATE DATABASE microservice_db;
Query OK, 1 row affected (0.04 sec)

mysql> use microservice_db;
Database changed
mysql> CREATE TABLE users(
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(100),
    -> email VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.12 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_microservice_db |
+-----+
| users                      |
+-----+
1 row in set (0.00 sec)

mysql> CREATE USER 'user'@'192.168.1.%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES ON microservice_db.* TO 'user'@'192.168.1.%';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>

```



3. Edit MySQL config to allow remote access:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Change:

```
bind-address = 0.0.0.0
```

4. Restart MySQL:

```
sudo systemctl restart mysql
```

## 4.3 Configuring Nginx Reverse Proxy on VM3

1. Install Nginx:

```
sudo apt install -y nginx
sudo systemctl enable --now nginx
systemctl status nginx
```

```
vboxvm1@vm1:/etc/netplan$ sudo systemctl status nginx
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-02-11 10:22:44 UTC; 4min 17s ago
     Docs: man:nginx(8)
  Main PID: 27817 (nginx)
    Tasks: 3 (limit: 2276)
   Memory: 2.4M (peak: 2.5M)
      CPU: 30ms
   CGroup: /system.slice/nginx.service
           └─27817 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           └─27818 "nginx: worker process"
           └─27819 "nginx: worker process"

Feb 11 10:22:44 vm1 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Feb 11 10:22:44 vm1 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
vboxvm1@vm1:/etc/netplan$
```

2. Configure Nginx:

```
sudo nano /etc/nginx/sites-available/microservice
```

Add:

```
server {
    listen 80;
    server_name 192.168.1.103;
    location / {
        proxy_pass http://192.168.1.101:3000;
    }
}
```

3. Enable and restart:

```
sudo ln -s /etc/nginx/sites-available/microservice /etc/nginx/sites-enabled/  
sudo systemctl restart nginx
```

## 5. Testing the Deployment

### 5.1 API Testing

Run these commands from **VM3** or a client machine:

1. **Store Data & Fetch Data**

```
</html>  
vboxvm1@vm1:/var/log/nginx$ curl -X GET http://192.168.1.101:3000/users  
[{"id":1,"name":"Rama","email":"Rama@gmail.com"}]vboxvm1@vm1:/var/log/nginx$  
vboxvm1@vm1:/var/log/nginx$  
vboxvm1@vm1:/var/log/nginx$  
vboxvm1@vm1:/var/log/nginx$ curl -X POST http://192.168.1.101:3000/users \  
> -H "Content-Type: application/json" \  
> -d '{"name": "Shyam", "email": "shyam@gmail.com"}'  
{"message":"User added successfully","userId":2}vboxvm1@vm1:/var/log/nginx$
```

2. Check in VM2

```
mysql> Select * from users;  
+-----+  
id | name | email |  
+-----+  
1 | Rama | Rama@gmail.com |  
2 | Shyam | shyam@gmail.com |  
+-----+  
2 rows in set (0.02 sec)  
  
mysql>
```

## 6. Architecture Design

**VM1 (Microservice - Node.js):** Hosts the REST API and processes business logic.

**VM2 (Database - MySQL):** Stores and manages data.

**VM3 (Reverse Proxy - Nginx):** Forwards client requests to the microservice.

Action	Source	Destination
Client requests API	Client	VM3(Nginx)
Nginx forwards request	VM3	VM1(Node.js)
Microservice queries database	VM1	VM2(MYSQL)
Database returns data	VM2	VM1
Microservice sends response	VM1	VM3
Nginx forwards response	VM3	Client

## 7. Deliverables

- **GitHub Repo:** [https://github.com/M23CSA520/VCC\\_PA1](https://github.com/M23CSA520/VCC_PA1)
- **Video Demonstration:** Links

[https://drive.google.com/file/d/1unl-hPaUZhkCTmVX5dk0oBvr\\_ozcVlco/view?usp=sharing](https://drive.google.com/file/d/1unl-hPaUZhkCTmVX5dk0oBvr_ozcVlco/view?usp=sharing)

[https://drive.google.com/file/d/1OtM7BiUT9zzx2kHIWycZGenPvmL\\_j3fN/view?usp=drive\\_link](https://drive.google.com/file/d/1OtM7BiUT9zzx2kHIWycZGenPvmL_j3fN/view?usp=drive_link)

## 8. Conclusion

This project successfully demonstrates how to deploy a microservice-based application across multiple VirtualBox VMs. The architecture includes a Node.js API, MySQL database, and Nginx reverse proxy, enabling efficient communication between services. The implementation provides a scalable and modular foundation for future enhancements.