

Assignment - 1 (Virtualization and cloud computing)

Name - Sachchida nand singh

Roll number - G24AI1048

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.

Step-by-Step Instructions for Implementation:

Local system used - macbook pro M1 chip (16GB RAM)

1. Installation of VirtualBox -

Download virtual box from below link -

<https://www.virtualbox.org/wiki/Downloads>

Refer the below screenshot -



Since I am using an M1 chip, I downloaded the **macOS / Apple Silicon hosts**. Now, using the above downloaded file, I completed the installation of Virtual box.

2. Download the ISO image -

This is the OS which will be used for VMs.

Initially I downloaded **Ubuntu (ubuntu-22.04.5-live-server-arm64.iso)** but this doesn't work due to some limitations on Mac os, then I moved to **Centos (CentOS-Stream-10-latest-aarch64-dvd1.iso)** image and that worked. Below is the link to centos image -

<https://www.centos.org/download/>

Refer the below screenshot -

CentOS Stream

Continuously delivered distro that tracks just ahead of Red Hat Enterprise Linux (RHEL) development, positioned as a midstream between Fedora Linux and RHEL. For anyone interested in participating and collaborating in the RHEL ecosystem, CentOS Stream is your reliable platform for innovation.

Architecture	ISOs	RPMs	Cloud	Containers	Vagrant
x86_64	Mirrors	Mirrors	Images	Images	Boxes
ARM64 (aarch64)	Mirrors	Mirrors	Images	Images	
IBM Power (ppc64le)	Mirrors	Mirrors	Images	Images	
IBM Z (s390x)	Mirrors	Mirrors	Images	Images	

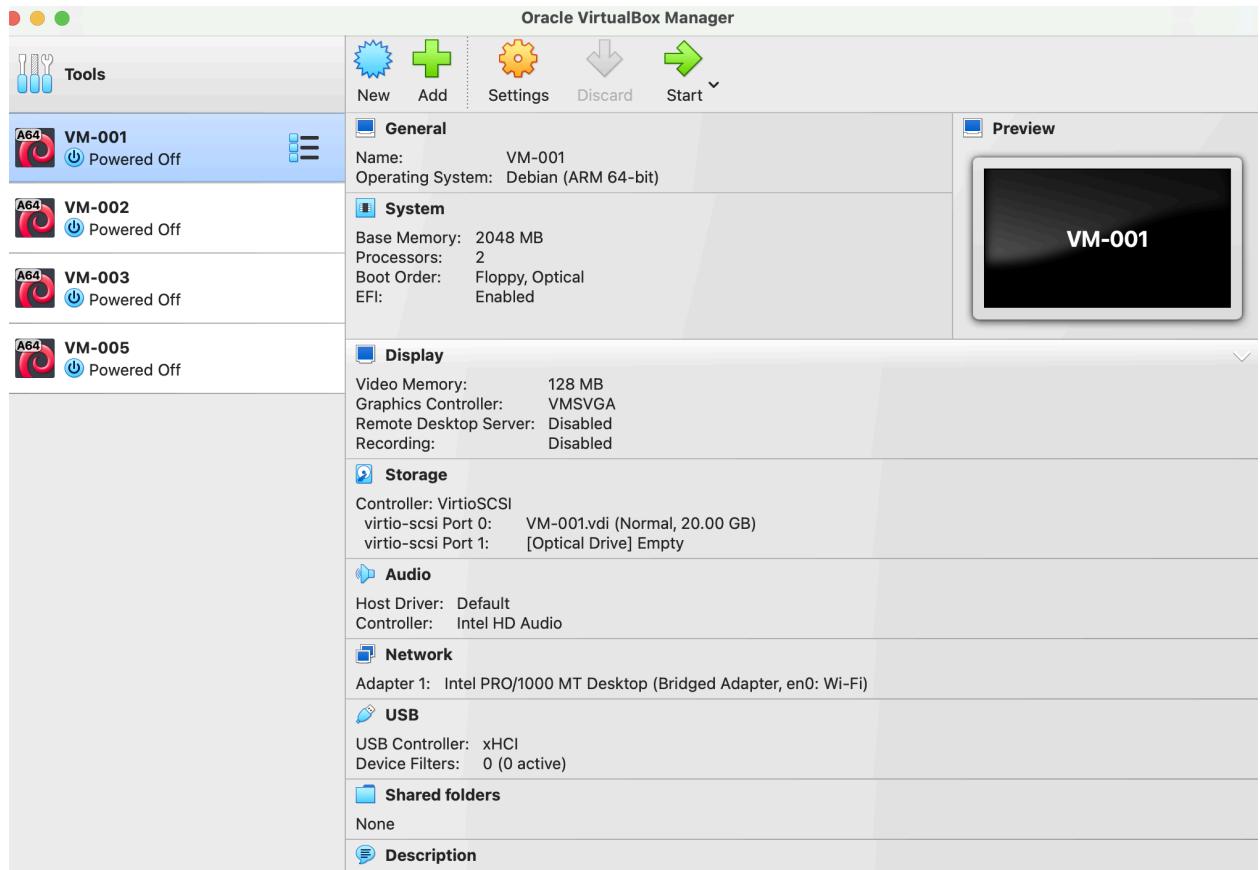
I used **ARM64 (aarch64)**.

3. Creation of VM -

After the successful installation of Virtual Box, I created multiple VM using the below configurations -

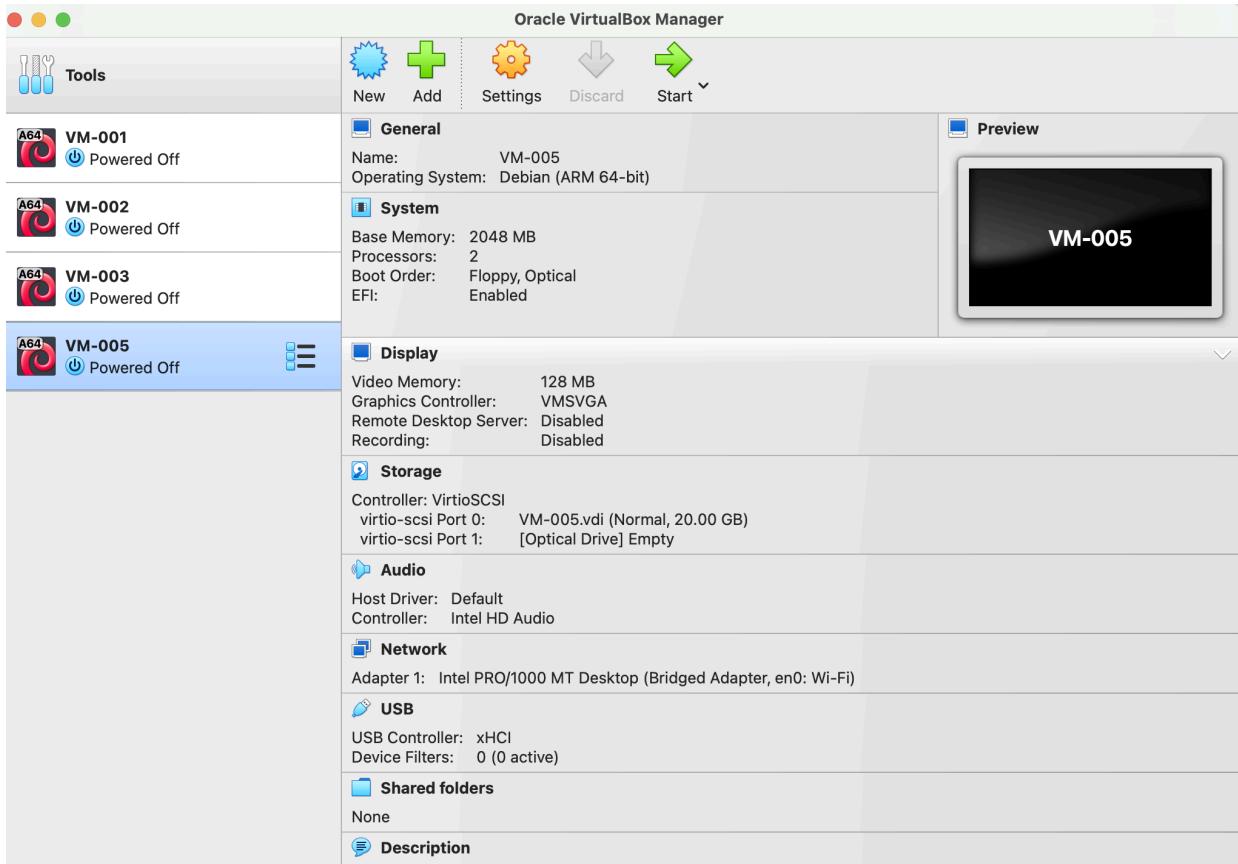
a. VM-001 -

Refer the below screenshot for configurations -



b. VM-005 -

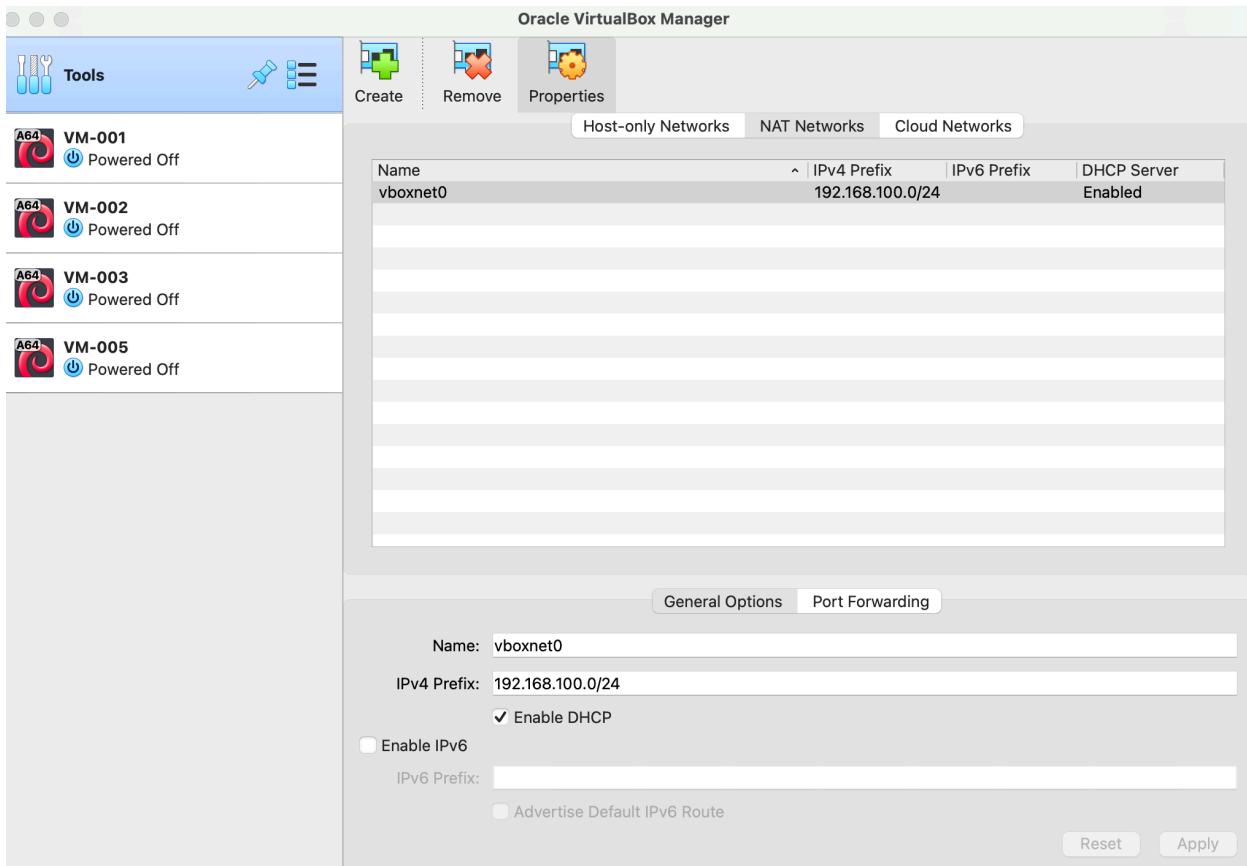
Refer the below screenshot for configurations -



Both the VMs have similar configurations.

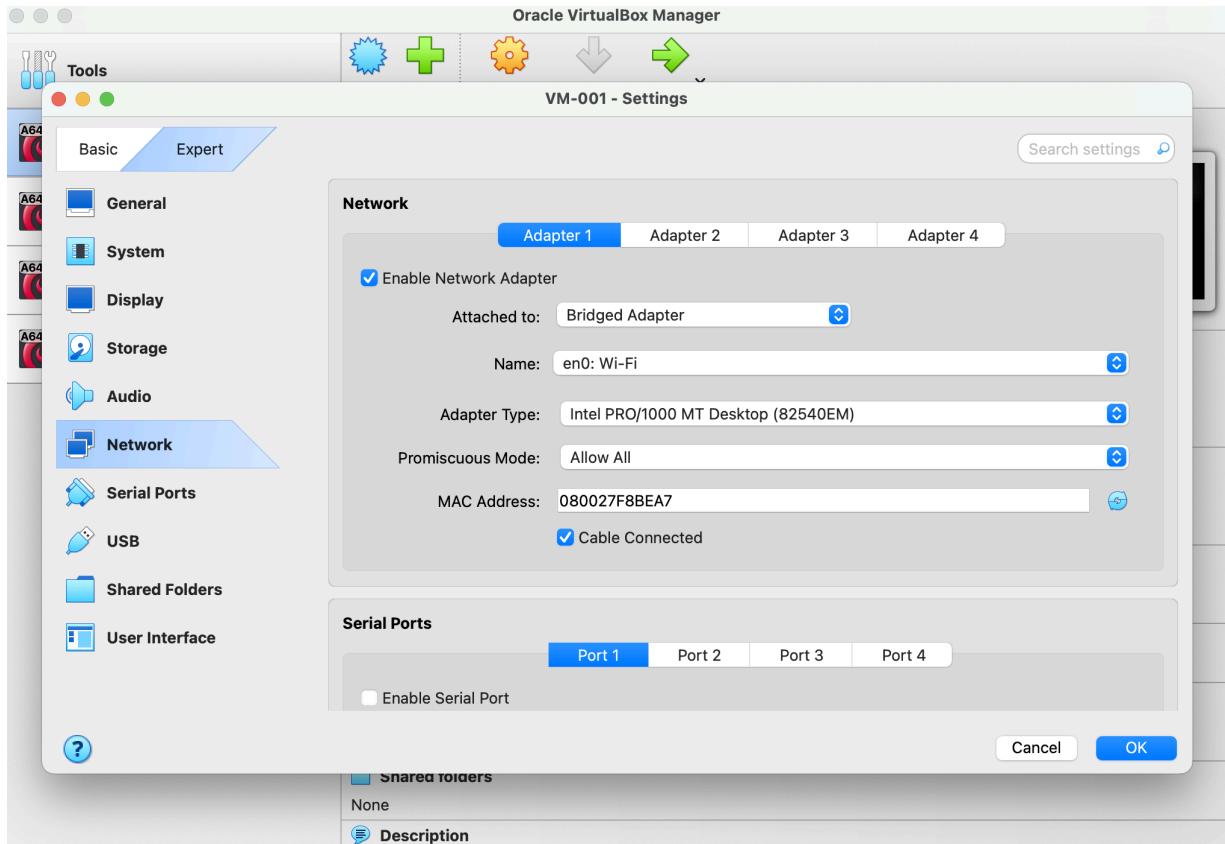
4. Network configuration -

First step we need to create an entry in Virtual box's network tab, below is the screenshot for the same -



Once this is done, save the settings and now move to VM settings, and under network tab we can enable the network settings under **Adapter 1**.

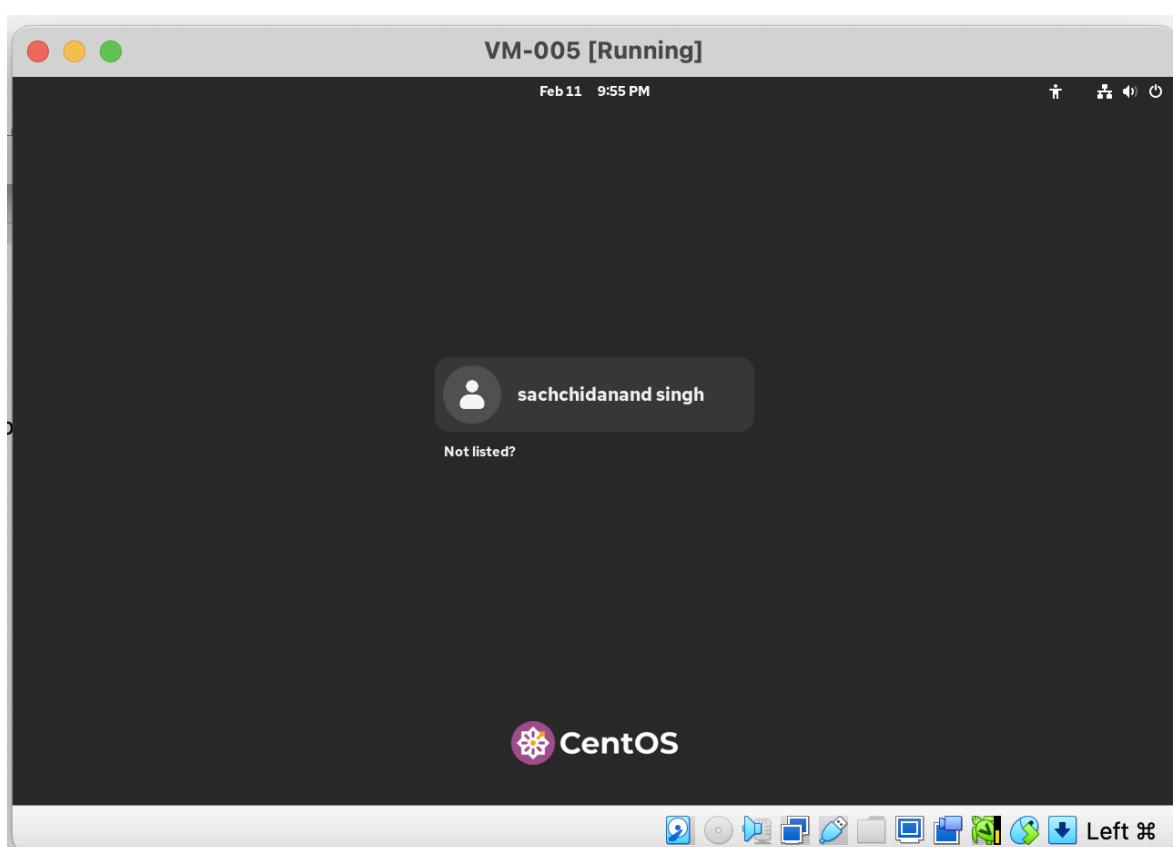
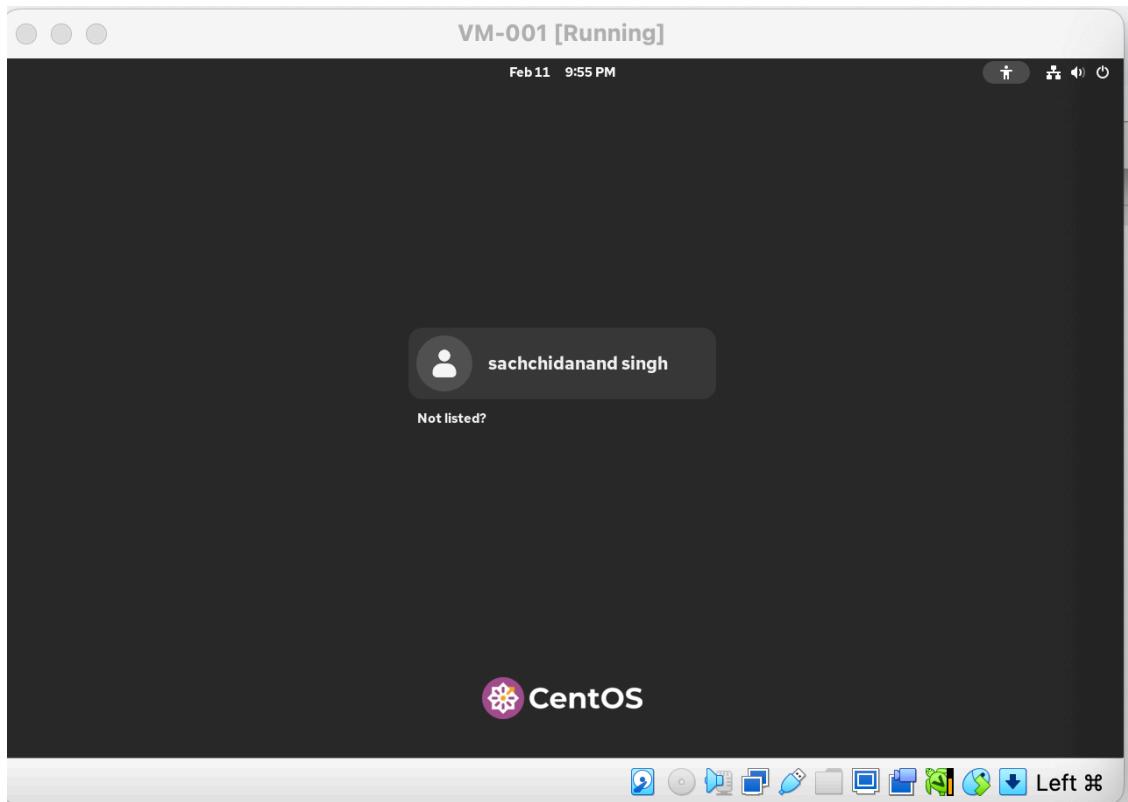
Below is the screenshot for network settings -



5. Start the VMs -

Now start both the VMs and install the OS.

After the installation of OS both the VMs should look like below -



Now check the connectivity between both the vms.

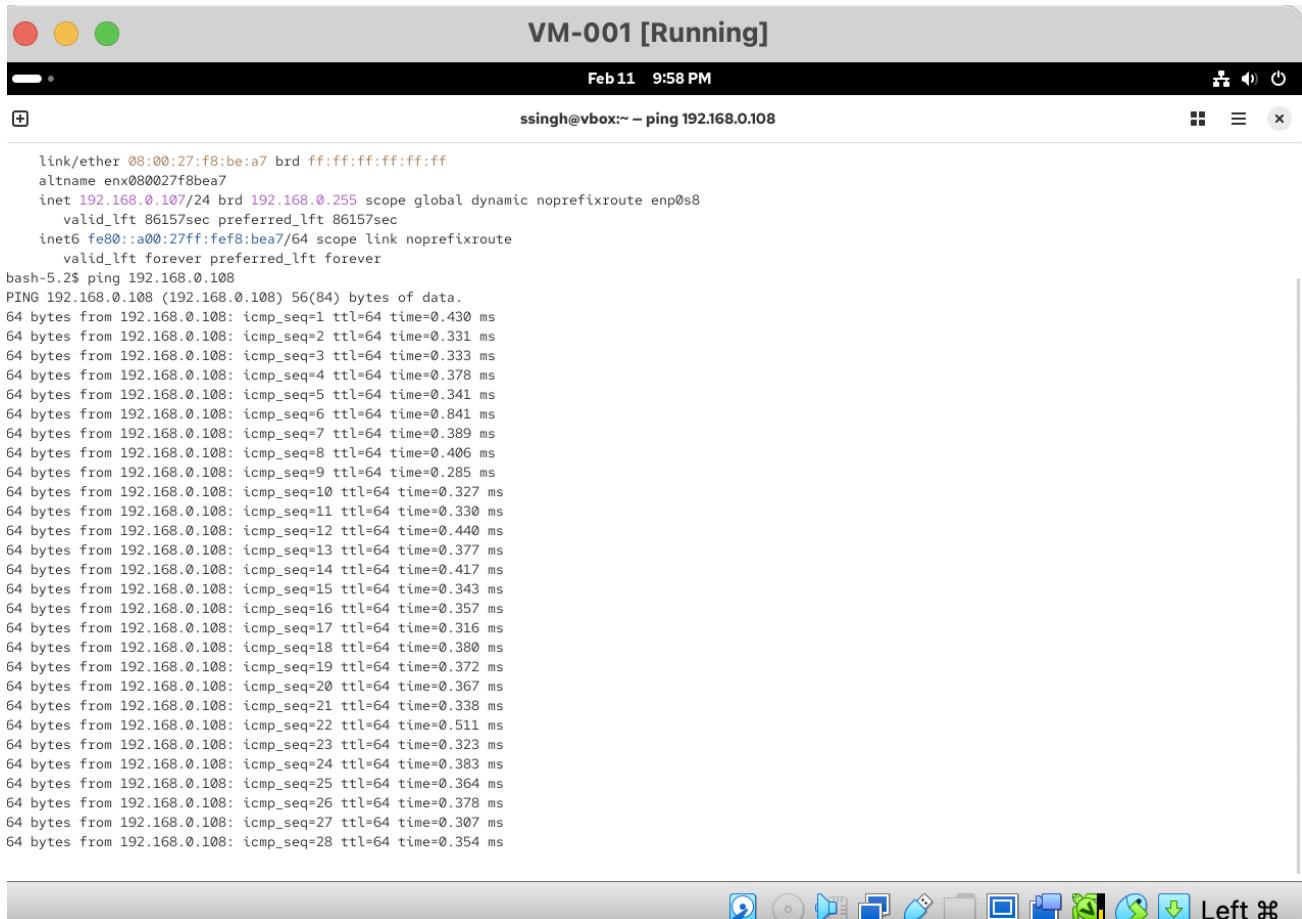
Check the ip by below command -

```
ip a
```

then try ping command to check if they are able to communicate each other -

```
ping 192.168.0.107
```

We can see both the VMs are able to communicate to each other, refer the below screenshot -



```
VM-001 [Running]
Feb 11 9:58 PM
ssingh@vbox:~ - ping 192.168.0.108

link/ether 08:00:27:f8:be:a7 brd ff:ff:ff:ff:ff:ff
  altname enx080027f8bea7
  inet 192.168.0.107/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s8
    valid_lft 86157sec preferred_lft 86157sec
  inet6 fe80::a00:27ff:fe8:bea7/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
bash-5.2$ ping 192.168.0.108
PING 192.168.0.108 (192.168.0.108) 56(84) bytes of data.
64 bytes from 192.168.0.108: icmp_seq=1 ttl=64 time=0.430 ms
64 bytes from 192.168.0.108: icmp_seq=2 ttl=64 time=0.331 ms
64 bytes from 192.168.0.108: icmp_seq=3 ttl=64 time=0.333 ms
64 bytes from 192.168.0.108: icmp_seq=4 ttl=64 time=0.378 ms
64 bytes from 192.168.0.108: icmp_seq=5 ttl=64 time=0.341 ms
64 bytes from 192.168.0.108: icmp_seq=6 ttl=64 time=0.841 ms
64 bytes from 192.168.0.108: icmp_seq=7 ttl=64 time=0.389 ms
64 bytes from 192.168.0.108: icmp_seq=8 ttl=64 time=0.406 ms
64 bytes from 192.168.0.108: icmp_seq=9 ttl=64 time=0.285 ms
64 bytes from 192.168.0.108: icmp_seq=10 ttl=64 time=0.327 ms
64 bytes from 192.168.0.108: icmp_seq=11 ttl=64 time=0.330 ms
64 bytes from 192.168.0.108: icmp_seq=12 ttl=64 time=0.440 ms
64 bytes from 192.168.0.108: icmp_seq=13 ttl=64 time=0.377 ms
64 bytes from 192.168.0.108: icmp_seq=14 ttl=64 time=0.417 ms
64 bytes from 192.168.0.108: icmp_seq=15 ttl=64 time=0.343 ms
64 bytes from 192.168.0.108: icmp_seq=16 ttl=64 time=0.357 ms
64 bytes from 192.168.0.108: icmp_seq=17 ttl=64 time=0.316 ms
64 bytes from 192.168.0.108: icmp_seq=18 ttl=64 time=0.380 ms
64 bytes from 192.168.0.108: icmp_seq=19 ttl=64 time=0.372 ms
64 bytes from 192.168.0.108: icmp_seq=20 ttl=64 time=0.367 ms
64 bytes from 192.168.0.108: icmp_seq=21 ttl=64 time=0.338 ms
64 bytes from 192.168.0.108: icmp_seq=22 ttl=64 time=0.511 ms
64 bytes from 192.168.0.108: icmp_seq=23 ttl=64 time=0.323 ms
64 bytes from 192.168.0.108: icmp_seq=24 ttl=64 time=0.383 ms
64 bytes from 192.168.0.108: icmp_seq=25 ttl=64 time=0.364 ms
64 bytes from 192.168.0.108: icmp_seq=26 ttl=64 time=0.378 ms
64 bytes from 192.168.0.108: icmp_seq=27 ttl=64 time=0.307 ms
64 bytes from 192.168.0.108: icmp_seq=28 ttl=64 time=0.354 ms
```

VM-005 [Running]

Feb 11 9:58 PM

ssingh@vbox:~ - ping 192.168.0.107

```

bash-5.2$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:07:1d:37 brd ff:ff:ff:ff:ff:ff
    altname enx080027071d37
    inet 192.168.0.108/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s8
        valid_lft 86190sec preferred_lft 86190sec
    inet6 fe80::a00:27ff:fe07:1d37/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
bash-5.2$ ping 192.168.0.107
PING 192.168.0.107 (192.168.0.107) 56(84) bytes of data.
64 bytes from 192.168.0.107: icmp_seq=1 ttl=64 time=0.676 ms
64 bytes from 192.168.0.107: icmp_seq=2 ttl=64 time=0.336 ms
64 bytes from 192.168.0.107: icmp_seq=3 ttl=64 time=0.362 ms
64 bytes from 192.168.0.107: icmp_seq=4 ttl=64 time=0.328 ms
64 bytes from 192.168.0.107: icmp_seq=5 ttl=64 time=0.362 ms
64 bytes from 192.168.0.107: icmp_seq=6 ttl=64 time=0.349 ms
64 bytes from 192.168.0.107: icmp_seq=7 ttl=64 time=0.379 ms
64 bytes from 192.168.0.107: icmp_seq=8 ttl=64 time=0.352 ms
64 bytes from 192.168.0.107: icmp_seq=9 ttl=64 time=0.346 ms
64 bytes from 192.168.0.107: icmp_seq=10 ttl=64 time=0.312 ms
64 bytes from 192.168.0.107: icmp_seq=11 ttl=64 time=0.394 ms
64 bytes from 192.168.0.107: icmp_seq=12 ttl=64 time=0.388 ms
64 bytes from 192.168.0.107: icmp_seq=13 ttl=64 time=0.868 ms
64 bytes from 192.168.0.107: icmp_seq=14 ttl=64 time=0.292 ms
64 bytes from 192.168.0.107: icmp_seq=15 ttl=64 time=0.340 ms
64 bytes from 192.168.0.107: icmp_seq=16 ttl=64 time=0.423 ms
64 bytes from 192.168.0.107: icmp_seq=17 ttl=64 time=0.364 ms
64 bytes from 192.168.0.107: icmp_seq=18 ttl=64 time=0.367 ms
64 bytes from 192.168.0.107: icmp_seq=19 ttl=64 time=0.348 ms
64 bytes from 192.168.0.107: icmp_seq=20 ttl=64 time=0.437 ms
64 bytes from 192.168.0.107: icmp_seq=21 ttl=64 time=0.352 ms
64 bytes from 192.168.0.107: icmp_seq=22 ttl=64 time=0.355 ms

```

Left

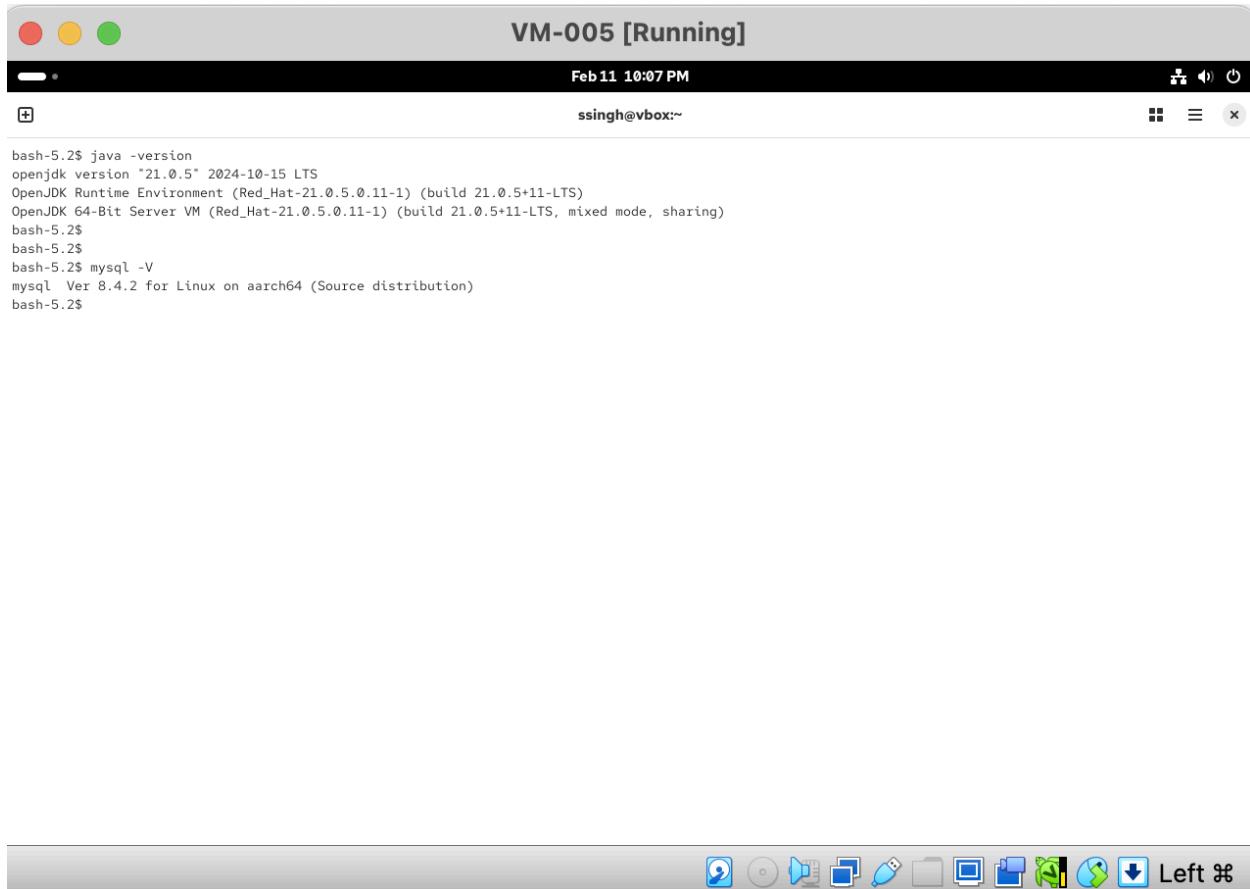
6. Installation of software needed -

Since I am deploying java based microservice application, below are the software needed -

- Java 21 (sudo yum install openjdk-21-jdk -y)**
- Mysql (sudo yum install mysql)**

I would be needing a database to store the data for rest service, hence I am using mysql for the same.

Below screenshot shows, java and mysql installation details -



```
VM-005 [Running]
Feb 11 10:07 PM
ssingh@vbox:~
```

```
bash-5.2$ java -version
openjdk version "21.0.5" 2024-10-15 LTS
OpenJDK Runtime Environment (Red Hat-21.0.5.0.11-1) (build 21.0.5+11-LTS)
OpenJDK 64-Bit Server VM (Red Hat-21.0.5.0.11-1) (build 21.0.5+11-LTS, mixed mode, sharing)
bash-5.2$
bash-5.2$ mysql -V
mysql Ver 8.4.2 for Linux on aarch64 (Source distribution)
bash-5.2$
```

7. Application deployment -

Application deployment contains two part -

Database setup -

Below are the script to setup database -

Login to mysql console -

```
mysql -u root -p
```

Create database and tables -

```
CREATE DATABASE `order_management`;
```

```
use database order_management;
```

Creating the table -

```

CREATE TABLE `customers` (
`id` bigint NOT NULL AUTO_INCREMENT,
`email` varchar(255) DEFAULT NULL,
`name` varchar(255) DEFAULT NULL,
`contact_no` varchar(255) DEFAULT NULL,
`address` varchar(255) DEFAULT NULL,
PRIMARY KEY (`id`)
)

```

Now, i want the database to be running in one vm and microservice running in another VM, for that I need to execute below script to enable mysql to be connected from different hosts -

```

mysql>
mysql> CREATE USER 'cloud_01'@'192.168.0.108' IDENTIFIED BY 'Phar@123';
Query OK, 0 rows affected (0.03 sec)
mysql> GRANT ALL PRIVILEGES ON *.* TO 'cloud_01'@'192.168.0.108' WITH GRANT OPTION;
Query OK, 0 rows affected (0.02 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
mysql> SELECT user, host FROM mysql.user;
+-----+-----+
| user      | host       |
+-----+-----+
| ssingh    | 192.168.0.102 |
| cloud_01  | 192.168.0.108 |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys    | localhost |
| root        | localhost |
+-----+-----+
6 rows in set (0.01 sec)

```

After running these steps, my VM-005 can call the database setup on VM-001

Microservice deployment -

I have created a java spring boot microservice, which contains rest services for order management. I will explain the details in the architecture diagram.

Once the code is done and I am able to successfully build the application in my local machine, I moved the executable jar to VM and deployed the jar.

For the deployment purpose I am simply running it as an executable jar not creating any docker image, since this is just to test the basic feature hence I chose this way.

Below is the command to build the executable jar in local machine -

mvn clean package

This will generate the executable jar. Below is the screenshot for successful build -

The screenshot shows a Java IDE interface with the following details:

- Project View:** Shows the project structure under "springboot-microservice [microservice]". It includes a ".idea" folder, a "src" folder containing "main" (with "java", "com.example.microservice" (containing "controller", "entity", "Customer", and "repository"), and "application.properties").
- Code Editor:** Displays the "application.properties" file with the following configuration:

```
spring.datasource.url=jdbc:mysql://192.168.0.107:3306/order_management
spring.datasource.username=cloud_01
spring.datasource.password=Phar@123
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto:update
```
- Terminal:** Shows the command-line output of the Maven build:

```
[INFO] --- spring-boot:3.1.0:repackage (repackage) @ microservice ---
[INFO] Replacing main artifact /Users/sachchidanandsingh/Documents/cloud_computing/springboot-microservice/target/microservice-0.0.1-SNAPSHOT.jar with repackaged archive
, adding nested dependencies in BOOT-INF.
[INFO] The original artifact has been renamed to /Users/sachchidanandsingh/Documents/cloud_computing/springboot-microservice/target/microservice-0.0.1-SNAPSHOT.jar.origi
nal
[INFO]
[INFO] --- spring-boot:3.1.0:repackage (default) @ microservice ---
[INFO] Replacing main artifact /Users/sachchidanandsingh/Documents/cloud_computing/springboot-microservice/target/microservice-0.0.1-SNAPSHOT.jar with repackaged archive
, adding nested dependencies in BOOT-INF.
[INFO] The original artifact has been renamed to /Users/sachchidanandsingh/Documents/cloud_computing/springboot-microservice/target/microservice-0.0.1-SNAPSHOT.jar.origi
nal
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.244 s
[INFO] Finished at: 2025-02-11T20:07:43+05:30
[INFO] -----
```

Once the build is successfully done, we need to push the jar to VM where we want to deploy this.

Using scp command we can transfer the jar to desired location -

```
springboot-microservice % scp -P 22
/Users/sachchidanandsingh/Documents/cloud_computing/springboot-microservice/tar
get/microservice-0.0.1-SNAPSHOT.jar
ssingh@192.168.0.108:/home/ssingh/Documents/cloud_computing/Project
```

Once teh jar is transferred to the Vm we can run the jar with below command -

java -jar microservice-0.0.1-SNAPSHOT.jar

This command will run the spring boot application and exposes all the endpoints.

Below is the screenshot for running logs -

The screenshot shows a terminal window with the following details:

- Terminal Title:** ssingh@localhost:~/Documents/cloud_computing/Project\$
- Command:** java -jar microservice-0.0.1-SNAPSHOT.jar
- Logs:** The log output is displayed in a monospaced font, showing the application's startup process, configuration, and database connection setup. Key entries include:
 - INFO 3305 --- [main] c.e.m.MicroserviceApplication : Starting MicroserviceApplication v0.0.1-SNAPSHOT using Java 21.0.5 with PID 3305 (/home/ssingh/Documents/cloud_computing/Project/microservice-0.0.1-SNAPSHOT.jar started by ssingh in /home/ssingh/Documents/cloud_computing/Project)
 - No active profile set, falling back to 1 default profile: "default"
 - Bootstrapping Spring Data JPA repositories in DEFAULT mode.
 - Finished Spring Data repository scanning in 20 ms. Found 1 JPA repository interfaces.
 - Tomcat initialized with port(s): 8080 (http)
 - Starting service [Tomcat]
 - Starting Servlet engine: [Apache Tomcat/10.1.8]
 - Initializing Spring embedded WebApplicationContext
 - Root WebApplicationContext: initialization completed in 866 ms
 - HHH000204: Processing PersistenceUnitInfo [name: default]
 - HHH000412: Hibernate ORM core version 6.2.2.Final
 - HHH000406: Using bytecode reflection optimizer
 - HHH00021: Bytecode provider name : bytebuddy
 - No LoadTimeWeaver setup: ignoring JPA class transformer
 - HikariPool-1 - Starting...
 - HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionImpl@1473b8c0
 - HikariPool-1 - Start completed.
 - HHH035001: Using dialect: org.hibernate.dialect.MySQLDialect, version: 8.0
 - HHH0000026: MySQLDialect has been deprecated; use org.hibernate.dialect.MySQLDialect instead
 - HHH00021: Bytecode provider name : bytebuddy
 - HHH000498: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
 - Initialized JPA EntityManagerFactory for persistence unit 'default'
 - JpaBaseConfiguration\$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
 - Tomcat started on port(s): 8080 (http) with context path ''
 - Started MicroserviceApplication in 3.039 seconds (process running for 3.373)
 - Initializing Spring DispatcherServlet 'dispatcherServlet'
 - Initializing Servlet 'dispatcherServlet'
 - Completed initialization in 2 ms
 - Closing JPA EntityManagerFactory for persistence unit 'default'
 - HikariPool-1 - Shutdown initiated...

Now the application is running and we can test the apis.

8. Testing -

For testing I have used Postman on my macbook, which will connect to VM as I have made the network settings in such a way that it can be connected.

Api specs -

1. GET API to get the list of customers -

API type	GET
URL	http://192.168.0.108:8080/customers
Sample output	[{ "id": 1, "name": "Sachchida", "email": "g24ai1048@iitj.ac.in", "contact_no": "9999999999", "address": "JP nagar, bangalore" }, { "id": 2, "name": "Sachchida", "email": "g24ai1048@iitj.ac.in", "contact_no": "9999999999", "address": "JP nagar, bangalore" }]

2. POST API to create a customer -

API type	POST
Url	http://192.168.0.108:8080/customers
Content-type	application/json
Request body	{ "name": "Sachchida", "email": "g24ai1048@iitj.ac.in", "contact_no": "9999999999", "address": "JP nagar, bangalore" }

Below is the screenshot of Postman -

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing 'Collections' (Assignment), 'Environments' (aws, GCP), 'Flows', and 'History'. The main area displays an 'Assignment' collection with a 'POST http://localhost:8080/orders' and a 'GET http://localhost:8080/orders' endpoint. A specific POST request is selected, targeting 'http://192.168.0.108:8080/customers'. The 'Headers' tab is active, showing a Content-Type header set to 'application/json'. Below the headers, the 'Body' tab is selected, showing a JSON response with the following content:

```
1 {  
2   "id": 2,  
3   "name": "Sachchida",  
4   "email": "g24ai1048@iitj.ac.in",  
5   "contact_no": "9999999999",  
6   "address": "JP nagar, bangalore"  
7 }
```

The response status is '200 OK' with a duration of '364 ms' and a size of '280 B'. At the bottom, there are various navigation and utility buttons like Postbot, Runner, Start Proxy, Cookies, Vault, Trash, etc.

Home Workspaces API Network

My Workspace

New Import

Overview POST http://localhost:8081 GET http://localhost:8080/Assignment / http://localhost:8080/orders

HTTP Assignment / http://localhost:8080/orders

Save Share

POST http://localhost:8080/orders

GET http://localhost:8080/orders

Params Auth Headers (7) Body Scripts Tests Settings Cookies

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

Body 200 OK 112 ms 399 B

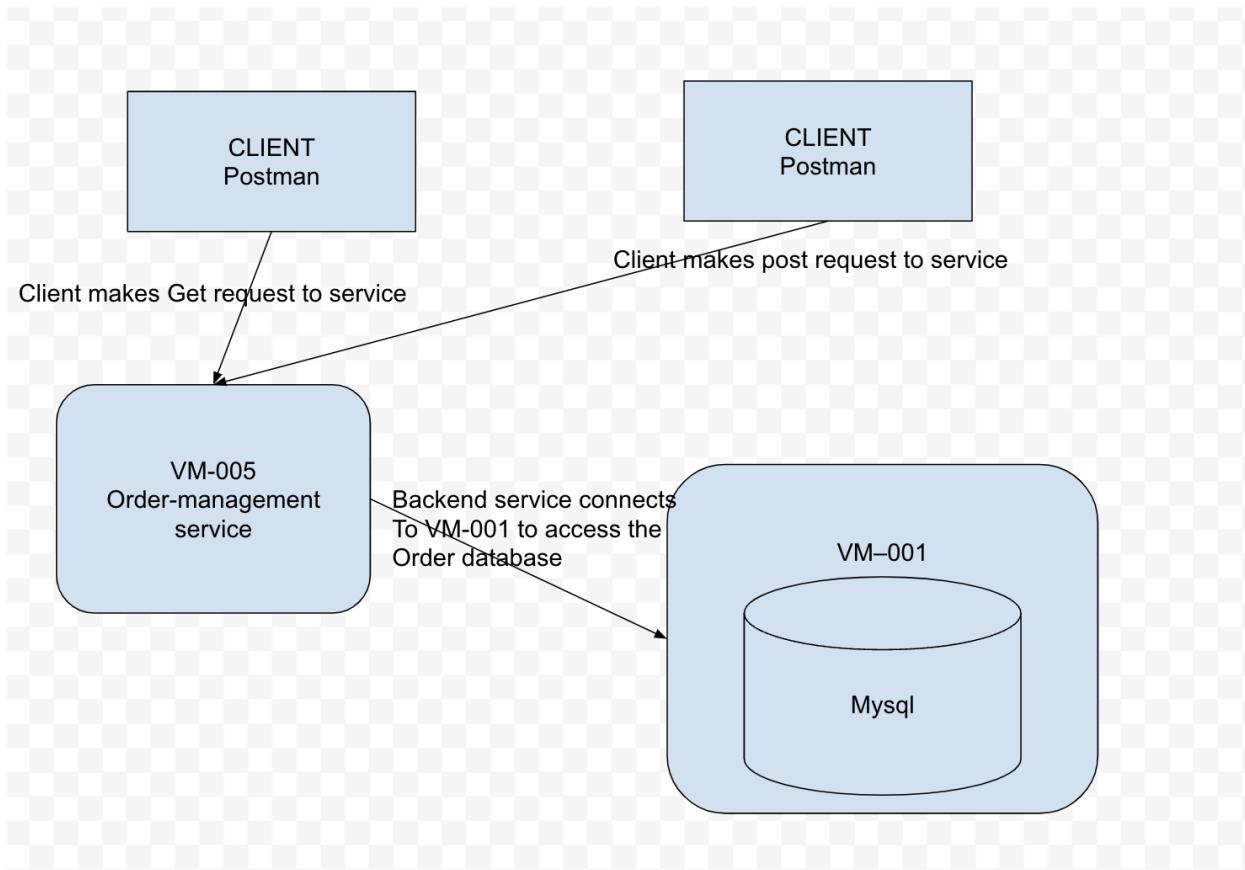
Pretty Raw Preview Visualize JSON

```
1 [  
2 {  
3   "id": 1,  
4   "name": "Sachchida",  
5   "email": "g24ai1048@iitj.ac.in",  
6   "contact_no": "9999999999",  
7   "address": "JP nagar, bangalore"  
8 },  
9 {  
10   "id": 2,  
11   "name": "Sachchida",  
12   "email": "g24ai1048@iitj.ac.in",  
13   "contact_no": "9999999999",  
14   "address": "JP nagar, bangalore"  
15 }
```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

9. Architecture Diagram -



9. Github link -

Below is the Github link where code is pushed -

<https://github.com/singhsn/spring-microservice/tree/main>

I have exported the VM configuration file but the size is 5.32 GB, I won't be able to upload it on Github. I am sharing the details here -

appliance.ova	21 Oct 2024 at 10:55 PM	0 KB	Printable...archive
Appliance.ova	Today at 10:57 PM	5.32 GB	Open V...Archive

Video Recording -

The recording is broken, I will update in sometime.

<https://www.youtube.com/watch?v=4oBOM0h9V6s>

