

## **Three-tier Architecture Project**    ([Giturl](#))

### **Frontend (Presentation Layer)**

- **What it is:** The user interface (UI) — what users see and interact with.
- **Built with:** Nginx
- **Responsibilities:**
  - Display data to the user
  - Capture user input (forms, buttons, navigation)
  - Make API requests to the backend

### **Backend (Application Layer)**

- **What it is:** The logic layer — handles requests, processes data, enforces rules.
- **Built with:** Tomcat
- **Responsibilities:**
  - Receives and processes frontend requests
  - Validates data
  - Applies business logic
  - Talks to the database
  - Handles authentication, authorization, sessions

### **Database (Data Layer)**

- **What it is:** Where data is stored and queried.
- **Types:**
  - SQL: MySQL
- **Responsibilities:**
  - Store persistent data (users, products, orders)
  - Enforce data constraints
  - Serve data to backend via queries

Tree structure for 3-tier architecture with a list of

### 1.Resource Group

In cloud computing (especially in platforms like Microsoft Azure, AWS, and Google Cloud), a Resource Group is a logical container that holds related resources for a specific project, application, or workload.

### 2.Virtual Network

A Virtual Network (often called VNet in Azure or VPC in AWS) is a private, isolated network in the cloud that allows resources (like VMs, databases, and apps) to securely communicate with each other, the internet, and on-premises networks.

### 3.Subnets

A subnet (short for subnetwork) is a smaller segment of a virtual network (VNet or VPC) that divides the network into manageable and secure sections.

### 4.Virtual Machines

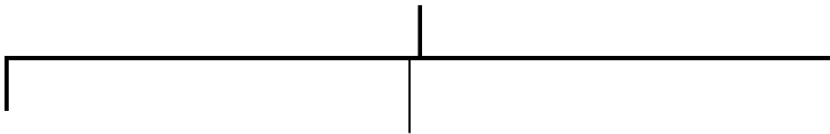
A Virtual Machine (VM) is a software-based simulation of a physical computer. It runs an operating system (like Windows or Linux) and applications just like a real machine, but it's hosted on a physical server using virtualization technology.



Resource Group



Virtual Network



Subnet01



VM01



Subnet02



VM02



Subnet03



VM03

Procedure:

1.Create a Resource group in any region with good name

## Create a resource group ...

Basics   Tags   Review + create

**Resource group** - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

Subscription * ⓘ	Azure for Students
Resource group name * ⓘ	3-tier-RGC
Region * ⓘ	(US) East US

2.Create a Virtual Network with same of resource group region

## Create virtual network ...

Basics   Security   IP addresses   Tags   Review + create

Resource group *	3-tier-RGC
------------------	------------

[Create new](#)

### Instance details


Virtual network name *	3-tier-VNC
Region * ⓘ	(US) East US

[Deploy to an Azure Extended Zone](#)

### 3. With in the Virtual Network create three subnets

- a) Frontendsub
- b) Backendsubnet
- c) DBsubnet

[Home](#) > [3-tier-VNC](#)

 **3-tier-VNC | Subnets** ☆ ...  
Virtual network


[+ Subnet](#) [Refresh](#) | [Manage users](#) [Delete](#)

Create subnets to segment the virtual network address space into smaller ranges for use by your applications. When you d subnet.

<input type="checkbox"/>	Name ↑	IPv4	IPv6	Available IPs
<input type="checkbox"/>	Frontendsub	10.0.0.0/24	-	250
<input type="checkbox"/>	Backendsubnet	10.0.1.0/24	-	250
<input type="checkbox"/>	DBsubnet	10.0.2.0/24	-	250

### 4. Create Virtual Machine with in the Frontend subnet

#### Create a virtual machine ...

 [Help me create a low cost VM](#) [Help me create a VM optimized for high availability](#) [Help me choose the right VM size for my workload](#)

Resource group \* ⓘ

3-tier-RGC  
[Create new](#)

Instance details

Virtual machine name \* ⓘ

FrontVM ✓

Region \* ⓘ

(US) East US


Availability options ⓘ

No infrastructure redundancy required

Security type ⓘ

Standard

Image \* ⓘ

 Ubuntu Server 24.04 LTS - x64 Gen2  
[See all images](#) | [Configure VM generation](#)

< Previous

Next : Disks >

Review + create

## Create a virtual machine ...



Help me create a low cost VM

Help me create a VM optimized for high availability

Help me choose the right VM size for my workload

When creating a virtual machine, a network interface will be created for you.

Virtual network \* ⓘ

3-tier-VNC

[Create new](#)

Subnet \* ⓘ

Frontendsub (10.0.0.0/24)

[Manage subnet configuration](#)

Public IP ⓘ

(new) FrontVM-ip

[Create new](#)

NIC network security group ⓘ

☐ None

☒ Basic

☐ Advanced



By using the public ip of Frontvm connected to the Mobaxterm and apply the following commands

sudo su

apt update

apt install nginx -y (installation of nginx)

nginx -version

```
root@FrontVM:/home/Chaithu# nginx -version
nginx version: nginx/1.24.0 (Ubuntu)
root@FrontVM:/home/Chaithu#
```

Now add the inbound security rule in

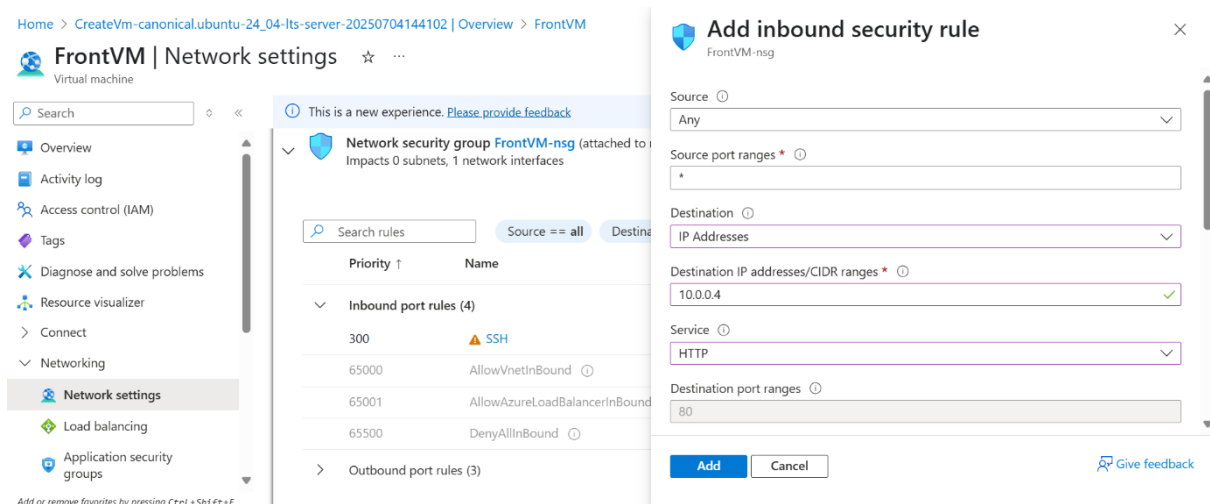
Source : Any

Destination : Ip Address of FrontVm

Service : HTTP

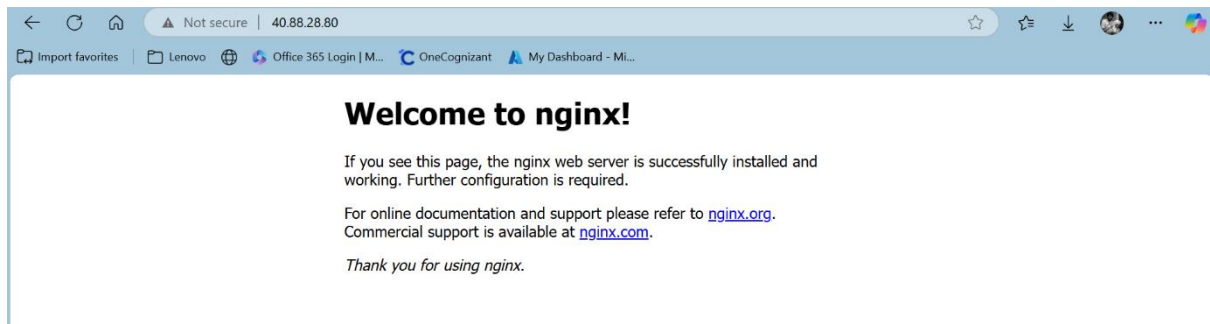
Dport : 80

Action : Allow



Now check the whether the server is up and run or not by using the


Browse the Public IP address of Frontvm it shows like that



## 5. Create another Virtual Machine with in the Backend subnet

[Home](#) > [Compute infrastructure | Virtual machines](#) >

### Create a virtual machine ...

 [Help me create a low cost VM](#) [Help me create a VM optimized for high availability](#) [Help me choose the right VM size for my workload](#)

Resource group \* ⓘ

3-tier-RGC

[Create new](#)

Instance details

Virtual machine name \* ⓘ

BackendVM

Region \* ⓘ

(US) East US


Availability options ⓘ

No infrastructure redundancy required






Security type ⓘ

Standard

Image \* ⓘ


 Ubuntu Server 24.04 LTS - x64 Gen2

[See all images](#) | [Configure VM generation](#)

 **Microsoft Azure**     

[Home](#) > [Compute infrastructure | Virtual machines](#) >

### Create a virtual machine ...

 [Help me create a low cost VM](#) [Help me create a VM optimized for high availability](#) [Help me choose the right VM size for my workload](#)

Virtual network \* ⓘ

3-tier-VNC

[Create new](#)

Subnet \* ⓘ

Backendsubnet (10.0.1.0/24)

[Manage subnet configuration](#)

Public IP ⓘ

(new) BackendVM-ip

[Create new](#)

NIC network security group ⓘ

☐ None

☒ Basic

☐ Advanced

Public inbound ports \* ⓘ

☐ None

☒ Allow selected ports

< Previous

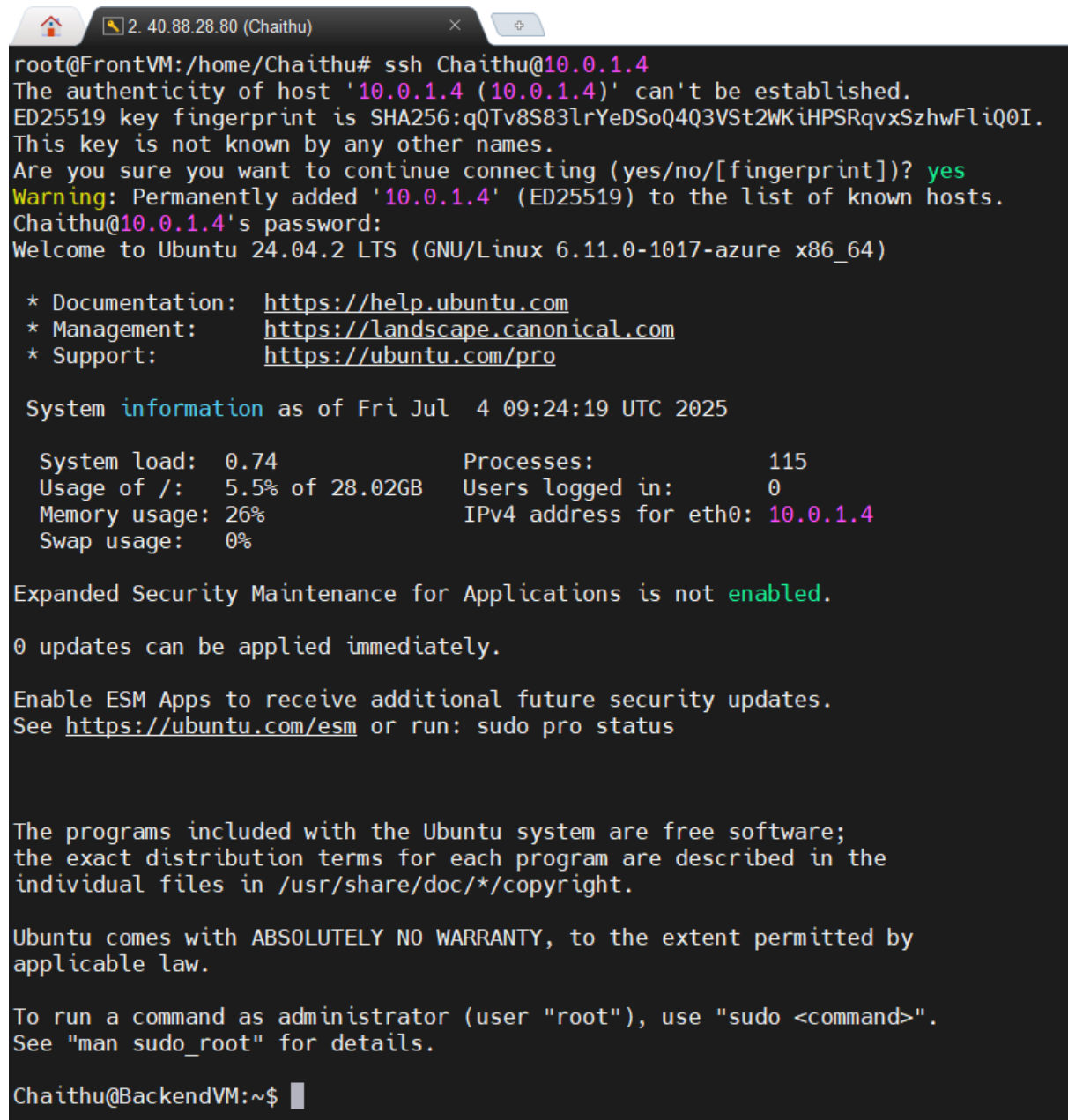
Next : Management >

Review + create



And we connect to this machine by using the following commands

```
ssh username@ip address of Backendvm
```



```
root@FrontVM:/home/Chaithu# ssh Chaithu@10.0.1.4
The authenticity of host '10.0.1.4 (10.0.1.4)' can't be established.
ED25519 key fingerprint is SHA256:qQTV8S83lrYeDSOQ4Q3VSt2WKiHPSRqvxSzhwFliQ0I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.4' (ED25519) to the list of known hosts.
Chaithu@10.0.1.4's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.11.0-1017-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Jul  4 09:24:19 UTC 2025

System load:  0.74           Processes:            115
Usage of /:   5.5% of 28.02GB Users logged in:        0
Memory usage: 26%           IPv4 address for eth0: 10.0.1.4
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Chaithu@BackendVM:~$
```

Install Tomcat by following steps

```
sudo su
```

```
apt update
```

```
apt install default-jdk -y (Tomcat requires Java. Install OpenJDK)
```

```
java -version (Verify the installation)
```

(Download Tomcat)

```
wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.43/bin/apache-tomcat-10.1.43.tar.gz
```

```
ls
```

(Extract and Configure Tomcat)

```
tar -xvzf apache-tomcat-10.1.43.tar.gz
```

```
ls
```

```
mv apache-tomcat-10.1.43 tomcat
```

```
cd /tomcat/bin (Navigate to the Tomcat Directory)
```

```
./startup.sh (Start Tomcat)
```

Now add the inbound security rule in

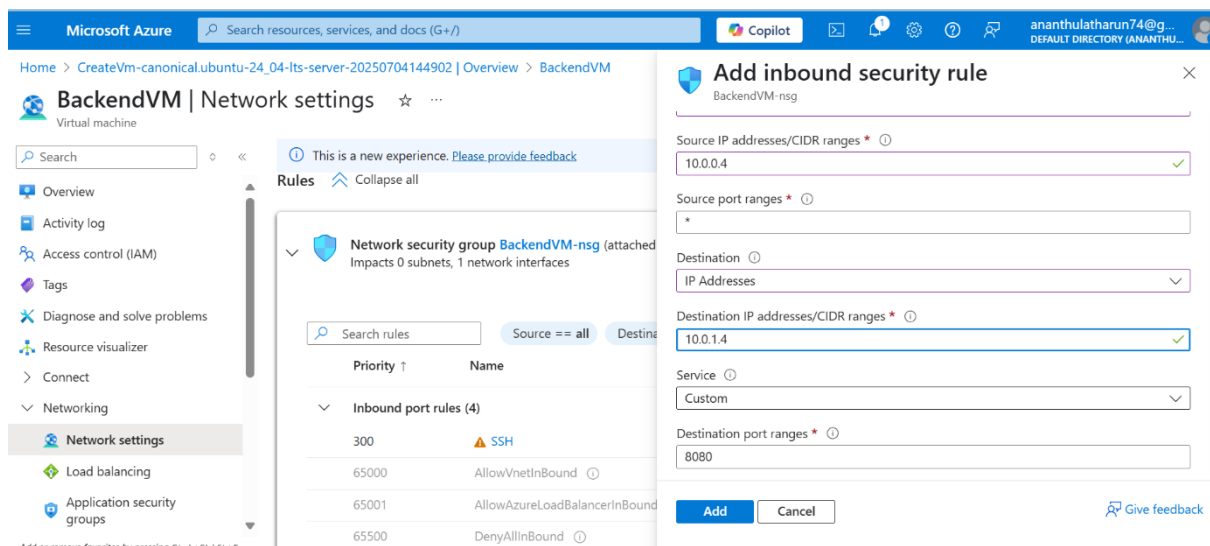
Source : Ip Address of Frontendvm

Destination : Ip Address of Backendvm

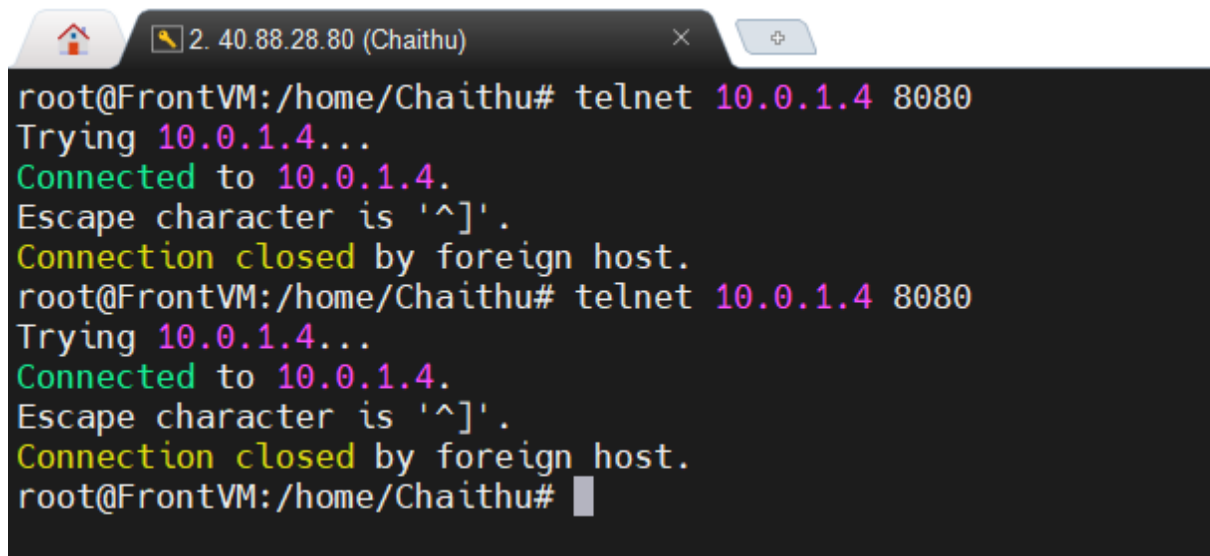
Service : Custom

Dport : 8080

Action : Allow



Now check whether it is connected or not by using the telnet





```
root@FrontVM:/home/Chaithu# telnet 10.0.1.4 8080
Trying 10.0.1.4...
Connected to 10.0.1.4.
Escape character is '^]'.
Connection closed by foreign host.
root@FrontVM:/home/Chaithu# telnet 10.0.1.4 8080
Trying 10.0.1.4...
Connected to 10.0.1.4.
Escape character is '^]'.
Connection closed by foreign host.
root@FrontVM:/home/Chaithu#
```

## 6. Create another Virtual Machine with in the DBsubnet

[Home](#) > [Compute infrastructure](#) | [Virtual machines](#) >

### Create a virtual machine ...

 [Help me create a low cost VM](#) [Help me create a VM optimized for high availability](#) [Help me choose the right VM size for my workload](#)

 Resource group \* ⓘ

3-tier-RGC

▼

[Create new](#)

Instance details

Virtual machine name \* ⓘ

DBVM

✓

Region \* ⓘ

(US) East US

▼

Availability options ⓘ

No infrastructure redundancy required


▼

Security type ⓘ

Standard

▼


Image \* ⓘ


 Ubuntu Server 24.04 LTS - x64 Gen2


▼

[See all images](#) | [Configure VM generation](#)

## Create a virtual machine ...

 Help me create a low cost VM

 Help me create a VM optimized for high availability

 Help me choose the right VM size for my workload

Virtual network \* ⓘ

3-tier-VNC

[Create new](#)

Subnet \* ⓘ

DBsubnet (10.0.2.0/24)

[Manage subnet configuration](#)

Public IP ⓘ

(new) DBVM-ip

[Create new](#)

NIC network security group ⓘ

☐ None

☒ Basic

☐ Advanced

Public inbound ports \* ⓘ

☐ None

☒ Allow selected ports

And we connect to this machine by using the following commands

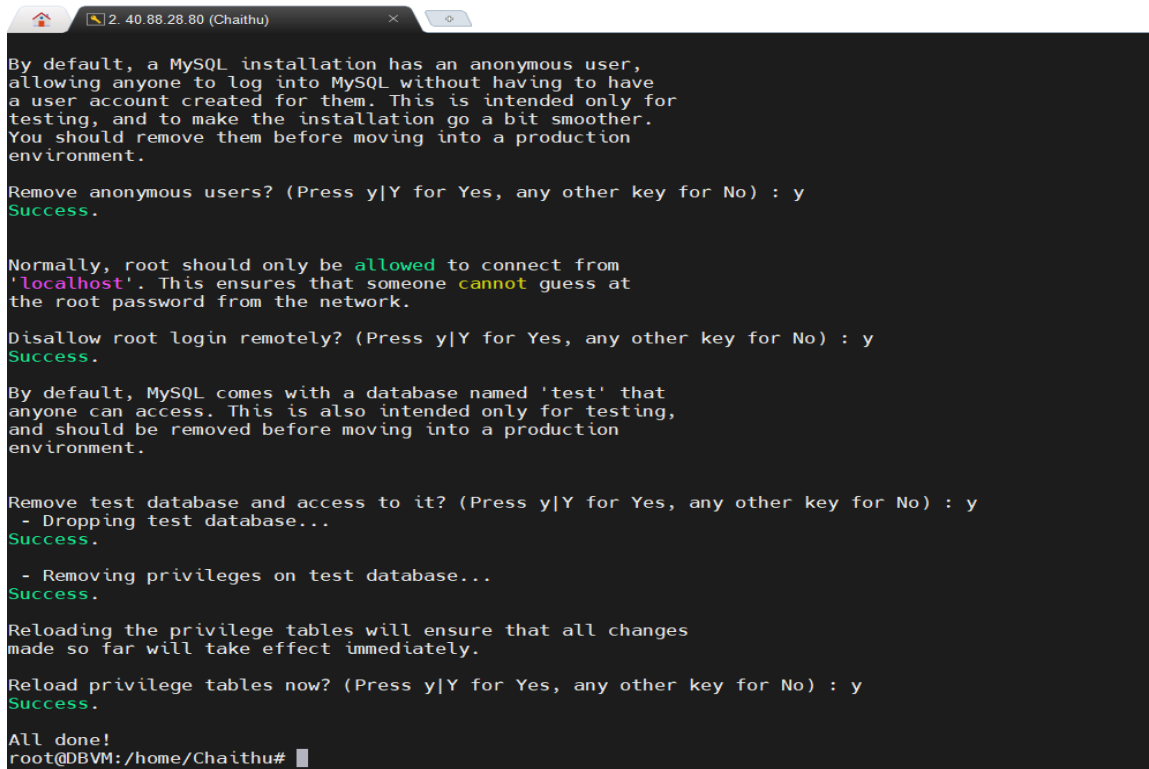
```
ssh username@ip address of DBVM
```

Install MYSQL by following steps

```
sudo su
```

```
apt update
```

```
apt install mysql-server -y (install MySQL)
```



```
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
root@DBVM:/home/Chaithu#
```

mysql\_secure\_installation (To improve security, run the following command)

```
root@DBVM:/home/Chaithu# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary      file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.
```

nano /etc/mysql/mysql.conf.d/mysqld.cnf (Edit the MySQL configuration file using a text editor)

bind-address = 0.0.0.0

Save and exit (CTRL + X, then Y, then Enter)

systemctl restart mysql

systemctl status mysql

Now add the inbound security rule in

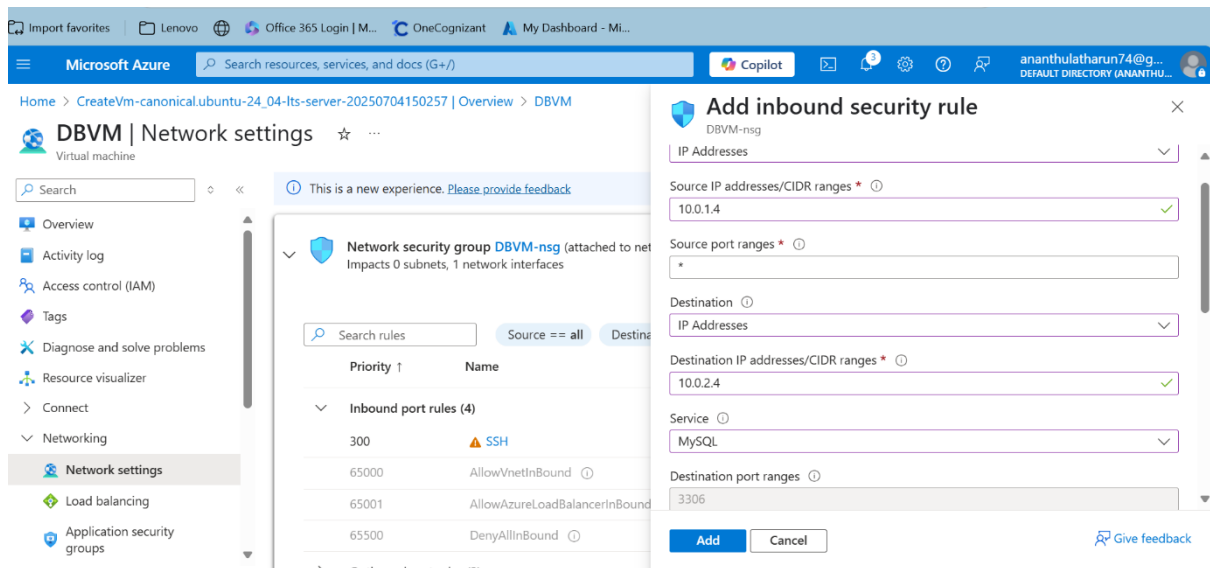
Source : Ip Address of Backendvm

Destination : Ip Address of DBvm

Service : MYSQL

Dport : 3306

Action : Allow



Now add another inbound security rule in

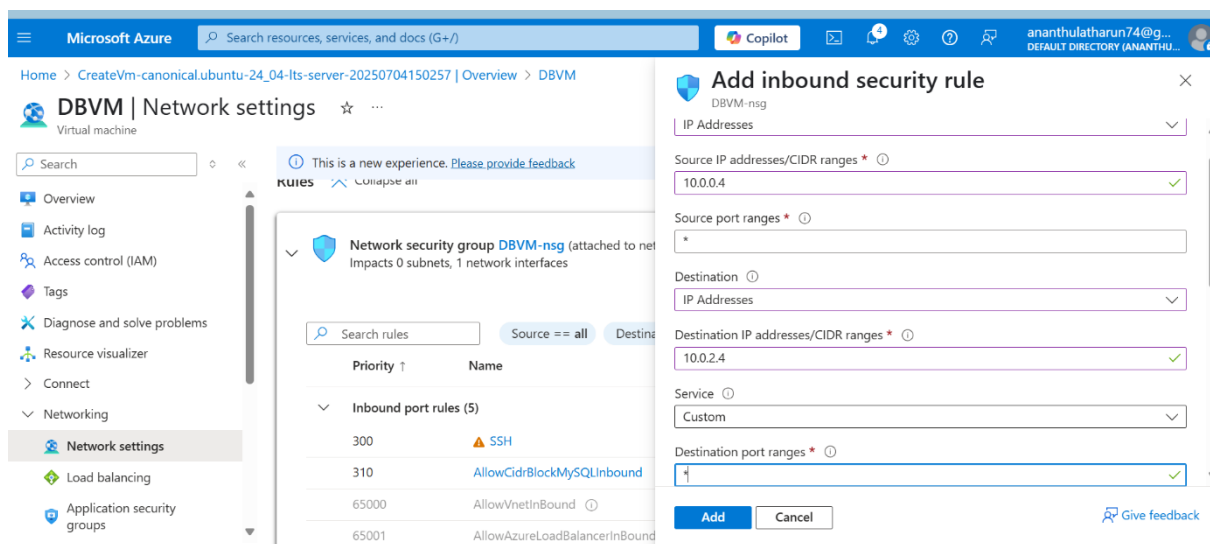
Source : Ip Address of Frontendvm

Destination : Ip Address of DBvm

Service : custom

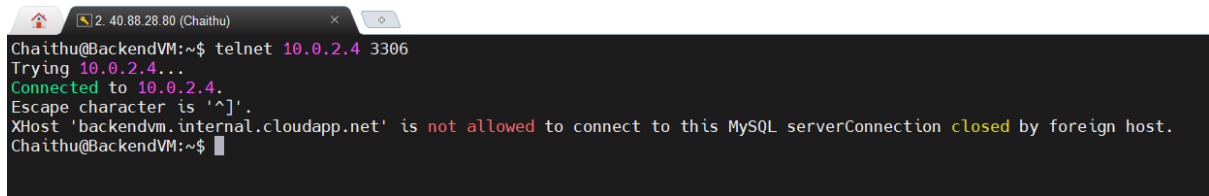
Dport : \*

Action : Deny



ssh username@ip address of Backendvm

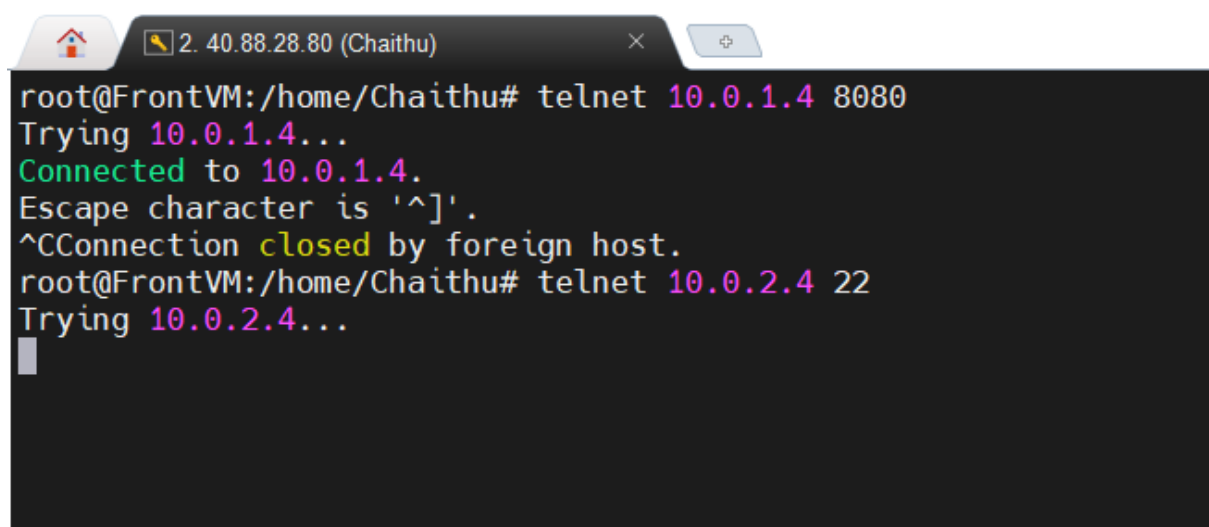
check whether it is connected or not by using telnet

A terminal window titled '2. 40.88.28.80 (Chaithu)' showing a telnet session. The user 'Chaithu@BackendVM' runs 'telnet 10.0.2.4 3306'. The output shows a successful connection to 10.0.2.4, but then an error: 'XHost 'backendvm.internal.cloudapp.net' is not allowed to connect to this MySQL serverConnection closed by foreign host.'

```
Chaithu@BackendVM:~$ telnet 10.0.2.4 3306
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
XHost 'backendvm.internal.cloudapp.net' is not allowed to connect to this MySQL serverConnection closed by foreign host.
Chaithu@BackendVM:~$
```

ssh username@ip address of Frontendvm

now check the app and db connected or not by using telnet

A terminal window titled '2. 40.88.28.80 (Chaithu)' showing two telnet sessions from 'root@FrontVM:/home/Chaithu'. The first session connects to 10.0.1.4 on port 8080, but the connection is closed by the foreign host. The second session connects to 10.0.2.4 on port 22.

```
root@FrontVM:/home/Chaithu# telnet 10.0.1.4 8080
Trying 10.0.1.4...
Connected to 10.0.1.4.
Escape character is '^]'.
^CConnection closed by foreign host.
root@FrontVM:/home/Chaithu# telnet 10.0.2.4 22
Trying 10.0.2.4...
```

## OUTPUTS:

- First login into Frontendvm here backend is connected because it gives the following advantages.

### 1. Dynamic Content

Content can be fetched from a database and displayed dynamically.

### 2. Data Storage & Persistence

Backend allows storing user data (e.g., sign-ups, form inputs) in databases.

### 3. User Authentication & Authorization

Backend handles secure login, session management, and role-based access.

### 4. Scalability

A connected backend allows handling growing data, users, and traffic efficiently (especially with cloud-based infrastructure).

- Login into Backendvm here Db is connected because it gives the following advantages.

### 1. Data Persistence

Data remains stored even after a server restart or user closes the app.

### 2. Structured Data Storage

Databases (especially relational ones) organize data in tables with relationships.

### 3. Centralized Data Management

All clients (web, mobile, admin panel) can access and sync with the same backend and database.

### 4. Security & Access Control

Backend can control *who* can read/write to the database.

### 5. Data Integrity

Databases can enforce rules (e.g., no empty emails, unique usernames).

- Login into Frontend here it is not connected to DB because

### 1. Security Risk

Exposes database credentials (username, password) to anyone.

Allows potential attackers to:

- Modify or delete data
- Bypass access controls
- Execute SQL injection or NoSQL injection attacks

### 2. No Business Logic Control






The frontend can't enforce validation, permissions, or workflows.

You lose centralized control over how data is handled.

### 3. No Audit or Logging

You can't easily track *who* did *what* when the frontend bypasses the backend.

## Why Use Three-Tier Architecture?

Benefit	Description
 <b>Security</b>	Database is never exposed to users directly
 <b>Separation of Concerns</b>	Each layer has its own job — easier to debug and maintain
 <b>Scalability</b>	You can scale frontend, backend, and DB independently
 <b>Reusability</b>	Backend APIs can serve web, mobile, or other clients
 <b>Maintainability</b>	Changes in one tier (e.g., UI redesign) don't affect others