**Project-1**

**Three-tier architecture:** It is a software development model that separates an application into three logical tiers or layers:

1. Presentation layer (front-end).
2. Application layer (Middle-tier/Back-end)
3. Data Layer (Database/Storage-tier)
4. **Presentation tier**:

The user interface that allows users to interact with the application. This tier can be a web browser or a graphical user interface (GUI).

Handles user interactions (UI/UX).

**Examples:** Web applications, mobile apps, or a front-end hosted in Azure App Service, Azure Static Web Apps, or Azure Front Door.

1. **Application tier**:

The middle tier that handles the application's logic, enforces business rules, and performs calculations.

It acts as a bridge between the presentation and data tiers.

**Examples:** APIs or micro services deployed on Azure Functions, Azure Kubernetes Service (AKS), or Azure App Service (API Apps).

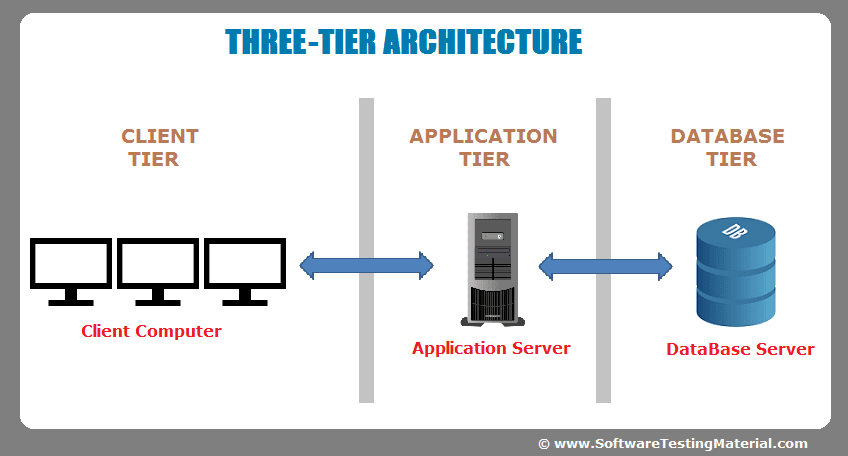
1. **Data tier**:

Stores and manages application data.

**Examples:** Azure SQL Database, Azure Cosmos DB, Azure Blob Storage, or Azure Data Lake.

Three-tier architecture is widely used in:

* **Web Applications**: E-commerce sites, social media platforms, and online banking systems often utilize this pattern.
* **Enterprise Applications**: Large-scale software systems for businesses, such as ERP or CRM systems.
* **Client-Server Applications**: Traditional applications where clients (e.g., desktop computers) interact with a central server.
* **Cloud-Based Applications**: Modern applications deployed on cloud platforms often follow this architecture.



**Fig:** Basic Block diagram of three tier architecture**.**

**NSG (network security group):**

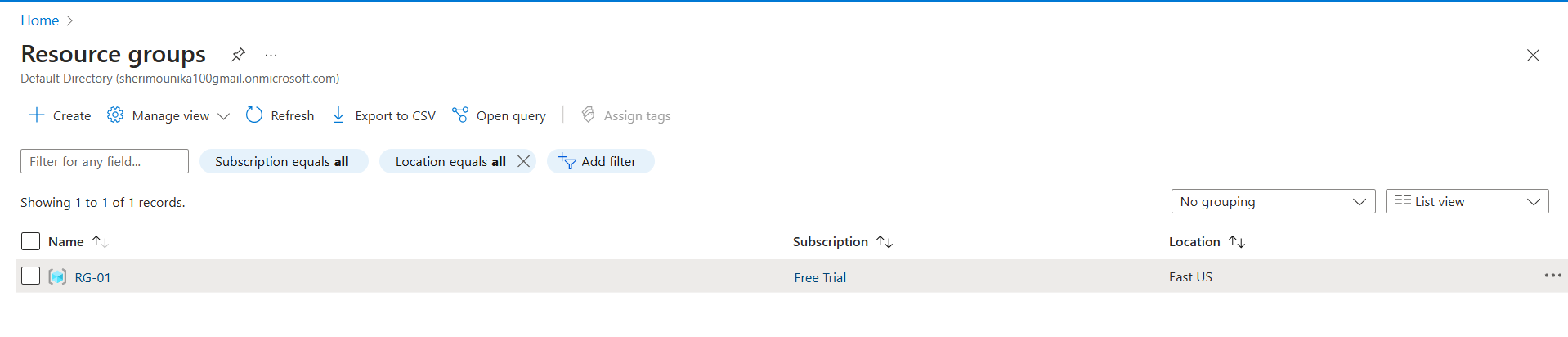
In Azure, an **NSG (Network Security Group)** is a fundamental security service that acts as a firewall for your virtual network resources. It's essentially a set of rules that control incoming and outgoing network traffic.

That means we can define rules within an NSG that specify whether to allow or deny traffic.

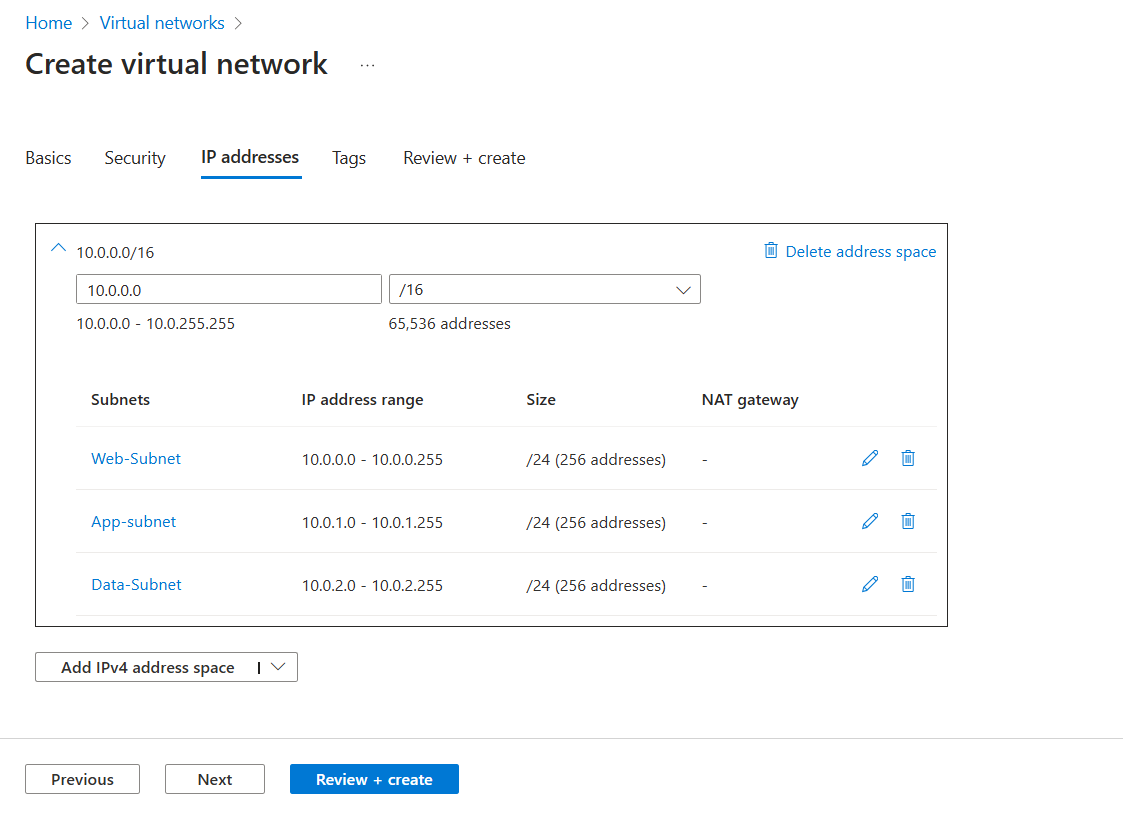
This NSG rules can be associated with NIC level or Subnet Level or both.

**Project Title:** Implementing of Three Tier architecture by install Nginx in web server, installing “Tomcat” in App server and “MySQL” in Data Server.

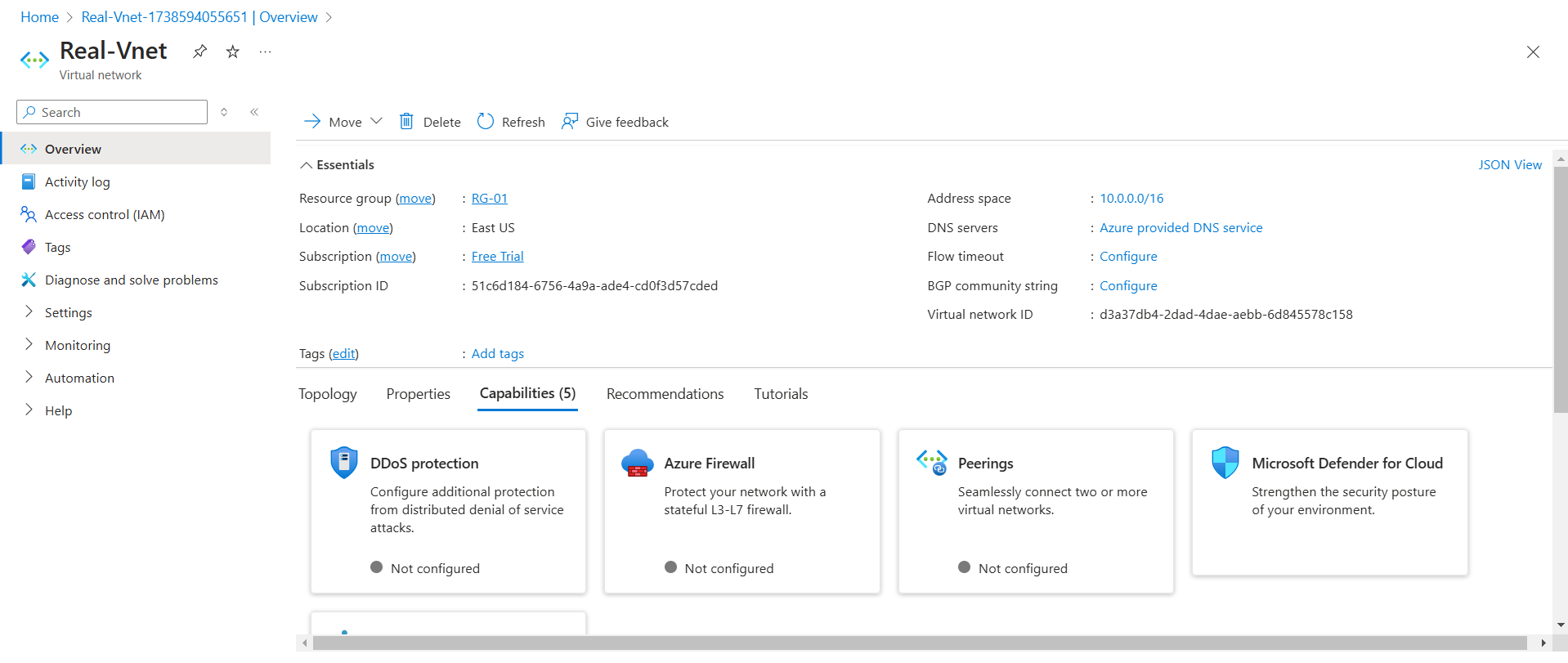
Step1: Create a resource group (RG-01).



**Step2:** Create the Virtual Network (Real-Vnet) with Three Subnets in it as shown below figure.

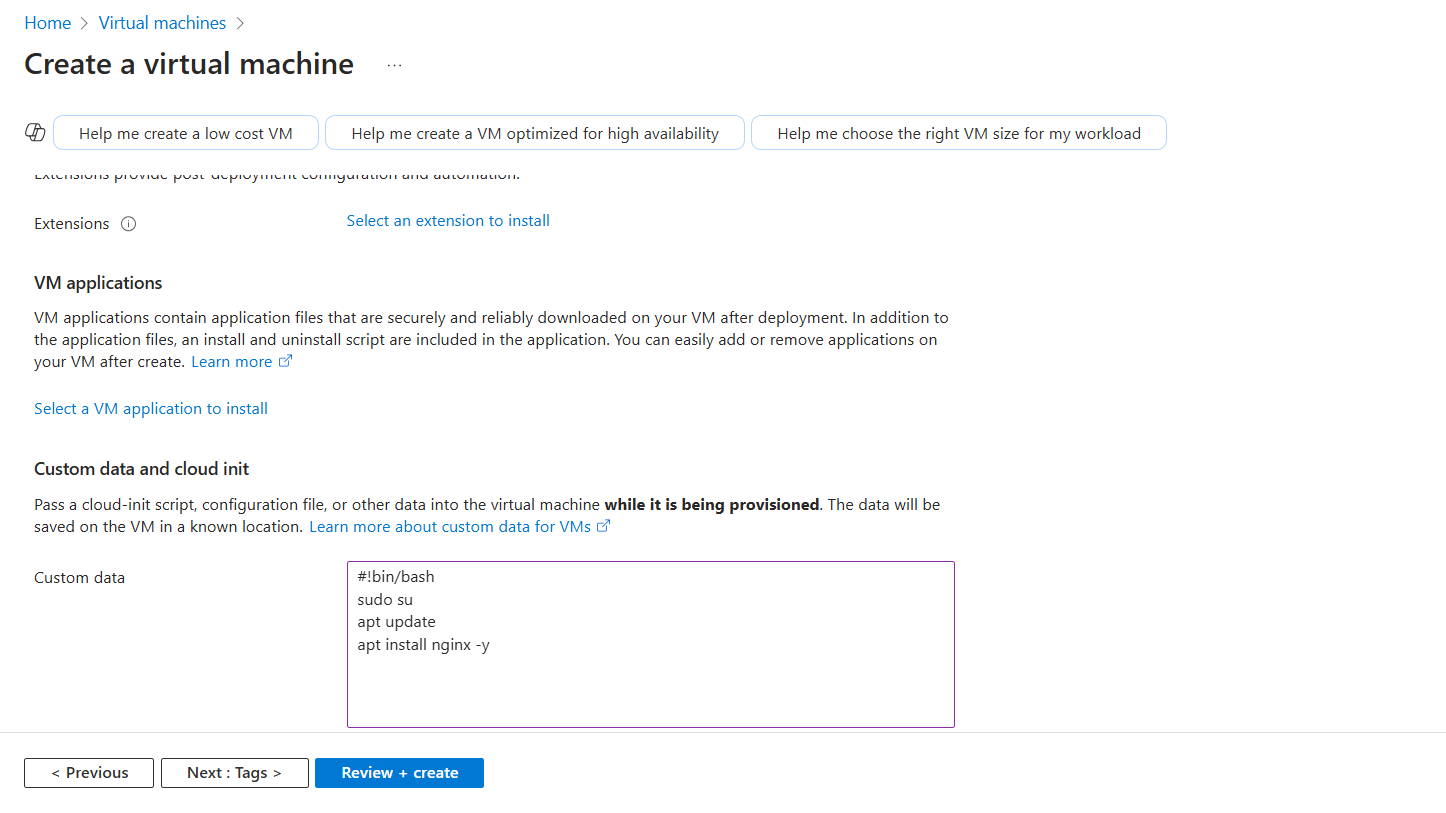


**Fig:** A Vnet with Three Subnets.

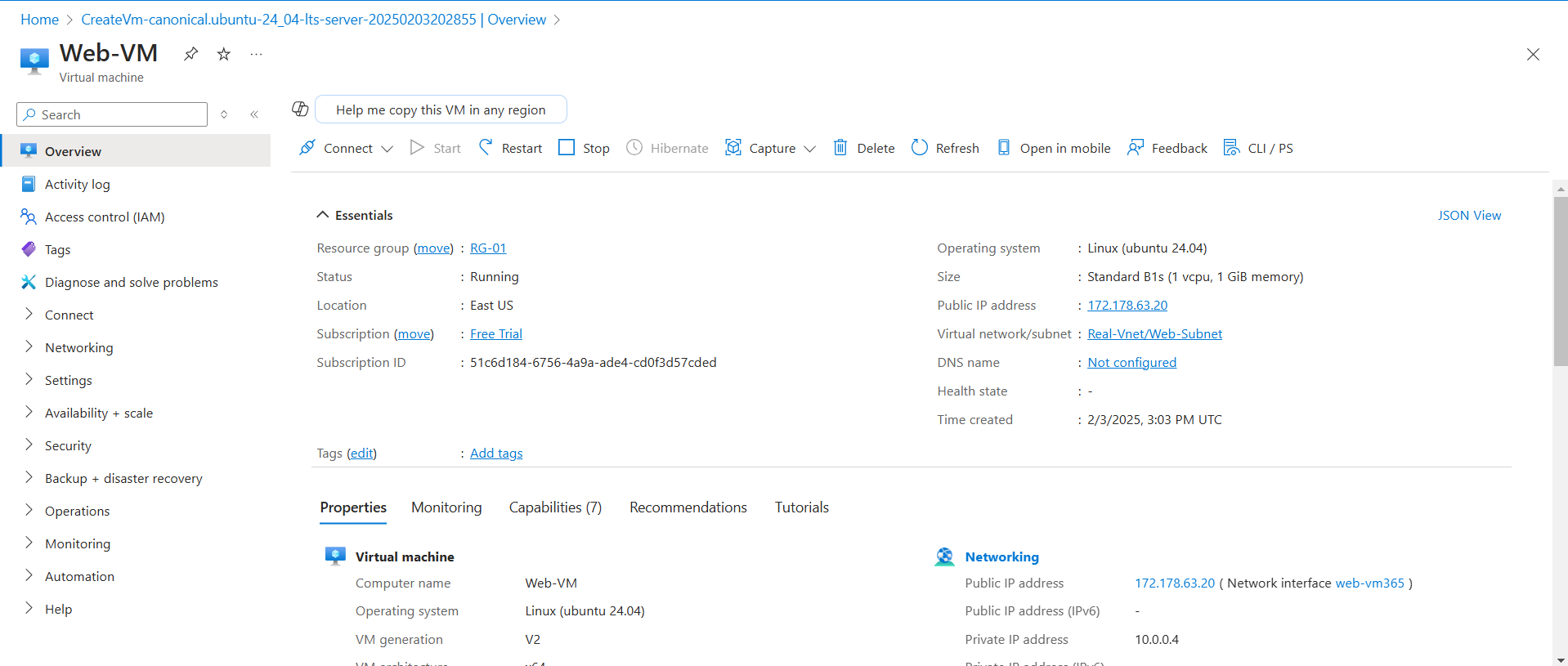


**Fig:** Virtual Network (Real-Vnet).

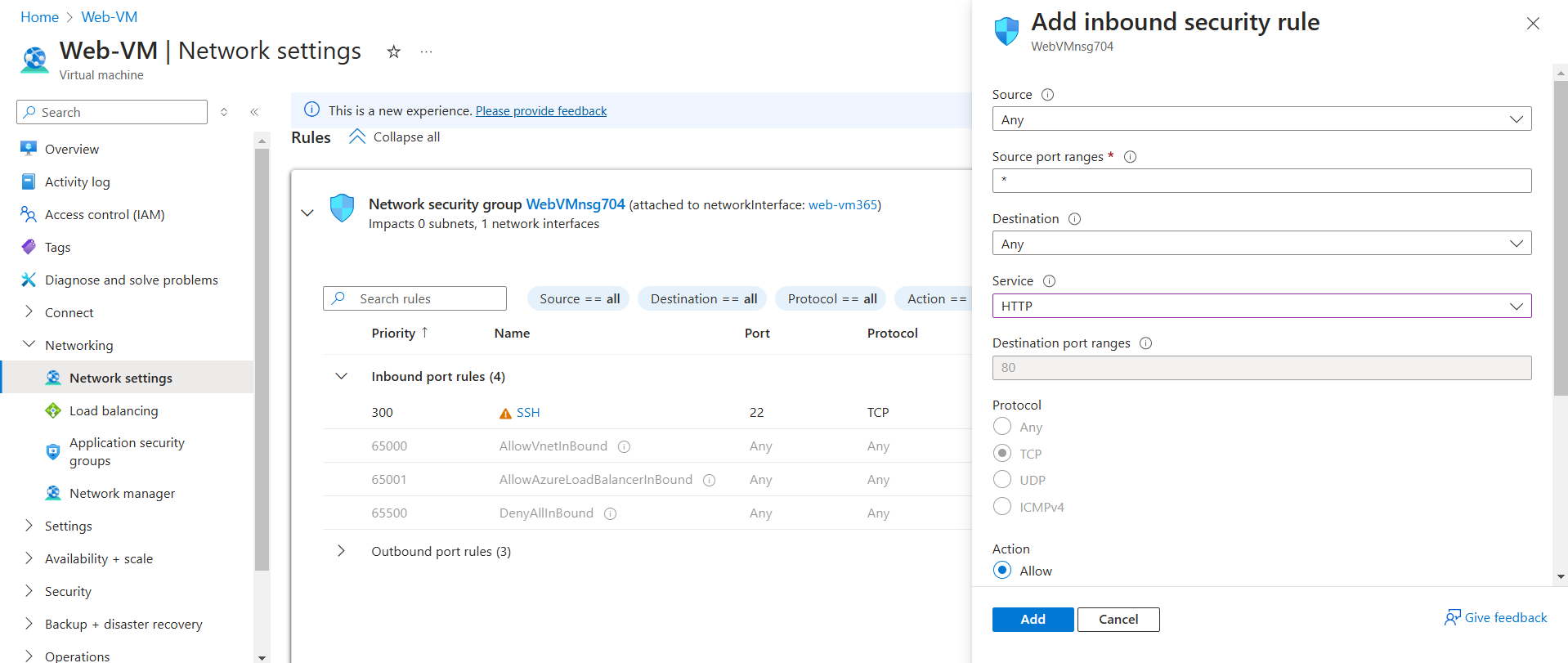
**Step3:** Create an Ubuntu VM (act as Web-server) and install Nginx in it.



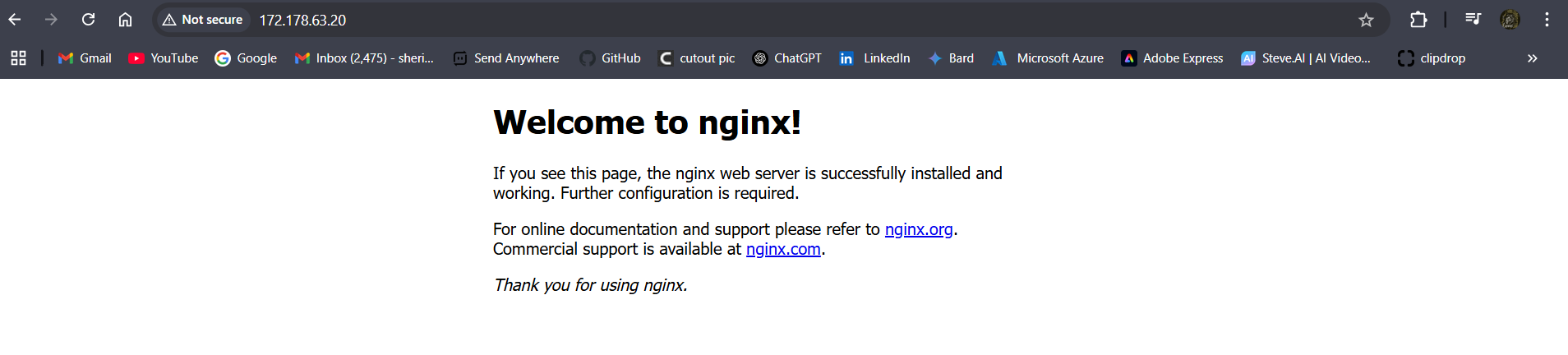
**Fig**: Installing of Nginx in the Ubuntu VM.



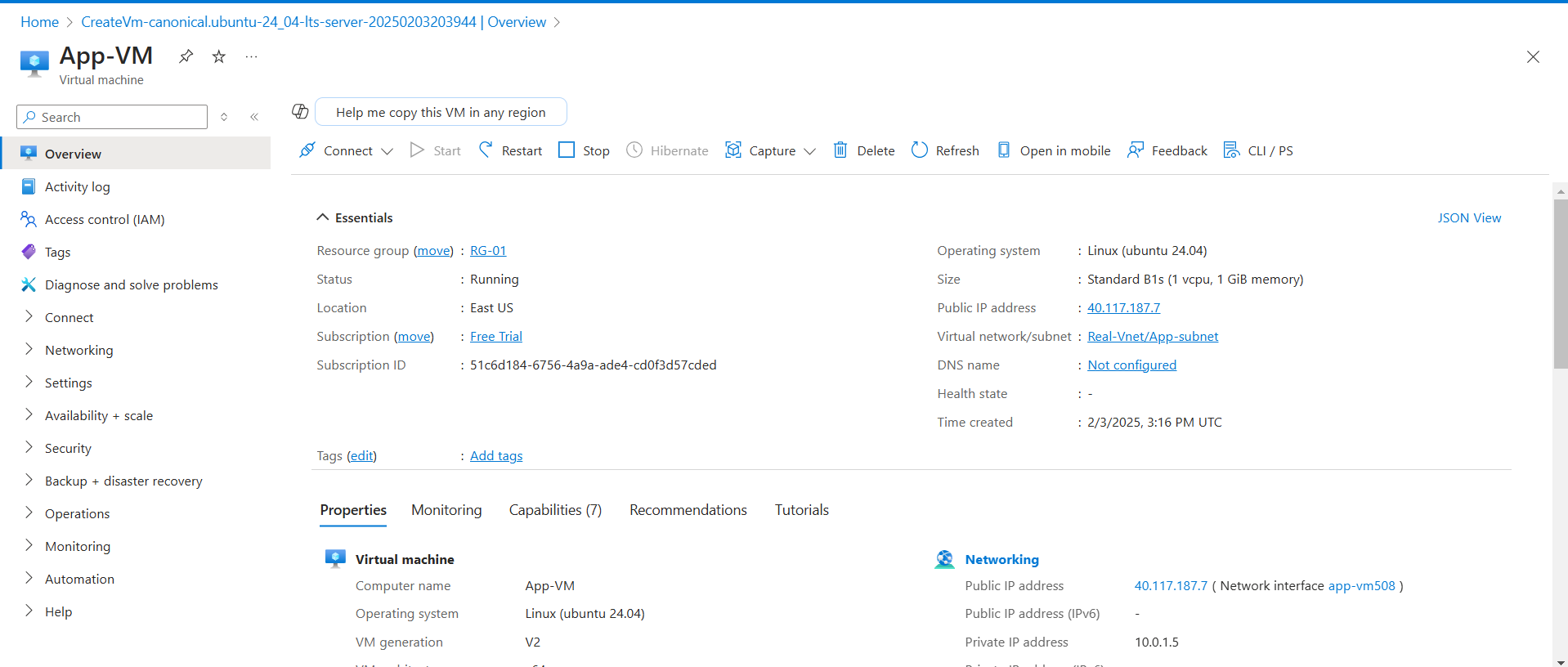
**Fig:** web-server is created with Nginx application.



**Fig:** Adding of NSG inbound rule to allow traffic from port 80 (HTTP).

If we allow the “port 80” (HTTP) for everyone then only we can browse our application in any browser.

Step4: Create the Ubuntu VM (act as App-Server) and install Tomcat in it.



Installing of Tomcat and adding manger role to the normal user:

1. Login into App-VM

ssh username@public IP

1. **Set up the environment for tomcat:** install the JDK (java development kit).You need the Java Development Kit (JDK) to install and run Tomcat because Tomcat is a Java-based application.

**[**Sudo install default-jdk].

1. Check whether jdk installed or not:

[java -version]

1. Install tomcat:

Sudo wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.2/bin/apache-tomcat-11.0.2.tar.gz

1. Extract the tomcat tar file:

[sudo tar -xzf < apache-tomcat-10.1.33.tar.gz>]

1. Create the tomcat directory inside “opt” folder.

Cd /opt/ 🡺press enter

Mkdir tomcat 🡺press enter

1. Move the extracted folder of tomcat into the path of /opt/tomcat:

Cd ..

[Mv apache-tomcat-10.1.33/\* /opt/tomcat/].

1. Running tomcat as root is not recommended due to security risk so add the normal user:

[sudo useradd -m -d /opt/tomcat -U -s /bin/false tomcat] it adds user called tomcat.

1. Create the manager for tomcat in order access tomcat server:

Cd /opt/tomcat/conf/ 🡺press enter

Vi tomcat-users.xml 🡺press enter

<role rolename="manager-gui"/>

<user username="harish" password="Harish@12345" roles="manager-gui"/>

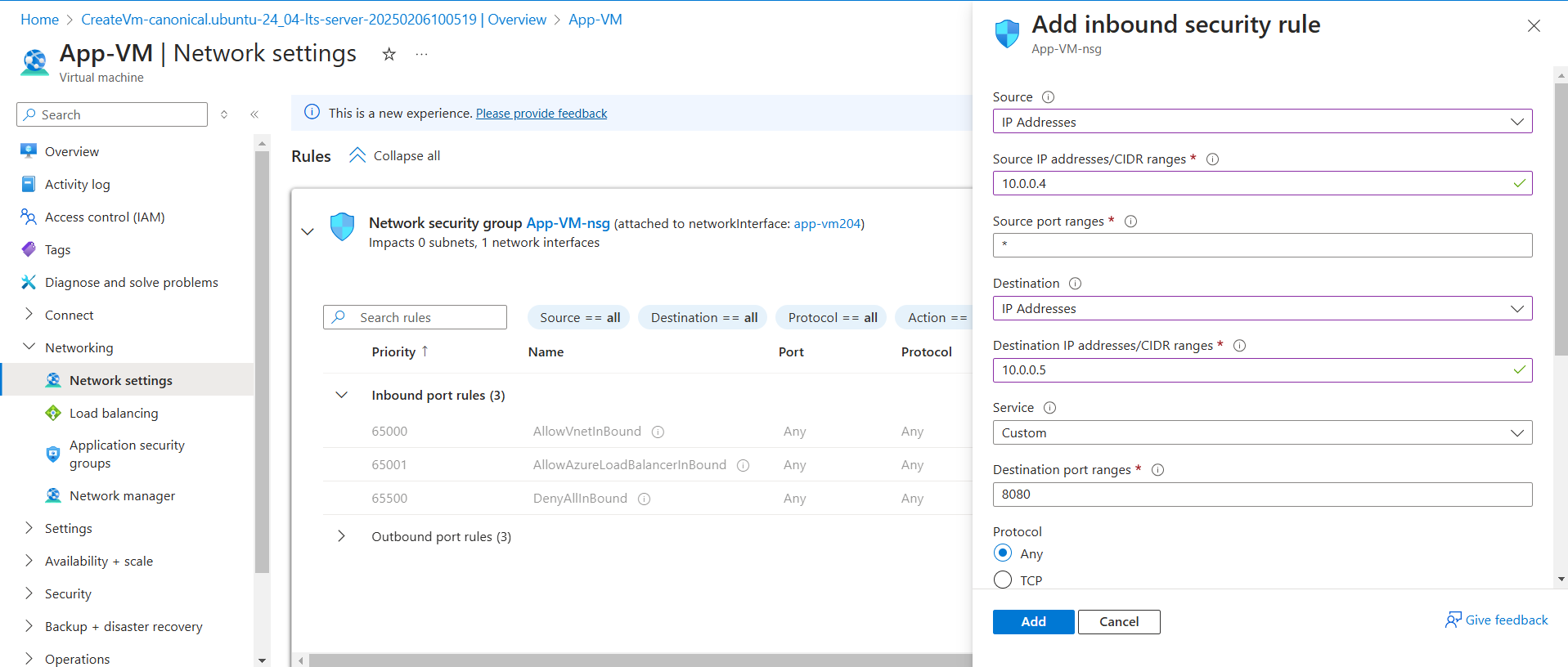
Then exit from the “vi” editor.

1. Now start the tomcat:

Cd /opt/tomcat/bin/ 🡺press enter

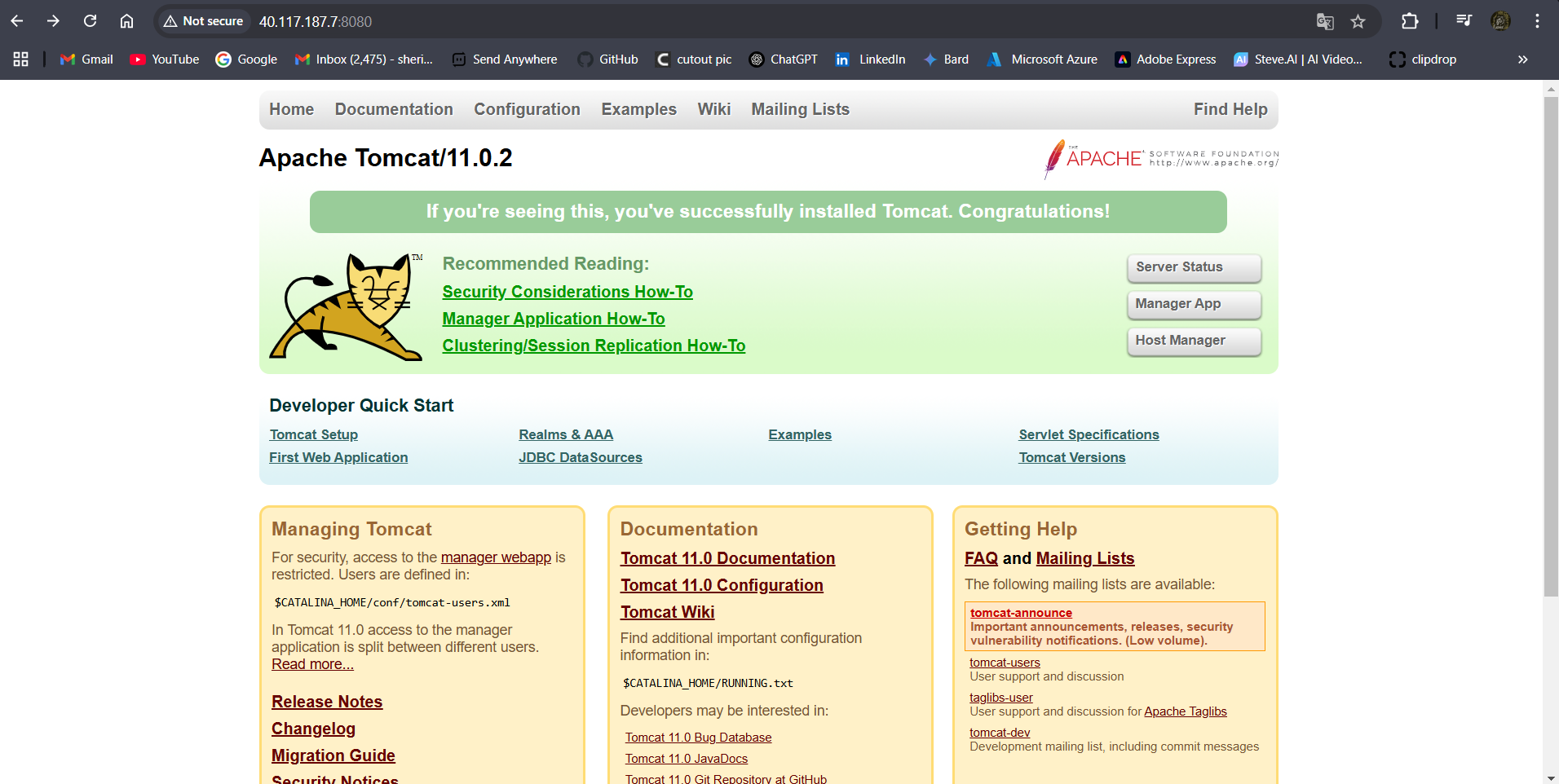
./startup.sh 🡺press enter.

**Note:** After installing Tomcat from NSG inbound rules deny or delete the SSH port traffic.



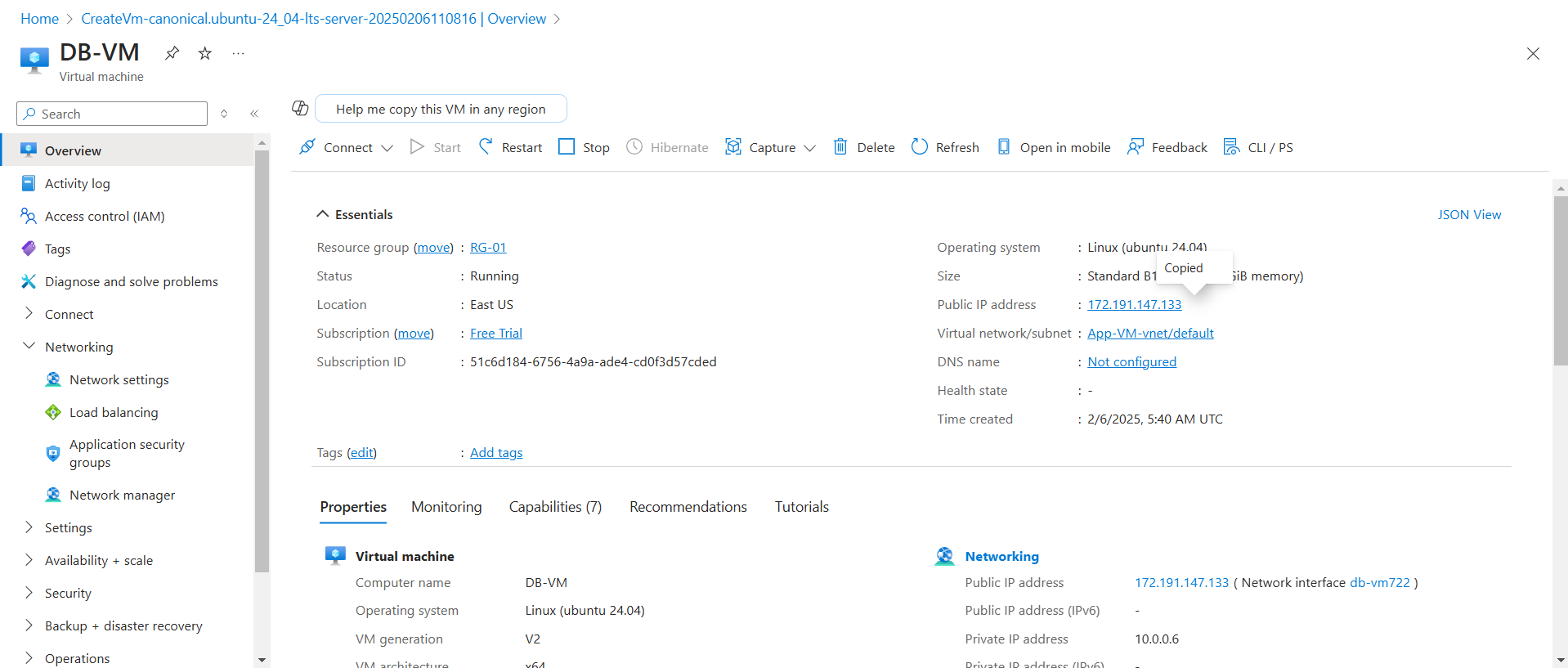
**Fig:** Allowing the traffic from web-VM (private IP: 10.0.0.4) to the App-VM (private IP: 10.0.0.5) on port 8080.

**Note:** Tomcat primarily operates on port **8080** by default. This is the port that clients (like web browsers) use to connect to Tomcat and access web applications running within it.



**Fig:** Tomcat is installed successfully.

**Step5:** create the Ubuntu VM (act as Data-server).



**Installing process of MySQL server in the DB-VM:**

So we have to **enable the SSH port** in order login into the Ubuntu VM to install MySQL in it.

1. Login into Ubuntu DB-VM.

ssh username@public IP

1. Update the DB-VM.

sudo apt update.

1. Install the MySQL server.

Sudo apt install mysql-server -y

1. Check status whether the server installed or not.

Sudo systemctl status mysql.service

1. If MySQL server is not activated or not started the start it.

Sudo systemctl start mysql.service

1. Setting of password and security related configuration.

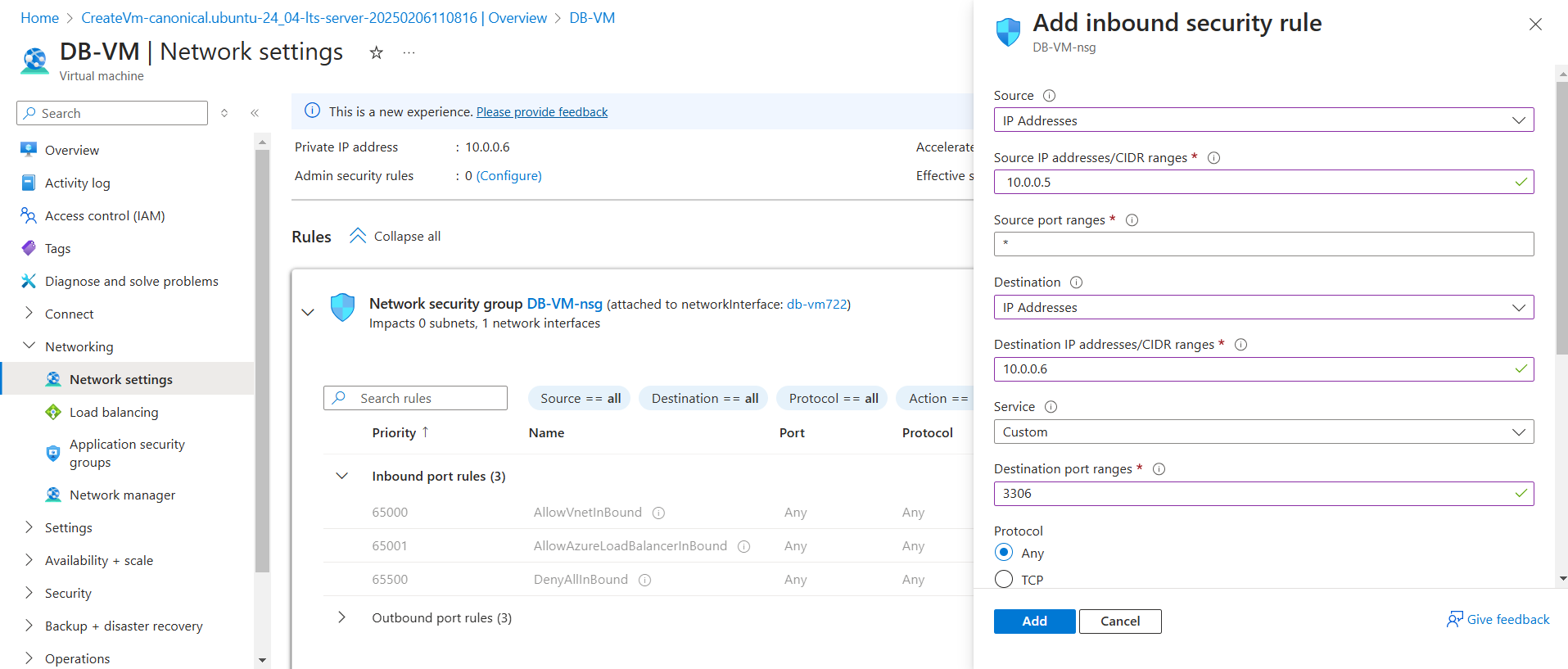
Sudo mysql\_secure\_installation

1. Press “y”
2. Press “2”
3. Press “y”
4. Press “y”
5. Press “y”
6. Press “y”
7. Login into the MySQL server.

Mysql 🡺press enter

Exit 🡺 to exit from mysql server.

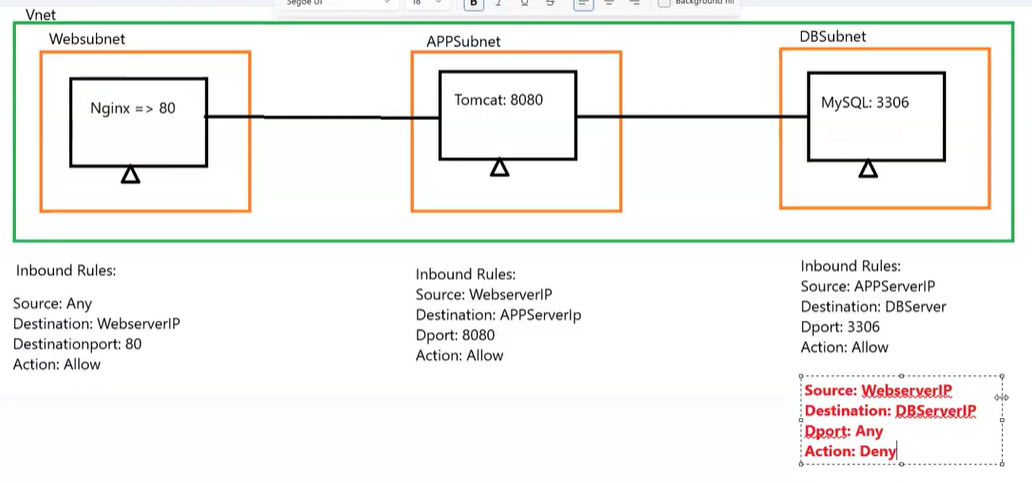
Step6: Configuring Of NSG inbound rules for DB-VM.



**Fig:** allowing the traffic from the App-VM (private IP: 10.0.0.5) to the DB-VM (private IP: 10.0.06).

**Note:** MySQL server runs on the port number 3306

**The block diagram of above project is given as:**

****