# PROJECT OVERVIEW

Project Tittle: 3-tier Azure deployment Project.

Aim: The aim of this project is to design and implement a 3-tier architecture infrastructure that separates the web interface, application logic, and database layers into distinct tiers. This architecture will enhance scalability, security, and manageability of the application environment.

1. Creat a Resource Group
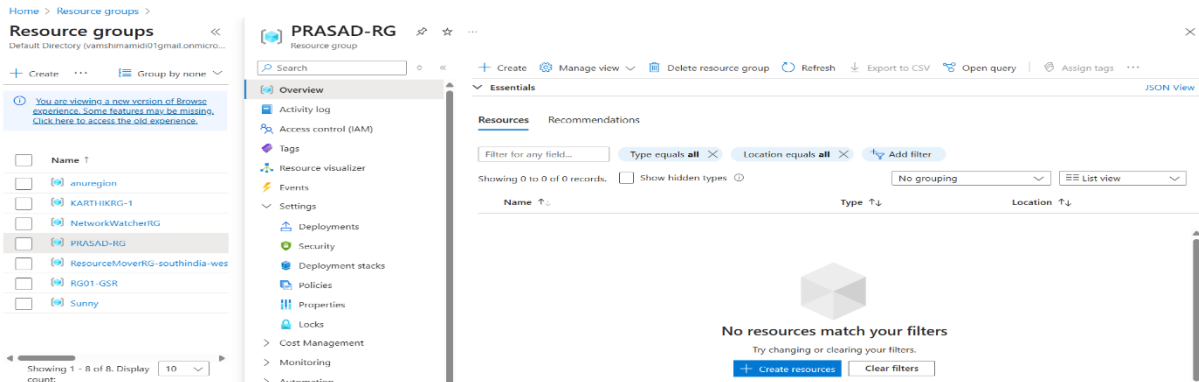
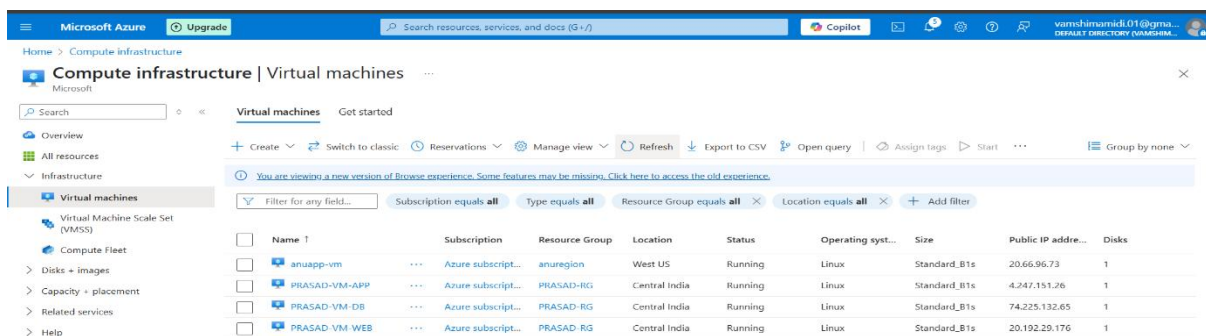## 2. Create 3 subnets in an Azure Virtual Network (VNet)—one each for a web server, app server, and DB server.
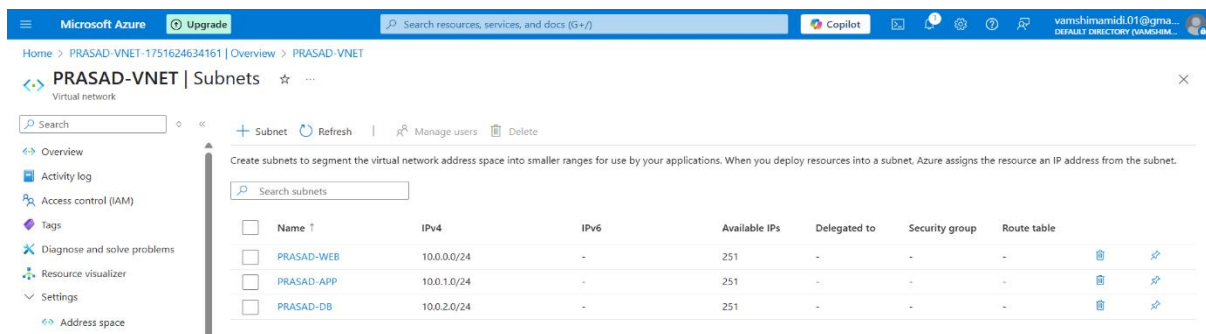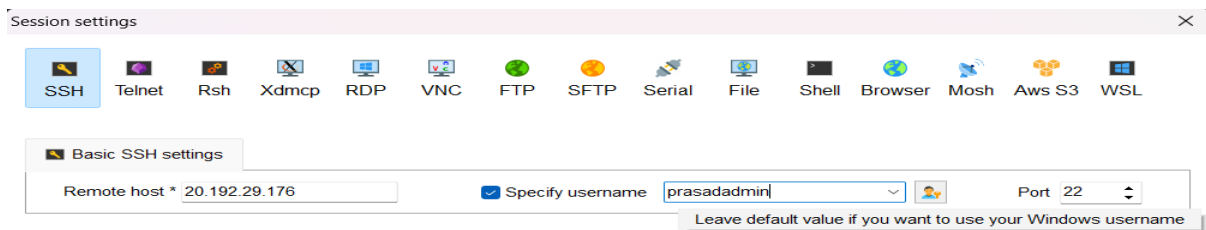

3. Deploy Virtual Machines in Each Subnet:

➢ **PRASAD-VM-WEB → PRASAD-WEB**

➢ **PRASAD-VM-APP → PRASAD-APP**

➢ **PRASAD-VM-DB → PRASAD-DB**

4. SSH into the PRASAD-VM-WEB [20.192.29.176] to Connect Mobaxterm:
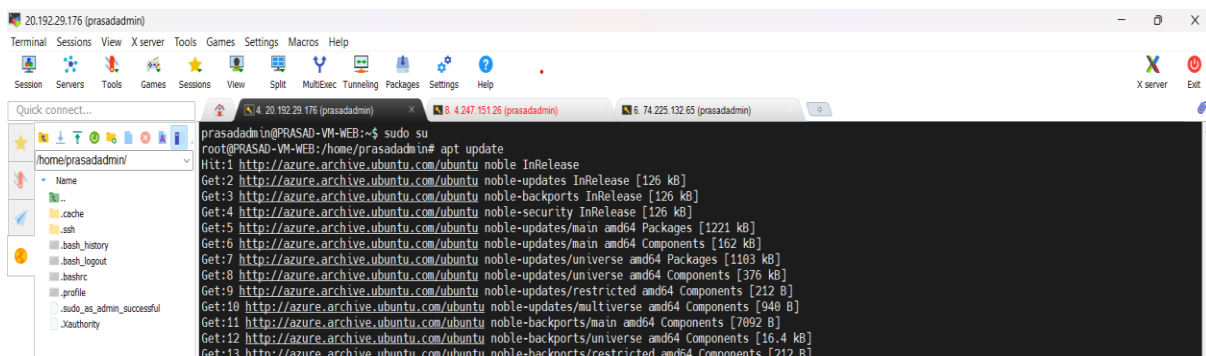

METHOD-1:

INSTALLATION OF NGINX IN PRASAD-VM-WEB

1: Using sudo su (Recommended)

• **sudo su**

      a. This switches you to the root user.

      b. The prompt will change from $ to #.
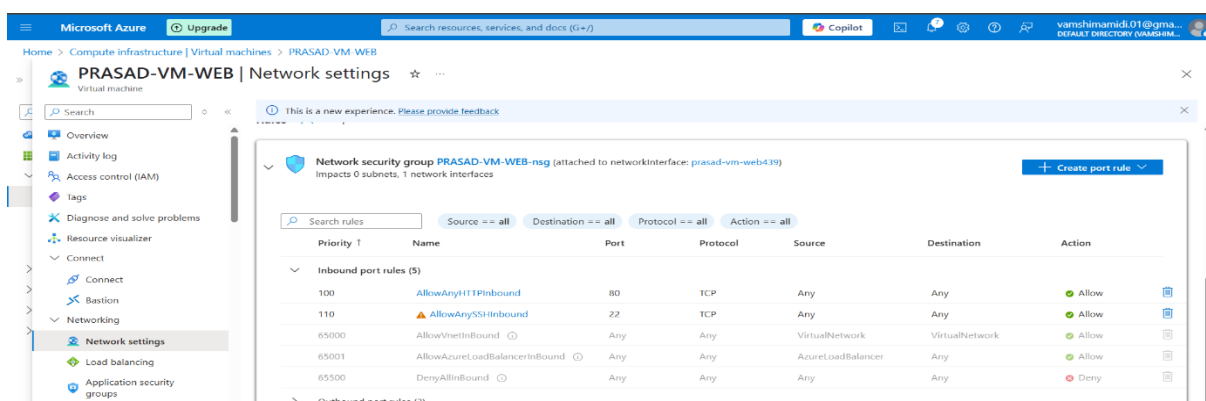
2. Update System Packages

• **apt update**

3.Install nginx

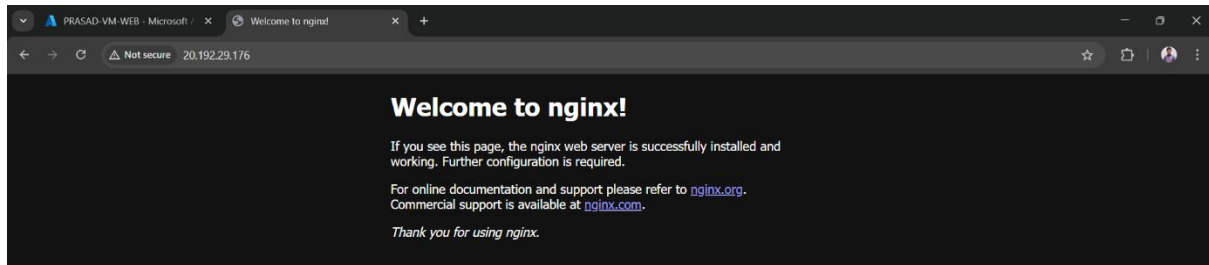•**apt install nginx -y**

4.Allow HTTP Traffic via NSG

Note: [If you're not seeing the NGINX page, your Network Security Group (NSG) may be blocking port 80. Run this to allow HTTP (port 80) on the NSG]

5.Access NGINX Web Page

Open your browser and go to: PRASAD-VM-WEB <20.192.29.176 > Copy and Paste

http://< 20.192.29.176 >80

> You should see the default NGINX welcome page.

METHOD-2:
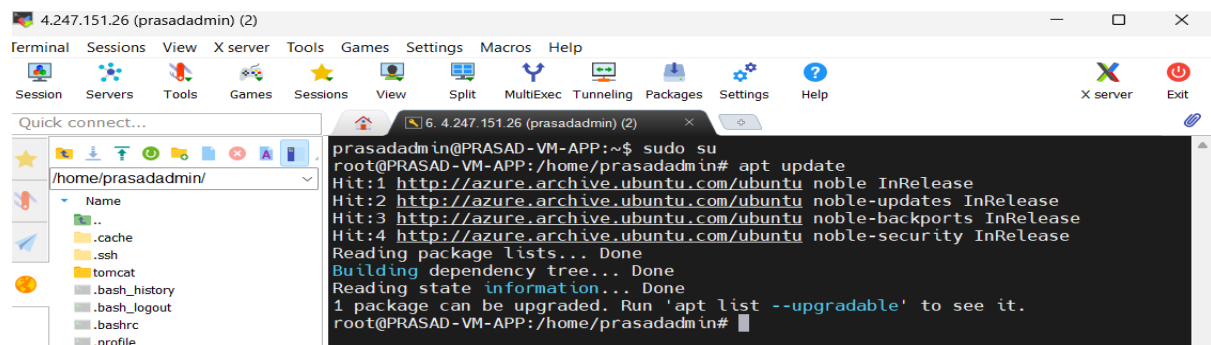
Installation of tomcat in PRASAD-VM-APP

1: Using sudo su (Recommended)

• <mark>sudo su</mark>

> a. This switches you to the root user.

> b. The prompt will change from $ to #.

2. Update System Packages

• <mark>apt update</mark>

3.Install Java

> Tomcat requires Java. Install OpenJDK:

• <mark>apt install default-jdk -y</mark>

Verify the installation:

• <mark>java -version</mark>

4.Download Tomcat

    • **Go to the Apache Tomcat official website and get the latest version.**

Run:

• **wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.41/bin/apache-tomcat-10.1.41.tar.gz**

    (Replace the version number with the latest one if needed.)

• **ls (to list the files)**

5. Extract and Configure Tomcat

• **tar -xvzf apache-tomcat-10.1.18.tar.gz**

• **ls**

• **mv apache-tomcat-10.1.18 tomcat**

• **ls**

To start Apache Tomcat manually from the bin directory, follow these steps:

6.Navigate to the Tomcat Directory

• **cd /tomcat/bin**

7. Start Tomcat

Run the startup script: (./ is to execute a command)

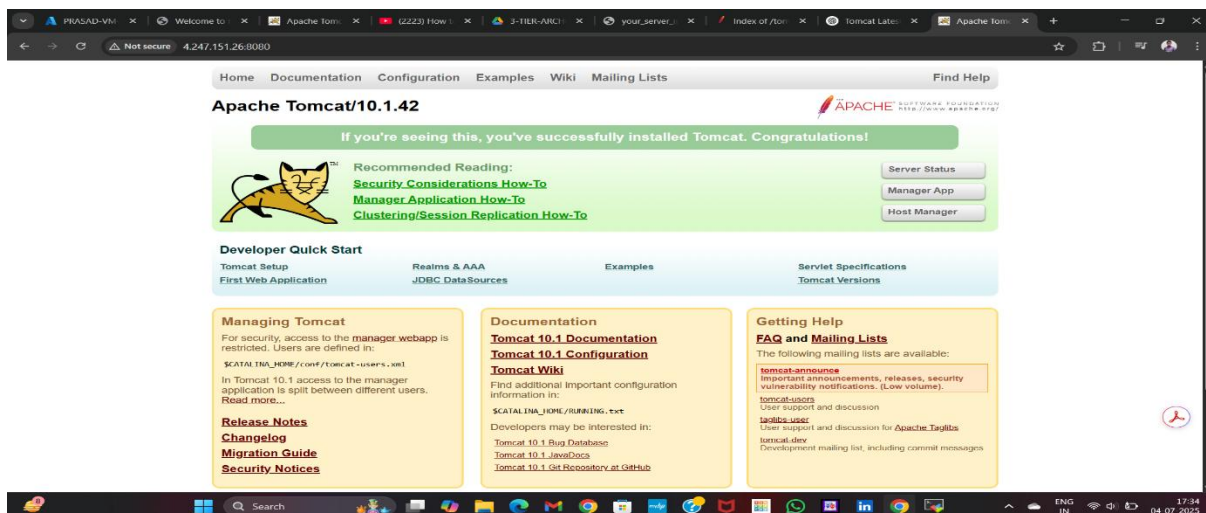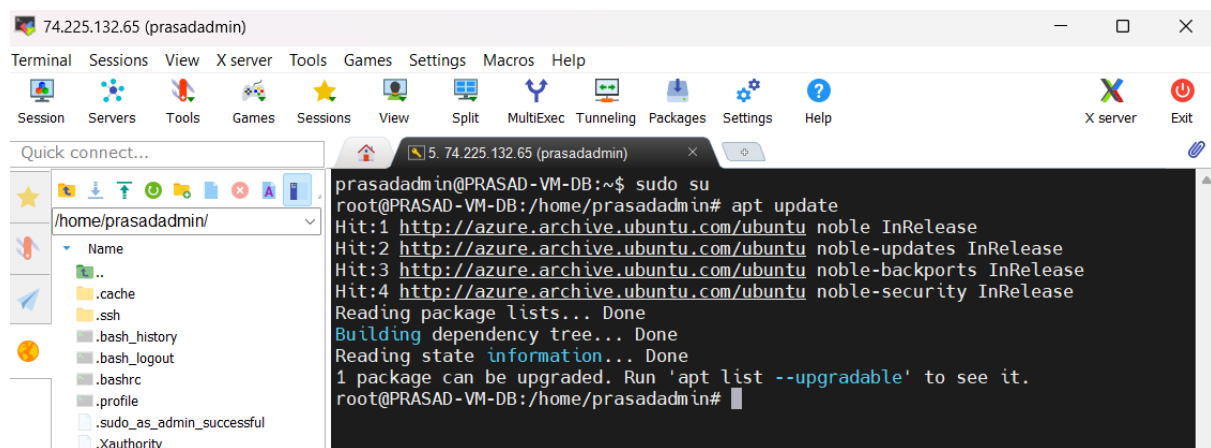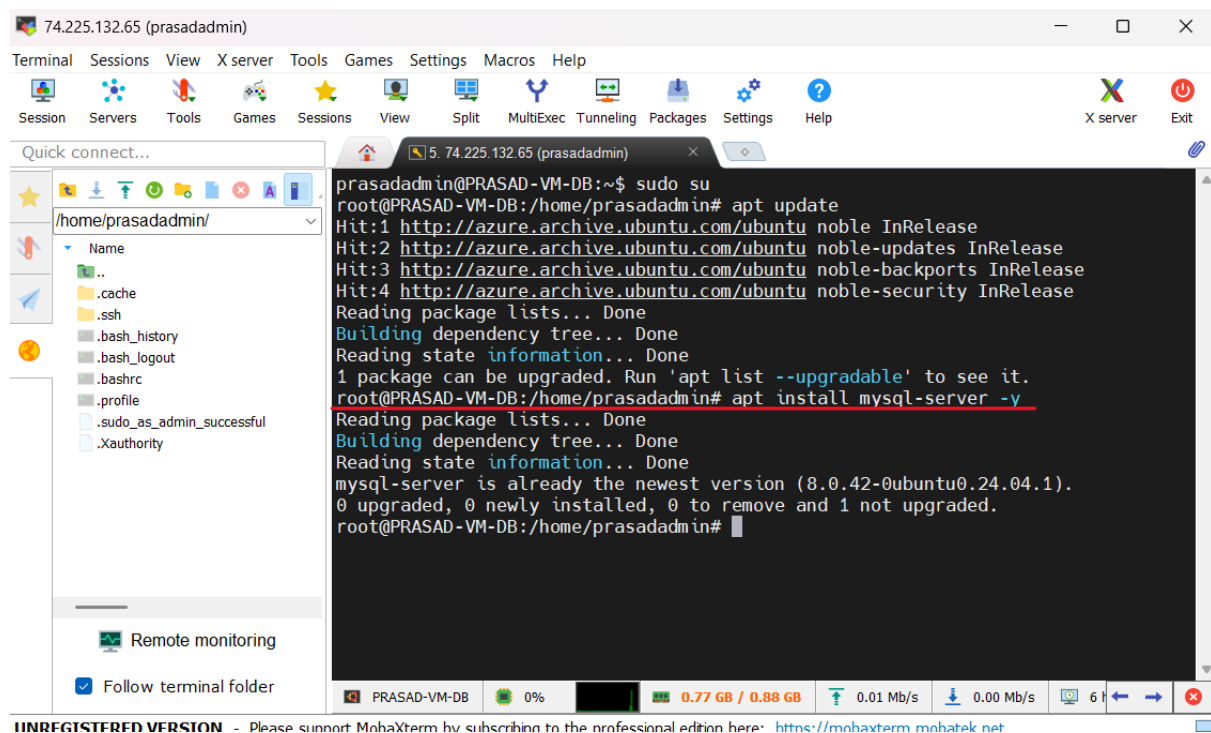• **./startup.sh**

8. Allow HTTP Traffic via NSG

   Note: [If you're not seeing the Tomcat page, your Network Security Group (NSG) may be blocking port 8080. Run this to allow HTTP (port 8080) on the NSG]
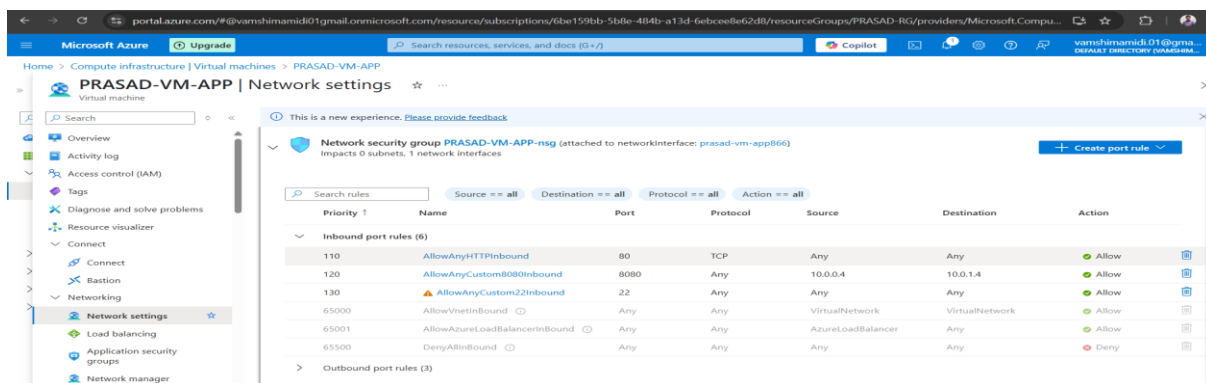
9. Access Tomcat

Open your browser and go to:

http:// 4.247.151.26:8080

METHOD-3:

Installation of tomcat in PRASAD-VM-DB

1: Using sudo su (Recommended)

• <mark>sudo su</mark>

>   **a. This switches you to the root user.**

>   **b. The prompt will change from $ to #.**

2. Update System Packages

• <mark>apt update</mark>

3.Run the following command to install MySQL:

• <mark>apt install mysql-server –y</mark>

4.To improve security, run the following command:

• <mark>mysql_secure_installation</mark>

5.Edit the MySQL configuration file using a text editor:

• <mark>nano /etc/mysql/mysql.conf.d/mysqld.cnf</mark>

6. Locate the following line: bind-address = 127.0.0.1

• <mark>bind-address = 0.0.0.0</mark>

7. EX: Modify it based on your requirement:

Scenario Configuration

Listen on all interfaces (for remote access) bind-address = 0.0.0.0

Listen on a specific IP (e.g., 192.168.1.100) bind-address = 192.168.1.100

• <mark>Save and exit (CTRL + X, then Y, then Enter).</mark>

8. Apply the changes by restarting MySQL:

• <mark>systemctl restart mysql</mark>

9.Check if MySQL is running:

• <mark>systemctl status mysql</mark>

Recommended NSG Rules for 3-Tier Setup :

(Network Security Group) is like a firewall for your Azure VMs or subnets. It controls inbound and outbound traffic using allow/deny rules based on:

• Source IP

• Destination IP

• Port

•Protocol (TCP/UDP)
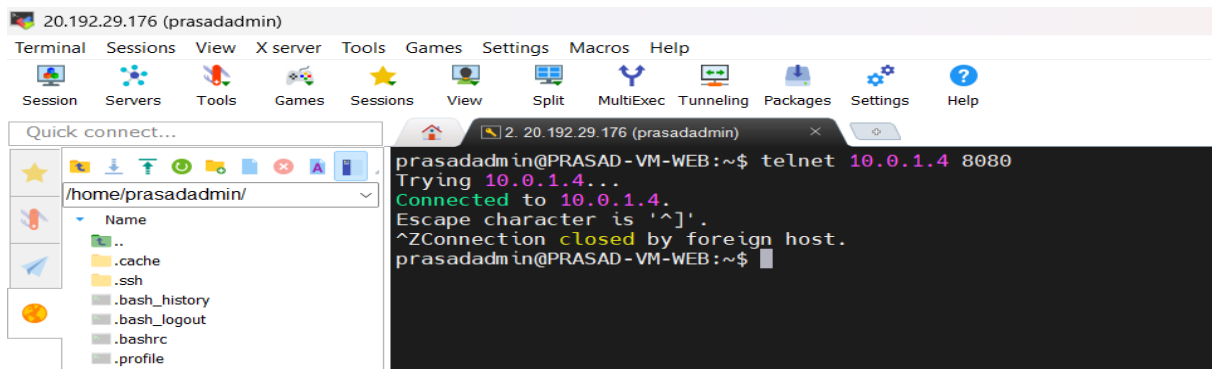
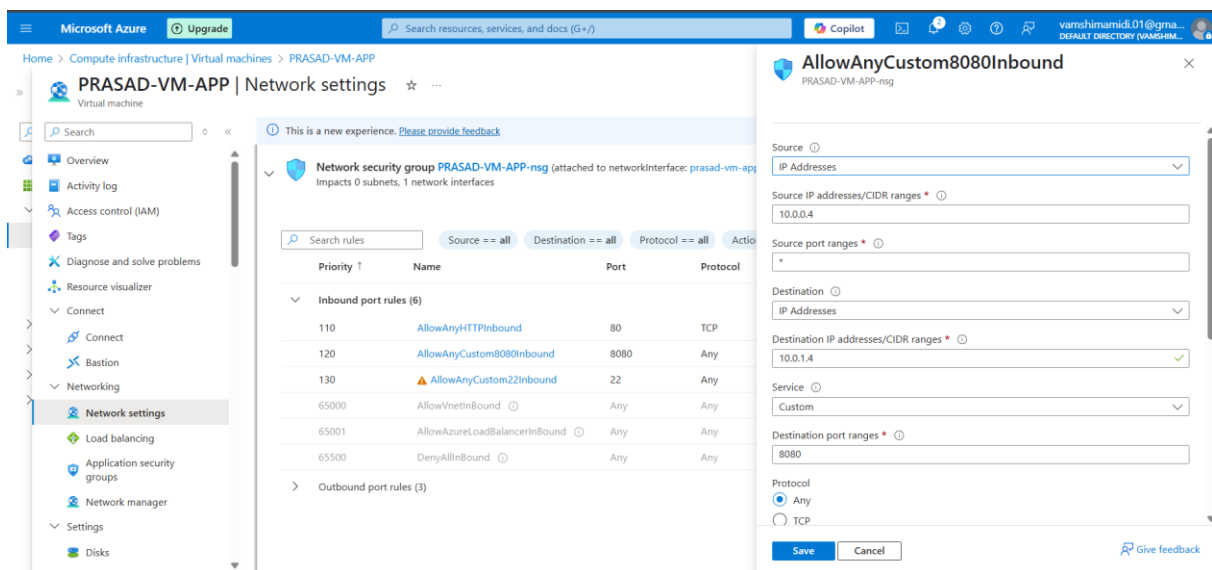### 1.PRASAD-VM-WEB Setup NSG:

Inbound Rules:

Source: Any

Destination: PRASAD-VM-WEB→ 10.0.0.4

Dport: http,https [80,443]

Action: Allow

telnet<APPVM02>8080

2.PRASAD-VM-APP Setup NSG:

Inbound Rules

Source: PRASAD-VM-WEB→ 10.0.0.4
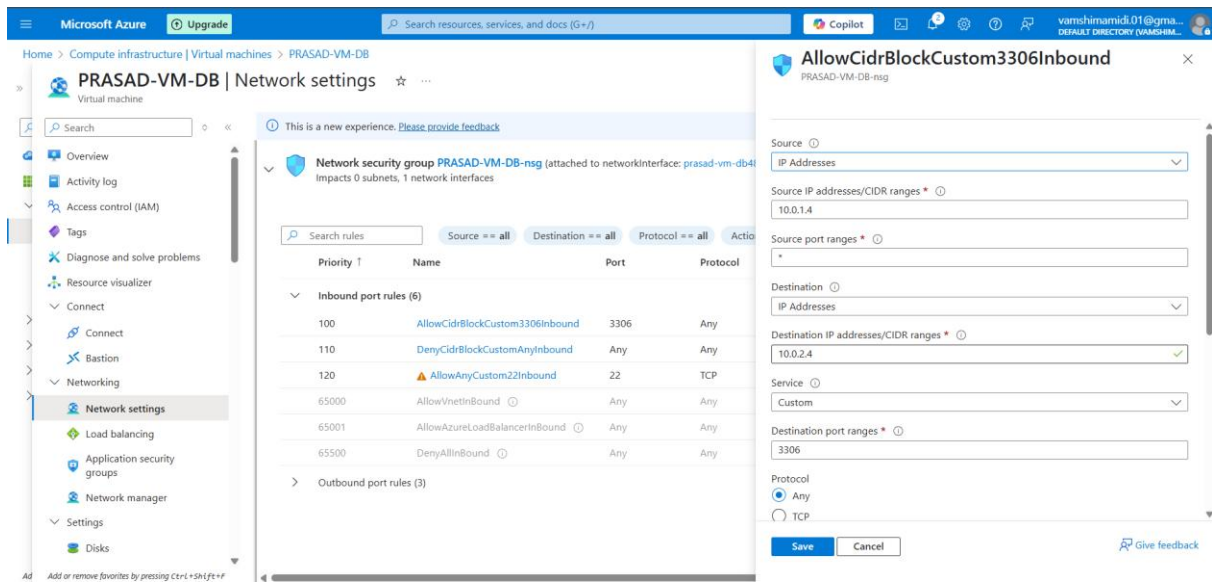
Destination: PRASAD-VM-APP→10.0.1.4

Dport: [8080]

Action: Allow

telnet<PRASAD-VM-DB>3306

3.PRASAD-VM-DB Setup NSG:

Inbound Rules :

Source: PRASAD-VM-APP→ 10.0.1.4

Destination:PRASAD-VM-BD→ 10.0.2.4

Dport:[3306]

Action: Allow

telnet<PRASAD-VM-WEB>80

Perfect — if your VMs (PRASAD-VM-WEB, PRASAD-VM-APP, and PRASAD-VM-DB) are successfully connecting to each other.

❖ DbVM03 denies access from WebVM01 by design, because the Web tier should not talk to the database. The web tier doesn't need direct access to the database, so we deny it.

❖ This limits the impact of a compromised web server — attackers can't jump straight to your database.
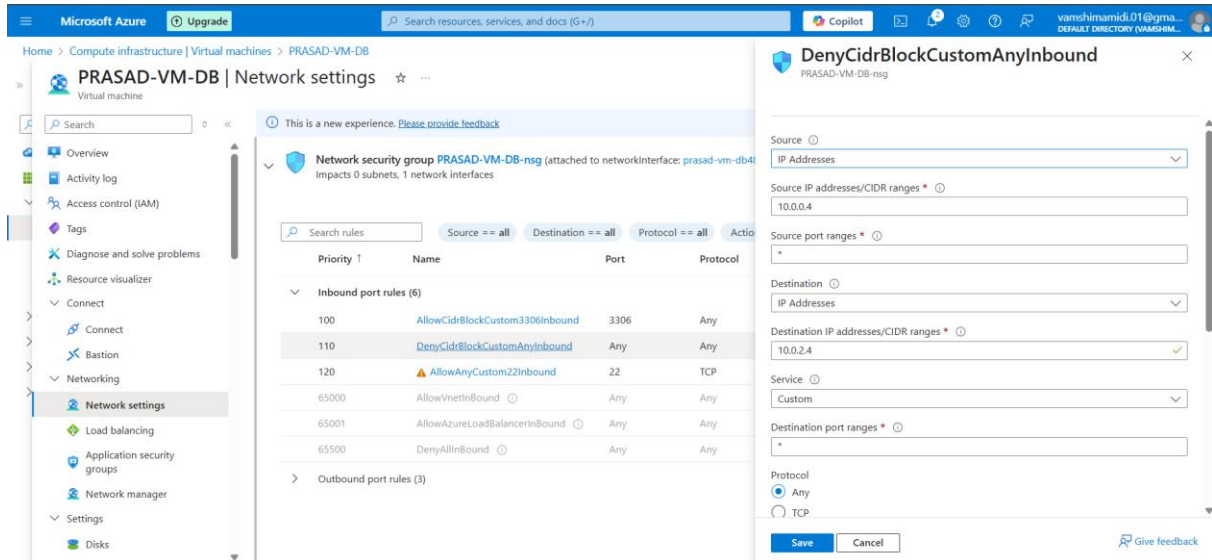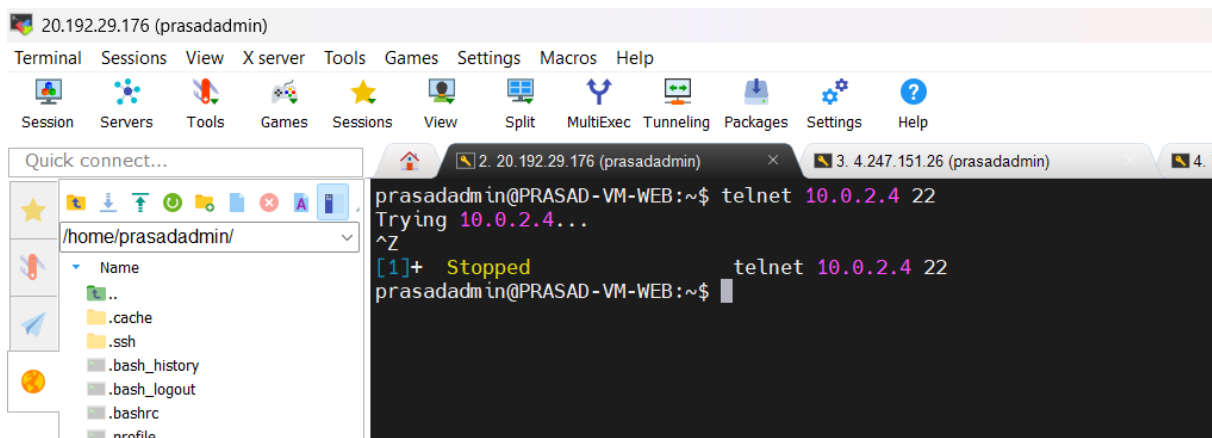

Inbound Rules
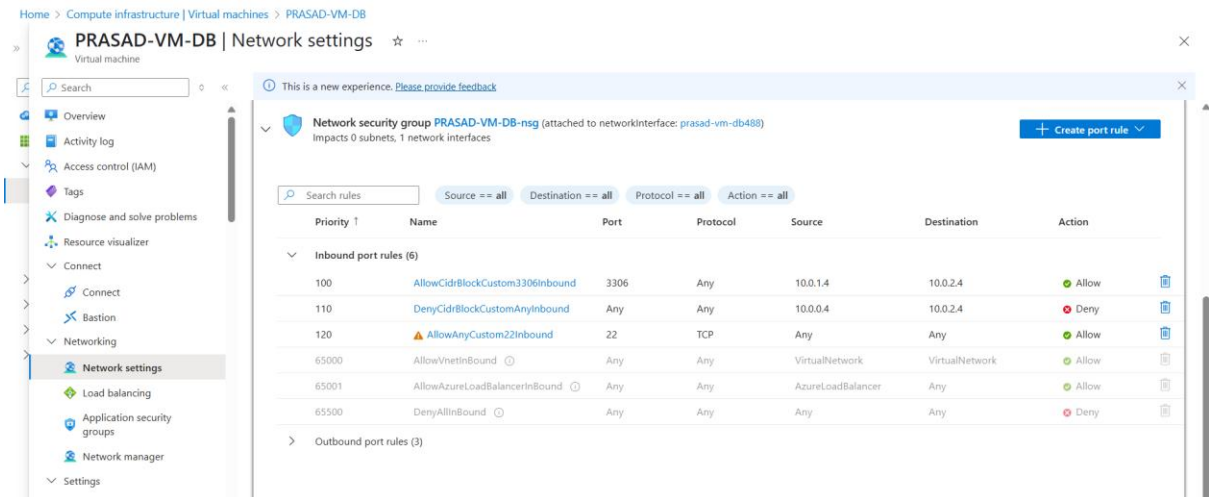
Source: PRASAD-VM-WEB→10.0.0.4

Destination: PRASAD-VM-DB→1.0.2.4

Dport: [*]

Acess: Deny

telnet<PRASAD-VM-DB>22

Conclusion:

1. NSGs protect your VMs by controlling which traffic is allowed or denied.

2. Deny rules help block unwanted access between tiers (e.g., Web → DB).

3. This ensures a secure and well-isolated 3-tier architecture in Azure.