

NETWORK SECURITY GROUP

In a 3-tier architecture on Azure, you'll typically separate your application into three distinct layers: presentation, application, and database. Here's how you might set this up with Azure VMs:

Resource group:

Step 1: First step created a resource group named Task1.

Step 2: Second step created Virtual network named VNET01.

With the following subnets:

For web server - 10.0.1.0/24

For App server - 10.0.2.0/24

For Database - 10.0.3.0/24

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓	Security group ↑↓	Route table ↑↓	
Appserver	10.0.2.0/24	-	250	-	-	-	...
Database	10.0.3.0/24	-	250	-	-	-	...
Webserver	10.0.1.0/24	-	250	-	-	-	...

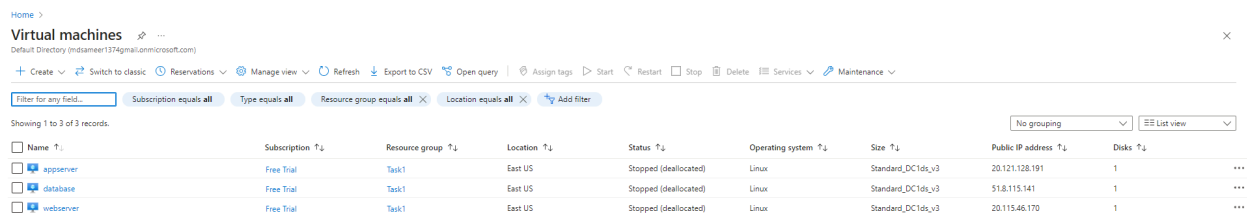
Step3: Created 3 individual virtual machines for web server, Appserver and Database.

With the following Details:

VM Setup:

Create an Azure VM: Use a 3 Linux-based VM for this purpose, such as Ubuntu.

Here we created a Linux based ubuntu 24.04 LTS version.



Name ↑↓	Subscription ↑↓	Resource group ↑↓	Location ↑↓	Status ↑↓	Operating system ↑↓	Size ↑↓	Public IP address ↑↓	Disks ↑↓
appserver	Free Trial	Task1	East US	Stopped (deallocated)	Linux	Standard_DC16is_v3	20.121.128.191	1
database	Free Trial	Task1	East US	Stopped (deallocated)	Linux	Standard_DC16is_v3	51.8.115.141	1
webserver	Free Trial	Task1	East US	Stopped (deallocated)	Linux	Standard_DC16is_v3	20.115.46.170	1

INSTALLATION Of Web Server:

We are going to use nginx for the web server.

The default port numbers for NGINX are 80 and 443.

We are gonna use port 80 for http connection.

First we need to login our web server using ssh port 22.

We need to allow use of port number 22 in inbound rules for connection purpose.

I Am using a putty application for secure connection of web servers through ssh port.
PUTTY is a free, open-source application that allows users to connect to remote computers and devices using a variety of protocols:

Secure socket shell (SSH), Telnet, Login, Rlogin, Raw, and Serial

Install Nginx:

- Connect to your Web server via SSH by using our web server public ip address.
- Update the package index:

sudo apt update

- Install Nginx:

sudo apt install nginx -y

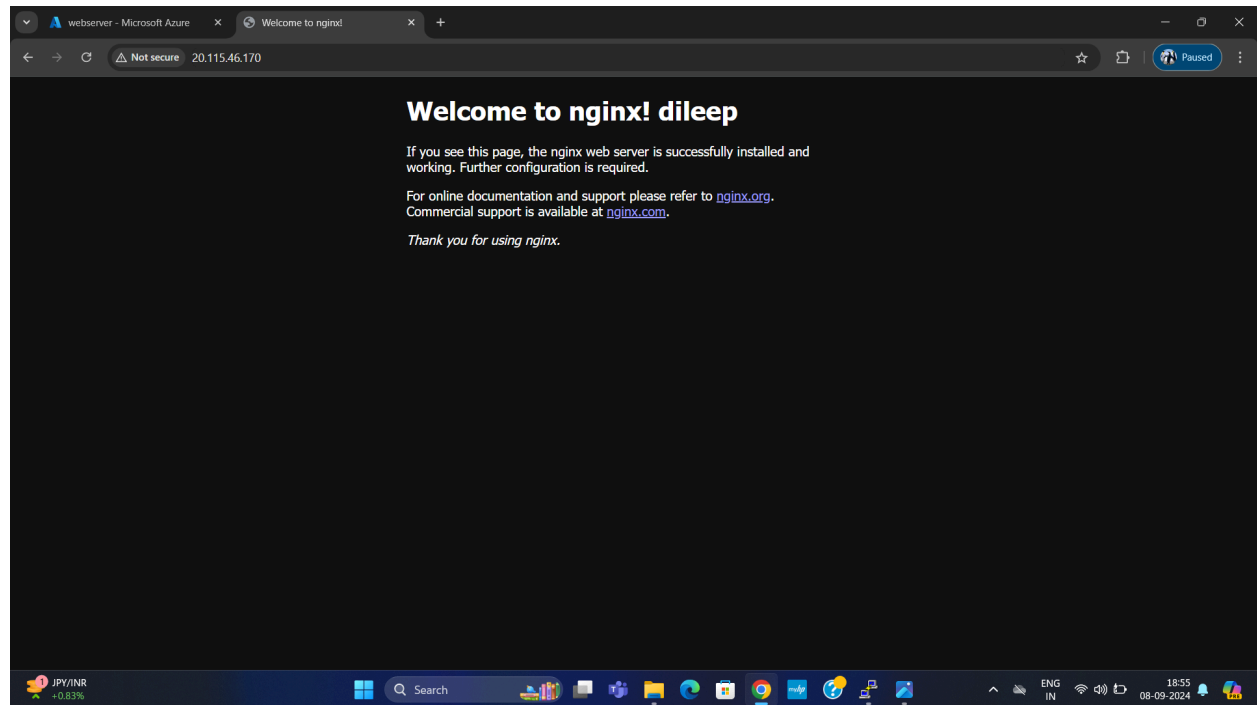
- Configure Nginx as needed for our application.
- Verify whether Nginx is running using systemctl command

sudo systemctl status nginx

```
dileep@webserver:~$ 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 Sun 2024-09-08 13:15:44 UTC; lmin 37s ago
     Docs: man:nginx(8)
   Process: 2346 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 2347 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 2349 (nginx)
    Tasks: 2 (limit: 9458)
   Memory: 1.7M (peak: 1.9M)
      CPU: 5ms
   CGroup: /system.slice/nginx.service
           └─2349 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2350 "nginx: worker process"

Sep 08 13:15:44 webserver systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Sep 08 13:15:44 webserver systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
dileep@webserver:~$
```

We can view our nginx web server using localhost:80



TOMCAT INSTALLATION:

Installing Apache Tomcat 10 on Ubuntu 24.04 is a straightforward process. Here's a step-by-step guide to help you get it up and running:

Prerequisites

- Ensure you have sudo privileges on the Ubuntu system.
- You should have Java Development Kit (JDK) installed, as Tomcat requires Java. If not installed, you can use OpenJDK.

Step 1: Install Java Development Kit (JDK)

Tomcat 10 requires JDK 8 or later. To install OpenJDK 21, you can use the following commands:

```
sudo apt update
sudo apt install default-jdk
```

Verify the installation:

```
java -version
```

```
appserver@appserver:~$ java -version
openjdk version "21.0.4" 2024-07-16
OpenJDK Runtime Environment (build 21.0.4+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
```

Step 2: Download Tomcat 10

1. Navigate to the Tomcat download page: [Apache Tomcat 10](#).
2. Get the link for the latest Tomcat 10 binary distribution (usually a `.tar.gz` file).

Download Tomcat 10. For example:

```
wget
https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.28/bin/apache-tomcat-10.1.28.tar.gz
```

Replace the URL with the latest version if necessary.

Step 3: Extract Tomcat

Create a directory for Tomcat (optional but recommended):

```
sudo mkdir -p /opt/tomcat
```

Extract the downloaded archive to the Tomcat directory:

```
sudo tar xzf apache-tomcat-10.1.15.tar.gz -C /opt/tomcat
--strip-components=1
```

Step 4: Set Up Tomcat User

Create a Tomcat user (optional but recommended for security):

```
sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
```

Change the ownership of the Tomcat directory to the Tomcat user:

```
sudo chown -R tomcat: /opt/tomcat
```

Step 5: Configure Tomcat as a Service

Create a service file for Tomcat:

```
sudo nano /etc/systemd/system/tomcat.service
```

Add the following content to the file:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Ensure `JAVA_HOME` points to your JDK installation and `CATALINA_HOME` is correctly set.

Reload systemd to recognize the new service:

```
sudo systemctl daemon-reload
```

Start and enable Tomcat:

```
sudo systemctl start tomcat
sudo systemctl enable tomcat
```

1.

Check the status to ensure Tomcat is running:

```
sudo systemctl status tomcat
```

```
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-09-08 14:15:58 UTC; 55s ago
     Main PID: 4941 (java)
       Tasks: 29 (limit: 9458)
      Memory: 112.0M (peak: 115.5M)
         CPU: 2.295s
    CGroup: /system.slice/tomcat.service
            └─4941 /usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoad

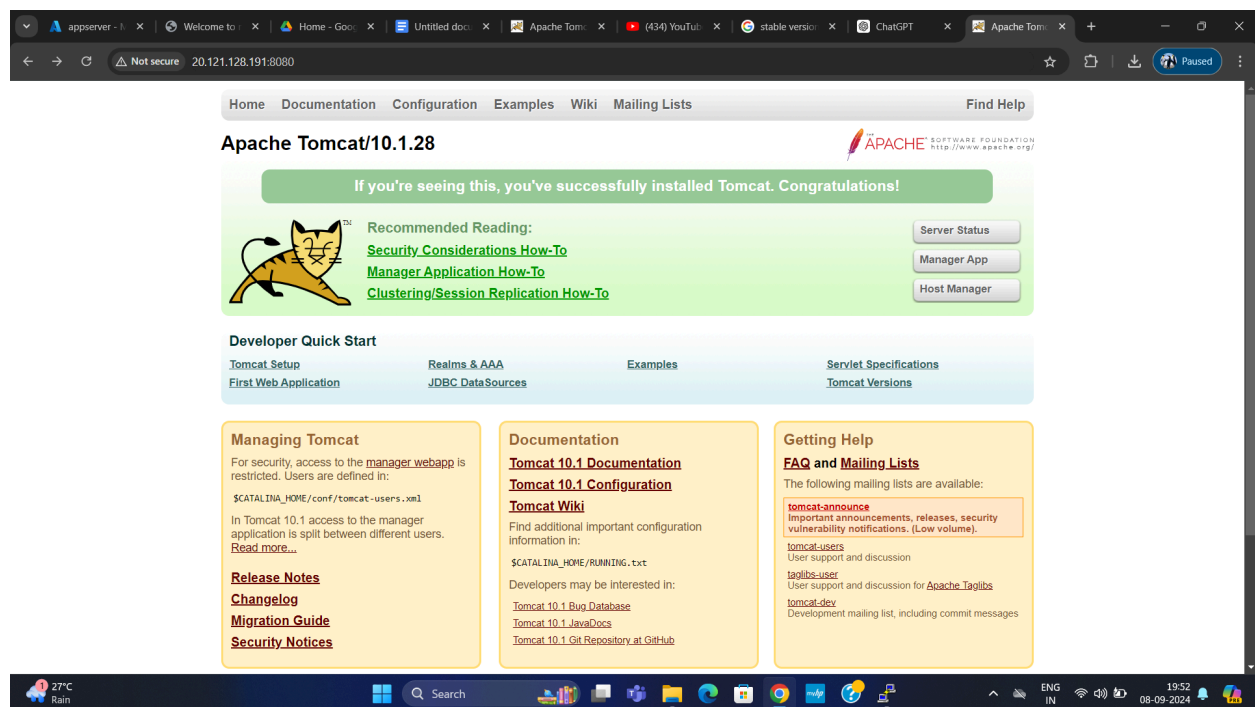
Sep 08 14:15:58 appserver systemd[1]: Starting tomcat.service - Apache Tomcat Web Application Container...
Sep 08 14:15:58 appserver startup.sh[4934]: Tomcat started.
Sep 08 14:15:58 appserver systemd[1]: Started tomcat.service - Apache Tomcat Web Application Container.
Since 1=13/13 (END)
```

Step 6: Access Tomcat

Open a web browser and navigate to:

```
http://your_server_ip:8080
```

1. You should see the Tomcat welcome page.



Step 7: Configure Firewall (if applicable)

If you have a firewall enabled, make sure to allow traffic on port 8080:

```
sudo ufw allow 8080/tcp
```

That's it! You now have Tomcat 10 installed and running on your Ubuntu 24.04 system.

INSTALLATION OF MYSQL DATABASE:

To install MySQL on Ubuntu 24.04, follow these steps:

Step 1: Update Package Index

First, update your package index to ensure you get the latest version of MySQL available in the Ubuntu repositories.

```
sudo apt update
```

Step 2: Install MySQL Server

Install the MySQL server package using the following command:

```
sudo apt install mysql-server
```

Step 3: Secure MySQL Installation

After installation, it's a good practice to run the security script that comes with MySQL. This script will help you set a root password, remove test databases, and apply other security settings.

Run the security script:

```
sudo mysql_secure_installation
```

Follow the prompts:

1. You'll be asked to configure the **VALIDATE PASSWORD PLUGIN**. This plugin helps improve MySQL user account security by enforcing strong passwords. You can choose to enable or disable it based on your needs.
2. Set a root password if you didn't set one during installation.
3. Remove anonymous users (recommended for security).
4. Disallow remote root login (recommended for security).
5. Remove the test database (recommended for security).
6. Reload privilege tables to ensure all changes take effect.

Step 4: Verify MySQL Service

Ensure that the MySQL service is running and enabled to start at boot:

```
sudo systemctl status mysql
```

```
dileep@database:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-09-08 14:38:38 UTC; 6min ago
     Process: 2258 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 2267 (mysqld)
      Status: "Server is operational"
        Tasks: 38 (limit: 9458)
     Memory: 381.7M (peak: 389.0M)
        CPU: 1.597s
      CGroup: /system.slice/mysql.service
             └─2267 /usr/sbin/mysqld

Sep 08 14:38:36 database systemd[1]: Starting mysql.service - MySQL Community Server...
Sep 08 14:38:38 database systemd[1]: Started mysql.service - MySQL Community Server.
dileep@database:~$
```

If it's not running, you can start it with:

```
sudo systemctl start mysql
```

And enable it to start on boot:

```
sudo systemctl enable mysql
```

Step 5: Access MySQL

Log in to the MySQL root account:

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

You'll be prompted to enter the root password you set up during the `mysql_secure_installation` process.

Step 6: Create a New Database and User (Optional)

Once logged into MySQL, you can create a new database and user. Here's an example of how to create a new database and user:

```
-- Create a new database
```

```
CREATE DATABASE mydatabase;
```

```
-- Create a new user with a password
```

```
CREATE USER 'appserver'@'10.0.2.4' IDENTIFIED BY '*';
```



```
-- Grant all privileges on the database to the new user
GRANT ALL PRIVILEGES ON mydatabase.* TO 'appserver'@'10.0.2.4';
```

```
-- Flush privileges to ensure that the changes take effect
FLUSH PRIVILEGES;
```

Step 7: Configure Remote Access (Optional)

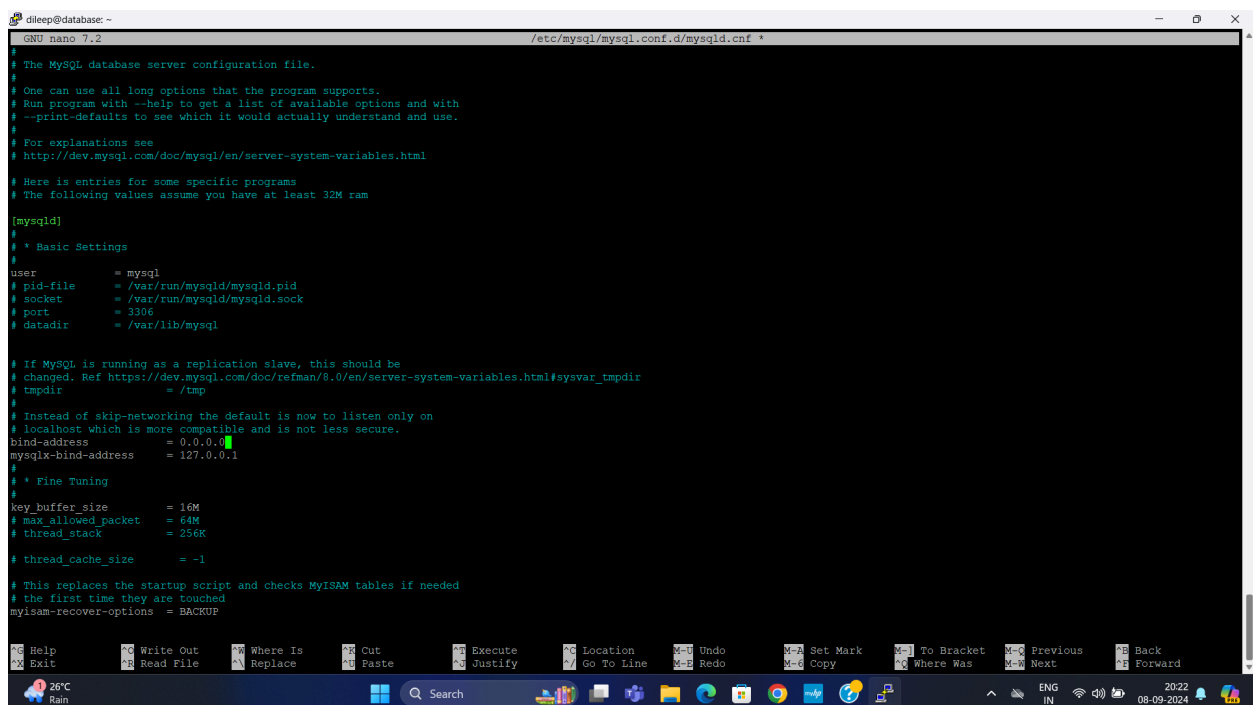
By default, MySQL only accepts connections from `localhost`. To allow remote access, you need to modify the MySQL configuration file.

Open the MySQL configuration file:

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

Find the line that starts with `bind-address` and change it to:

```
bind-address = 0.0.0.0
```



(it's for testing purposes. Not secured if we allow 0.0.0.0. Need to specify ip address of specific machine)

1. This change allows MySQL to accept connections from any IP address. If you want to restrict it to specific IPs, replace `0.0.0.0` with the specific IP address.

Restart MySQL for the changes to take effect:

```
sudo systemctl restart mysql
```

Ensure your firewall allows traffic on MySQL's default port (3306):

```
sudo ufw allow 3306/tcp
```

With these steps, you'll have MySQL installed, configured, and ready to use on your Ubuntu 24.04 system.

Networking and Security

1. Network Setup:

- Use Azure Virtual Network (VNET01) to securely connect the VMs.
- Configure Network Security Groups (NSGs) to control inbound and outbound traffic.
- Ensure that the Nginx server can communicate with the Tomcat server and the Tomcat server can access the MySQL server.


2. Firewall Rules:

- Open necessary ports:
 - Nginx: Typically port 80 (HTTP) and 443 (HTTPS).
 - Tomcat: Typically port 8080 (HTTP) or any custom port you configure.
 - MySQL: Port 3306 (make sure it's restricted to the IP addresses or VMs that need access).

3. Database Configuration:


- Ensure the database server allows connections from the application server's IP address.

Here is the Network security group of Web Server.
Following details as:







 **Network Interface: [webserver243_z1](#)** [Effective security rules](#) [Troubleshoot VM connection issues](#) [Topology](#)

Virtual network/subnet: [VNET01/Webserver](#) NIC Public IP: **20.115.46.170** NIC Private IP: **10.0.1.4** Accelerated networking: **Enabled**


[Inbound port rules](#) [Outbound port rules](#) [Application security groups](#) [Load balancing](#)

 Network security group [webserver-nsg](#) (attached to network interface: [webserver243_z1](#))
Impacts 0 subnets, 1 network interfaces

[Add inbound port rule](#)


Priority	Name	Port	Protocol	Source	Destination	Action	
300	 SSH	22	TCP	Any	Any	 Allow	...
310	AllowAnyHTTPInbound	80	TCP	Any	Any	 Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	 Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	 Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	 Deny	...

Here is the Network security group of App Server.
Following details as:
Port allowed are: 8080 for tomcat server





 **Network Interface: [appserver645_z1](#)** [Effective security rules](#) [Troubleshoot VM connection issues](#) [Topology](#)

Virtual network/subnet: [VNET01/Appserver](#) NIC Public IP: **20.121.128.191** NIC Private IP: **10.0.2.4** Accelerated networking: **Enabled**


[Inbound port rules](#) [Outbound port rules](#) [Application security groups](#) [Load balancing](#)

 Network security group [appserver-nsg](#) (attached to network interface: [appserver645_z1](#))
Impacts 0 subnets, 1 network interfaces

[Add inbound port rule](#)


Priority	Name	Port	Protocol	Source	Destination	Action	
100	AllowCidrBlockCustom8080Inbound	8080	Any	10.0.1.4	10.0.2.4	 Allow	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	 Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	 Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	 Deny	...

Here is the Network security group of Mysql Server.
Following details as:
Port allowed are: 3306 for mysql server
Port deny are: web server to mysql server from any port.






 **Network Interface: [database74_z1](#)** [Effective security rules](#) [Troubleshoot VM connection issues](#) [Topology](#)

Virtual network/subnet: [VNET01/Database](#) NIC Public IP: **51.8.115.141** NIC Private IP: **10.0.3.4** Accelerated networking: **Enabled**

[Inbound port rules](#) [Outbound port rules](#) [Application security groups](#) [Load balancing](#)

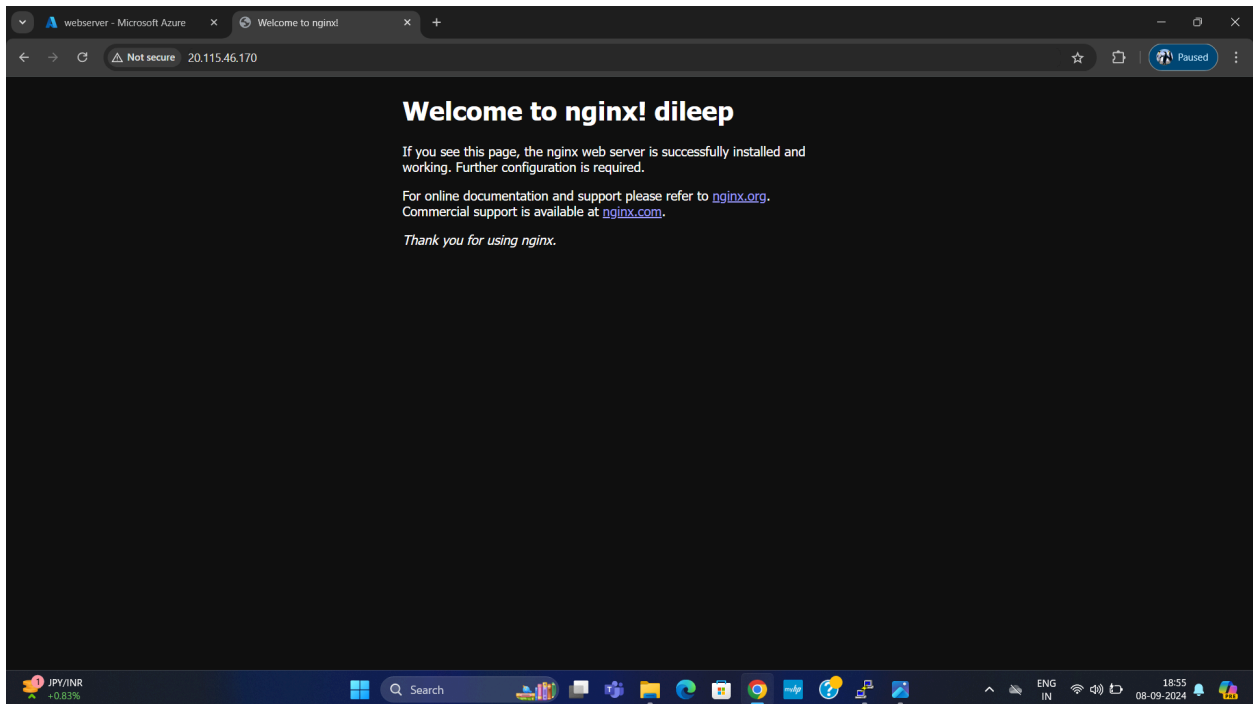
 Network security group [database-nsg](#) (attached to network interface: [database74_z1](#))
Impacts 0 subnets, 1 network interfaces

[Add inbound port rule](#)

Priority	Name	Port	Protocol	Source	Destination	Action	
100	AllowCidrBlockMySQLInbound	3306	TCP	10.0.2.4	10.0.3.4	 Allow	...
110	AllowCidrBlockCustomAnyInbound	Any	Any	10.0.1.4	10.0.3.4	 Deny	...
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	 Allow	...
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	 Allow	...
65500	DenyAllInBound	Any	Any	Any	Any	 Deny	...

Results:

1. Anyone should be able to access webserver on port 80 [it should be accessible via browser]



2. By using telnet command from webserver , Result ☐ should be able to get connect to appserver on port 8080

```
root@webserver:/home/dileep# telnet 10.0.2.4 8080
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
[
```

3. telnet from appserver to db server on port 3306/1433
result should be connected

```
appserver@appserver:~$ telnet 10.0.3.4 3306
Trying 10.0.3.4...
Connected to 10.0.3.4.
Escape character is '^]'.
[
wN(J.Gh%cyHcaching_sha2_password
```

4. By using telnet command from webserver , Result ☐ should not get connect to DbServer on any port

Result ⇒no connection, or connection refused

```
dileep@webserver:~$ telnet 10.0.3.4 3306
Trying 10.0.3.4...
telnet: Unable to connect to remote host: Connection timed out
dileep@webserver:~$ █
```