**NAME : V.SURAJ KUMAR**

**BATCH NO:16**

**EMAIL ID: VSURAJKUMAR2003@GMAIL.COM**

# 3-TIER

# ARCHITECTURE

# 3-TIER ARCHITECTURE

## Tree structure:

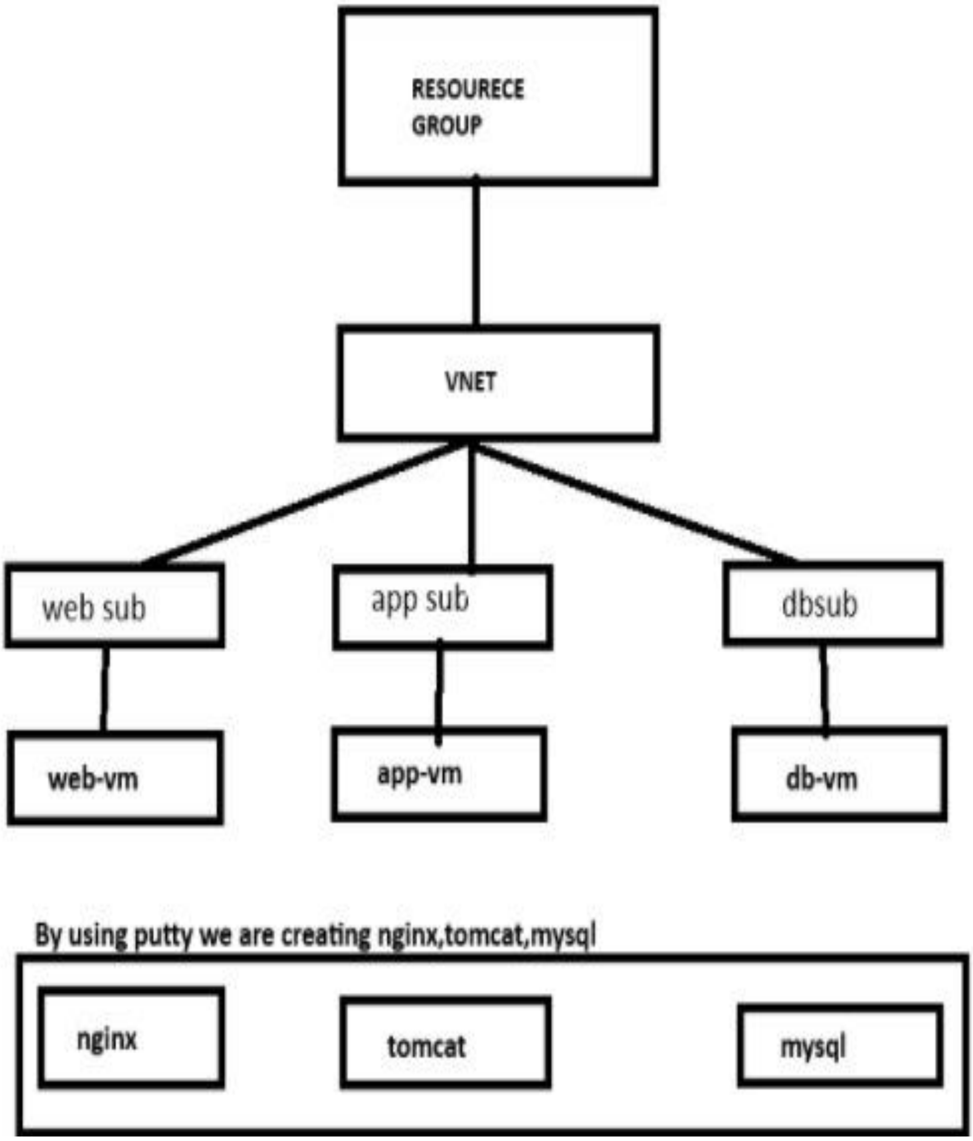
**The below is the tree structure for the 3-tier architecture with a list of**

**1.Resource group**

**2.Virtual network**

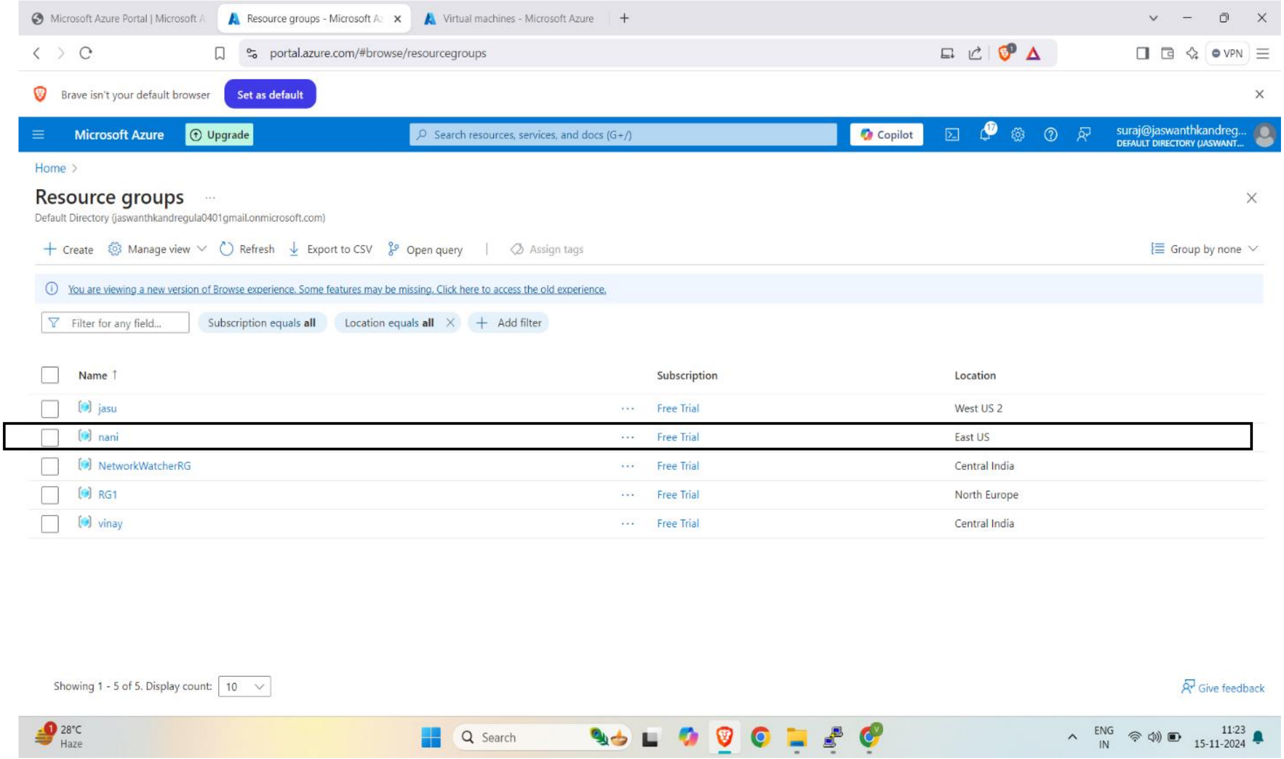
**3.Subnets**

**4.Virtual machines**



## Procedure :

1.create a resource group in any region .My resource group name is nani and region is east us



Resource groups

Default Directory (jaswanthkandregula0401gmail.onmicrosoft.com)

You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

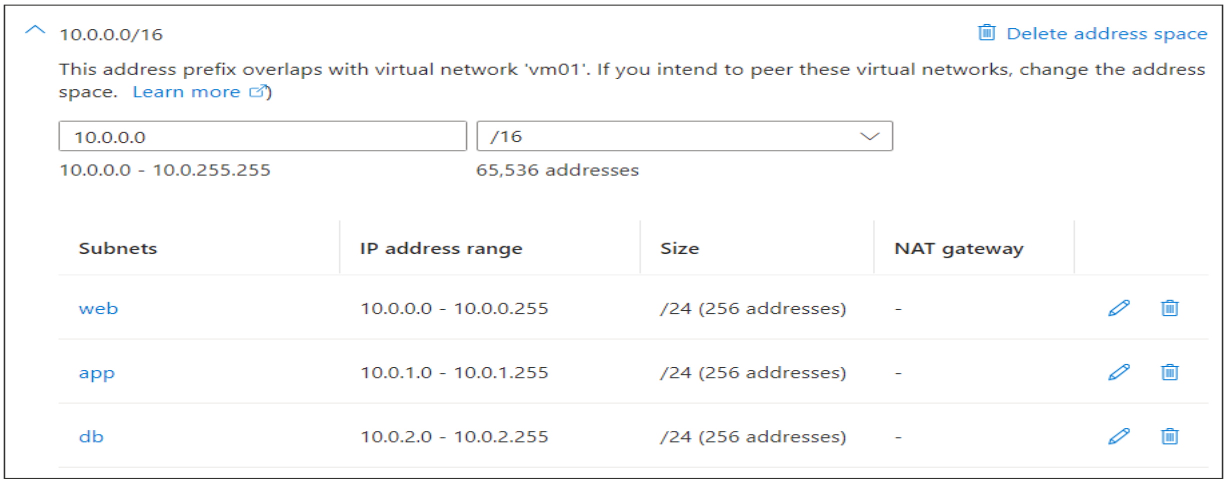
| Name ↑ | Subscription | Location |
|---|---|---|
| jasu | Free Trial | West US 2 |
| nani | Free Trial | East US |
| NetworkWatcherRG | Free Trial | Central India |
| RG1 | Free Trial | North Europe |
| vinay | Free Trial | Central India |

Showing 1 - 5 of 5. Display count: 10

2.Now create a virtual network in with VMN name . Including three subnets within it and name them as

1.websubnet 3.dbsubnet

2.appsubnet



10.0.0.0/16

This address prefix overlaps with virtual network 'vm01'. If you intend to peer these virtual networks, change the address space. Learn more

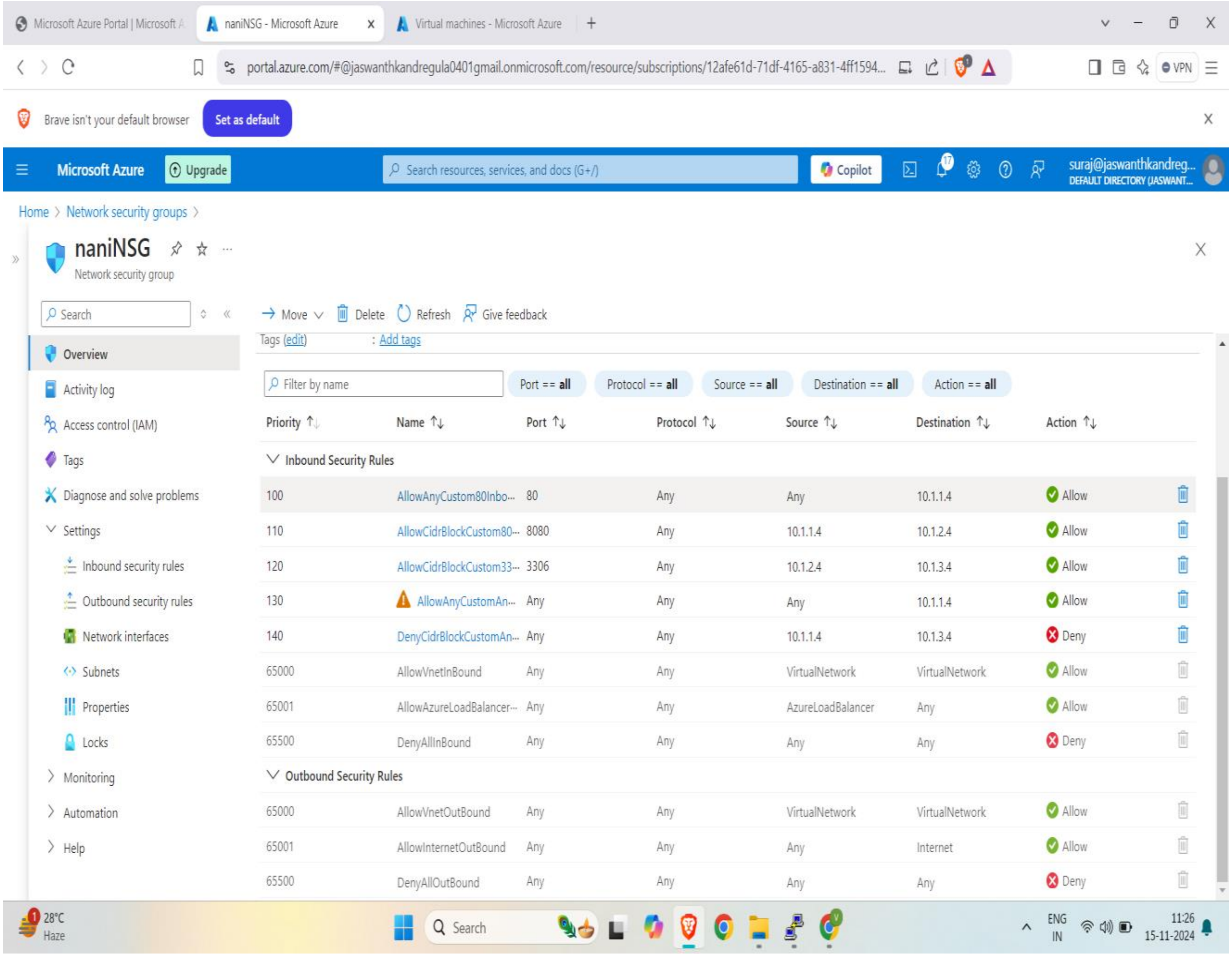
10.0.0.0 /16

10.0.0.0 - 10.0.255.255 65,536 addresses

| Subnets | IP address range | Size | NAT gateway |
|---|---|---|---|
| web | 10.0.0.0 - 10.0.0.255 | /24 (256 addresses) | - |
| app | 10.0.1.0 - 10.0.1.255 | /24 (256 addresses) | - |
| db | 10.0.2.0 - 10.0.2.255 | /24 (256 addresses) | - |

## Create a NSG with rules

- Create NSG with inbound and outbound rules.
- NSG (Network security group) is a basic firewall which allows or denys the traffic .
- NSG is associated at the NIC card level (network interface card),subnet level or both.
- It has 3 inbound and outbound rules by default .
- We need to create extra rules for adding  other virtual machines with port numbers and other details.



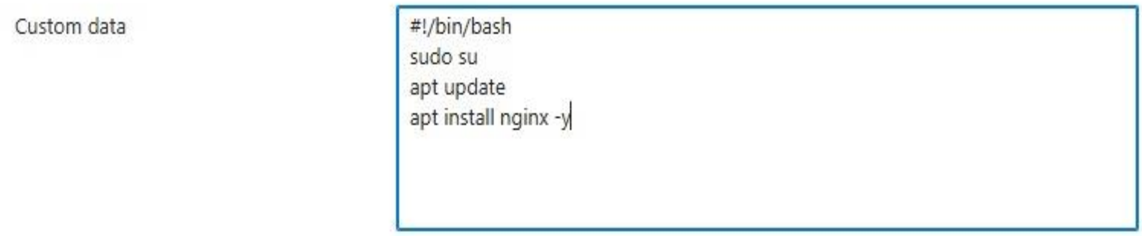
| Priority | Name | Port | Protocol | Source | Destination | Action |
|---|---|---|---|---|---|---|
| Inbound Security Rules | | | | | | |
| 100 | AllowAnyCustom80Inbo... | 80 | Any | Any | 10.1.1.4 | Allow |
| 110 | AllowCidrBlockCustom80... | 8080 | Any | 10.1.1.4 | 10.1.2.4 | Allow |
| 120 | AllowCidrBlockCustom33... | 3306 | Any | 10.1.2.4 | 10.1.3.4 | Allow |
| 130 | ⚠ AllowAnyCustomAn... | Any | Any | Any | 10.1.1.4 | Allow |
| 140 | DenyCidrBlockCustomAn... | Any | Any | 10.1.1.4 | 10.1.3.4 | Deny |
| 65000 | AllowVnetInBound | Any | Any | VirtualNetwork | VirtualNetwork | Allow |
| 65001 | AllowAzureLoadBalancer... | Any | Any | AzureLoadBalancer | Any | Allow |
| 65500 | DenyAllInBound | Any | Any | Any | Any | Deny |
| Outbound Security Rules | | | | | | |
| 65000 | AllowVnetOutBound | Any | Any | VirtualNetwork | VirtualNetwork | Allow |
| 65001 | AllowInternetOutBound | Any | Any | Any | Internet | Allow |
| 65500 | DenyAllOutBound | Any | Any | Any | Any | Deny |

## 3.**Now create virtual machines :web server ,application server, database server.**

## 1.WEB SERVER

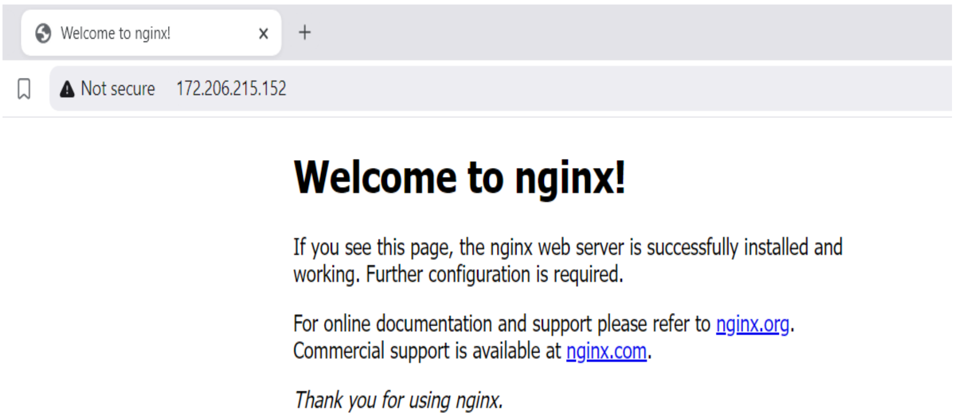
- In web server install nginx.
- In application server install Tomcat application.
- In database server install My SQL.

In web server we need to install nginx . I am writing the below code for installation of nginx in web server in custom data itself while creating .



```
#!/bin/bash
sudo su
apt update
apt install nginx -y
```

- We use nginx as web server for delivering web pages handling HTTP requests, and serving static content .
- Install the web server virtual machine in the web subnet .
- After installation the virtual machine itself gets a public Ip address so that can be accessible over the internet.
- You need to get as below



# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

*Thank you for using nginx.*

## 2.Application server

- Application server must created as same as web server but in advanced code we don’t use nginx commands because we are installing tomcat inside it.
- Application server must be in the app subnet and after creation of application server we get public Ip addresses .
- Open putty tool and add public Ip addresses of application server and then accept the request .
- While creation of virtual machine in terms of authentication we have two types one is password and another on is public key .
- If we use public key then after creation we use to get one public key downloaded in . pem format . Then by using putty gen tool we use to convert the .pem key to .ppk by loading and saving the key.
- Instead we use password authentication method .while creation select password for all virtual machines which is easy accessible by using username and password .
- Add username and password in putty tool after giving public address of app server and check it by using ip :portnumber
- In order to download tomcat follow the below commands.

## COMMANDS FOR TOMCAT INSTALLATION.

Login into ubuntu machine

Step 1 : sudo apt update

Step 2 : sudo apt install default-jdk

Step 3 : java -Version (in order to check whether the java is installed or not)

Step 4 : cd /opt/ (inside this folder create a tomcat directory)

Step 5 : sudo mkdir tomcat

Step 6 : cd ..

Step 7 : cd /tmp/ (where we download the tomcat tar file)

Step 8 : wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.33/bin/apache-tomcat-10.1.33.tar.gz

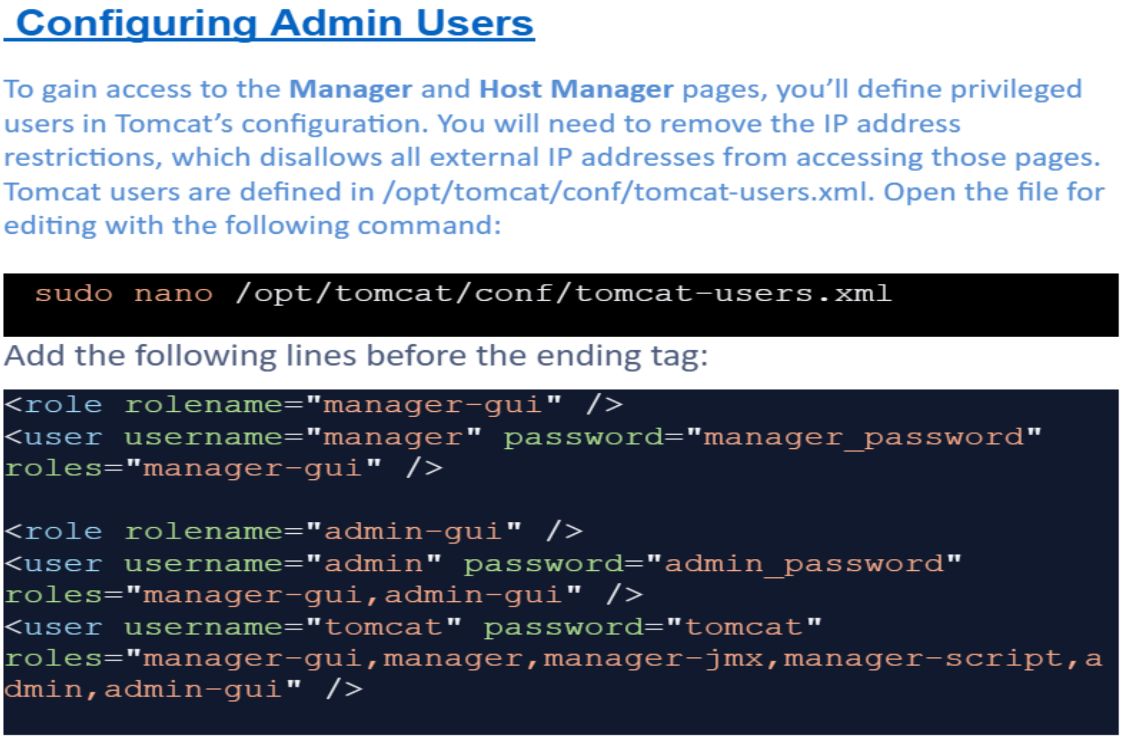
Step 9 : sudo tar xvaf apache-tomcat-10.1.33.tar.gz -C /opt/tomcat --strip-components=1(to unzip the tar file and move the contents of that file into previously created tomcat directory in opt)

Step 10 :sudo useradd -m -d /opt/tomcat -U -s /bin/false tomcat (Run the command to create a user called Tomcat)

Since you have already created a user, you can now
grant tomcat ownership over the extracted installation by running:

1. sudo chown -R tomcat:tomcat /opt/tomcat/

2.sudo chmod -R u+x /opt/tomcat/bin



## Configuring Admin Users

To gain access to the **Manager** and **Host Manager** pages, you'll define privileged users in Tomcat's configuration. You will need to remove the IP address restrictions, which disallows all external IP addresses from accessing those pages. Tomcat users are defined in /opt/tomcat/conf/tomcat-users.xml. Open the file for editing with the following command:

```
 sudo nano /opt/tomcat/conf/tomcat-users.xml
```

Add the following lines before the ending tag:

```
<role rolename="manager-gui" />
<user username="manager" password="manager_password"
roles="manager-gui" />

<role rolename="admin-gui" />
<user username="admin" password="admin_password"
roles="manager-gui,admin-gui" />
<user username="tomcat" password="tomcat"
roles="manager-gui,manager,manager-jmx,manager-script,a
dmin,admin-gui" />
```

By default, Tomcat is configured to restrict access to the admin pages, unless the connection comes from the server itself. To access those pages with the users you just defined, you will need to edit config files for those pages.

To remove the restriction for the **Manager** page, open its config file for editing:

```
sudo nano
  /opt/tomcat/webapps/manager/META-INF/context.xml
```

comment Value line ➔ as which is highlighted in yellow colour

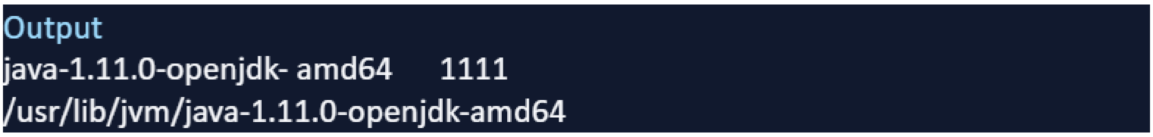
```
<Context antiResourceLocking="false" privileged="true" >

  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"

                   sameSiteCookies="strict" />

<!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"

         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->

  <Manager
sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|or
g\.apache\.catalina\.filters\.Csr>

</Context>
```

## Creating a systemd service

The systemd service that you will now create will keep Tomcat quietly running in the background. The systemd service will also restart Tomcat automatically in case of an error or failure.

Tomcat, being a Java application itself, requires the Java runtime to be present, which you installed with the JDK in step 1. Before you create the service, you need to know where Java is located. You can look that up by running the following command:

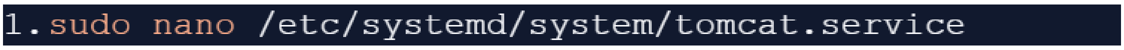
```
sudo update-java-alternatives -l
```



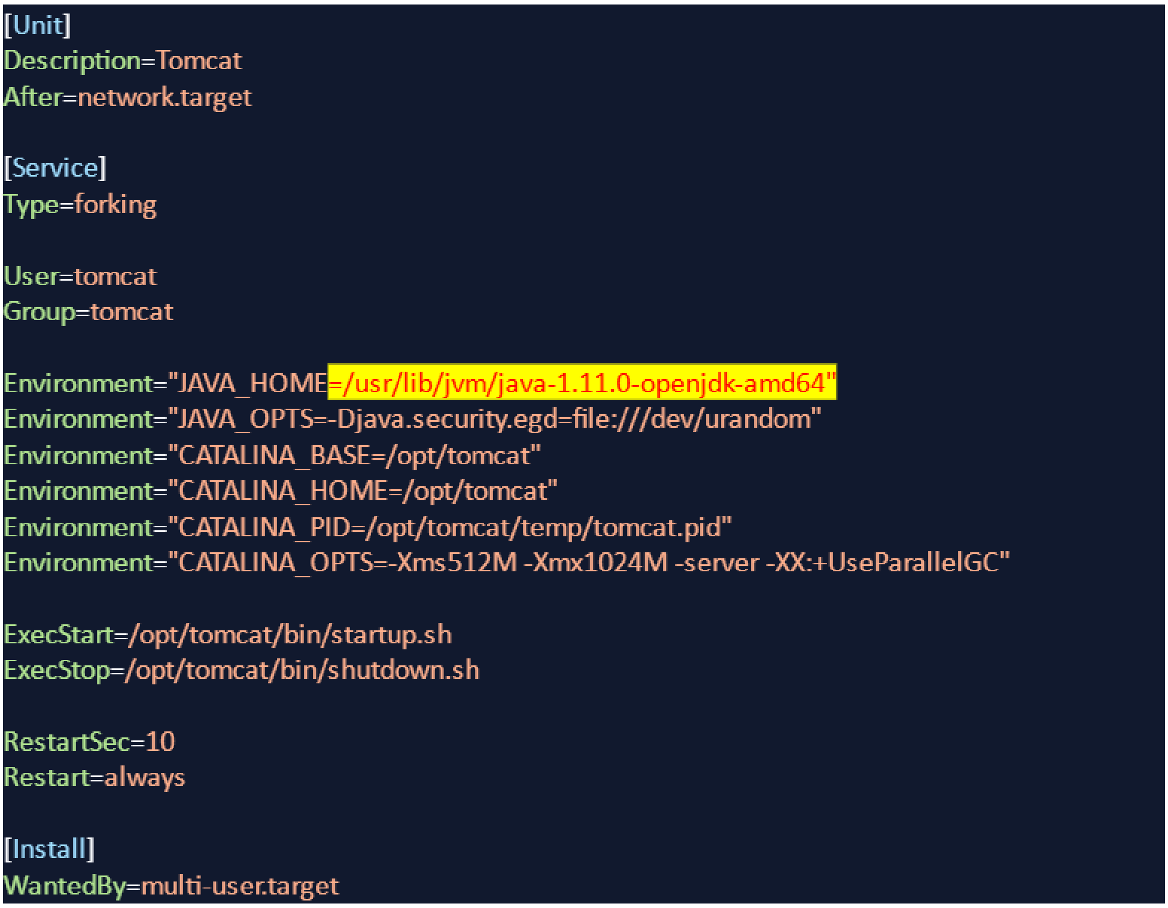
```
Output
java-1.11.0-openjdk- amd64     1111
/usr/lib/jvm/java-1.11.0-openjdk-amd64
```

Note the path where Java resides, listed in the last column. You'll need the path momentarily to define the service.

You'll store the tomcat service in a file named tomcat.service, under /etc/systemd/system. Create the file for editing by running:



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



```
[Unit]
Description=Tomcat
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

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

RestartSec=10
Restart=always

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

Modify the highlighted value of JAVA_HOME if it differs from the one you noted previously.

Here, you define a service that will run Tomcat by executing the startup and shutdown scripts it provides. You also set a few environment variables to define its home directory (which is /opt/tomcat as before) and limit the amount of memory that the Java VM can allocate (in CATALINA_OPTS). Upon failure, the Tomcat service will restart automatically.

When you’re done, save and close the file.

## 3.Database server

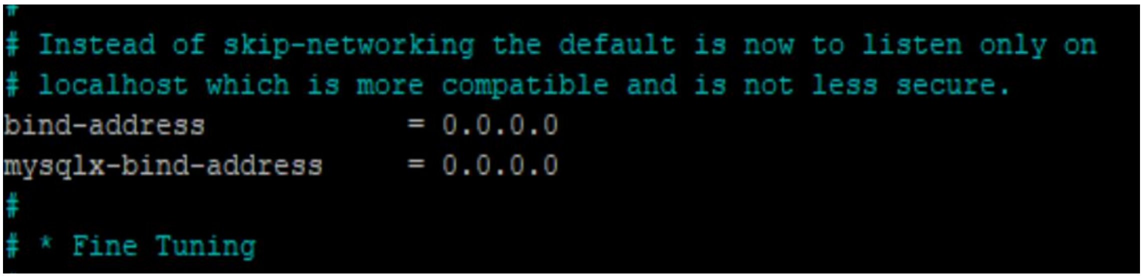
- create database virtual machine under db subnet .
- As same as application server you need not to add code in the advanced .
- Login into putty tool by using public Ip address you get after creating virtual machine of data server.
- After login add username and password in the putty as you mentioned in while creation of virtual machine.
- After login follow the below commands for installation of my SQL in the server.
- Sudo su (to enter into root user).

### MY SQL installation process

Install MY SQL DB on Linux machine using the below commands:

1. **apt update** —update the machine
2. **apt install mysql-server -y** ---- install my sql server
3. **systemctl start mysql.service** --- start the my sql server
4. **systemctl status mysql** ---- check the status whether it is in active &amp; running or not
5. **systemctl enable mysql** ----- enable my sql in your system
6. **mysql_secure_installation**
7. y
8. y
9. 2

10. Y
11. Y
12. Y
13. y
14. mysql --- login into the mysql &amp; check the , exit from my sql
15. ctrl+z
16. nano /etc/mysql/mysql.conf.d/mysqld.cnf -----◊edit the file as (blind-address&amp- 0.0.0.0&mysqlx-blind-address- 0.0.0.0)



```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 0.0.0.0
#
# * Fine Tuning
```

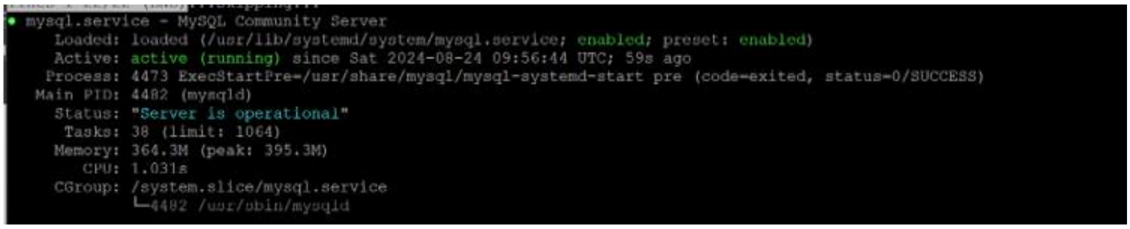
After this do ctrl+x to save
Yes-y
enter

17.systemctl restart mysql -restart my sql

18.service mysql restart -- restart my sql service

19.systemctl status mysql.service --check the status

20.after successful installation you will be able to see the putty console like below.



```
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-08-24 09:56:44 UTC; 59s ago
    Process: 4473 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 4482 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1064)
     Memory: 364.3M (peak: 395.3M)
        CPU: 1.031s
     CGroup: /system.slice/mysql.service
             └─4482 /usr/sbin/mysqld
```