***Cloud Project***

***create 3 Tier architecture:*** *to create the 3 tier architecture we nee to install the 3 Virtual machines*

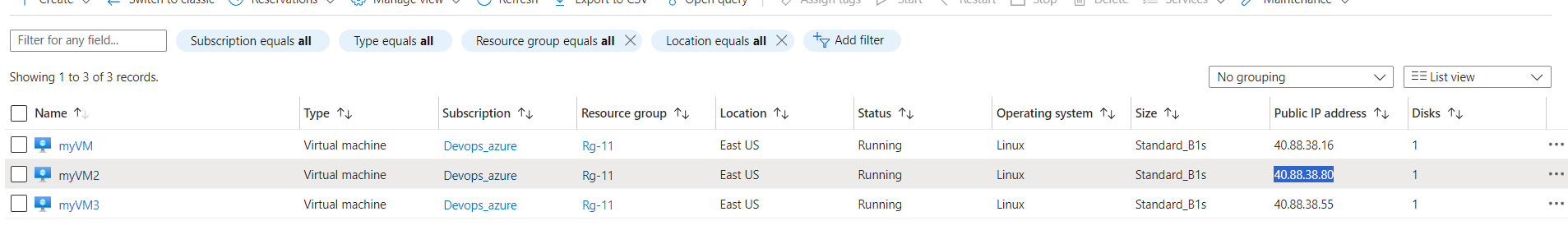
*one is for web server*

*one is for app server*

*one is for DB server*

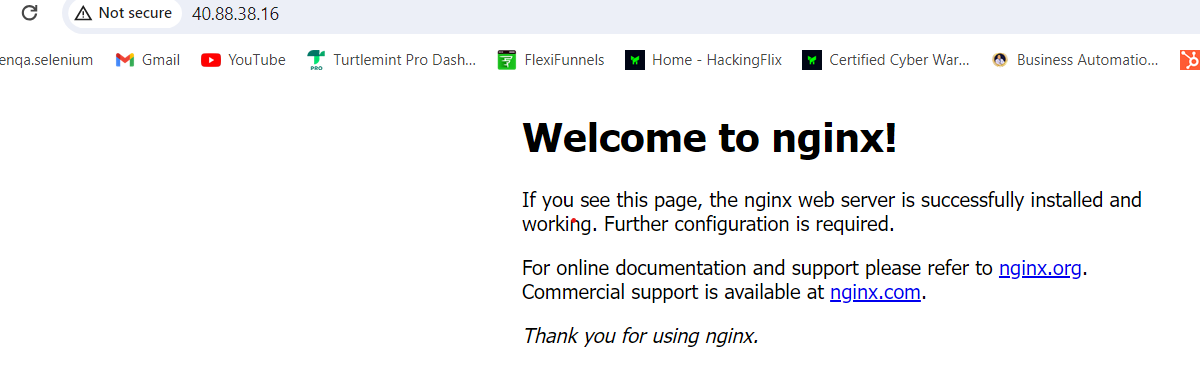
*Here I installed the 3 virtual machines using the terraform code & installed the nginx on all the 3 machines which are working fine.*

*Please refer the screenshot for the virtual machines status*

**

***Webserver:*** *consider vm-1 as web server(primary machine)*

*Linux machine installed which has the ip address 0f http://40.88.38.16/ & working fine*

**

***Application server*** *: consider vm-2 as APP server(secondary machine)*

*Next login to putty and install the Tomcat on Ubuntu machine, tomcat application will run on Port 8080*

*Use the Below commands to install tomcat on your machine:*

*Login into ubuntu machine*

*# sudo apt update*

*#sudo apt install default-jdk*

*#java -version*

*First, navigate to the /tmp directory:*

*# cd /tmp*

*# wget* [*https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.20/bin/apache-tomcat-10.1.20.tar.gz*](https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.20/bin/apache-tomcat-10.1.20.tar.gz)

*Then, extract the archive you downloaded by running:*

1. *sudo tar xzvf apache-tomcat-10\*tar.gz -C /opt/tomcat --strip-components=1*

*For security purposes, Tomcat should run under a separate, unprivileged user. Run the following command to create a user called tomcat:*

1. *sudo useradd -m -d /opt/tomcat -U -s /bin/false 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/*

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

## [*Step 2 — Configuring Admin Users*](file:///C:\Users\User\Downloads\TOMCAT%20INSTALLATION%20ON%20UBUNTU.docx#step-2-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:*

1. *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,admin,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:*

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

*comment Value line è 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)|org\.apache\.catalina\.filters\.Csr>*

*</Context>*

## [*Step 3 — Creating a systemd service*](file:///C:\Users\User\Downloads\TOMCAT%20INSTALLATION%20ON%20UBUNTU.docx#step-3-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:*

1. *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.*

*Reload the systemd daemon so that it becomes aware of the new service:*

*sudo systemctl daemon-reload*

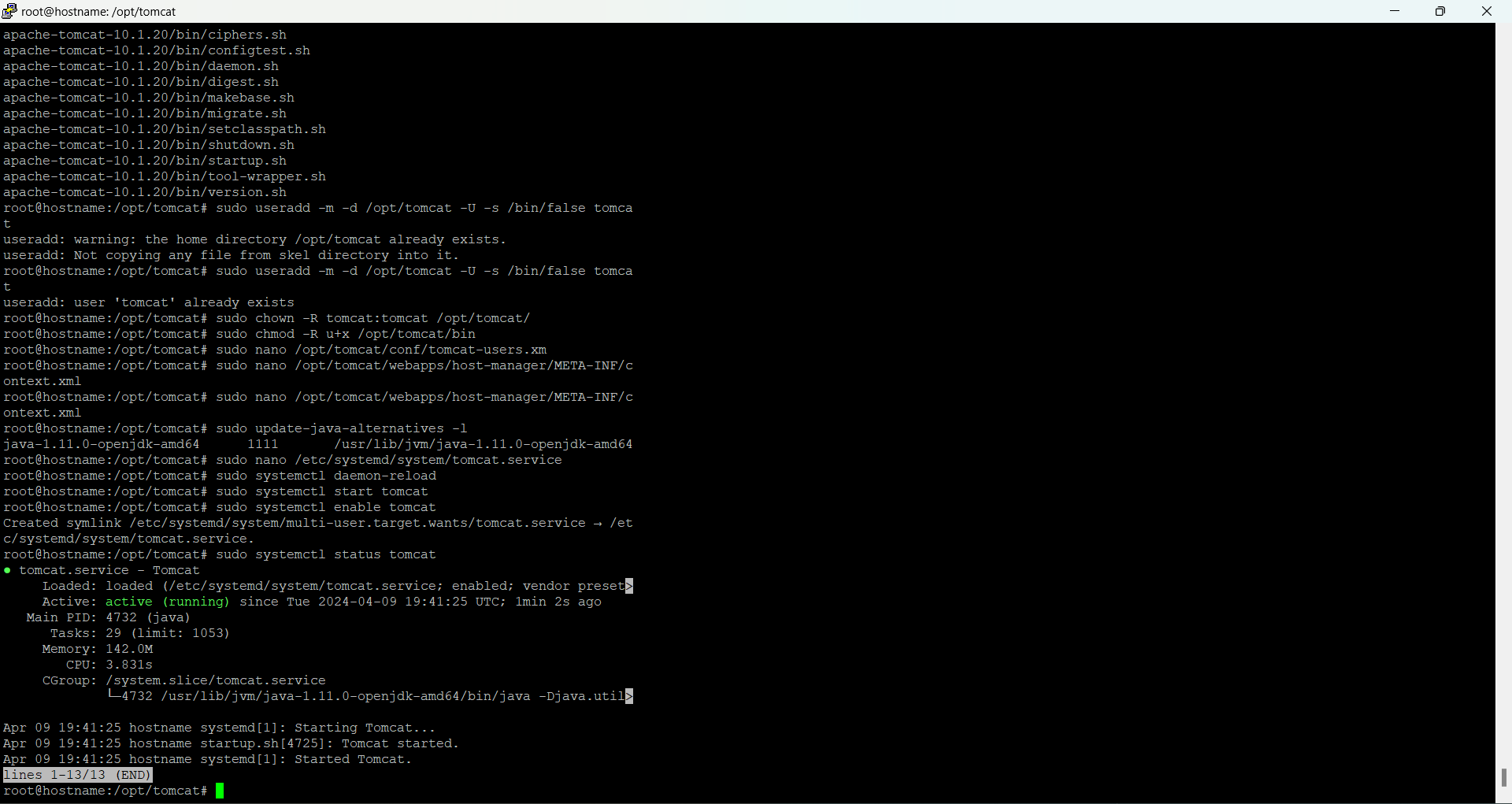
*sudo systemctl start tomcat*

*Then, look at its status to confirm that it started successfully:  
  
sudo systemctl status tomcat*

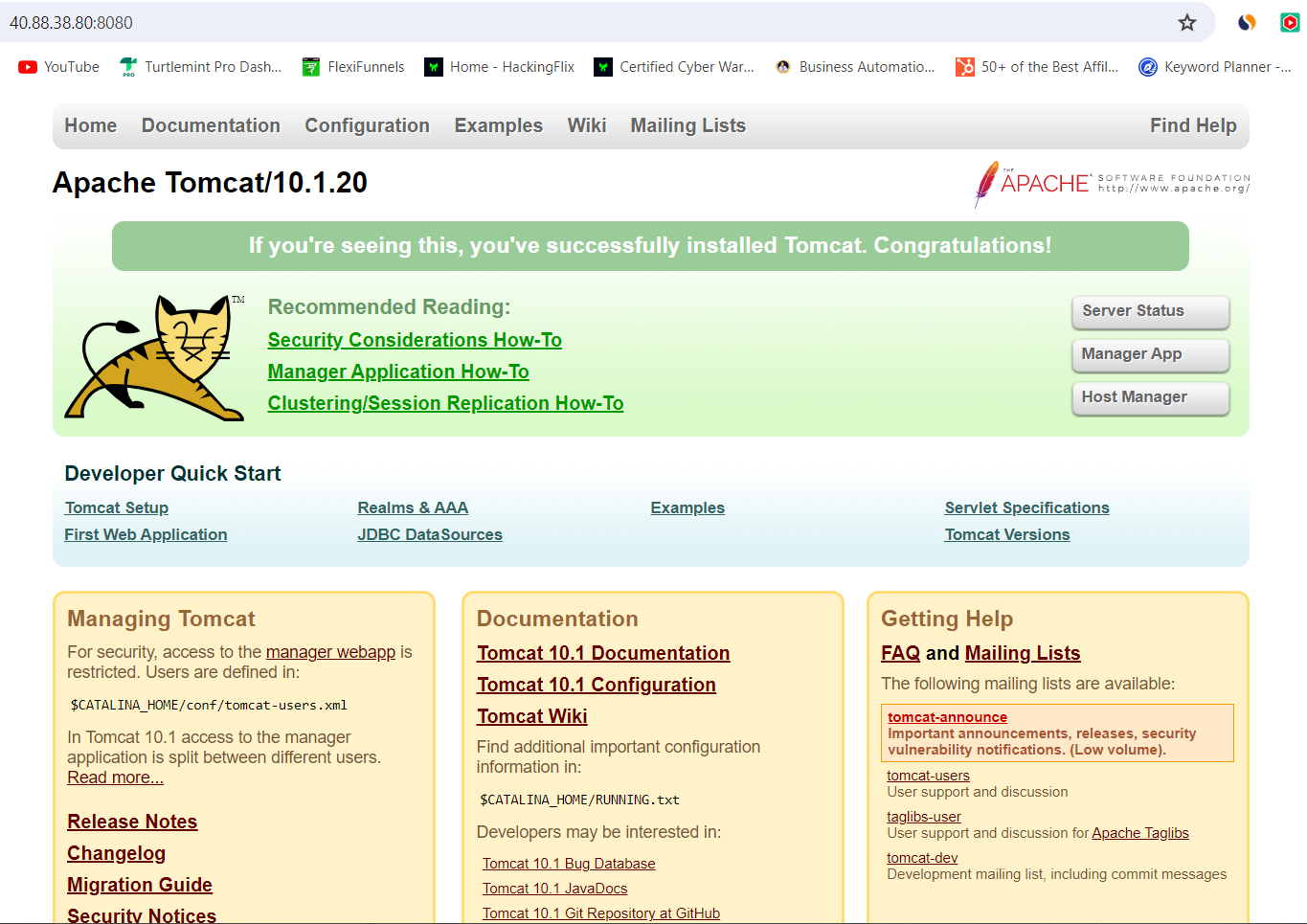
*To enable Tomcat starting up with the system, run the following command:*

1. *sudo systemctl enable tomcat*

*you will get the putty console like this*

**

*after successful installation you will be able to see the output like below screen shot*

**

***DB Server*** *: consider vm-3 as DB server(Third machine)*

*install My Sql DB on Linux machine using the below commands*

*sudo apt update—update the machine*

*sudo apt install mysql-server---- install my sql server*

*sudo systemctl start mysql.service--- start the my sql server*

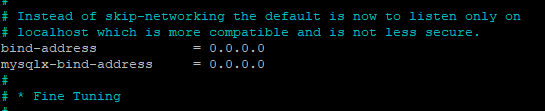
*sudo systemctl status mysql---- check the status whether it is in active & running or not*

*sudo systemctl enable mysql ----- enable my sql in your system*

*sudo mysql\_secure\_installation*

*sudo mysql--- login into the mysql & check the , exit from my sql*

*sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf-----🡪edit the file (blind-address& mysqlx-blind-address to 0.0.0.0)*

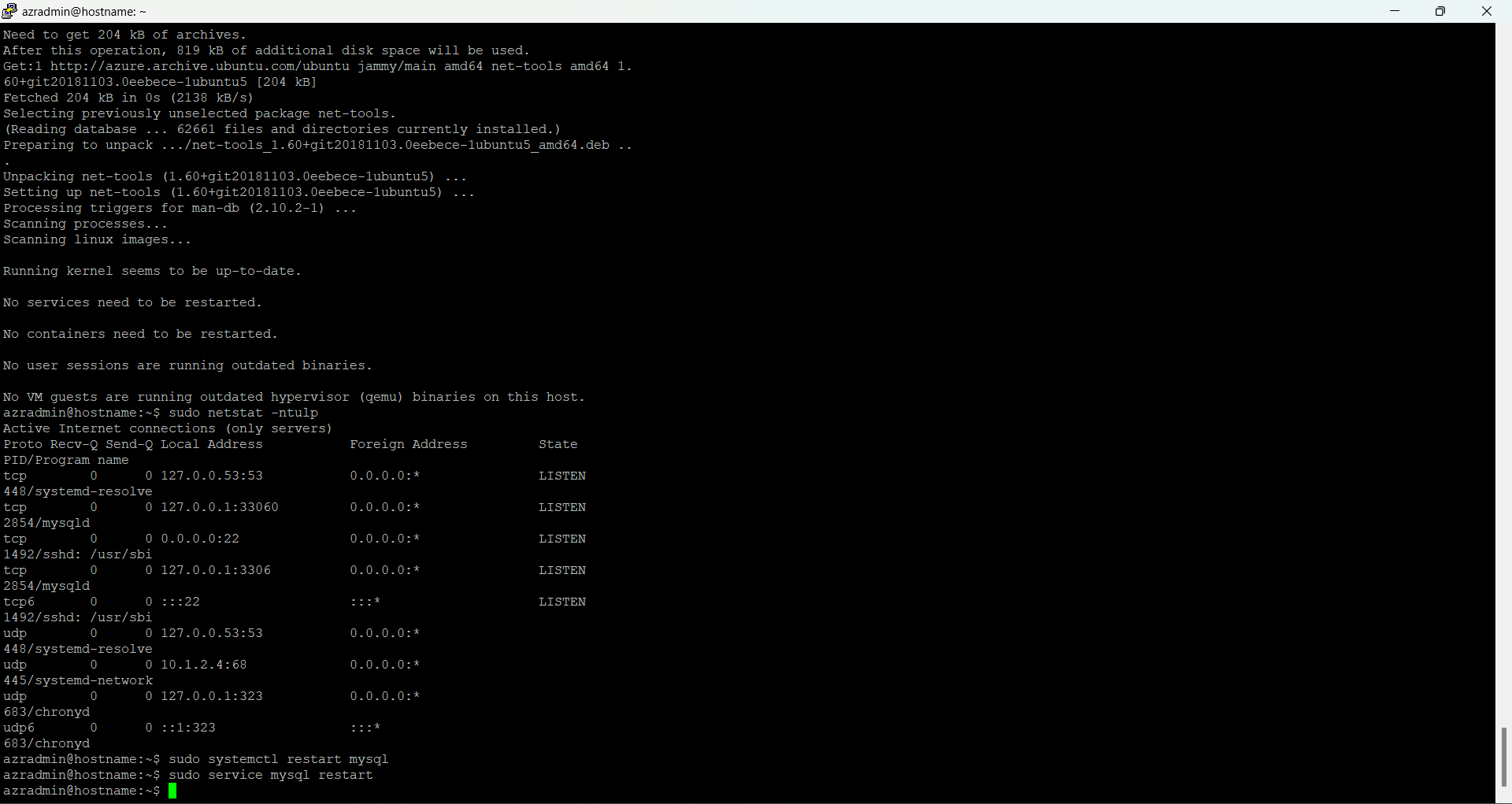
**

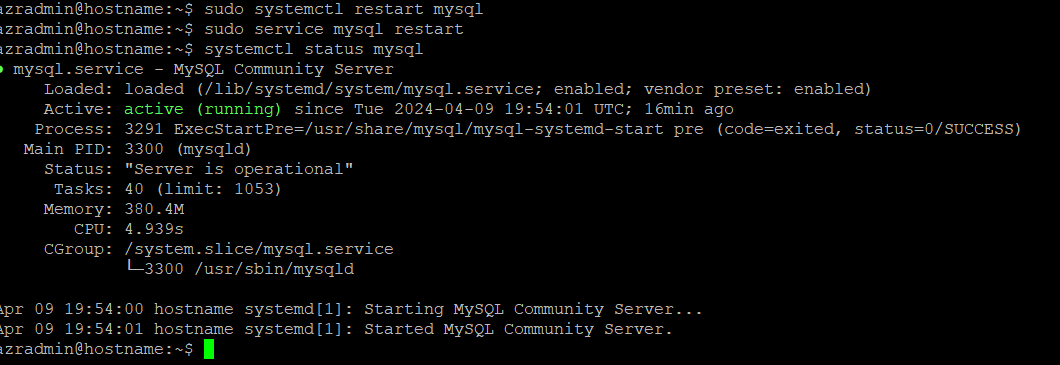
*sudo systemctl restart mysql-🡪restart my sql*

*sudo service mysql restart--🡪 restart my sql service*

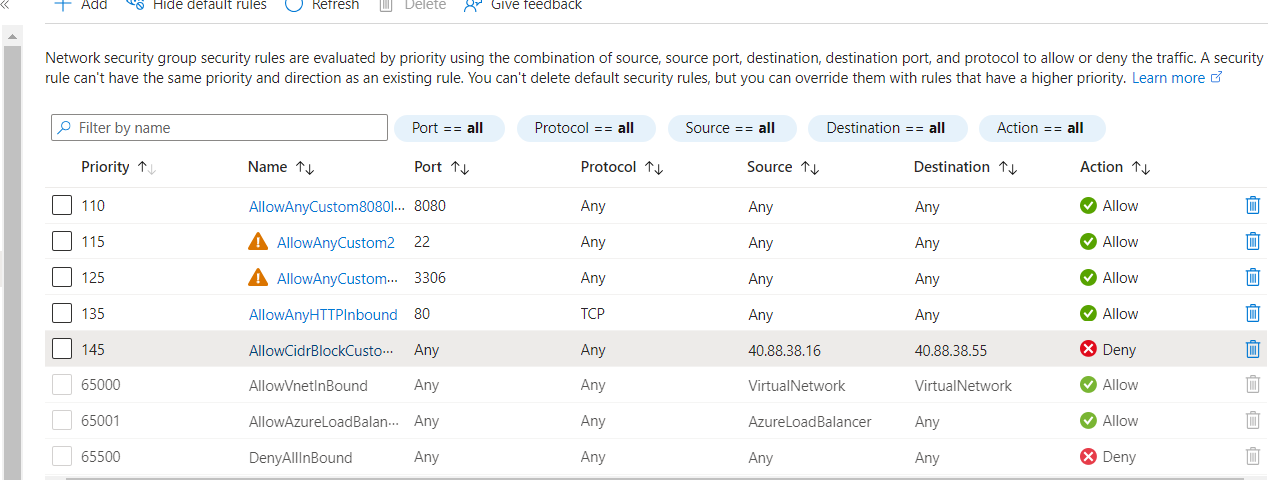
*systemctl status mysql.service--🡪check the status*

*after successful installation you will be able to se the putty console like below*

**

**

*now go to your azure portal write the NSG rules as below as per your requirement*

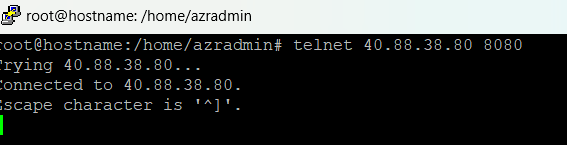
**

*Finally verifiy the out put using the telenet<ipaddress> <port number>*

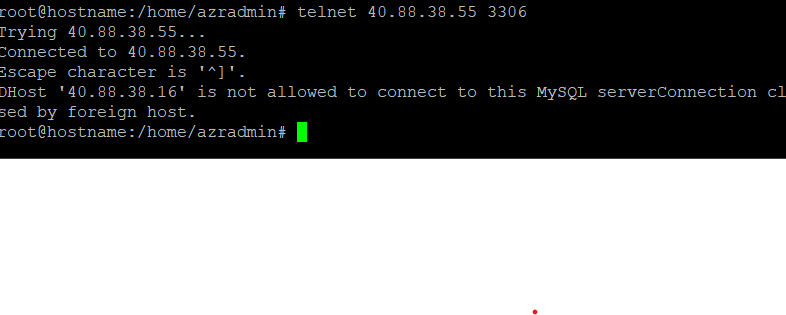
*Results:*

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

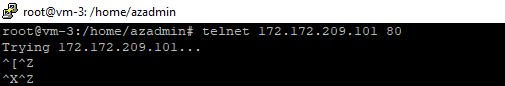
1. *By using telnet command from webserver , I should be able to get connect to appserver on port 8080*

**

1. *By using telnet command from APPserver , I should be able to get connect to DbServer on port 3306*

**

1. *By using telnet command from DBserver , I should not able to get connect to Web Server on port 80*

**