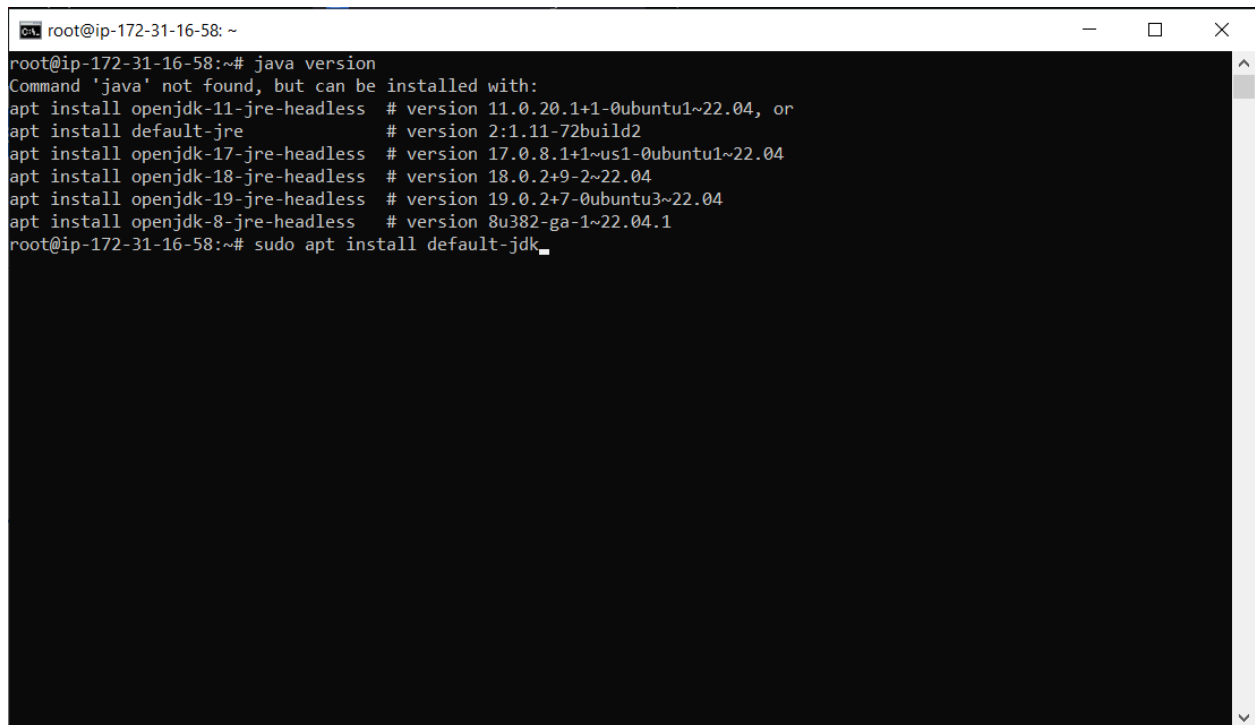


TOMCAT SERVER SETUP AND HOSTING A WEB-APPLICATION

Step 1 : Install Java

To set up the tomcat server we first need to install java on our machine. Check if java is installed on the machine using the following command.

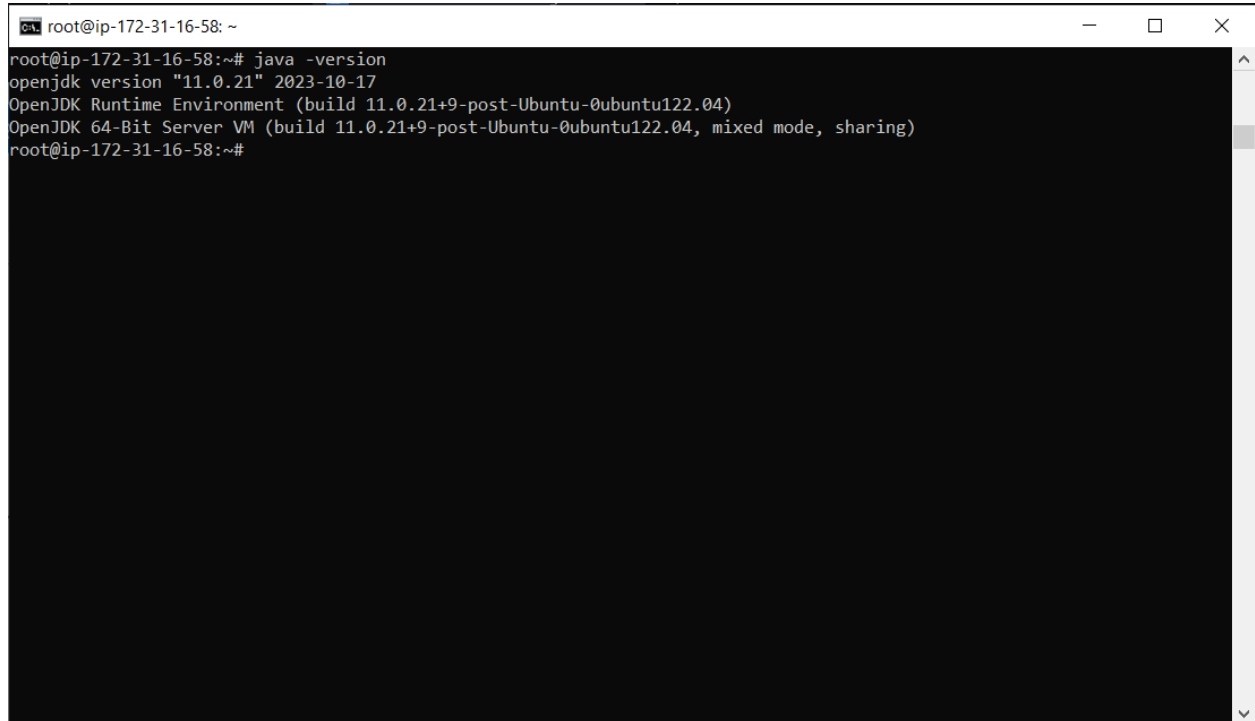
```
java -version
```



```
root@ip-172-31-16-58: ~  
root@ip-172-31-16-58:~# java version  
Command 'java' not found, but can be installed with:  
apt install openjdk-11-jre-headless # version 11.0.20.1+1-0ubuntu1~22.04, or  
apt install default-jre # version 2:1.11-72build2  
apt install openjdk-17-jre-headless # version 17.0.8.1+1~us1-0ubuntu1~22.04  
apt install openjdk-18-jre-headless # version 18.0.2+9-2~22.04  
apt install openjdk-19-jre-headless # version 19.0.2+7-0ubuntu3~22.04  
apt install openjdk-8-jre-headless # version 8u382-ga-1~22.04.1  
root@ip-172-31-16-58:~# sudo apt install default-jdk
```

If java is not installed, run the following command in the terminal to install the default-jdk package.

```
apt-get install java* -y
```

A terminal window with a black background and white text. The window title is 'root@ip-172-31-16-58: ~'. The command 'java -version' has been executed, resulting in the following output: 'openjdk version "11.0.21" 2023-10-17', 'OpenJDK Runtime Environment (build 11.0.21+9-post-Ubuntu-0ubuntu122.04)', and 'OpenJDK 64-Bit Server VM (build 11.0.21+9-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)'.

```
root@ip-172-31-16-58: ~  
root@ip-172-31-16-58:~# java -version  
openjdk version "11.0.21" 2023-10-17  
OpenJDK Runtime Environment (build 11.0.21+9-post-Ubuntu-0ubuntu122.04)  
OpenJDK 64-Bit Server VM (build 11.0.21+9-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)  
root@ip-172-31-16-58:~#
```

Step 2 : Install Tomcat server

Run following command to download Tomcat archive from internet

```
curl -O https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.tar.gz
```

Now extract the downloaded .tar.gz file using following command

```
tar -xvzf apache-tomcat-8.5.99.tar.gz -C /opt/
```

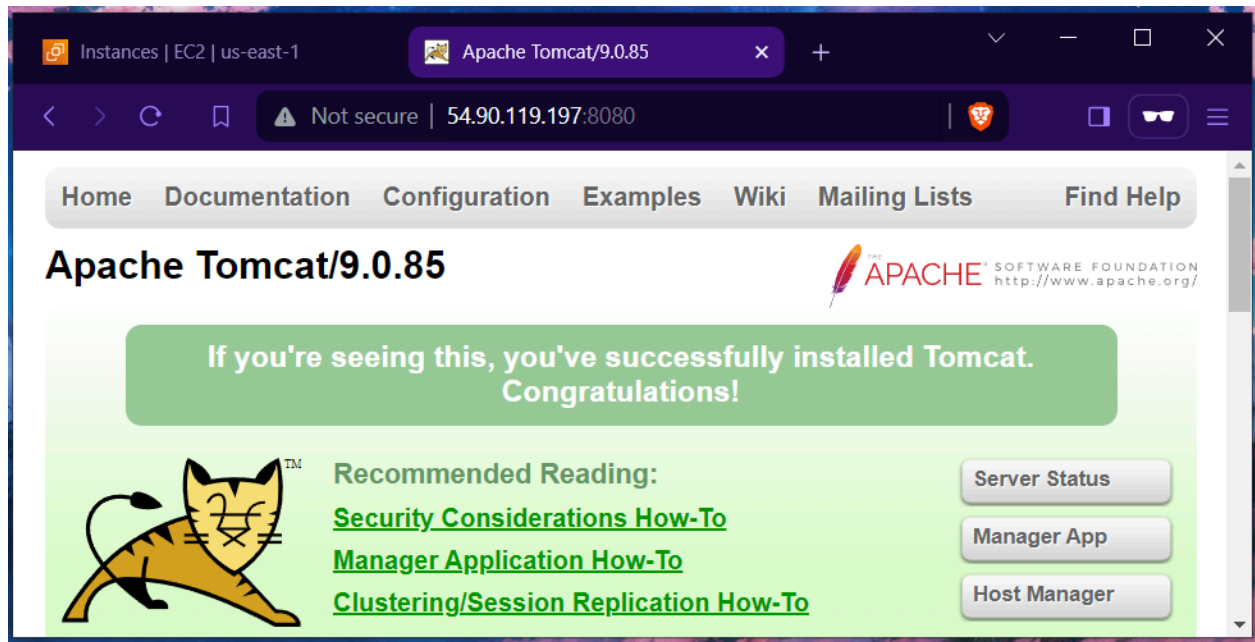
Use the following command to start Tomcat service in bin folder of the apache-tomcat directory.

```
./catalina.sh start
```

```
root@ip-172-31-45-226: ~  
root@ip-172-31-45-226:~# curl -O https://d1cdn.apache.org/tomcat/tomcat-8/v8.5.99/bin/apache-tomcat-8.5.99.tar.gz  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
           % Dload  % Upload   Total   Spent    Left   Speed  
100 10.3M  100 10.3M    0     0    50.1M      0  --:--:-- --:--:-- --:--:-- 50.3M  
root@ip-172-31-45-226:~# ls  
apache-tomcat-8.5.99.tar.gz  snap  
root@ip-172-31-45-226:~#
```

```
root@ip-172-31-16-58: /opt/tomcat  
root@ip-172-31-16-58:/opt/tomcat# systemctl daemon-reload  
root@ip-172-31-16-58:/opt/tomcat# systemctl start tomcat  
root@ip-172-31-16-58:/opt/tomcat# systemctl status tomcat  
● tomcat.service - Apache Tomcat Web Application Container  
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)  
   Active: active (running) since Thu 2024-02-15 15:03:04 UTC; 4s ago  
     Process: 17018 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)  
    Main PID: 17025 (java)  
      Tasks: 29 (limit: 1121)  
     Memory: 131.1M  
        CPU: 3.615s  
    CGroup: /system.slice/tomcat.service  
            └─17025 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties  
Feb 15 15:03:04 ip-172-31-16-58 systemd[1]: Starting Apache Tomcat Web Application Container...  
Feb 15 15:03:04 ip-172-31-16-58 startup.sh[17018]: Tomcat started.  
Feb 15 15:03:04 ip-172-31-16-58 systemd[1]: Started Apache Tomcat Web Application Container.  
lines 1-14/14 (END)
```

Now hit the tomcat server using the public IP of the instance.



Step 3 : Hosting the web-application

There are certain conditions to host the web-application on the tomcat server. We need to put all files in their respective environment on tomcat server to successfully host the application.

According to the tomcat server environment all .jar files goes in *lib* directory and all .war files goes in *webapps* directory. We have the files with names student.war and mysql-connector.jar.

To put these files in their respective locations we first need those files on our EC2 instance. This can be done by running following commands in the terminal of our local system.

```
Windows PowerShell

PS C:\Users\Dell\Downloads> scp -i id_rsa student.war ubuntu@54.90.119.197:/home/ubuntu
student.war                                     100% 87KB 28.3KB/s 00:03
PS C:\Users\Dell\Downloads> scp -i id_rsa mysql-connector.jar ubuntu@54.90.119.197:/home/ubuntu
mysql-connector.jar                             100% 984KB 364.7KB/s 00:02
PS C:\Users\Dell\Downloads>
```

Now that we have required files on our EC2 instance, we can copy them to their desired locations.

```
root@ip-172-31-16-58: /home/ubuntu# ls
mysql-connector.jar  student.war
root@ip-172-31-16-58: /home/ubuntu# mv mysql-connector.jar /opt/tomcat/lib/
root@ip-172-31-16-58: /home/ubuntu# mv student.war /opt/tomcat/webapps/
root@ip-172-31-16-58: /home/ubuntu# ls /opt/tomcat/lib/ | grep mysql-connector.jar
mysql-connector.jar
root@ip-172-31-16-58: /home/ubuntu# ls /opt/tomcat/webapps/ | grep student.war
student.war
root@ip-172-31-16-58: /home/ubuntu#
```

After completing the entire process we can access our web-application using the public IP of our EC2 instance.

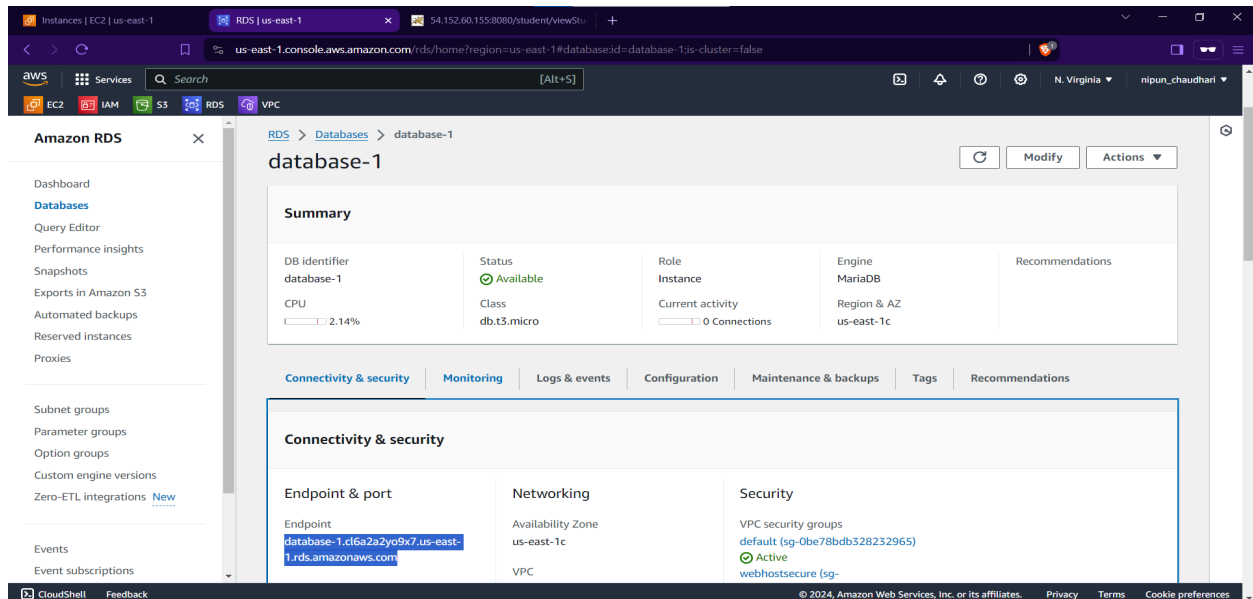
The screenshot shows a web browser window with the title 'Instances | EC2 | us-east-1' and a tab labeled 'User Data'. The address bar displays 'Not secure | 54.90.119.197:8080/student/'. The main content area features a 'Student Registration Form' with the following fields and a button:

Student Name	<input type="text"/>
Student Address	<input type="text"/>
Student Age	<input type="text"/>
Student Qualification	<input type="text"/>
Student Percentage	<input type="text"/>
Year Passed	<input type="text"/>
<input type="button" value="register"/>	

This is the entire process of hosting the web-application on the tomcat server.

Step 4 : Setting connection with database.

First create a database in Amazon RDS(Relational Database Service).



Copy the highlighted endpoint, and paste the following content in /opt/apache-tomcat-8.5.99/conf/context.xml file. Change the endpoint with your database endpoint.

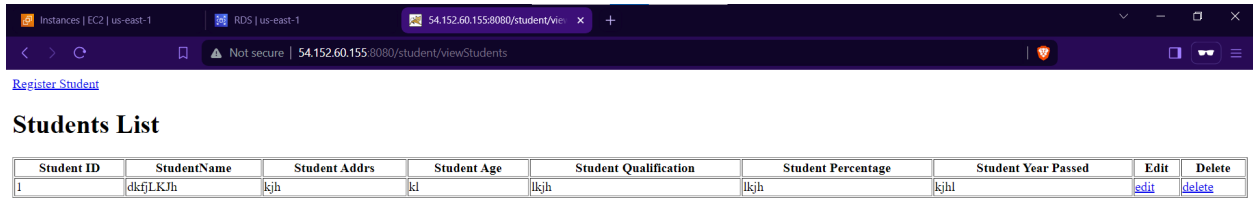
```
root@ip-172-31-45-226: /opt/apache-tomcat-8.5.99# cat conf/context.xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context>
    <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
        maxTotal="500" maxIdle="30" maxWaitMillis="1000"
        username="admin" password="password" driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://database-1.cl6a2a2yo9x7.us-east-1.rds.amazonaws.com:3306/studentapp"/>

    <!-- Default set of monitored resources. If one of these changes, the
    <!-- web application will be reloaded. -->
```

Now, after setting up the database connection, restart the tomcat server using catalina.sh file. And check whether your application is hosted properly or not.



[Register Student](#)

Students List

Student ID	StudentName	Student Addr	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	dkfjLKJh	kjh	kl	lkjh	lkjh	kjhl	edit	delete