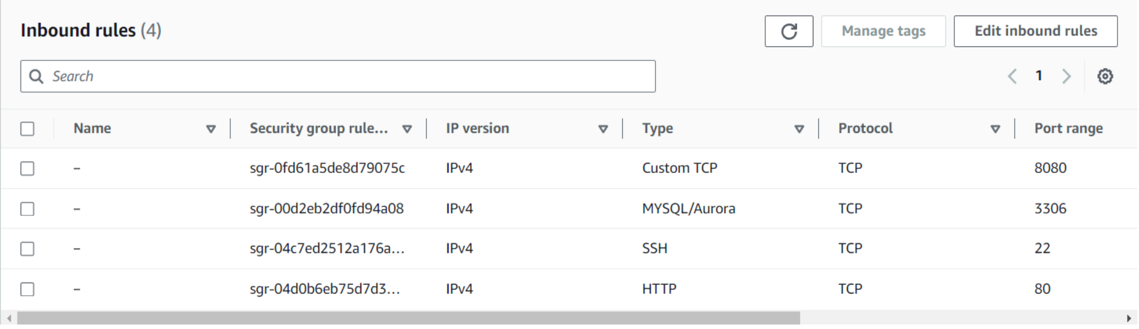
AWS Project

Description: -

"In this project, I developed a student registration form to capture and store student information. The backend is powered by AWS RDS (Relational Database Service) to manage and store the data securely. I deployed the website using an Apache Tomcat server, ensuring smooth hosting and accessibility of the application. This project demonstrates my ability to integrate AWS services for database management and web hosting, creating a seamless full-stack solution."

Steps:

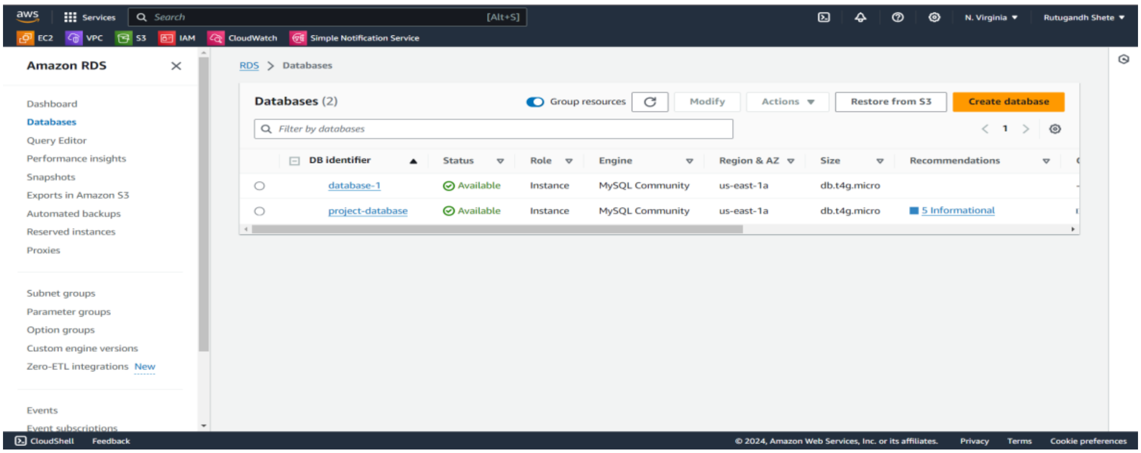
- Create one RDS by normal process. (refer RDS doc). While creating attach same security group to RDS and instance, select same availability zone.
- Add this inbound rule to security group. **Port 8080** is the default port used by Apache Tomcat to listen for HTTP requests. When you access a web application hosted on Tomcat, you typically use http://localhost:8080 (or the server's IP address with port 8080).



Inbound rules (4)

□	Name	Security group rule...	IP version	Type	Protocol	Port range
□	–	sgr-0fd61a5de8d79075c	IPv4	Custom TCP	TCP	8080
□	–	sgr-00d2eb2df0fd94a08	IPv4	MYSQL/Aurora	TCP	3306
□	–	sgr-04c7ed2512a176a...	IPv4	SSH	TCP	22
□	–	sgr-04d0b6eb75d7d3...	IPv4	HTTP	TCP	80

- After creating databases.



Amazon RDS

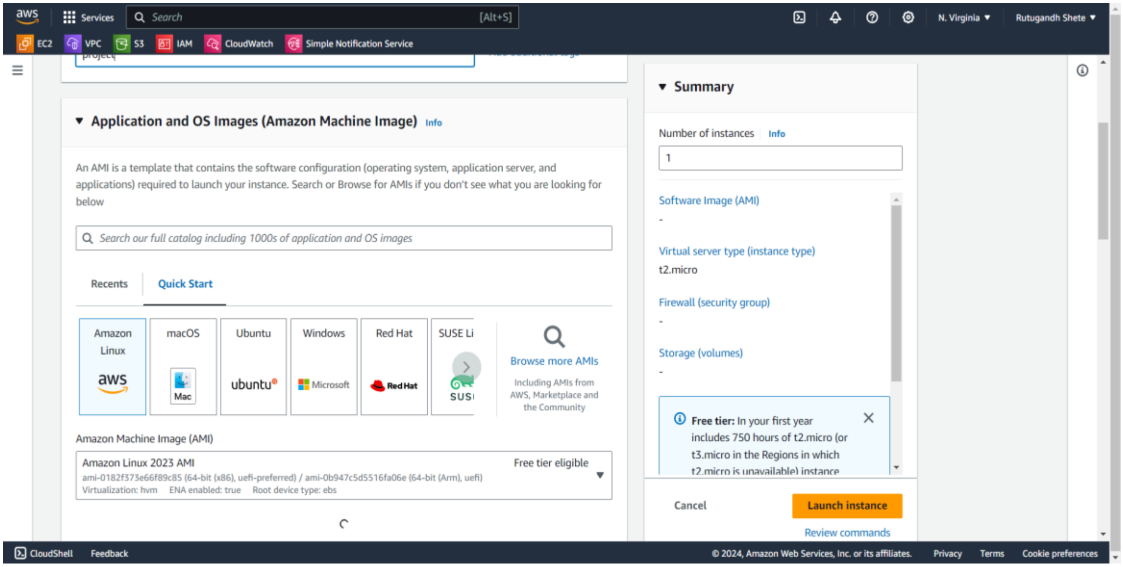
- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations New
- Events
- Event subscriptions

RDS > Databases

Databases (2)

	DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations
○	database-1	Available	Instance	MySQL Community	us-east-1a	db.t4g.micro	-
○	project-database	Available	Instance	MySQL Community	us-east-1a	db.t4g.micro	5 Informational

- Now create one instance→select Amazon machine →select subnet of same availability zone→select same security that we have attached to RDS.



Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0182f373e66f89c85 (64-bit (x86), uefi-preferred) / ami-0b947c5d5516fa06e (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances: 1

Software Image (AMI): -

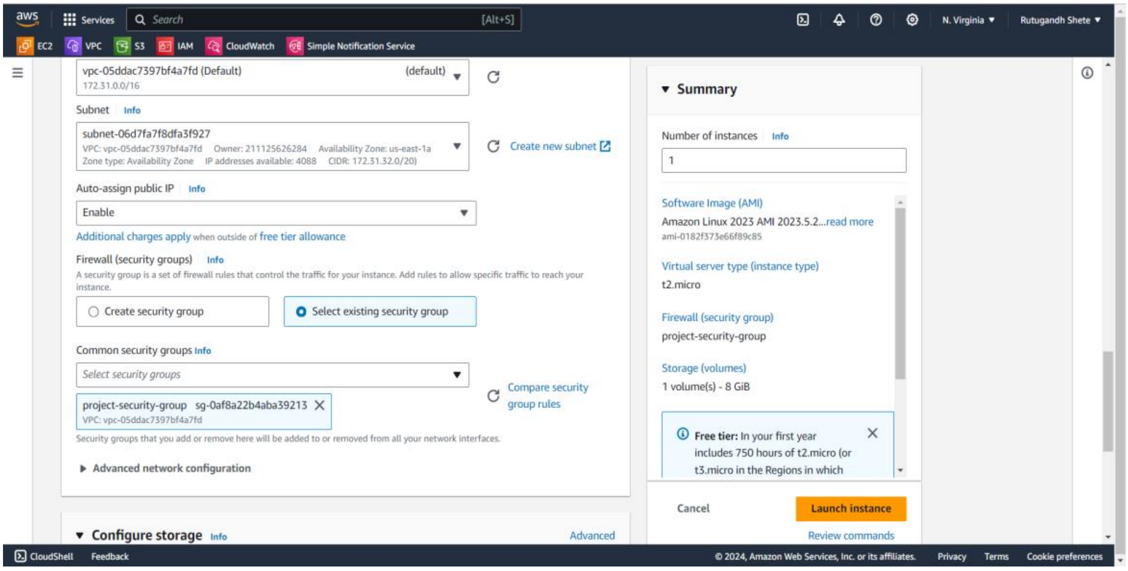
Virtual server type (instance type): t2.micro

Firewall (security group): -

Storage (volumes): -

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance

Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Cancel
Launch instance
Review commands



vpc-05ddac7397bf4a7fd (Default) (default)
172.31.0.0/16

Subnet Info

subnet-06d7fa7f8dfa3f927
VPC: vpc-05ddac7397bf4a7fd Owner: 211125626284 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 4088 CIDR: 172.31.32.0/20)

Auto-assign public IP Info: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups Info

project-security-group sg-0af8a22b4aba39213
VPC: vpc-05ddac7397bf4a7fd

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more
ami-0182f373e66f89c85

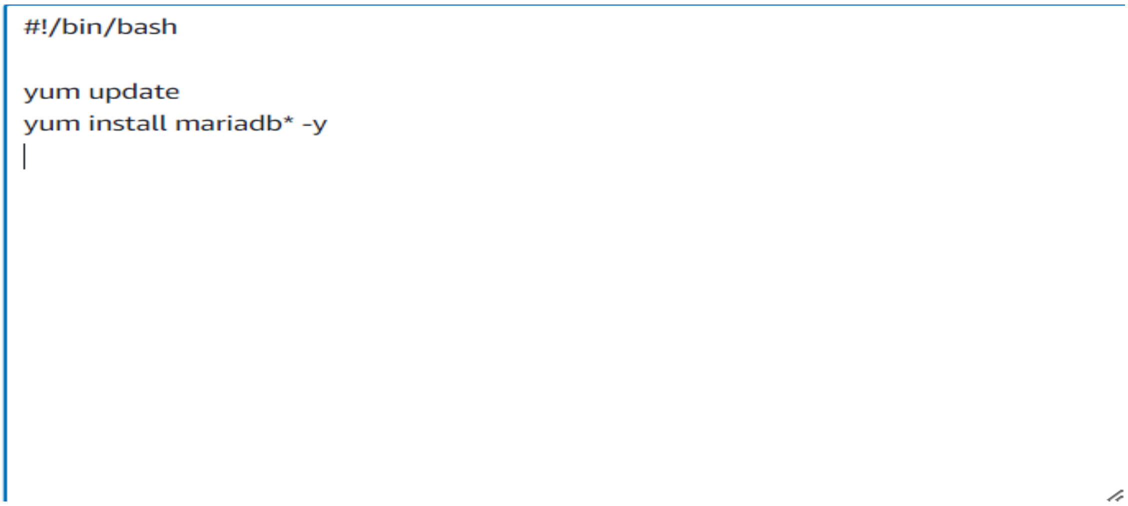
Virtual server type (instance type): t2.micro

Firewall (security group): project-security-group

Storage (volumes): 1 volume(s) - 8 GiB

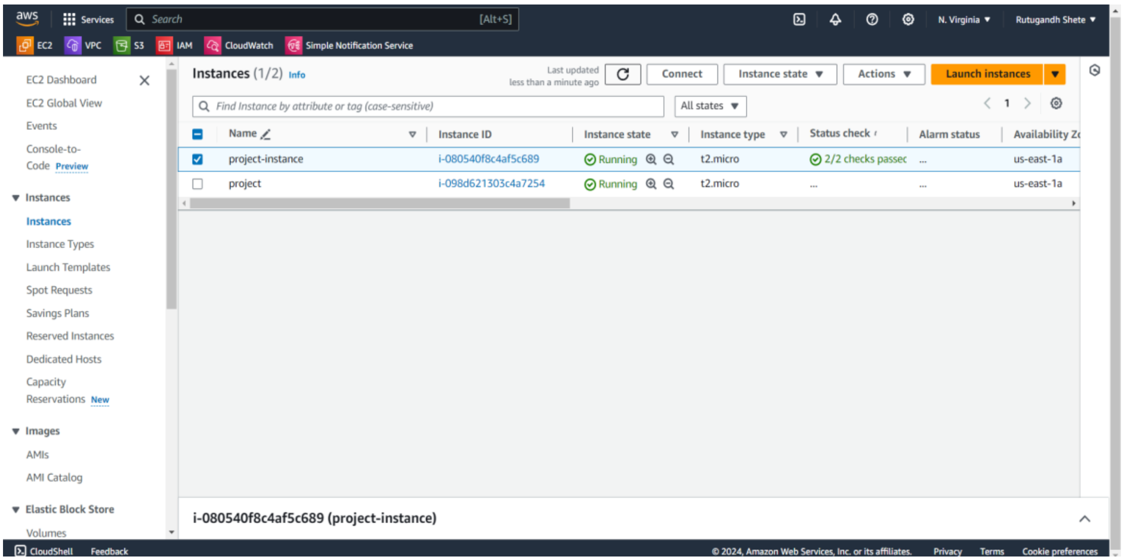
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which

- Add script in user data.



```
#!/bin/bash

yum update
yum install mariadb* -y
```

- After launching instance.



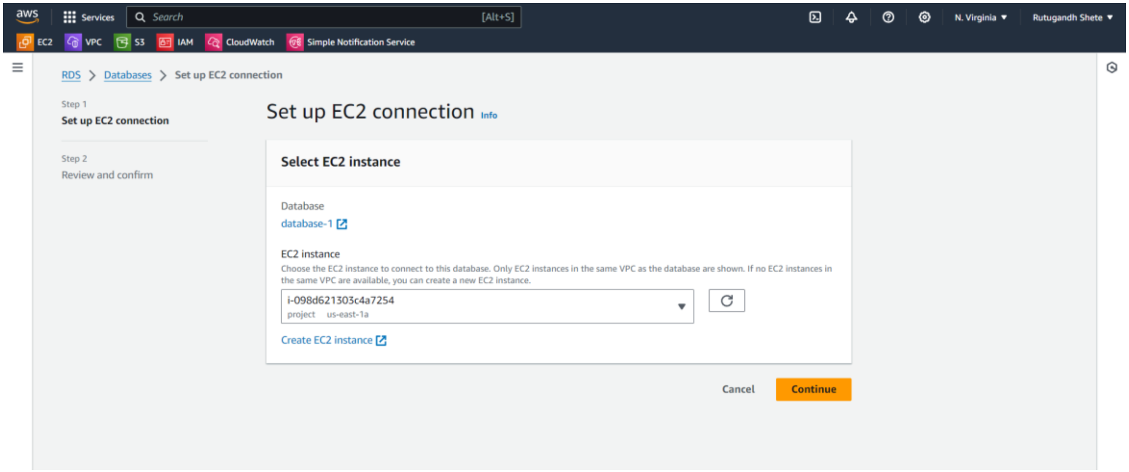
Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zc
☑	project-instance	i-080540f8c4af5c689	Running	t2.micro	2/2 checks passec	...	us-east-1a
□	project	i-098d621303c4a7254	Running	t2.micro	...	...	us-east-1a

i-080540f8c4af5c689 (project-instance)

- Now go to RDS service click on database and set up an EC2 connection.



Set up EC2 connection Info

Select EC2 instance

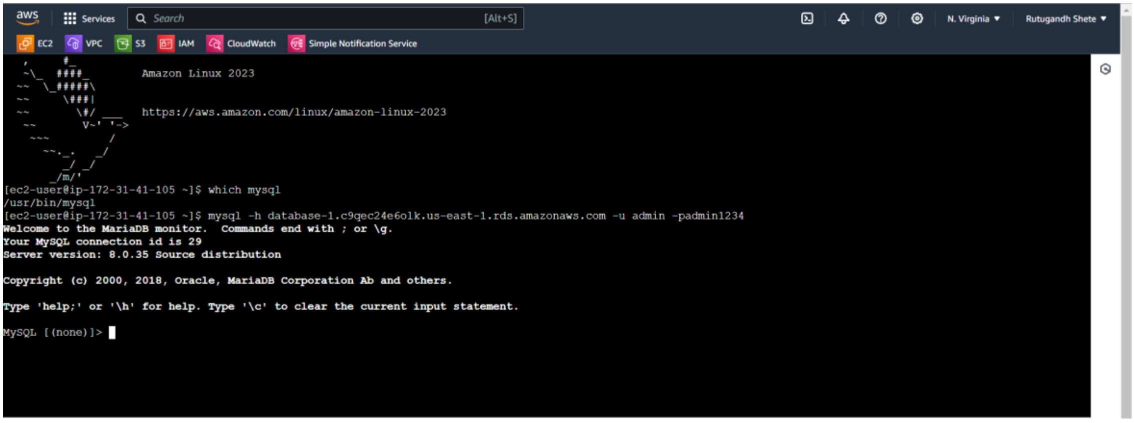
Database
database-1

EC2 instance
Choose the EC2 instance to connect to this database. Only EC2 instances in the same VPC as the database are shown. If no EC2 instances in the same VPC are available, you can create a new EC2 instance.

i-098d621303c4a7254
project us-east-1a

Create EC2 instance

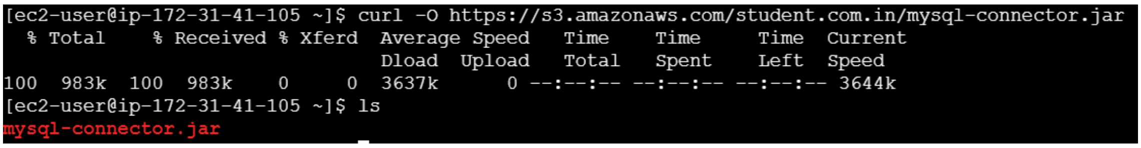
- Connect your EC2 terminal and Check connection using “mysql -h database-1.c9qec24e6olk.us-east-1.rds.amazonaws.com -u admin -p”.



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-41-105 ~]$ which mysql
/usr/bin/mysql
[ec2-user@ip-172-31-41-105 ~]$ mysql -h database-1.c9qec24e6olk.us-east-1.rds.amazonaws.com -u admin -padmin1234
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

- Now install these links curl -O https://s3.amazonaws.com/student.com.in/mysql-connector.jar

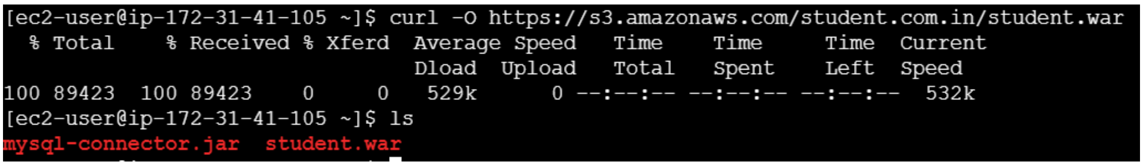
The mysql-connector.jar file is a Java library (JAR file) that allows Java applications to connect and interact with MySQL databases. It acts as a JDBC (Java Database Connectivity) driver, enabling Java programs to establish a connection, execute SQL queries, and retrieve data from MySQL databases.



```
[ec2-user@ip-172-31-41-105 ~]$ curl -O https://s3.amazonaws.com/student.com.in/mysql-connector.jar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  983k  100  983k    0     0  3637k      0 --:--:-- --:--:-- --:--:-- 3644k
[ec2-user@ip-172-31-41-105 ~]$ ls
mysql-connector.jar
```

2.curl -O https://s3.amazonaws.com/student.com.in/student.war

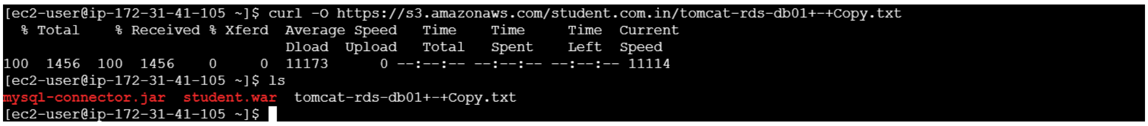
WAR files are deployed on web servers (like Apache Tomcat) or Java EE application servers. The server extracts and runs the WAR file, making the web application available to users.



```
[ec2-user@ip-172-31-41-105 ~]$ curl -O https://s3.amazonaws.com/student.com.in/student.war
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 89423  100 89423    0     0   529k      0 --:--:-- --:--:-- --:--:--  532k
[ec2-user@ip-172-31-41-105 ~]$ ls
mysql-connector.jar  student.war
```

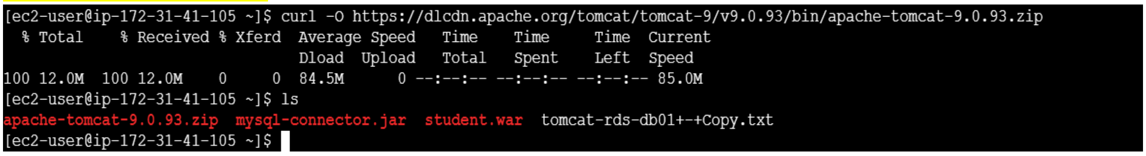
3.curl -O https://s3.amazonaws.com/student.com.in/tomcat-rds-db01+-+Copy.txt

This file contains all the changes that we have to do.



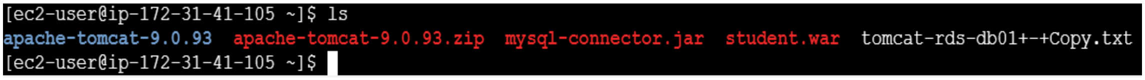
```
[ec2-user@ip-172-31-41-105 ~]$ curl -O https://s3.amazonaws.com/student.com.in/tomcat-rds-db01+-+Copy.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1456  100  1456    0     0  11173      0 --:--:-- --:--:-- --:--:-- 11114
[ec2-user@ip-172-31-41-105 ~]$ ls
mysql-connector.jar  student.war  tomcat-rds-db01+-+Copy.txt
[ec2-user@ip-172-31-41-105 ~]$
```

- Now install Latest version of apache tomcat "https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.zip"



```
[ec2-user@ip-172-31-41-105 ~]$ curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 12.0M  100 12.0M    0     0  84.5M      0 --:--:-- --:--:-- --:--:-- 85.0M
[ec2-user@ip-172-31-41-105 ~]$ ls
apache-tomcat-9.0.93.zip  mysql-connector.jar  student.war  tomcat-rds-db01+-+Copy.txt
[ec2-user@ip-172-31-41-105 ~]$
```

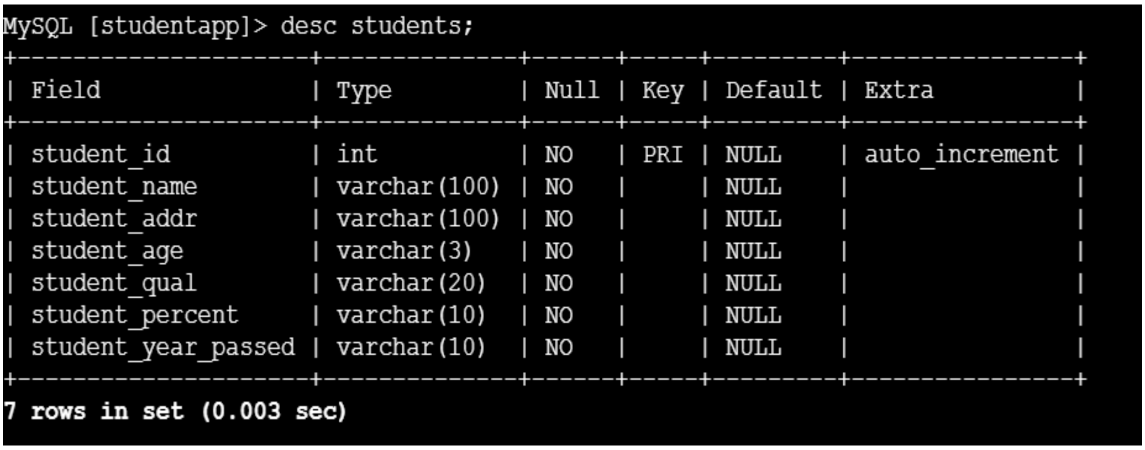
- Unzip apache tomcat file. After extracting file



```
[ec2-user@ip-172-31-41-105 ~]$ ls
apache-tomcat-9.0.93  apache-tomcat-9.0.93.zip  mysql-connector.jar  student.war  tomcat-rds-db01+-+Copy.txt
[ec2-user@ip-172-31-41-105 ~]$
```

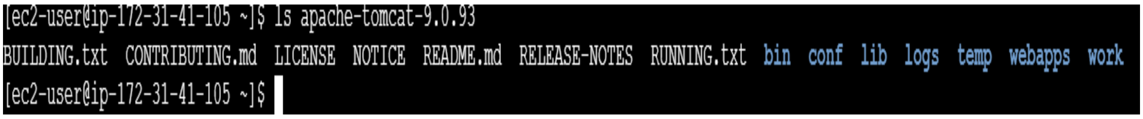
- Now Add one Schema into your database. "mysql -h DB-Endpoint -u admin -p". Add this query

```
create database studentapp;
use studentapp;
CREATE TABLE if not exists students(student_id INT NOT NULL AUTO_INCREMENT,
	student_name VARCHAR(100) NOT NULL,
    student_addr VARCHAR(100) NOT NULL,
	student_age VARCHAR(3) NOT NULL,
	student_qual VARCHAR(20) NOT NULL,
	student_percent VARCHAR(10) NOT NULL,
	student_year_passed VARCHAR(10) NOT NULL,
	PRIMARY KEY (student_id) );
```



```
MySQL [studentapp]> desc students;
+----------------------+--------------+------+-----+---------+----------------+
| Field                | Type         | Null | Key | Default | Extra          |
+----------------------+--------------+------+-----+---------+----------------+
| student_id           | int          | NO   | PRI | NULL    | auto_increment |
| student_name         | varchar(100) | NO   |     | NULL    |                |
| student_addr         | varchar(100) | NO   |     | NULL    |                |
| student_age          | varchar(3)   | NO   |     | NULL    |                |
| student_qual         | varchar(20)  | NO   |     | NULL    |                |
| student_percent      | varchar(10)  | NO   |     | NULL    |                |
| student_year_passed  | varchar(10)  | NO   |     | NULL    |                |
+----------------------+--------------+------+-----+---------+----------------+
7 rows in set (0.003 sec)
```

- Listing all files in apache-tomcat file



```
[ec2-user@ip-172-31-41-105 ~]$ ls apache-tomcat-9.0.93
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
[ec2-user@ip-172-31-41-105 ~]$
```

- Files within apache-tomcat file

  BUILDING.txt: Instructions for building Tomcat from source code.

  CONTRIBUTING.md: Guidelines for contributing to the development of Tomcat.

  LICENSE: The legal license under which Tomcat is distributed.

  NOTICE: Information about the licenses of included software components.

  README.md: A basic introduction and overview of Tomcat.

  RELEASE-NOTES: Notes about the current version's updates, changes, and fixes.

  RUNNING.txt: Instructions for starting and running Tomcat.

- Directories within file. □

  bin: Contains scripts and executables to start and stop Tomcat (e.g., startup.sh, shutdown.sh).

  Use case: Starting or stopping the Tomcat server.

  conf: Contains configuration files like server.xml and web.xml that define how Tomcat runs.

  Use case: Modify settings such as ports, security, and other server configurations.

  lib: Contains Java libraries (JAR files) required by Tomcat.

  Use case: Adding additional libraries that the server needs.

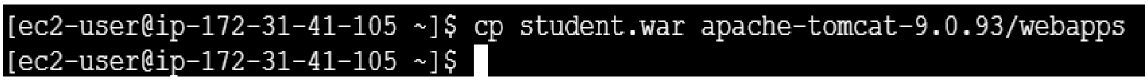
logs: Stores log files, including error messages and access logs.
Use case: Checking logs to troubleshoot issues or monitor server activity.

temp: Temporary files created by Tomcat while running.
Use case: Used for processing during runtime but usually not interacted with manually.

webapps: This is where your web applications (like your WAR files) are placed for deployment.
Use case: Deploy your web apps by adding WAR files here.

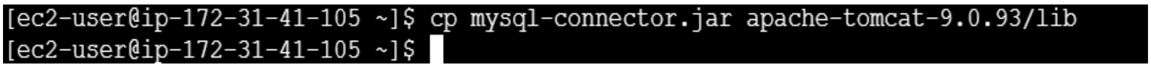
work: Tomcat uses this directory to store files generated during the runtime, such as JSP compiled code.
Use case: Tomcat manages this automatically, and you usually don't need to modify it manually.

- Copy student.war to apache-tomcat/webapps directory.



```
[ec2-user@ip-172-31-41-105 ~]$ cp student.war apache-tomcat-9.0.93/webapps
[ec2-user@ip-172-31-41-105 ~]$
```

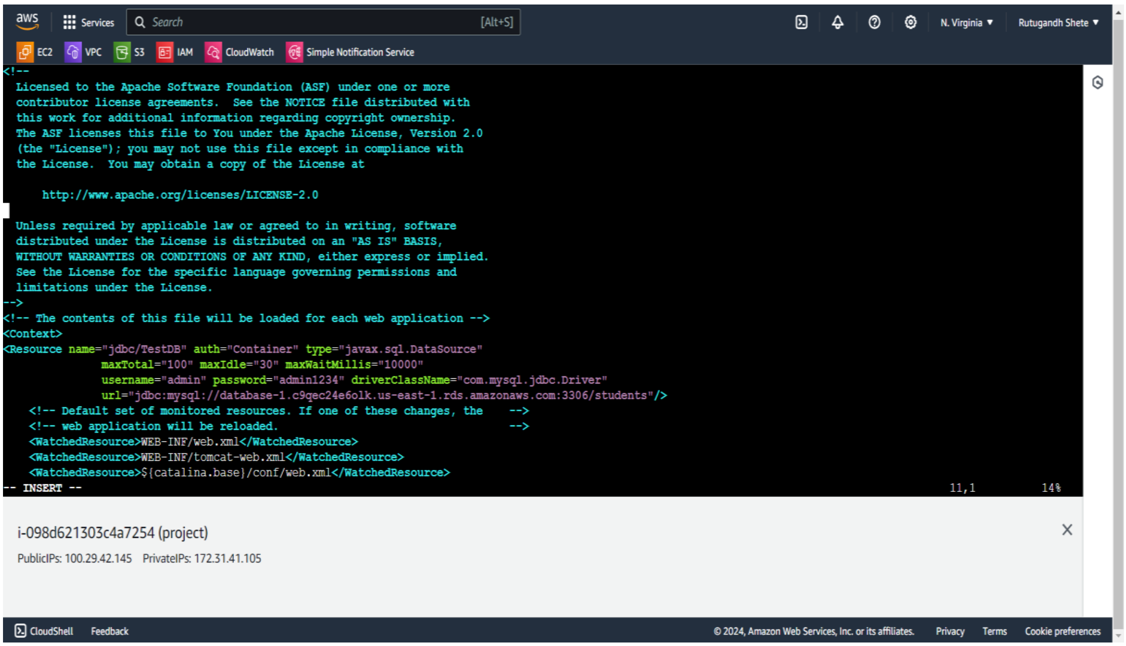
- Copy student.war to apache-tomcat/webapps directory.



```
[ec2-user@ip-172-31-41-105 ~]$ cp mysql-connector.jar apache-tomcat-9.0.93/lib
[ec2-user@ip-172-31-41-105 ~]$
```

- vim apache-tomcat-9.0.93/conf/context.xml(line no 21)

```
<Resource name="jdbc/TestDB" auth="Container"
type="javax.sql.DataSource"
 maxTotal="100" maxIdle="30" maxWaitMillis="10000"
username="USERNAME" password="PASSWORD"
driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://DB-
ENDPOINT:3306/DATABASE"/>
```

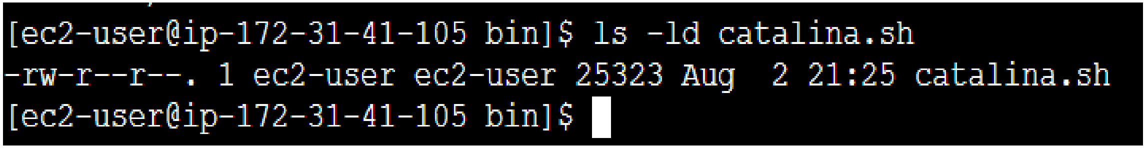


```
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "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="100" maxIdle="30" maxWaitMillis="10000"
               username="admin" password="admin1234" driverClassName="com.mysql.jdbc.Driver"
               url="jdbc:mysql://database-1.c9qec24e6olk.us-east-1.rds.amazonaws.com:3306/students"/>
    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
-- INSERT --                                                                       11,1          14%
```

i-098d621303c4a7254 (project)

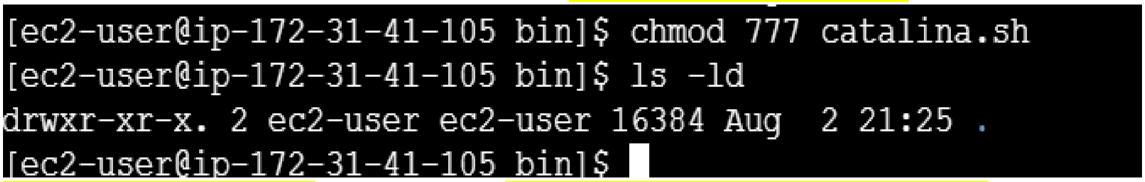
PublicIPs: 100.29.42.145   PrivateIPs: 172.31.41.105

- Catalina.sh file:
  Starting Tomcat: You can use catalina.sh start to start the Tomcat server.
  Stopping Tomcat: You can use catalina.sh stop to stop the Tomcat server.
  Restarting Tomcat: You can restart the server using catalina.sh restart.
  Running Tomcat in the Foreground: With catalina.sh run, you can run Tomcat in the foreground, useful for development.
- This file is present in “apache-tomcat-9.0.93/bin/Catalina.sh”. if try to execute this file there is no permissions for this file.



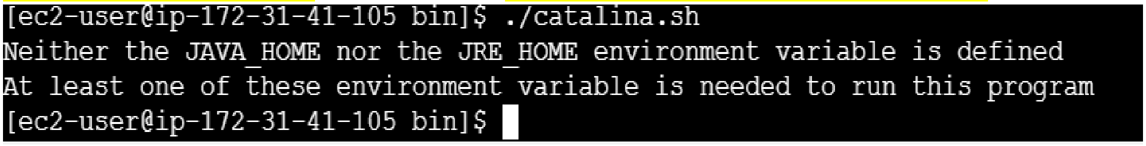
```
[ec2-user@ip-172-31-41-105 bin]$ ls -ld catalina.sh
-rw-r--r--. 1 ec2-user ec2-user 25323 Aug  2 21:25 catalina.sh
[ec2-user@ip-172-31-41-105 bin]$ 
```

- Give execute permissions to this file “chmod 777 catalina.sh”



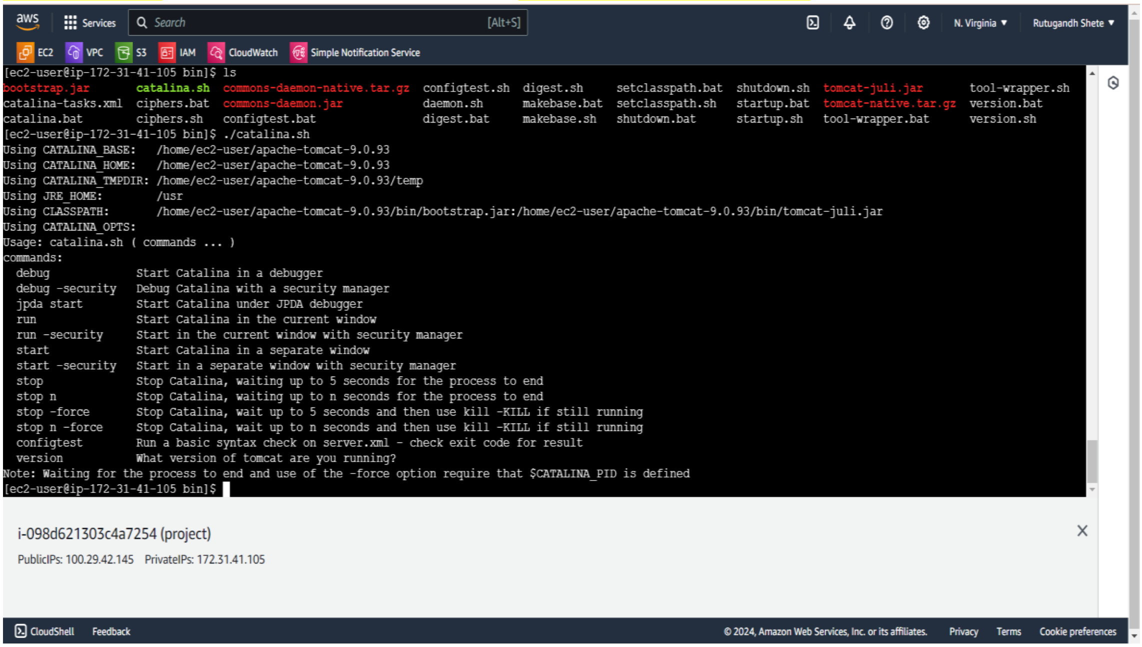
```
[ec2-user@ip-172-31-41-105 bin]$ chmod 777 catalina.sh
[ec2-user@ip-172-31-41-105 bin]$ ls -ld
drwxr-xr-x. 2 ec2-user ec2-user 16384 Aug  2 21:25 .
[ec2-user@ip-172-31-41-105 bin]$ 
```

- Now file is excutable but there is no java environment for this file.



```
[ec2-user@ip-172-31-41-105 bin]$ ./catalina.sh
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variable is needed to run this program
[ec2-user@ip-172-31-41-105 bin]$ 
```

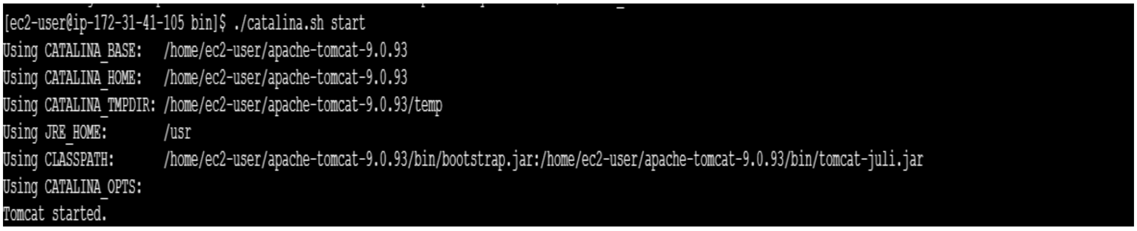
- Install java and start tomcat using "./catalina.sh start"



```
[ec2-user@ip-172-31-41-105 bin]$ ls
bootstrap.jar        catalina.sh   commons-daemon-native.tar.gz  configtest.sh  digest.sh     setclasspath.bat  shutdown.sh  tomcat-juli.jar       tool-wrapper.sh
catalina-tasks.xml   ciphers.bat   commons-daemon.jar            daemon.sh      makebase.bat  setclasspath.sh   startup.bat  tomcat-native.tar.gz  version.bat
catalina.bat         ciphers.sh    configtest.bat                digest.bat     makebase.sh   shutdown.bat      startup.sh   tool-wrapper.bat      version.sh
[ec2-user@ip-172-31-41-105 bin]$ ./catalina.sh
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.93
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.93
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.93/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.93/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-9.0.93/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Usage: catalina.sh ( commands ... )
commands:
  debug             Start Catalina in a debugger
  debug -security   Debug Catalina with a security manager
  jpda start        Start Catalina under JPDA debugger
  run               Start Catalina in the current window
  run -security     Start in the current window with security manager
  start             Start Catalina in a separate window
  start -security   Start in a separate window with security manager
  stop              Stop Catalina, waiting up to 5 seconds for the process to end
  stop n            Stop Catalina, waiting up to n seconds for the process to end
  stop -force       Stop Catalina, wait up to 5 seconds and then use kill -KILL if still running
  stop n -force     Stop Catalina, wait up to n seconds and then use kill -KILL if still running
  configtest        Run a basic syntax check on server.xml - check exit code for result
  version           What version of tomcat are you running?
Note: Waiting for the process to end and use of the -force option require that $CATALINA_PID is defined
[ec2-user@ip-172-31-41-105 bin]$
```

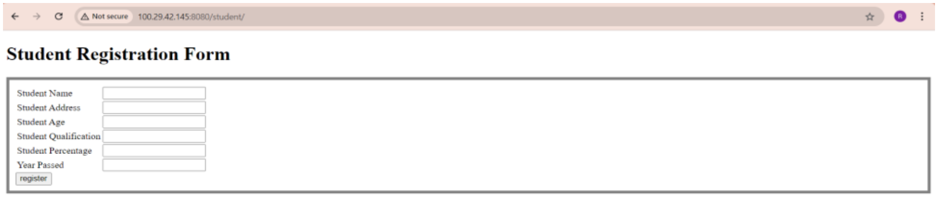
i-098d621303c4a7254 (project)

PublicIPs: 100.29.42.145 PrivateIPs: 172.31.41.105



```
[ec2-user@ip-172-31-41-105 bin]$ ./catalina.sh start
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.93
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.93
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.93/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.93/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-9.0.93/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

- Hit public IP of your instance. "IP:8080/student/"



Student Registration Form

Student Name	
Student Address	
Student Age	
Student Qualification	
Student Percentage	
Year Passed	