Deployment of Dynamic Website Using AWS EC2 and Store Data in AWS RDS Database.

Introduction to Cloud Computing:

Cloud computing is a modern technology that enables users to access computing resources—such as servers, storage, databases, networking, software, and analytics—over the internet on a pay-as-you-go basis. Instead of investing in expensive physical infrastructure, individuals and organizations can rent resources from cloud service providers like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud. This model offers high scalability, flexibility, cost-efficiency, and global accessibility. Cloud computing is typically categorized into service models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), allowing users to choose the level of control and abstraction they need based on their requirements.

Amazon Web Services (AWS) Overview:

Amazon Web Services (AWS) is a comprehensive and widely adopted cloud computing platform offered by Amazon, providing a broad set of on-demand services such as computing power, storage, databases, machine learning, and networking. Launched in 2006, AWS enables individuals, startups, and enterprises to build and scale applications quickly without the need for upfront hardware investment. Its global infrastructure, spanning multiple regions and availability zones, ensures high availability, fault tolerance, and low latency. AWS supports various service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), making it suitable for a wide range of use cases—from simple web hosting to complex data analytics and enterprise-level deployments.

What is a Dynamic Website:

A dynamic website is a type of website that generates and displays different content and allows user interaction in real-time, based on inputs, user behavior, or other variables. Unlike static websites, where content remains the same for every visitor, dynamic websites use server-side technologies like PHP, Node.js, or Python in combination with databases such as MySQL or PostgreSQL to fetch, update, and display content dynamically. This means users can perform actions like logging in, submitting forms, posting comments, or viewing personalized data. Examples of dynamic websites include e-commerce platforms, social media sites, and content management systems, making them ideal for applications that require frequent updates and user interaction.

What is Amazon EC2 (Elastic Compute Cloud):

Amazon EC2 (Elastic Compute Cloud) is a core service offered by Amazon Web Services (AWS) that provides scalable and resizable computing capacity in the cloud. EC2 allows users to rent virtual machines, called instances, to run applications and workloads without having to invest in physical hardware. It offers a wide range of instance types optimized for

various use cases, such as compute-intensive tasks, memory-heavy applications, or general-purpose workloads. With EC2, users can scale their computing resources up or down based on demand, making it an ideal solution for dynamic and fluctuating workloads. EC2 instances can run on multiple operating systems, including Linux and Windows, and can be managed through the AWS Management Console or AWS CLI. Additionally, EC2 integrates with other AWS services such as Elastic Load Balancing, Auto Scaling, and Amazon RDS, enabling the creation of highly scalable and fault-tolerant applications. Its flexible pricing models, including On-Demand, Reserved, and Spot Instances, allow users to optimize costs based on their usage needs.

What is Amazon RDS (Relational Database Service):

Amazon RDS (Relational Database Service) is a fully managed relational database service provided by Amazon Web Services (AWS). It simplifies the process of setting up, operating, and scaling a relational database in the cloud. RDS supports several popular database engines, including MySQL, PostgreSQL, Oracle, SQL Server, and Amazon Aurora, allowing users to choose the database that best fits their needs. With RDS, AWS handles routine database management tasks such as backups, patch management, and scaling, so users can focus on application development instead of database maintenance.

One of the key benefits of RDS is its scalability. It enables automatic storage scaling and provides options for multi-AZ (Availability Zone) deployments for high availability, as well as Read Replicas to offload read traffic and improve performance. RDS also offers built-in security features, including encryption at rest and in transit, network isolation through VPC, and IAM-based access control. Additionally, RDS integrates seamlessly with other AWS services, such as EC2 and Lambda, allowing for a fully managed, secure, and scalable database solution that can support mission-critical applications.

How EC2 and RDS Work Together:

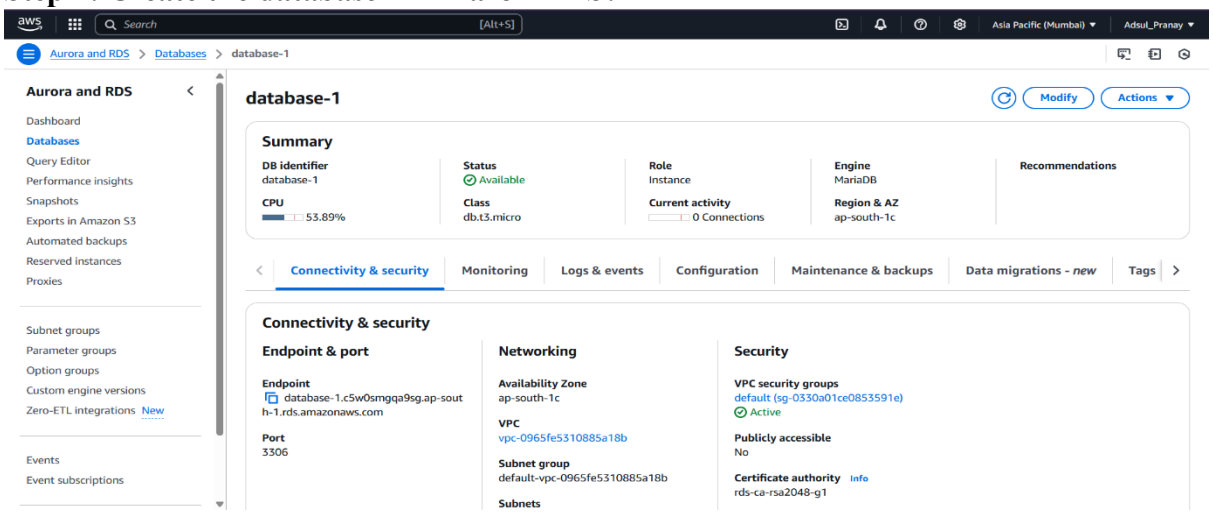
Amazon EC2 (Elastic Compute Cloud) and Amazon RDS (Relational Database Service) are often used together to build scalable, secure, and high-performance web applications. EC2 serves as the computing resource where the web application or backend services are hosted, while RDS acts as the managed database solution that stores and manages the application's data. Here’s how they interact:

1. **EC2 as the Application Host**:
    - EC2 instances host the dynamic website or application. These instances run the web server (e.g., Apache or Nginx) and application server (e.g., PHP, Node.js, Python), handling the front-end user requests and server-side processing.
    - When users interact with the website (e.g., submitting forms, logging in, or viewing content), the EC2 instance processes these requests and generates dynamic responses.
2. **RDS as the Data Store**:

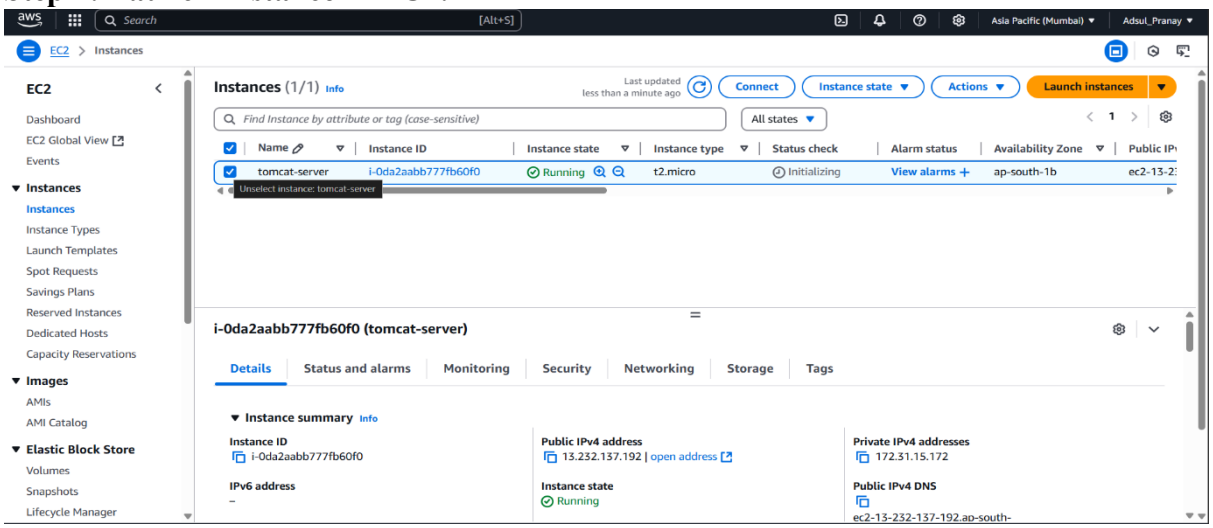
- Amazon RDS is used to manage and store relational data, such as user information, product details, orders, etc. The database can be MySQL, PostgreSQL, or any other supported engine, depending on the application's needs.
- The EC2 instance sends queries to RDS using SQL (Structured Query Language) to retrieve, update, or insert data into the database.

3. **Secure Communication Between EC2 and RDS**:
    - EC2 instances and RDS databases are typically deployed within the same Virtual Private Cloud (VPC) for enhanced security and performance. This ensures that communication between the EC2 instance and the RDS instance is private and does not travel over the public internet.
    - The EC2 instance connects to the RDS instance using standard database protocols (e.g., MySQL's port 3306) while being secured by **security groups** that control access to the RDS instance.
4. **Scalability and High Availability**:
    - EC2 and RDS can be scaled independently based on the application's needs. For example, if web traffic increases, more EC2 instances can be added, or an Auto Scaling Group can be set up to automatically manage scaling.
    - For high availability, RDS offers Multi-AZ deployments, where the database is replicated in different Availability Zones. This ensures that if one AZ fails, the application can still access the database from another AZ, minimizing downtime.
5. **Fault Tolerance and Backup**:
    - EC2 instances can be configured to automatically back up their data or be part of an **Auto Scaling Group** to ensure high availability.
    - RDS also automatically handles backups and can perform **point-in-time recovery**, allowing data to be restored in case of corruption or data loss.

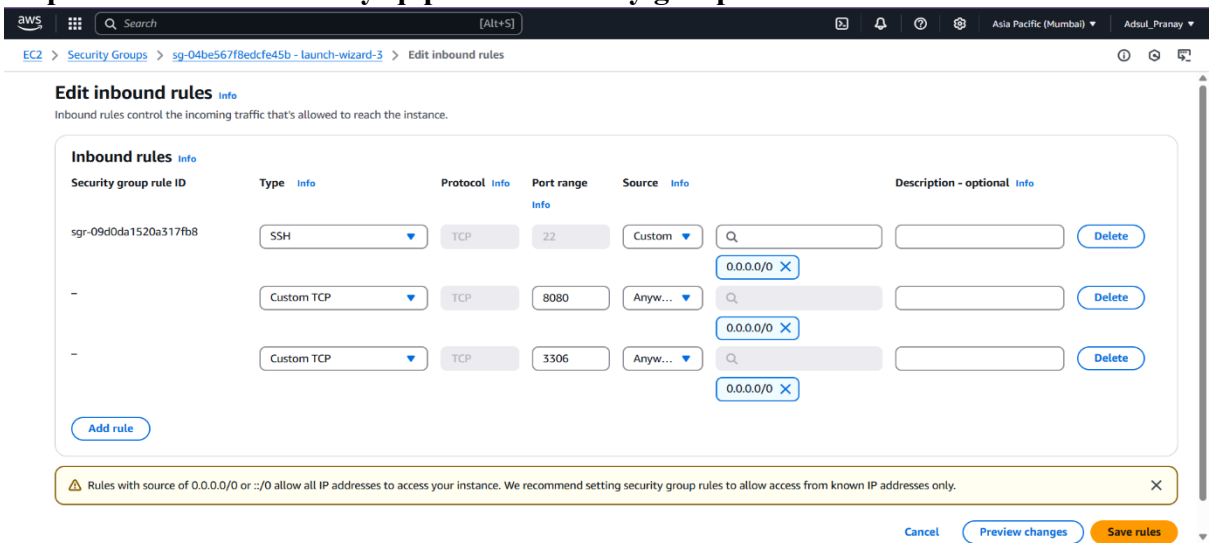
Step 1: Create the database in Amazon RDS:



Step 2: Launch Instance in EC2:



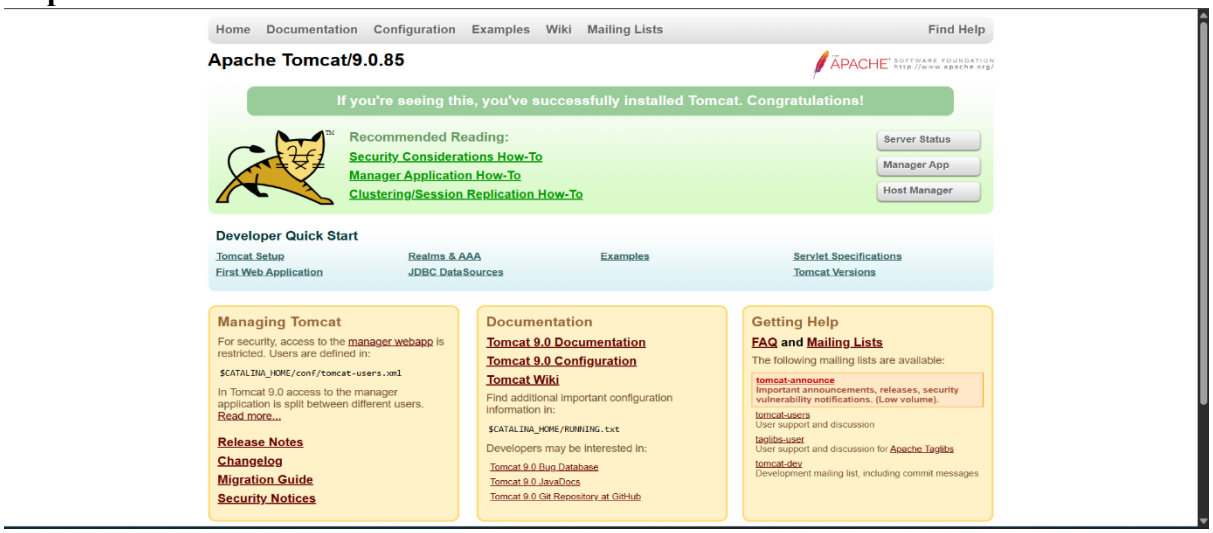
Step 3: Add tomcat and mysql ports in security group of instance:



Step 4: Install Java and Tomcat Server:

```
apache-tomcat-9.0.85/webapps/host-manager/WEB-INF/web.xml
apache-tomcat-9.0.85/webapps/host-manager/css/manager.css
apache-tomcat-9.0.85/webapps/host-manager/images/asf-logo.svg
apache-tomcat-9.0.85/webapps/host-manager/images/tomcat.svg
apache-tomcat-9.0.85/webapps/host-manager/index.jsp
apache-tomcat-9.0.85/webapps/manager/META-INF/context.xml
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/401.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/403.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/404.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/connectorCerts.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/connectorCiphers.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/connectorTrustedCerts.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/sessionDetail.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/jsp/sessionsList.jsp
apache-tomcat-9.0.85/webapps/manager/WEB-INF/web.xml
apache-tomcat-9.0.85/webapps/manager/css/manager.css
apache-tomcat-9.0.85/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.85/webapps/manager/images/tomcat.svg
apache-tomcat-9.0.85/webapps/manager/index.jsp
apache-tomcat-9.0.85/webapps/manager/status.xsd
apache-tomcat-9.0.85/webapps/manager/xform.xsl
apache-tomcat-9.0.85/bin/catalina.sh
apache-tomcat-9.0.85/bin/ciphers.sh
apache-tomcat-9.0.85/bin/configtest.sh
apache-tomcat-9.0.85/bin/daemon.sh
apache-tomcat-9.0.85/bin/digest.sh
apache-tomcat-9.0.85/bin/makebase.sh
apache-tomcat-9.0.85/bin/setclasspath.sh
apache-tomcat-9.0.85/bin/shutdown.sh
apache-tomcat-9.0.85/bin/startup.sh
apache-tomcat-9.0.85/bin/tool-wrapper.sh
apache-tomcat-9.0.85/bin/version.sh
[root@ip-172-31-11-196 opt]# ls
apache-tomcat-9.0.85  apache-tomcat-9.0.85.tar.gz  aws
[root@ip-172-31-11-196 opt]#
```

Step 5: Hit the Public IP:



Step 6: Load the dynamic website jar file in tomcat server:

```
apache-tomcat-9.0.85/webapps/manager/xform.xsl
apache-tomcat-9.0.85/bin/catalina.sh
apache-tomcat-9.0.85/bin/ciphers.sh
apache-tomcat-9.0.85/bin/configtest.sh
apache-tomcat-9.0.85/bin/daemon.sh
apache-tomcat-9.0.85/bin/digest.sh
apache-tomcat-9.0.85/bin/makebase.sh
apache-tomcat-9.0.85/bin/setclasspath.sh
apache-tomcat-9.0.85/bin/shutdown.sh
apache-tomcat-9.0.85/bin/startup.sh
apache-tomcat-9.0.85/bin/tool-wrapper.sh
apache-tomcat-9.0.85/bin/version.sh
[root@ip-172-31-11-196 opt]# ls
apache-tomcat-9.0.85  apache-tomcat-9.0.85.tar.gz  aws
[root@ip-172-31-11-196 opt]# cd apache-tomcat-9.0.85/
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# cd bin
[root@ip-172-31-11-196 bin]# ./catalina.sh start
Using CATALINA_BASE:   /opt/apache-tomcat-9.0.85
Using CATALINA_HOME:   /opt/apache-tomcat-9.0.85
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.85/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-9.0.85/bin/bootstrap.jar:/opt/apache-tomcat-9.0.85/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-11-196 bin]# cd..
-bash: cd..: command not found
[root@ip-172-31-11-196 bin]# cd ..
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# cd webapps/
[root@ip-172-31-11-196 webapps]# ls
ROOT  docs  examples  host-manager  manager
[root@ip-172-31-11-196 webapps]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 89423  100 89423    0     0  38366      0  0:00:02  0:00:02 --:--:-- 38378
[root@ip-172-31-11-196 webapps]#
```

Step7: Preview of website:

Student Registration Form

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

Step 8: Install mysql-connector:

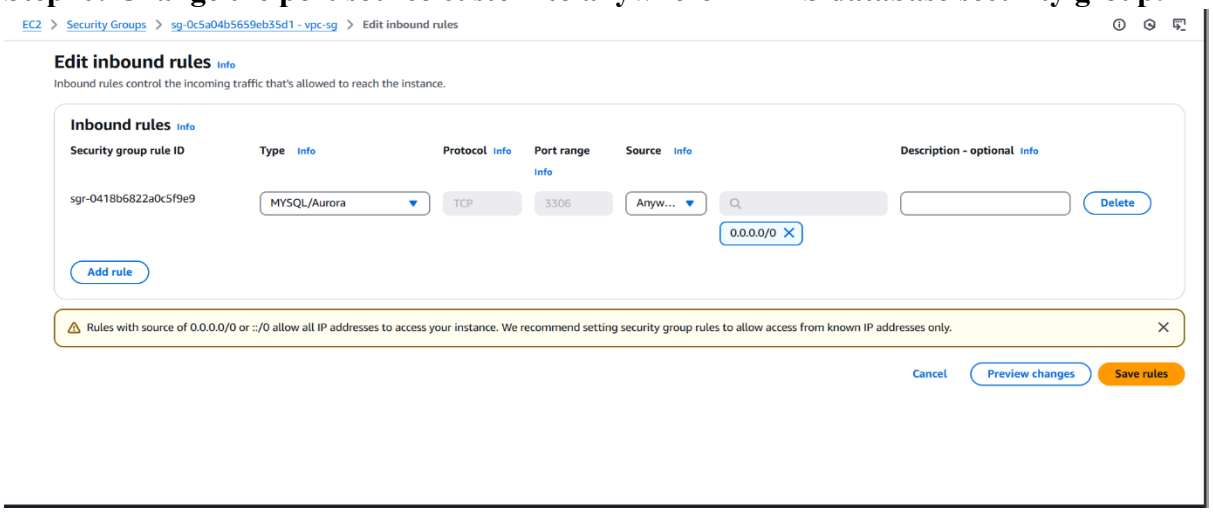
```
Using CATALINA_HOME:   /opt/apache-tomcat-9.0.85
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.85/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-9.0.85/bin/bootstrap.jar:/opt/apache-tomcat-9.0.85/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-11-196 bin]# cd..
-bash: cd..: command not found
[root@ip-172-31-11-196 bin]# cd ..
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# cd webapps/
[root@ip-172-31-11-196 webapps]# ls
ROOT  docs  examples  host-manager  manager
[root@ip-172-31-11-196 webapps]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 89423  100 89423    0     0  38366      0  0:00:02  0:00:02 --:--:-- 38378
[root@ip-172-31-11-196 webapps]# cd ..
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# cd lib
[root@ip-172-31-11-196 lib]# ls
annotations-api.jar       catalina-tribes.jar  jasper.jar        tomcat-coyote.jar   tomcat-i18n-fr.jar     tomcat-i18n-zh-CN.jar  tomcat-websocket.jar
catalina-ant.jar          catalina.jar         jaspic-api.jar    tomcat-dbcp.jar     tomcat-i18n-ja.jar     tomcat-jdbc.jar        websocket-api.jar
catalina-ha.jar           ecj-4.20.jar         jsp-api.jar       tomcat-i18n-cs.jar  tomcat-i18n-ko.jar     tomcat-jni.jar
catalina-ssi.jar          el-api.jar           servlet-api.jar   tomcat-i18n-de.jar  tomcat-i18n-pt-BR.jar  tomcat-util-scan.jar
catalina-storeconfig.jar  jasper-el.jar        tomcat-api.jar    tomcat-i18n-es.jar  tomcat-i18n-ru.jar     tomcat-util.jar
[root@ip-172-31-11-196 lib]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  983k  100  983k    0     0   144k      0  0:00:06  0:00:06 --:--:--  201k
[root@ip-172-31-11-196 lib]# ls
annotations-api.jar       catalina-tribes.jar  jasper.jar            tomcat-api.jar      tomcat-i18n-es.jar     tomcat-i18n-ru.jar     tomcat-util.jar
catalina-ant.jar          catalina.jar         jaspic-api.jar        tomcat-coyote.jar   tomcat-i18n-fr.jar     tomcat-i18n-zh-CN.jar  tomcat-websocket.jar
catalina-ha.jar           ecj-4.20.jar         jsp-api.jar           tomcat-dbcp.jar     tomcat-i18n-ja.jar     tomcat-jdbc.jar        websocket-api.jar
catalina-ssi.jar          el-api.jar           mysql-connector.jar   tomcat-i18n-cs.jar  tomcat-i18n-ko.jar     tomcat-jni.jar
catalina-storeconfig.jar  jasper-el.jar        servlet-api.jar       tomcat-i18n-de.jar  tomcat-i18n-pt-BR.jar  tomcat-util-scan.jar
[root@ip-172-31-11-196 lib]#
```

Step 9: Install mariadb:

```
Installed:
  Judy-1.0.5-25.amzn2023.0.3.x86_64                                     libdatrie-0.2.13-1.amzn2023.0.2.x86_64
  libpq-17.4-1.amzn2023.0.1.x86_64                                      libsphinxclient-2.2.11-24.amzn2023.0.4.x86_64
  libthai-0.1.28-6.amzn2023.0.2.x86_64                                  mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64
  mariadb-connector-c-config-3.3.10-1.amzn2023.0.1.noarch               mariadb-connector-c-devel-3.3.10-1.amzn2023.0.1.x86_64
  mariadb-connector-c-doc-3.3.10-1.amzn2023.0.1.noarch                  mariadb105-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-backup-3:10.5.25-1.amzn2023.0.1.x86_64                     mariadb105-common-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-connect-engine-3:10.5.25-1.amzn2023.0.1.x86_64             mariadb105-cracklib-password-check-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-devel-3:10.5.25-1.amzn2023.0.1.x86_64                      mariadb105-errmsg-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-gssapi-server-3:10.5.25-1.amzn2023.0.1.x86_64              mariadb105-oggraph-engine-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-pam-3:10.5.25-1.amzn2023.0.1.x86_64                        mariadb105-rocksdb-engine-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-server-3:10.5.25-1.amzn2023.0.1.x86_64                     mariadb105-server-utils-3:10.5.25-1.amzn2023.0.1.x86_64
  mariadb105-sphinx-engine-3:10.5.25-1.amzn2023.0.1.x86_64              mariadb105-test-3:10.5.25-1.amzn2023.0.1.x86_64
  mysql-selinux-1.0.4-2.amzn2023.0.3.noarch                             openssl-devel-1:3.2.2-1.amzn2023.0.1.x86_64
  patch-2.7.6-14.amzn2023.0.2.x86_64                                    perl-B-1.80-477.amzn2023.0.6.x86_64
  perl-DBD-MariaDB-1.22-1.amzn2023.0.4.x86_64                           perl-DBI-1.643-7.amzn2023.0.3.x86_64
  perl-Data-Dumper-2.174-460.amzn2023.0.2.x86_64                        perl-English-1.11-477.amzn2023.0.6.noarch
  perl-Env-1.04-458.amzn2023.0.2.noarch                                 perl-File-Copy-2.34-477.amzn2023.0.6.noarch
  perl-File-Find-1.37-477.amzn2023.0.6.noarch                           perl-FileHandle-2.03-477.amzn2023.0.6.noarch
  perl-Importer-0.026-2.amzn2023.0.2.noarch                             perl-JSON-PP-1:4.06-2.amzn2023.0.2.noarch
  perl-MIME-Charset-1.012.2-13.amzn2023.0.2.noarch                      perl-Math-BigInt-1:1.9998.39-2.amzn2023.0.2.noarch
  perl-Math-BigRat-0.2614-458.amzn2023.0.2.noarch                       perl-Math-Complex-1.59-477.amzn2023.0.6.noarch
  perl-Memoize-1.03-477.amzn2023.0.6.noarch                             perl-NDBM_File-1.15-477.amzn2023.0.6.x86_64
  perl-Object-HashBase-0.009-5.amzn2023.0.2.noarch                      perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64
  perl-Term-Size-Any-0.002-33.amzn2023.0.2.noarch                       perl-Term-Size-Perl-0.031-10.amzn2023.0.2.x86_64
  perl-Term-Table-0.015-6.amzn2023.0.2.noarch                           perl-Test-Simple-3:1.302183-2.amzn2023.0.2.noarch
  perl-Tie-4.6-477.amzn2023.0.6.noarch                                  perl-Time-1.03-477.amzn2023.0.6.noarch
  perl-Time-HiRes-4:1.9764-460.amzn2023.0.2.x86_64                      perl-Unicode-LineBreak-2019.001-9.amzn2023.0.2.x86_64
  perl-base-2.27-477.amzn2023.0.6.noarch                                perl-lib-0.65-477.amzn2023.0.6.x86_64
  perl-threads-1:2.25-458.amzn2023.0.3.x86_64                           perl-threads-shared-1.61-458.amzn2023.0.2.x86_64
  sombok-2.4.0-14.amzn2023.0.2.x86_64                                   sphinx-2.2.11-24.amzn2023.0.4.x86_64
  zlib-devel-1.2.11-33.amzn2023.0.5.x86_64

Complete!
[root@ip-172-31-11-196 lib]#
```

Step10: Change the port source custom to anywhere in RDS database security group:



Step 11: Connect and Create the database in mariadb:

```
Complete!
[root@ip-172-31-11-196 lib]# cd
[root@ip-172-31-11-196 ~]# mysql -h database-1.c5w0smgqa9sg.ap-south-1.rds.amazonaws.com -u admin -pPranay123
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 51
Server version: 11.4.4-MariaDB-log managed by https://aws.amazon.com/rds/

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

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

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.004 sec)

MariaDB [(none)]> create database studentapp;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]> use studentapp;
Database changed
MariaDB [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,
```

Step 12: Add resource name and details of database in context.xml file:

```
  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>

    <!-- 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>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
    <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
            maxTotal="500" maxIdle="30" maxWaitMillis="1000"
            username="admin" password="Pranay123" driverClassName="com.mysql.jdbc.Driver"
            url="jdbc:mysql://database-1.c5w0smgqa9sg.ap-south-1.rds.amazonaws.com:3306/studentapp?useUnicode=yes&amp;characterEncoding=utf8"/>

</Context>
-- INSERT (paste) --                                                    35,98         All
```

Step 13: Stop and Start the Tomcat Server:

```
MariaDB [studentapp]> exit
Bye
[root@ip-172-31-11-196 ~]# cd /opt
[root@ip-172-31-11-196 opt]# cd apache-tomcat-9.0.85/
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# cd conf
[root@ip-172-31-11-196 conf]# ls
Catalina          catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
catalina.policy  context.xml          jaspic-providers.xsd  server.xml          tomcat-users.xsd
[root@ip-172-31-11-196 conf]# vim context.xml
[root@ip-172-31-11-196 conf]# cd
[root@ip-172-31-11-196 ~]# cd /opt
[root@ip-172-31-11-196 opt]# ls
apache-tomcat-9.0.85  apache-tomcat-9.0.85.tar.gz  aws
[root@ip-172-31-11-196 opt]# cd apache-tomcat-9.0.85/
[root@ip-172-31-11-196 apache-tomcat-9.0.85]# cd bin
[root@ip-172-31-11-196 bin]# ./catalina.sh stop
Using CATALINA_BASE:   /opt/apache-tomcat-9.0.85
Using CATALINA_HOME:   /opt/apache-tomcat-9.0.85
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.85/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-9.0.85/bin/bootstrap.jar:/opt/apache-tomcat-9.0.85/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --
add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
[root@ip-172-31-11-196 bin]# ./catalina.sh start
Using CATALINA_BASE:   /opt/apache-tomcat-9.0.85
Using CATALINA_HOME:   /opt/apache-tomcat-9.0.85
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.85/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-9.0.85/bin/bootstrap.jar:/opt/apache-tomcat-9.0.85/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-11-196 bin]#
```

Step 14: Hit the Website and fill the form:

Register Student

Students List

Student ID	StudentName	Student Addrs	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	Pranay	Colony 7 ganeshnagar Pune	24	MCA	76	2025	edit	delete

Step 15: View the data in Mysql database:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 68
Server version: 11.4.4-MariaDB-log managed by https://aws.amazon.com/rds/

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

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

MariaDB [(none)]> use studentapp
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [studentapp]> show tables;
+----------------------+
| Tables_in_studentapp |
+----------------------+
| students             |
+----------------------+
1 row in set (0.001 sec)

MariaDB [studentapp]> use students;
ERROR 1049 (42000): Unknown database 'students'
MariaDB [studentapp]> show students;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'students'
at line 1
MariaDB [studentapp]> select * from students;
+------------+--------------+-------------------------+-------------+--------------+----------------+--------------------+
| student_id | student_name | student_addr            | student_age | student_qual | student_percent | student_year_passed |
+------------+--------------+-------------------------+-------------+--------------+----------------+--------------------+
|          1 | Pranay       | Colony 7 ganeshnagar Pune | 24          | MCA          | 76             | 2025               |
+------------+--------------+-------------------------+-------------+--------------+----------------+--------------------+
1 row in set (0.001 sec)

MariaDB [studentapp]>
```

Conclusion:

In this project, we successfully deployed a dynamic website using AWS EC2 (Elastic Compute Cloud) and implemented a scalable, reliable database solution using AWS RDS (Relational Database Service). The EC2 instance served as the web server, hosting the dynamic website, while AWS RDS was utilized for securely storing and managing the website's data in a managed relational database.

Key takeaways from the project include:

1. **Scalability**: AWS EC2 instances provided the flexibility to scale the web server as per the traffic demands, ensuring that the website can handle varying loads efficiently.
2. **Reliability and Availability**: By utilizing AWS RDS, we ensured that the database is highly available and can automatically scale according to the storage and performance requirements, reducing the chances of downtime or data loss.
3. **Security**: The integration of security groups and proper configuration of AWS IAM roles ensured that both the EC2 instance and RDS database were securely accessible, providing a secure environment for hosting sensitive data.
4. **Cost Efficiency**: AWS's pay-as-you-go pricing model allowed us to optimize costs by only paying for the resources that were actively being used, making it a cost-effective solution for deploying a dynamic website.

In conclusion, deploying a dynamic website using AWS EC2 and RDS has proven to be a robust and efficient solution for building scalable and secure web applications, providing the flexibility and tools required to manage and scale the website as it grows.