

Three-Tier Project Deployment on AWS.

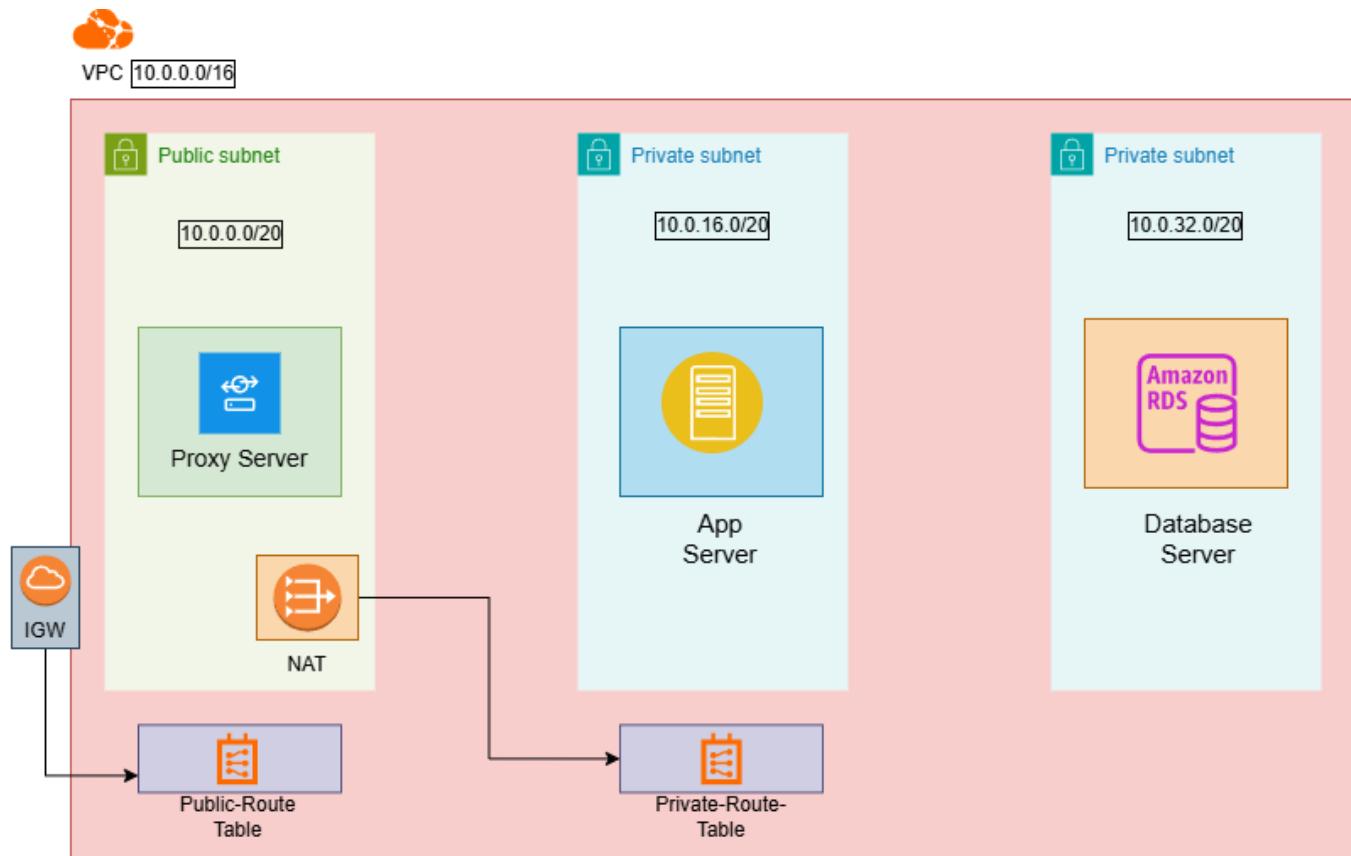
Introduction

This project presents a **Java-based Student Registration Web Application** deployed using a **Three-Tier Architecture on Amazon Web Services (AWS)**. The infrastructure features an **NGINX Proxy Server** within the public subnet, an **Apache Tomcat Application Server** in a private subnet, and a **MariaDB Database** hosted on **Amazon RDS**. The **Proxy Server** not only manages incoming HTTP requests but also functions as a **Bastion Host (Jump Server)** to enable secure SSH access to the private-tier instances. The entire environment is deployed within a **custom AWS VPC, utilizing subnets, route tables, a NAT gateway, and security groups** to ensure controlled and secure communication between layers. This setup effectively demonstrates how user requests travel through the proxy to the application and database layers, modeling a real-world, scalable, and secure cloud architecture.

Architecture Overview

The architecture consists of:

- **VPC (10.0.0.0/16)** – Custom Virtual Private Cloud hosting all resources.
- **Public Subnet (10.0.0.0/20)** – Hosts the Proxy (Web) Server.
- **Private Subnet 1 (10.0.16.0/20)** – Hosts the Application Server.
- **Private Subnet 2 (10.0.32.0/20)** – Hosts the Database (Amazon RDS).
- **NAT Gateway**: In Public Subnet with Elastic IP
- **Security Group Ports**:
 - 22 (SSH)
 - 80 (HTTP)
 - 8080 (Java Application)
 - 3306 (MySQL)



Tech Stack

Tier	Component	Technology Used
Presentation	Proxy / Web Server	Nginx / Apache / React / HTML
Application	App Server	Node.js / Flask / Spring Boot
Database	Database Server	Amazon RDS (MySQL / PostgreSQL)
Cloud Infrastructure	AWS Services	VPC, EC2, RDS, NAT Gateway, Route Tables, IGW, Subnets

Deployment Steps

PART 1: Create Networking Resources

Step 1: Create VPC

- Name tag: three-tier-vpc
- IPv4 CIDR block: 10.0.0.0/16
- Tenancy: Default

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.
three-tier-VPC

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block

Step 2: Create Subnet

- Public Subnet: 10.0.0.0/20 (for Proxy Server)
- Private Subnet 1: 10.0.16.0/20 (for App Server)
- Private Subnet 2: 10.0.32.0/20 (for DB Server)

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
public-subnet
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-1a (us-east-1a)

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.0.0/20 4,096 IPs

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
private-subnet-1
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-1b (us-east-1b)

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.16.0/20 4,096 IPs

Subnet 3 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
private-subnet-2
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-1c (us-east-1c)

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.32.0/20 4,096 IPs

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar for 'Virtual private cloud' with 'Your VPCs' and 'Subnets' selected. The main area displays a table of subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
private-subnet-1	subnet-0a6b5e7ae0df87b54	Available	vpc-0474ec95e956f7656 three-tier-vpc	Off	10.0.16.0/20
private-subnet-2	subnet-057b76af1cf638f15	Available	vpc-0474ec95e956f7656 three-tier-vpc	Off	10.0.32.0/20
public-subnet	subnet-014d0bb20a2edfabe	Available	vpc-0474ec95e956f7656 three-tier-vpc	Off	10.0.0.0/20

Step 3: Create Internet Gateway

- In the VPC Dashboard, select Internet Gateways and click "Create Internet Gateway."
 - Name tag: three-tier-igw
- After creation, choose the new Internet Gateway and click "Attach to VPC."
 - Select your previously created VPC (three-tier-vpc).

Create internet gateway [Info](#)
An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.
three-tier-IGW

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="three-tier-IGW"/> Remove

Add new tag You can add 49 more tags.

[Cancel](#) **Create internet gateway**

The following internet gateway was created: igw-04bcd1cb15078f8d6 - three-tier-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to VPC (igw-04bcd1cb15078f8d6) Info

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

Q vpc-0474ec95e956f7656 X

▶ AWS Command Line Interface command

Cancel Attach internet gateway

Step 4: Configure Public Route Table

Update Public Route Table to add an IGW route.

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute
	igw-04bcd1cb15078f8d6			
	igw-04bcd1cb15078f8d6 (three-tier-IGW)			

Add route Remove Cancel Preview Save changes

Step 5: Create NAT Gateway

Configure the NAT Gateway as follows:

- Name tag: three-tier-nat-gateway
- Subnet: Select the Public Subnet (where the Proxy Server resides).
- Elastic IP allocation: Choose “Allocate Elastic IP automatically.”

Elastic IP address 52.45.31.74 (eipalloc-0a6a9351f89266678) allocated.

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
Three-tier-NAT
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-014d0bb20a2edfabe (public-subnet)

Connectivity type
Select a connectivity type for the NAT gateway.
 Public
 Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-0a6a9351f89266678

Allocate Elastic IP

Step 6: Create Private Route table inside your VPC and add route of NAT gateway

Edit Subnet Association

Available subnets (2/3)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
private-subnet-1	subnet-0a6b5e7ae0df87b54	10.0.16.0/20	-	Main (rtb-02cc35846364c8dba / public...)
private-subnet-2	subnet-057b76af1cf638f15	10.0.32.0/20	-	Main (rtb-02cc35846364c8dba / public...)
public-subnet	subnet-014d0bb20a2edfabe	10.0.0.0/20	-	rtb-02cc35846364c8dba / public-RT

Selected subnets

- subnet-0a6b5e7ae0df87b54 / private-subnet-1
- subnet-057b76af1cf638f15 / private-subnet-2

Buttons: Cancel, Save associations

Add Route of NAT Gateway

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
0.0.0.0/0	NAT Gateway	-	No	CreateRoute

Buttons: Add route, Remove, Cancel, Preview, Save changes

Step 7: Create a Security Group

Create a Security Group with inbound rules:

- 22 (SSH)
- 80 (HTTP)
- 8080 (Tomcat)
- 3306 (MySQL/RDS)

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
SSH	TCP	22	Anyw... ▾	0.0.0.0/0 X
HTTP	TCP	80	Anyw... ▾	0.0.0.0/0 X
MySQL/Aurora	TCP	3306	Anyw... ▾	0.0.0.0/0 X
Custom TCP	TCP	8080	Anyw... ▾	0.0.0.0/0 X

[Add rule](#)

PART 2: Launch EC2 Instances and Create RDS

Tier	Subnet	Description	Security Group
Proxy Tier	Public Subnet	NGINX Reverse Proxy Server (also acts as Bastion Host for SSH access)	three-tier-sg
Application Tier	Private Subnet 1	Apache Tomcat Server hosting the Java-based Student Registration Web App	three-tier-sg
Database Tier	Private Subnet 2	Amazon RDS instance running MariaDB for secure and reliable data storage	three-tier-sg

Step 1: Create Proxy-server

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0474ec95e956f7656 (three-tier-VPC)
 10.0.0.0/16

Subnet [Info](#)

subnet-014d0bb20a2edfbabe
 VPC: vpc-0474ec95e956f7656 Owner: 952406520450
 Availability Zone: us-east-1a (use1-az6) Zone type: Availability Zone
 IP addresses available: 4090 CIDR: 10.0.0.0/20

public-subnet

Create new subnet

Auto-assign public IP [Info](#)

Enable

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](#)

Select security groups

three-tier-Sg sg-0e7ead3dbb0d937fe X
 VPC: vpc-0474ec95e956f7656

Compare security group rules

Step 2: Create App-server

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

app [Add additional tags](#)

Network settings [Info](#)

VPC - required [Info](#)

vpc-0474ec95e956f7656 (three-tier-VPC)
 10.0.0.0/16

Subnet [Info](#)

subnet-0a6b5e7ae0df87b54
 VPC: vpc-0474ec95e956f7656 Owner: 952406520450
 Availability Zone: us-east-1b (use1-az1) Zone type: Availability Zone
 IP addresses available: 4091 CIDR: 10.0.16.0/20

private-subnet-1

Create new subnet

Auto-assign public IP [Info](#)

Disable

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](#)

Select security groups

three-tier-Sg sg-0e7ead3dbb0d937fe X
 VPC: vpc-0474ec95e956f7656

Compare security group rules

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

Step 3: Create DB-server

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name [Add additional tags](#)

▼ Network settings Info

VPC - required Info

[Create new VPC](#)

Subnet Info

VPC: vpc-0474ec95e956f7656 Owner: 952406520450 Availability Zone: us-east-1c (use1-az2) Zone type: Availability Zone IP addresses available: 4091 CIDR: 10.0.32.0/20 [Create new subnet](#)

Auto-assign public IP Info

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

VPC: vpc-0474ec95e956f7656 [Compare security group rules](#)

Step 4: Create RDS instance (MariaDB) in the same VPC with the same security group.

Aurora and RDS Databases

Databases

- Dashboard
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups

Successfully created database three-tier-rds

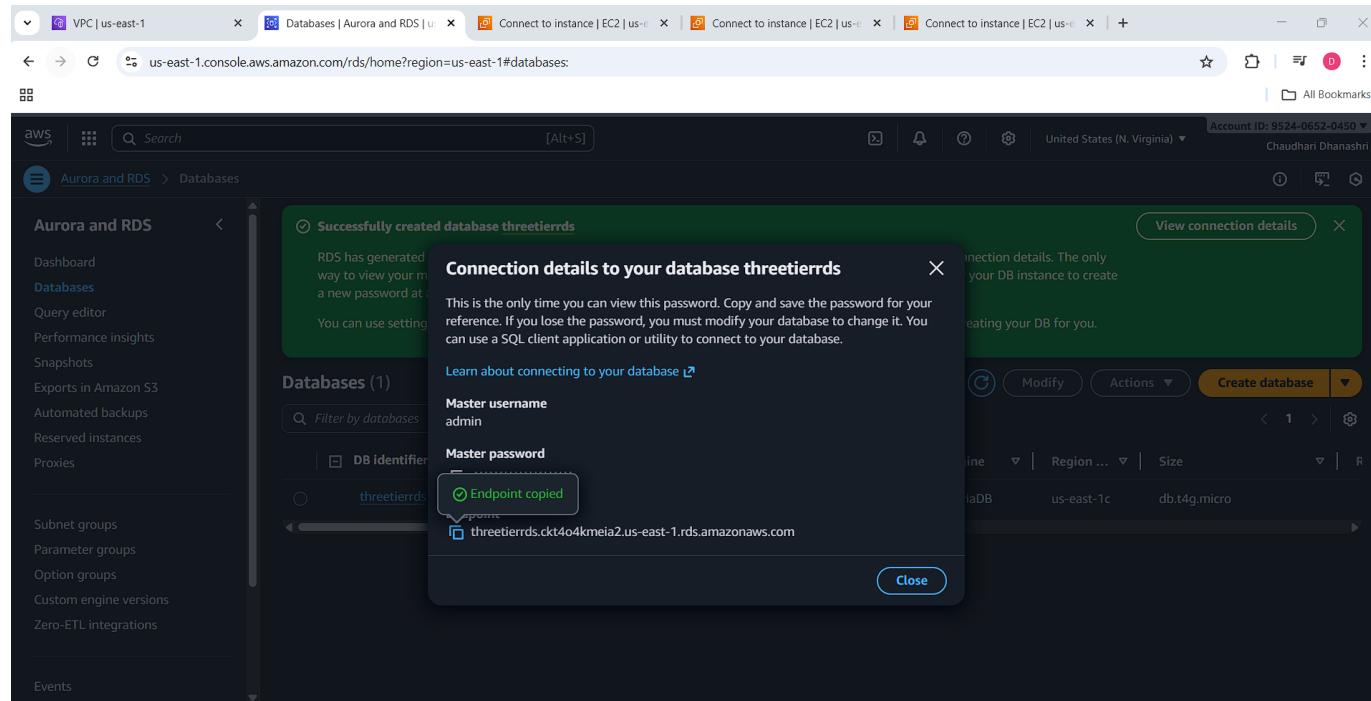
RDS has generated your database master password during the database creation and it will be displayed in the connection details. The only way to view your master password is to choose [View connection details](#) during database creation. You can modify your DB instance to create a new password at any time.

You can use settings from three-tier-rds to simplify configuration of [suggested database add-ons](#) while we finish creating your DB for you.

Databases (1)

DB identifier	Status	Role	Engine	Region ...	Size
three-tier-rds	Available	Instance	MariaDB	us-east-1c	db.t4g.micro

Copy the endpoint and password



PART 3: Proxy Server Setup (NGINX)

Step 1: Connect via SSH to the Proxy instance.

```
HP@LAPTOP-HFSFVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ ssh -i "key.pem" ec2-user@13.217.99.162
The authenticity of host '13.217.99.162 (13.217.99.162)' can't be established.
ED25519 key fingerprint is SHA256:B1P2Djr7sxv4l7SP5RqE7gtOfBjg+yoPhZZHpeLx+4E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.217.99.162' (ED25519) to the list of known hosts.

,      #
~\_\_ #####_      Amazon Linux 2023
~~ \_\_#####\_
~~ \#\#|
~~   \#/ ,__->
~~     V~' ,__->
~~     /   /
~~   ._.  /  /
~~   /_ /_/
~~   /m/ ,_
[ec2-user@ip-10-0-0-54 ~]$ sudo hostnamectl hostname proxy
[ec2-user@ip-10-0-0-54 ~]$ exit
Logout
Connection to 13.217.99.162 closed.

HP@LAPTOP-HFSFVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ ssh -i "key.pem" ec2-user@13.217.99.162
,      #
~\_\_ #####_      Amazon Linux 2023
~~ \_\_#####\_
~~ \#\#|
~~   \#/ ,__->
~~     V~' ,__->
~~     /   /
~~   ._.  /  /
~~   /_ /_/
~~   /m/ ,_
Last login: Fri Oct 31 05:34:40 2025 from 223.228.133.224
[ec2-user@proxy ~]$ |
```

Step 2: Install and start NGINX

```
sudo yum update -y
sudo yum install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
```

```
[ec2-user@proxy:~]
[ec2-user@proxy ~]$ sudo yum update
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@proxy ~]$ sudo yum install nginx -y
Last metadata expiration check: 0:00:12 ago on Fri Oct 31 05:36:37 2025.
Dependencies resolved.
=====
Package           Architecture      Version       Repository
=====
Installing:
nginx            x86_64          1:1.28.0-1.amzn2023.0.2   amazonlinux
Installing dependencies:
generic-logos-httd noarch          18.0.0-12.amzn2023.0.3   amazonlinux
gperf-tools-libs  x86_64          2.9.1-1.amzn2023.0.3    amazonlinux
libunwind         x86_64          1.4.0-5.amzn2023.0.3    amazonlinux
nginx-core        x86_64          1:1.28.0-1.amzn2023.0.2   amazonlinux
nginx-filesystem noarch          1:1.28.0-1.amzn2023.0.2   amazonlinux
nginx-mimetypes  noarch          2.1.49.3.amzn2023.0.3    amazonlinux
```

Step 3: Edit NGINX configuration

Go to /etc/nginx/nginx.conf

```
[ec2-user@proxy:~]
[ec2-user@proxy ~]$ sudo systemctl start nginx
[ec2-user@proxy ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service
[ec2-user@proxy ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@proxy ~]$
```

Inside the server block, add: location / { proxy_pass http://:8080/student/; }

```
error_page 500 502 503 504 /50x.html;
location = /50x.html {
}
location / {
proxy_pass http://10.0.28.121:8080/student/;
}
```

Step 4: Restart NGINX

```
sudo systemctl restart nginx
```

NGINX will now forward external traffic to your Tomcat server.

PART 4: Application Server Setup (Tomcat)

Step 1: From jump server(Proxy) Connect to your App instance.

```

HP@LAPTOP-HFSVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ scp -i key.pem key.pem ec2-user@13.217.99.162:/home/ec2-user/
key.pem                                         100%
HP@LAPTOP-HFSVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ ssh -i "key.pem" ec2-user@13.217.99.162
,
~\_ #_
~~ \####` Amazon Linux 2023
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~, '-->
~~ .-. / \
~~ / , / \
~~ /m/ /
Last login: Fri Oct 31 05:35:35 2025 from 223.228.133.224
[ec2-user@proxy ~]$ ls
key.pem
[ec2-user@proxy ~]$ sudo ssh -i key.pem ec2-user@10.0.28.121
The authenticity of host '10.0.28.121 (10.0.28.121)' can't be established.
ED25519 key fingerprint is SHA256:Kue6l8/dkTEjIRpk3YspBknnXdvU1G1BEpaWaYXRbsw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.28.121' (ED25519) to the list of known hosts.
,
~\_ #_
~~ \####` Amazon Linux 2023
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~, '-->
~~ .-. / \
~~ / , / \
~~ /m/ /

```

Step 2: Install Java and Tomcat

- update system
- install java
- install tomcat

```

sudo yum update -y
sudo yum install java -y
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.98/bin/apache-tomcat-
9.0.98.tar.gz
sudo tar -xvzf apache-tomcat-9.0.98.tar.gz -C /opt

```

```

ec2-user@app:~
[ec2-user@app ~]$ sudo yum update
Last metadata expiration check: 0:08:28 ago on Fri Oct 31 06:10:57 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@app ~]$ sudo yum install java -y
Last metadata expiration check: 0:08:41 ago on Fri Oct 31 06:10:57 2025.
Dependencies resolved.
-----

```

Package	Architecture	Version

```
ec2-user@app:~$ [ec2-user@app ~]$ sudo curl -o https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.111/bin/apache-tomcat-9.0.111.tar.gz
[ec2-user@app ~]$ ls apache-tomcat-9.0.111.tar.gz
[ec2-user@app ~]$ sudo tar -xvzf apache-tomcat-9.0.111.tar.gz -C /opt
apache-tomcat-9.0.111/conf/
apache-tomcat-9.0.111/conf/catalina.policy
apache-tomcat-9.0.111/conf/catalina.properties
apache-tomcat-9.0.111/conf/context.xml
apache-tomcat-9.0.111/conf/jaspic-providers.xml
apache-tomcat-9.0.111/conf/jaspic-providers.xsd
apache-tomcat-9.0.111/conf/logging.properties
apache-tomcat-9.0.111/conf/server.xml
apache-tomcat-9.0.111/conf/tomcat-users.xml
apache-tomcat-9.0.111/conf/tomcat-users.xsd
apache-tomcat-9.0.111/conf/web.xml
```

Step 3: Check that tomcat is installed correctly

```
root@app:/opt/apache-tomcat-9.0.111/webapps$ [root@app ~]# cd /opt/apache-tomcat-9.0.111/
[root@app apache-tomcat-9.0.111]# ls
BUILDING.txt      LICENSE  README.md      RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md   NOTICE   RELEASE-NOTES bin        lib    temp  work
[root@app apache-tomcat-9.0.111]# cd webapps
```

Step 4: Deploy your application WAR file

Deploy your web application inside webapps

```
cd /opt/apache-tomcat/webapps
curl -O <S3-Bucket-URL-to-App-Code>
```

```
[root@app bin]# cd ..
[root@app apache-tomcat-9.0.111]# cd webapps
[root@app webapps]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent   Left  Speed
100 89423  100 89423    0     0  231k      0 --:--:-- --:--:-- 231k
[root@app webapps]# ls
ROOT  docs  examples  host-manager  manager  student  student.war
[root@app webapps]#
```

Step 5: Restart Tomcat

```
cd /opt/apache-tomcat/bin
./catalina.sh stop
./catalina.sh start
```

Step 6: Check Java

- http://Proxy-Public-IP



Student Registration Form

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"/>	

PART 5: Database Setup (MariaDB / RDS)

Step 1: SSH into DB instance

```
[ec2-user@proxy ~]$ sudo ssh -i key.pem ec2-user@10.0.46.73
The authenticity of host '10.0.46.73 (10.0.46.73)' can't be established.
ED25519 key fingerprint is SHA256:j51X0PIRYxfZbym8LNmD5ciK4RQwWVbFHwzJ
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.46.73' (ED25519) to the list of known hosts.

,#
~\####
~~ \####\
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~, '-->
~~   /
~~ ._. / /
~~ ._. / /
~~ /m/ .

[ec2-user@ip-10-0-46-73 ~]$ sudo hostnamectl hostname db
[ec2-user@ip-10-0-46-73 ~]$ exit
Logout
Connection to 10.0.46.73 closed.
[ec2-user@proxy ~]$ sudo ssh -i key.pem ec2-user@10.0.46.73
,#
~\####
~~ \####\
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~, '-->
~~   /
~~ ._. / /
~~ ._. / /
~~ /m/ .

Last login: Fri Oct 31 06:45:13 2025 from 10.0.0.54
[ec2-user@db ~]$
```

Step 2: Take access of RDS

- Install mariadb

```
[ec2-user@db ~]$ sudo yum update
Amazon Linux 2023 repository
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Fri Oct 31 06:46:28 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@db ~]$ sudo yum install mariadb105-server -y
Last metadata expiration check: 0:00:25 ago on Fri Oct 31 06:46:28 2025.
Dependencies resolved.
=====
 Package                               Architecture Version
=====
Installing:
 mariadb105-server                   x86_64        3:10.5.29-1.amzn2023.0.1
Installing dependencies:
 mariadb-connector-c                 x86_64        3.3.10-1.amzn2023.0.1
```

- Take access of RDS

```
[ec2-user@db ~]$ sudo mysql -h threetierrds.ckt4o4kmeia2.us-east-1.rds.amazonaws.com -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 126
Server version: 11.4.8-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 (0.005 sec)

MariaDB [(none)]> |
```

Step 3:Create database and table

```
CREATE DATABASE studentapp;

USE studentapp;

CREATE TABLE 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)
);
```

```
MariaDB [(none)]> create database studentapp;
Query OK, 1 row affected (0.005 sec)

MariaDB [(none)]> use studentapp
Database changed
MariaDB [studentapp]> CREATE TABLE 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));
Query OK, 0 rows affected (0.022 sec)

MariaDB [studentapp]> show tables;
+-----+
| Tables_in_studentapp |
+-----+
| students              |
+-----+
1 row in set (0.003 sec)

MariaDB [studentapp]> select * from students;
Empty set (0.001 sec)

MariaDB [studentapp]> |
```

PART 6: Connect App Server to RDS

Step 1: Install JDBC connector in App server

```
root@app:/opt/apache-tomcat-9.0.111/lib#
[ec2-user@app ~]$ sudo -i
[root@app ~]# cd /opt/apache-tomcat-9.0.111/lib
[root@app lib]# ls
annotations-api.jar      ejc-4.20.jar      tomcat-api.jar      tomcat-i18n-fr.jar      tomcat-jni.jar
catalina-ant.jar          el-api.jar       tomcat-coyote-ffm.jar tomcat-i18n-ja.jar      tomcat-util-scan.jar
catalina-ha.jar           jasper-el.jar   tomcat-coyote.jar    tomcat-i18n-ko.jar      tomcat-util.jar
catalina-ssi.jar          jasper.jar      tomcat-dbcp.jar    tomcat-i18n-pt-BR.jar  tomcat-websocket.jar
catalina-storeconfig.jar  jaspic-api.jar  tomcat-i18n-cs.jar  tomcat-i18n-ru.jar      websocket-api.jar
catalina-tribes.jar       jsp-api.jar     tomcat-i18n-de.jar  tomcat-i18n-zh-CN.jar
catalina.jar               servlet-api.jar tomcat-i18n-es.jar  tomcat-jdbc.jar
[root@app lib]# curl -o https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 983k 100 983k 0 0 1556k 0 --:--:--:--:--:-- 1556k
[root@app lib]# ls
annotations-api.jar      ejc-4.20.jar      servlet-api.jar      tomcat-i18n-es.jar      tomcat-jdbc.jar
catalina-ant.jar          el-api.jar       tomcat-api.jar      tomcat-i18n-fr.jar      tomcat-jni.jar
catalina-ha.jar           jasper-el.jar   tomcat-coyote-ffm.jar tomcat-i18n-ja.jar      tomcat-util-scan.jar
catalina-ssi.jar          jasper.jar      tomcat-coyote.jar  tomcat-i18n-ko.jar      tomcat-util.jar
catalina-storeconfig.jar  jaspic-api.jar  tomcat-dbcp.jar   tomcat-i18n-pt-BR.jar  tomcat-websocket.jar
catalina-tribes.jar       jsp-api.jar     tomcat-i18n-cs.jar  tomcat-i18n-ru.jar      websocket-api.jar
catalina.jar               mysql-connector.jar tomcat-i18n-de.jar  tomcat-i18n-zh-CN.jar
[root@app lib]# |
```

Step 2: Edit the context file

```
cd /opt/apache-tomcat/conf
vim context.xml
```

```
[root@app lib]# cd ..
[root@app apache-tomcat-9.0.111]# ls
BUILDING.txt      LICENSE      README.md      RUNNING.txt  conf      logs      webapps
CONTRIBUTING.md   NOTICE       RELEASE-NOTES  bin        lib      temp      work
[root@app apache-tomcat-9.0.111]# cd conf
[root@app 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@app conf]# vim context.xml
[root@app conf]# |
```

Step 3: Add this configuration inside context block

```
<Resource name="jdbc/TestDB" auth="Container"
type="javax.sql.DataSource"
```

```
maxTotal="500" maxIdle="30" maxWaitMillis="1000"
username="admin" password="redhat123!"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://<RDS-ENDPOINT>:3306/studentapp?
useUnicode=yes&characterEncoding=utf8"/>
```

```
root@app-server:/opt/apache-tomcat-9.0.11/conf
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="P8LbTSLDJHKtvTjk3x4n" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://three-tier-db.ckt4o4kmeia2.us-east-1.rds.amazonaws.com:3306/studentapp?useUnicode=yes&characterEncoding=utf8"/>
</Context>
"context.xml" 35L, 1762B
```

34,112 Bot

Step 4: Restart Tomcat

```
cd /opt/apache-tomcat/bin
./catalina.sh stop
./catalina.sh start
```

PART 7: Access the Application and add entries

visit: <http://Proxy-Public-IP>

Student Registration Form

Student Name	<input type="text" value="abc"/>
Student Address	<input type="text" value="pune"/>
Student Age	<input type="text" value="21"/>
Student Qualification	<input type="text" value="btech"/>
Student Percentage	<input type="text" value="88"/>
Year Passed	<input type="text" value="2024"/>
<input type="button" value="register"/>	

Students List

Student ID	StudentName	Student Addrs	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	abc	pune	21	btech	88	2025	edit	delete

PART 8: Verify entries in the RDS

```
ec2-user@db-server:~$ MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| mysql |
| performance_schema |
| studentapp |
| sys |
+-----+
6 rows in set (0.001 sec)

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]> select * from students;
+-----+-----+-----+-----+-----+-----+-----+
| student_id | student_name | student_addr | student_age | student_qual | student_percent | student_year_passed |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | abc | pune | 21 | btech | 88 | 2025 |
| 2 | dhanashri | dhule | 22 | btech | 84 | 2025 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Final Result

Successfully deployed a **3-Tier Web Application on AWS** featuring:

- Isolated **VPC architecture** for secure networking
- **Public and private subnets** for controlled access
- **NAT Gateway** enabling private subnet internet access
- **NGINX Reverse Proxy** for traffic routing and load balancing
- **RDS (MariaDB)** integration for database storage
- Fully functional **Student Registration System** hosted on AWS