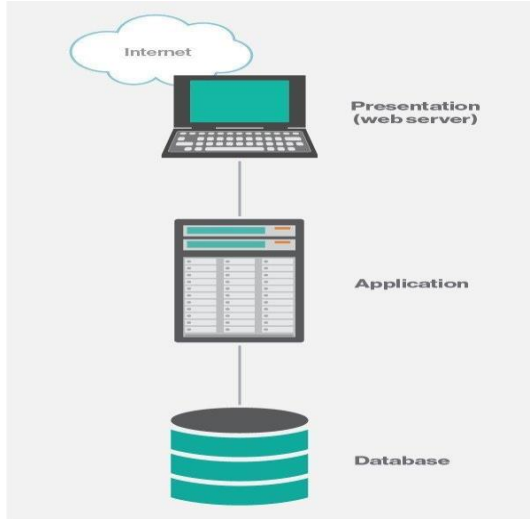# 3 Tier Architecture of AWS

Before start the practical I will explain you how 3 tier architecture works.

There are three lyres in 3 tier architecture:-

Presentation tier:-The presentation tier is the user interface and communication layer of the application, where the end user interacts with the application.

Application tier:-The application tier, also known as the logic tier or middle tier, is the heart of the application.

Database tier:-The data tier, sometimes called database tier, data access tier or back-end, is where the information processed by the application is stored and managed.

Step 1:-

First of all login to AWS account.
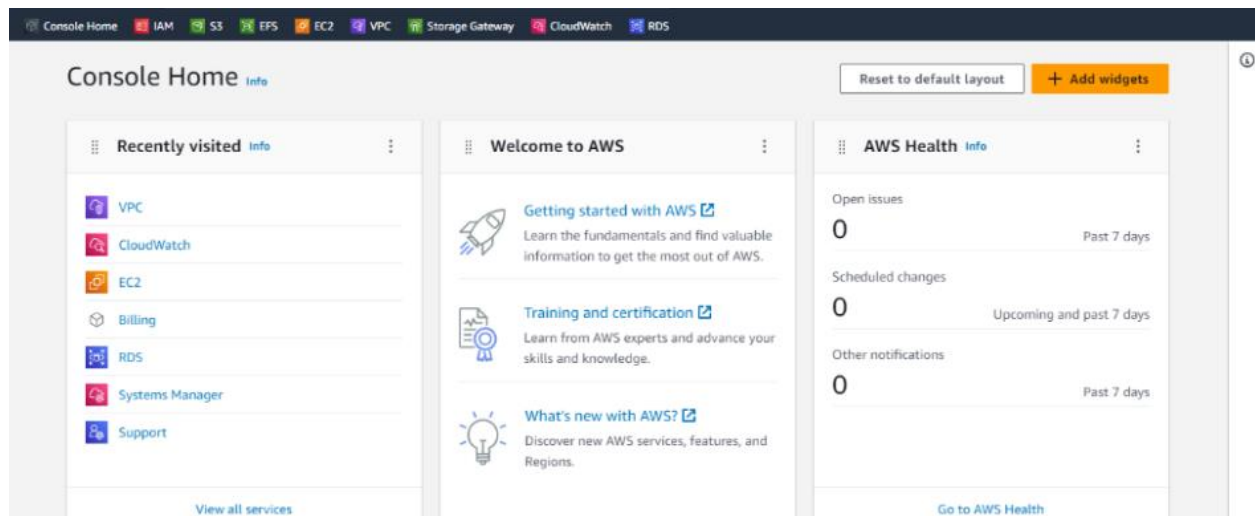
Step 2:-

Now we have to firstly create VPC so go to VPC service. And click on create VPC. Then click on VPC only and then gave VPC name.

Step 3:-

Now select IPv4 and then select IPv4 range and then tenancy as default.

IPv4 CIDR block Info

● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

```
192.168.0.0/24
```

IPv6 CIDR block Info

● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info

```
Default                                              ▼
```

Step 4:-

Now go to subnet session and click on create subnet and select your VPC.

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

```
vpc-0b737931ab7cc0112 (my-vpc)                        ▼
```

Associated VPC CIDRs

IPv4 CIDRs

192.168.0.0/24

Step 5:-

Now give a first subnet name as public subnet and then select any availability zone and then ipv4.

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

my-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.

Europe (Ireland) / eu-west-1a	▼

IPv4 CIDR block Info

Q 192.168.0.0/27	✕

Step 6:-

After that click on add new subnet and select availability zone and then gave IPv4 range.

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

my-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.

Europe (Ireland) / eu-west-1b	▼

IPv4 CIDR block Info

Q 192.168.0.32/27	✕

Step 7:-

And next again click on add new subnet and gave it name as private subnet 2 and then select availability zone after that gave IPv4.

Subnet 3 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

my-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.

Europe (Ireland) / eu-west-1c   ▼

IPv4 CIDR block Info

Q 192.168.0.64/27   ✕

Step 8:-

Now last one subnet you have to add as RDS subnet. So click on add new subnet and gave name RDS subnet and then gave availability zone and then gave IPv4 and click on create subnet.

Subnet 4 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

my-RDS-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.

Europe (Ireland) / eu-west-1a   ▼

Use: "192.168.0.96/27"

Q 192.168.0.96/27   ✕

Step 9:-

Now you have to create route table to attach public subnet and IGW.

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

```
my-RT-public
```

VPC
The VPC to use for this route table.

```
vpc-0b737931ab7cc0112 (my-vpc)                    ▼
```

Tags

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	
🔍 Name ✕	🔍 my-RT-public ✕	Remove

Add new tag
You can add 49 more tags.

Cancel   **Create route table**

Step 10:-

After that create rote table to connect private subnets and NAT connect.

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

```
my-private-RT
```

VPC
The VPC to use for this route table.

```
vpc-0b737931ab7cc0112 (my-vpc)                    ▼
```

Step 11:-

Now add last one route table to connect RDS subnet.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

```
my-RDS-RT
```

VPC
The VPC to use for this route table.

```
vpc-0b737931ab7cc0112 (my-vpc)              ▼
```

Tags

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*	
🔍 Name ✕	🔍 my-RDS-RT ✕	Remove

Add new tag

You can add 49 more tags.

Cancel | **Create route table**

Step 12:-

After that go to internet gateway session to create IGW for connect to public RT, get internet access to public instance.

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.

```
my-NAt-gateway
```

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*	
🔍 Name ✕	🔍 my-NAt-gateway ✕	Remove

Add new tag

You can add 49 more tags.

Cancel | **Create internet gateway**

Step 13:-

Now go to NAT session to create NAT gateway to get internet access to private server through the public server. click on create NAT gateway and then give it name and select subnet of public and then select connective type public and allocate it Elastic IP and save it.

Step 14:-

And your VPC was ready. Select the VPC and then click on map to see connection between VPC, Subnet, Route table and Network connection, So we have to connect step by step one.
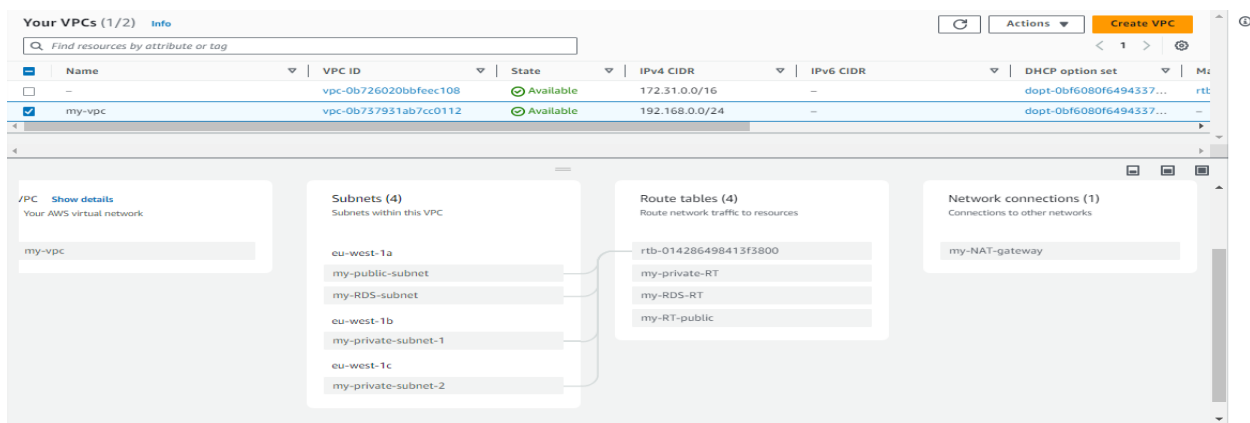
Step 15:-

First of all we connect subnet to route table. Go to subnet session and select public subnet and then click route table option and then click on edit route table association.

Step 16:-

Now select public Route Table and hit save option.

Step 17:-

As it is we do that steps to public subnet do same. Select private instance 1 and then click route table and then edit route table association.

Step 18:-

Now select the private Route table and save it.

Step 19:-

As it is we do that steps to private subnet do same. Select private instance 1 and then click route table and then edit route table association.

Step 20:-

Now select the private Route table and save it.

Step 21:-

Now you have to select RDS subnet and click on route table and then edit route table association. And then select RDS and then save it.

Subnet route table settings

Subnet ID

subnet-0c1f8f26a58008257

Route table ID

rtb-0514f7e46c45e7efb (my-RDS-RT)   ▲   ⟳

🔍 |

rtb-014286498413f3800
Main route table   eachability Analyzer   ✕

rtb-02d7e373ce675a004 (my-private-RT)

rtb-0514f7e46c45e7efb (my-RDS-RT)   ✓

rtb-0a1893f949af5e484 (my-RT-public)   ‹ 1 ›   ⚙

Destination ▽	Target ▽
192.168.0.0/24	local

Cancel   **Save**

Step 22:-

Now you have to create internet gateway to get internet access to public instance. And connect IGW to your VPC and by select it and click on action then click on attach to VPC.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

my-IGW-gateway

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*	
🔍 Name ✕	🔍 my-IGW-gateway ✕	Remove

Add new tag
You can add 49 more tags.

Cancel   **Create internet gateway**

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.

🔍 vpc-0b737931ab7cc0112 ✕

▶ **AWS Command Line Interface command**

Cancel   **Attach internet gateway**

Step 23:-

Now go to route table session and select RT public and click on action then click on edit routes and after that you have see add route option click on that option and then select first column and then select 0.0.0.0/0 subnet and then click on second column and select in that internet gateway option and then select your IGW and then click on save changes.

Step 24:-

 Now, you have to gave a internet connection to private instances so you have to create NAT gateway so go back to RT session and then select RT private and then click on action then select edit route option and then you have see new screen and then click on add route after that click on first column and select 0.0.0.0/0 and then nest click on second column and NAT gateway and then select your NAT gateway and then click on save changes. And your VPC was ready.

Step 25:-

Now you have to create a data base by RDS service so go to RDS service and then click on create data base option.
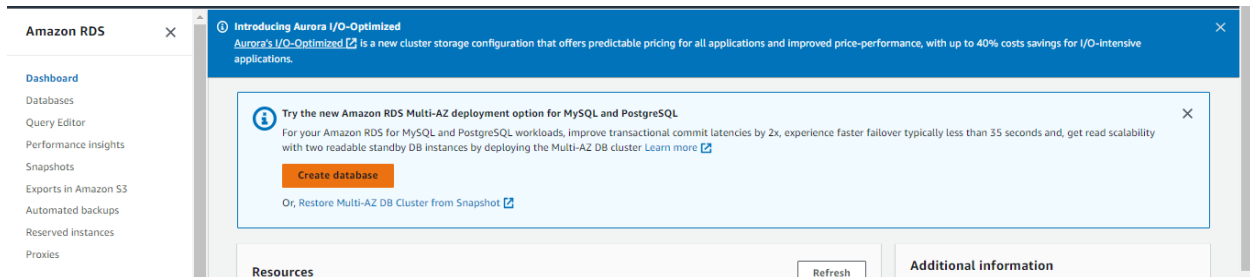
Step 26:-

Now you have see the create data base setting so choose a database creation method as standard create.
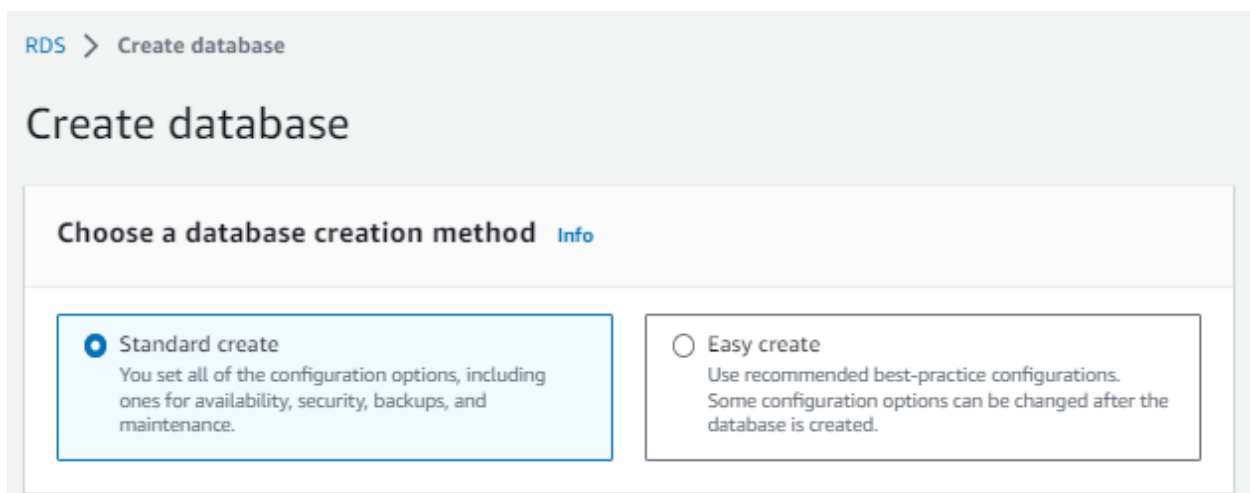
Step 27:-

After that select MariaDB engine to create data base. And next select of engine version as you want.

Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

○ MySQL

● MariaDB

○ PostgreSQL

○ Oracle

ORACLE®

○ Microsoft SQL Server

Microsoft SQL Server

▼ **Hide filters**

Show versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version

MariaDB 10.6.14   ▼

Step 28:-

Now select template type as production.

Templates
Choose a sample template to meet your use case.

● Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

○ Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

Step 29:-

Now gave database name and then gave username as that database access.

Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

```
my-mariad-database
```

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username Info
Type a login ID for the master user of your DB instance.

```
ani
```

1 to 16 alphanumeric characters. The first character must be a letter.

Step 30:-

Now lave manage credentials in AWS secrets manager as it as and then gave database user password.

☐ **Manage master credentials in AWS Secrets Manager**
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

ⓘ If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
Learn more ↗

☐ **Auto generate a password**
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

```
••••••••••••••
```

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password Info

```
••••••••••••
```

Step 31:

Now choose instance configuration. I leave as it.

Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

> ⓘ **Amazon RDS Optimized Writes - *new*** Info
> ⬤ Show instance classes that support Amazon RDS Optimized Writes

DB instance class **Info**
- ⦿ Standard classes (includes m classes)
- ◯ Memory optimized classes (includes r and x classes)
- ◯ Burstable classes (includes t classes)

db.m6gd.large (supports Amazon RDS Optimized Writes)
2 vCPUs   8 GiB RAM   Network: 4,750 Mbps   ▼

⬤ Include previous generation classes

Step 32:-

Now, select storage setting as you want and then scroll down.

Storage

Storage type   **Info**
Provisioned IOPS SSD (io1)
Flexibility in provisioning I/O   ▼

Allocated storage   **Info**
400   GiB
The minimum value is 100 GiB and the maximum value is 65,536 GiB

> ⓘ After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.
> Learn more ↗

Provisioned IOPS   **Info**
3000   IOPS
The minimum value is 1,000 IOPS and the maximum value is 160,000 IOPS. The IOPS to GiB ratio must be between 0.5 and 50

> ⓘ Your actual IOPS might vary from the amount that you provisioned based on your database workload and instance type. Learn more ↗

Storage autoscaling   **Info**
Provides dynamic scaling support for your database's storage based on your application's needs.
☑ Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold   **Info**
Charges will apply when your database autoscales to the specified threshold
1000   GiB
The minimum value is 440 GiB and the maximum value is 65,536 GiB

Step 33:-

Now select availability and durability to database I gave it and then scroll down.

Availability & durability

Multi-AZ deployment **Info**

○ Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

○ Do not create a standby instance

Step 34:-

Now you have to gave a connectivity setting properly. Now you have to select instance setting.

I have not connect instance now because I have to gave database a private subnet so I was choose don't connect ec2 option.

Connectivity Info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

○ Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

○ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Step 35:-

Then select network type ipv4.

Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

○ IPv4
Your resources can communicate only over the IPv4 addressing protocol.

○ Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Step 36:-

Now you have to select your own created VPC.

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

my-vpc (vpc-0b737931ab7cc0112)
4 Subnets, 3 Availability Zones ▼

Only VPCs with a corresponding DB subnet group are listed.

Step 37:-

Now select subnet group If you have crated or not then leave as it as and then important on is don't select public access to database so select no option.

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

Create new DB Subnet Group ▼

Public access Info

○ Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

● No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

Step 38:-

Now select VPC security group in that SG you have to add port 8080 for tomcat,80 for nginx or httpd and 3306 for MariaDB. And then scroll down leave other setting as it is. Only select you added port SG.

VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

● Choose existing	○ Create new
Choose existing VPC security groups	Create new VPC security group

Existing VPC security groups

Choose one or more options ▼

my_VPC-SG ✕

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

☐ Create an RDS Proxy   Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing ↗.

Certificate authority - optional   Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-2019 (default) ▼

If you don't select a certificate authority, RDS chooses one for you.

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
SSH ▼	TCP	22	Anywh... ▼ 🔍		Delete
			0.0.0.0/0 ✕		
Custom TCP ▼	TCP	8080	Anywh... ▼ 🔍		Delete
			0.0.0.0/0 ✕		
HTTP ▼	TCP	80	Anywh... ▼ 🔍		Delete
			0.0.0.0/0 ✕		

Add rule

Step 39:-

Now select database authentication for use database so, I have choose password authentication. And then click on create database and your database is ready.

Database authentication

Database authentication options   Info

○ **Password authentication**
Authenticates using database passwords.

○ Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Step 40:-

Now you have to launch 3 instance one public and remaining 2 are private. So first add public instance with following configuration. First all configure and then edit network setting and then select your VPC and after that select private subnet then enable public IP.

▼ Network settings   Info

VPC - *required* Info

vpc-0b737931ab7cc0112 (my-vpc)
192.168.0.0/24

Subnet Info

subnet-05817dc304b8238f9   my-public-subnet
VPC: vpc-0b737931ab7cc0112   Owner: 132076981423
Availability Zone: eu-west-1a   IP addresses available: 26   CIDR: 192.168.0.0/27)

Create new subnet ⬈

Auto-assign public IP Info

Enable

Step 41:-

After that select a security group remember that In that SG you have add port 80 ,3306 ,8080 ,22 because we have to host nginx on this instance. And after launch instance with as it. Only add user data of configuration nginx.

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 ▼

my_VPC-SG   sg-0d372d310264bb3ca ✕
VPC: vpc-0b737931ab7cc0112

Compare security group rules

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

▶ Advanced network configuration

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 ▼

my_VPC-SG   sg-0d372d310264bb3ca ✕
VPC: vpc-0b737931ab7cc0112

Compare security group rules

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

▶ Advanced network configuration

Step 42:-

Now you have to launch private instance one so configure it gave name and all and then click on edit network setting and then select your VPC and then select private subnet 1 after that disable public IP and then select security group in that security group you have to add port of tomcat 8080 and then scroll down and write user data of configure tomcat. And then launch instance.

▼ **Network settings** Info

VPC - *required* Info

```
vpc-0b737931ab7cc0112 (my-vpc)
192.168.0.0/24
```
▼   C

Subnet Info

```
subnet-00c068df3658ecf6c                     my-private-subnet-1
VPC: vpc-0b737931ab7cc0112    Owner: 132076981423
Availability Zone: eu-west-1b    IP addresses available: 27    CIDR: 192.168.0.32/27)
```
▼   C   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
```
▼

```
my_VPC-SG  sg-0d372d310264bb3ca  ✕
VPC: vpc-0b737931ab7cc0112
```
C   **Compare security group rules**

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

▶ **Advanced network configuration**

User data – *optional* Info
Upload a file with your user data or enter it in the field.

⊞ Choose file

```
#! /bin/bash
sudo -i

wget https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.92/bin/apache-tomcat-
8.5.92.tar.gz
tar -xzvf apache-tomcat-8.5.92.tar.gz -C /opt
|
```

Step 43:-

Now you have to launch a private instance 2 so, gave a instance name and then select instance configuration and then edit network setting and select your VPC after that select private subnet 2. This is private instance so, don't gave it public IP. After that select security group in which MariaDB port Is add 330 port number, and in user data write a installation and configuration process of MariaDB service. And then launch instance, and now your three instance is launched.

▼ **Network settings** Info

VPC – *required* Info

```
vpc-0b737931ab7cc0112 (my-vpc)
192.168.0.0/24
```

Subnet Info

```
subnet-0f3a2eddb269f00cd              my-private-subnet-2
VPC: vpc-0b737931ab7cc0112    Owner: 132076981423
Availability Zone: eu-west-1c    IP addresses available: 26    CIDR: 192.168.0.64/27)
```

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
```

```
my_VPC-SG   sg-0d372d310264bb3ca  ✕
VPC: vpc-0b737931ab7cc0112
```

Compare security group rules

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

▶ **Advanced network configuration**

User data – *optional* Info
Upload a file with your user data or enter it in the field.

⤒ Choose file

```
#! /bin/bash

sudo -i

yum install mariadb105-server -y
systemctl start mariadb
systemctl enable mariadb
```

	Name		Instance ID	Instance state		Instance type		Status check	Alarm status		Availability Zone		Public IPv4 DNS		Public IP
☐	public-instance	▼	i-0e0524ec68f5361f0	⊘ Running	⊕⊖	t2.micro		⊘ 2/2 checks passed	No alarms	+	eu-west-1a		–		34.244.4
☐	private-instance-2		i-037379a99dbab41cb	⊘ Running	⊕⊖	t2.micro		⊘ Initializing	No alarms	+	eu-west-1c		–		–
☐	private-instace-1		i-030088c8392b6281a	⊘ Running	⊕⊖	t2.micro		⊘ 2/2 checks passed	No alarms	+	eu-west-1b		–		–

Step 44:-

Now get remotely access of public instance on any third party app. And then enter command sudo –I to switch to root user and then gave machine name "public".

```
[ec2-user@ip-192-168-0-10 ~]$ sudo -i
[root@ip-192-168-0-10 ~]#
[root@ip-192-168-0-10 ~]# rpm -q nginx
nginx-1.24.0-1.amzn2023.0.1.x86_64
[root@ip-192-168-0-10 ~]#
[root@ip-192-168-0-10 ~]# sudo -i
[root@ip-192-168-0-10 ~]# hostnamectl set-hostname public
[root@ip-192-168-0-10 ~]# bash
[root@public ~]#
```

Step 45:-

After that get student.war and JDBC connecter file in your public instance and your private instance key also put in pubic instance.

```
[root@public ~]# cd /home/ec2-user/
[root@public ec2-user]# ls
ireland-key.pem  mysql-connector.jar  student.war
[root@public ec2-user]#
```

Step 46:-

Next you have to send JDBC connector and student.war file to your private-insatnce-1 because we have not get private instance so we have to send to that instance by scp command.

scp -i ireland-key.pem mysql-connector.jar student.war ec2-user@192.168.0.42:/home/ec2-user

```
[root@public ec2-user]# ls
ireland-key.pem  mysql-connector.jar  student.war
[root@public ec2-user]# scp -i ireland-key.pem mysql-connector.jar student.war ec2-user@192.168.0.42:/home/ec2-user
The authenticity of host '192.168.0.42 (192.168.0.42)' can't be established.
ED25519 key fingerprint is SHA256:Atud9xk4OyggkoWKGyAz9KKstZd/lo12pDMKfUitw28.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.42' (ED25519) to the list of known hosts.
mysql-connector.jar                                    100%  984KB  81.6MB/s   00:00
student.war                                            100%   87KB  33.2MB/s   00:00
[root@public ec2-user]#
```

Step 47:-

Now get access of private instance 1 to configure tomcat and connect it to nginx server.

So first get access of private-1 by ssh command. Run command

ssh -i ireland-key.pem ec2-user@192.168.0.42 and then gave that private instance machine name
"private-1".

```
[root@public ec2-user]# ssh -i ireland-key.pem ec2-user@192.168.0.42
     ,       #_
    ~\_     ####_        Amazon Linux 2023
   ~~  \_#####\
   ~~     \###|
   ~~       \#/ ___       https://aws.amazon.com/linux/amazon-linux-2023
    ~~       V~' '->
     ~~~         /
       ~~._.   _/
          _/ _/
        _/m/'
[ec2-user@ip-192-168-0-42 ~]$ sudo -i
[root@ip-192-168-0-42 ~]# hostnamectl set-hostname private-1
[root@ip-192-168-0-42 ~]# bash
[root@private-1 ~]#
```

step 48:-

After that see your tomcat package is configured or not in /opt directory.so now we have to copy
myself-connecter.jar and student.war files which we have get from private instance to tomcat service's
directories. JDBC connecter send to /opt/apache-tomcat-8.5.92/lib directory and student.war
/opt/apache-tomcat-8.5.92/webapps directory.

```
[root@private-1 ec2-user]# ls
mysql-connector.jar  student.war
[root@private-1 ec2-user]# cp mysql-connector.jar /opt/apache-tomcat-8.5.92/lib/
[root@private-1 ec2-user]# cp student.war /opt/apache-tomcat-8.5.92/webapps/
[root@private-1 ec2-user]#
```

Step 49:-

Next change directory /opt/apache-tomcat-8.5.92/bin. There was a file ./catalina.sh that file is for start
and restart tomcat service but that file was not run without java so we have to install it by run
command #yum install java —y. Then run that catalina.sh and startup.sh file.

./catalina.sh start and ./satartup.sh

Now your backend server is ready. Now you have to connect backend server to frontend sever.

```
[root@private-1 bin]# yum install java -y
Last metadata expiration check: 1:06:57 ago on Sun Aug 20 11:58:07 2023.
Dependencies resolved.
================================================================================
 Package               Architecture    Version              Repository    Size
================================================================================
Installing:
 java-17-amazon-corretto  x86_64       1:17.0.8+7-1.amzn2023.1   amazonlinux   188 k
Installing dependencies:
 alsa-lib                 x86_64       1.2.7.2-1.amzn2023.0.2    amazonlinux   504 k
 cairo                    x86_64       1.17.4-3.amzn2023.0.2     amazonlinux   674 k
```

```
[root@private-1 bin]# ls
bootstrap.jar        ciphers.bat                    configtest.bat  digest.sh       shutdown.sh       tomcat-native.tar.gz  version.sh
catalina-tasks.xml  ciphers.sh                     configtest.sh   setclasspath.bat  startup.bat      tool-wrapper.bat
catalina.bat        commons-daemon-native.tar.gz   daemon.sh       setclasspath.sh   startup.sh       tool-wrapper.sh
catalina.sh         commons-daemon.jar             digest.bat      shutdown.bat      tomcat-juli.jar  version.bat
[root@private-1 bin]# ./catalina.sh start
Using CATALINA_BASE:   /opt/apache-tomcat-8.5.92
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.92
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.92/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-8.5.92/bin/bootstrap.jar:/opt/apache-tomcat-8.5.92/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@private-1 bin]# ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat-8.5.92
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.92
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.92/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-8.5.92/bin/bootstrap.jar:/opt/apache-tomcat-8.5.92/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@private-1 bin]#
```

Step 50:-

So now go back to public server where nginx was installed. And then you have add some data in nginx configuration file. So run command vim /etc/nginx/nginx.conf

In that file you have to write some data on line number 47 48 and 49 (as per show in following image).

That data means you have to configure your tomcat servers location.

location / {

proxy_pass http:// tomcat servers IP:8080/student/;

}

After write this data save this file. And then restart nginx service by run command #systemctl restart nginx.

And your frontend and backend is connected and whenever you hit public ip then you see the backends page. See following image.

```
[root@public ec2-user]# vim /etc/nginx/nginx.conf
```

```
37      server {
38          listen       80;
39          listen       [::]:80;
40          server_name  _;
41          root         /usr/share/nginx/html;
42
43          # Load configuration files for the default server block.
44          include /etc/nginx/default.d/*.conf;
45
46          error_page 404 /404.html;
47          location / {
48          proxy_pass http://192.168.0.42:8080/student/;
49          }
50          location = /404.html {
51          }
52
53          error_page 500 502 503 504 /50x.html;
54          location = /50x.html {
55          }
56      }
57
58 # Settings for a TLS enabled server.
59 #
60 #      server {
:wq
```

```
[root@public ec2-user]# systemctl restart nginx
[root@public ec2-user]#
```

← → C ⚠ Not secure | 34.244.46.79

▶ YouTube ○ GitHub ⊙ Student/Parent Portal 🔶 AWS Management... ⊙ ChatGPT 📷 Photos - Google Ph... ⊗ Visual Subnet Calcu... ⊗ AWS Notes.pdf 📦 Installing the Cloud...
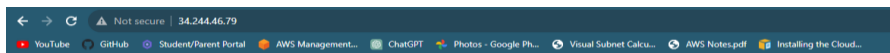
Student Registration Form

Student Name   [_____]
Student Address   [_____]
Student Age   [_____]
Student Qualification [_____]
Student Percentage   [_____]
Year Passed   [_____]
[register]

Step 51:-

To connect database to your backend firstly you have to add table to your database so we have to get access of database so go to RDS service and select your database and then click on actions and then click on set up EC2 instance.

Databases (1)   ○ Group resources ⟳ Modify Actions ▲ Restore from S3 **Create database**

🔍 Filter by databases   Quick Actions - New   ‹ 1 › ⚙

                                                               Convert to Multi-AZ deployment

⊞ **DB identifier** ▲ Status ▽ Role ▽ Engine ▽ Region & AZ ▽ Size ▽ Actions ▽ CPU ▽ Curr Stop temporarily   c   ▽

● my-mariad-database ⊘ Available Instance MariaDB eu-west-1a db.m6gd.large -   ▭ 1.77% ▭ Reboot -0b737931ab7cc0112

                                                                              Delete   ▶

                                                               Set up EC2 connection

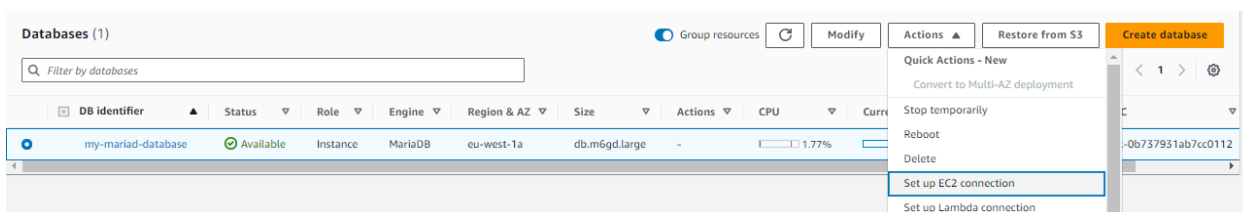   Set up Lambda connection

Step 52:-

Now select instance private-2 to your database was connected. Because whenever get access of database to get from this instance. And then click on continue . next you have see all configuration see it and click on set up.

Set up EC2 connection Info

Select EC2 instance

Database
my-mariad-database

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-037379a99dbab41cb   ▲ │   │ ⟳   │
│ private-instance-2   eu-west-1c   │   └──────┘
├───┤
│ 🔍 | │
├───┤
│ i-030088c8392b6281a   │
│ private-instace-1   eu-west-1b   │
├───┤
│ i-0e0524ec68f5361f0   │
│ public-instance   eu-west-1a   │   Cancel   **Continue**
├───┤
│ i-037379a99dbab41cb   │
│ private-instance-2   eu-west-1c   ✓ │
└───┘

Step 53:-

Now you have to add a database to backend. Get access of private-instance-2 instance where MariaDB is configured.

Run a command ssh –I private instance key and then user@private-2 instance IP.

And then gave machine name to private instance as name private-2.


```
[root@public ec2-user]# ssh -i ireland-key.pem ec2-user@192.168.0.89
The authenticity of host '192.168.0.89 (192.168.0.89)' can't be established.
ED25519 key fingerprint is SHA256:H+WjNlE32EG5yvZCrI2Lgx8T1lMI/U4qWPzPFCOArKo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.89' (ED25519) to the list of known hosts.
        #_
   ~\_  ####_          Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

```
[ec2-user@ip-192-168-0-89 ~]$ sudo -i
[root@ip-192-168-0-89 ~]# hostnamectl set-hostname private-2
[root@ip-192-168-0-89 ~]# bash
[root@private-2 ~]#
```

Step 54:-

Now get access of your database and create a table in it so your data is save it that table correctly. And that table save in database securely . so run command

#mysql –u username –h end point of RDS –p password

And you have get access of your RDS database.

```
[root@private-2 ~]# mysql -u ani -h my-mariad-database.cpt01kzfvpqi.eu-west-1.rds.amazonaws.com -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 219
Server version: 10.6.14-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.
```

Step 55:-

Now create table by run command in MariaDB #create table studentapp; and then get on that database by command #use studentapp;

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

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

Step 56:-

Now create table and configure your table what you want in it.

Run command #

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));

Then you have to see your table then run command

desc studenets;

Now your table and database is ready. Now connect RDS database to your tomcat sever.

```
MariaDB [studentapp]> create table students (student_id INT NOT NULL AUTO_INCREMENT,student_name VARCHAR(100) NOT NULL,student_addr VARCH
AR(100) NOT NULL,student_age VARCHAR(3) NOT NULL,student_qual VARCHAR(20) NOT NULL,student_percent VARCHAR(10) NOT NULL,student_year_pass
ed VARCHAR(10) NOT NULL,PRIMARY KEY (student_id));
Query OK, 0 rows affected (0.014 sec)

MariaDB [studentapp]> desc student;
ERROR 1146 (42S02): Table 'studentapp.student' doesn't exist
MariaDB [studentapp]> desc students;
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| student_id          | int(11)      | 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 (0.004 sec)
```

Step 57:-

So you have to connect database to tomcat sever so get access of private instance 1 .

And change directory to /opt/apache-tomcat-8.5.92/conf

In that directory you see the context.xml file you have to write end point of RDS database in it to connect your RDS data.

Step 58:-

So run command

#vim context.xml

After that go on line number 20 and paste data there.

<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"/>

In that data you have make some changes on line no 22 write a username and then password.

After that change on line number 23 paste your RDS end point and at the end gave a name of your database.

Now save that file.

```
17  -->
18  <!-- The contents of this file will be loaded for each web application -->
19  <Context>
20  <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
21            maxTotal="100" maxIdle="30" maxWaitMillis="10000"
22            username="ani" password="anilagad12345" driverClassName="com.mysql.jdbc.Driver"
23            url="jdbc:mysql://my-mariad-database.cpt01kzfvpqi.eu-west-1.rds.amazonaws.com:3306/studentapp"/>
24      <!-- Default set of monitored resources, if one of these changes, the   -->
25      <!-- web application will be reloaded.                                   -->
26      <WatchedResource>WEB-INF/web.xml</WatchedResource>
27      <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
28
29      <!-- Uncomment this to disable session persistence across Tomcat restarts -->
30      <!--
```

Step 59:-

 Now run command cd .. to go backward.

And then change directory cd bin.

There where you will see catalina.sh file and startup.sh file.

Run that scripted files ./catalina.sh start and then ./startup.sh .

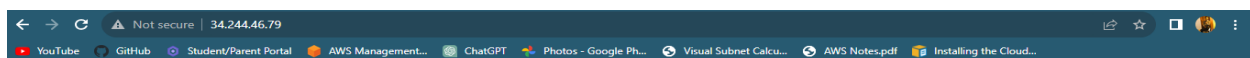And your backend has connected to database.

```
[root@private-1 conf]# cd ..
[root@private-1 apache-tomcat-8.5.92]# ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
[root@private-1 apache-tomcat-8.5.92]# cd  bin/
[root@private-1 bin]# ls
bootstrap.jar       ciphers.bat              configtest.bat  digest.sh       shutdown.sh      tomcat-native.tar.gz  version.sh
catalina-tasks.xml  ciphers.sh               configtest.sh   setclasspath.bat  startup.bat    tool-wrapper.bat
catalina.bat        commons-daemon-native.tar.gz  daemon.sh  setclasspath.sh   startup.sh     tool-wrapper.sh
catalina.sh         commons-daemon.jar       digest.bat      shutdown.bat      tomcat-juli.jar  version.bat
[root@private-1 bin]# ./catalina.sh start
Using CATALINA_BASE:   /opt/apache-tomcat-8.5.92
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.92
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.92/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-8.5.92/bin/bootstrap.jar:/opt/apache-tomcat-8.5.92/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@private-1 bin]# ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat-8.5.92
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.92
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.92/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-8.5.92/bin/bootstrap.jar:/opt/apache-tomcat-8.5.92/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Step 60:-

Now hit public IP and then fill there column. And click on register. And your data was saved.

← → C   ⚠ Not secure | 34.244.46.79   ⤢ ☆ ☐ 🦊 :
▶ YouTube 🔘 GitHub ⊙ Student/Parent Portal ⬛ AWS Management... 🔘 ChatGPT ✥ Photos - Google Ph... 🔵 Visual Subnet Calcu... 🔵 AWS Notes.pdf 🎁 Installing the Cloud...

Student Registration Form

Student Name	aniket
Student Address	at.post telgaon ,beed
Student Age	24
Student Qualification	BBA
Student Percentage	84
Year Passed	2022
register	

Register Student

Students List

Student ID	StudentName	Student Addrs	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	aniket	at.post telgaon ,beed	24	BBA	84	2022	edit	delete