

Task : Secure 3-Tier AWS Deployment : Angular Frontend, Java Backend, RDS

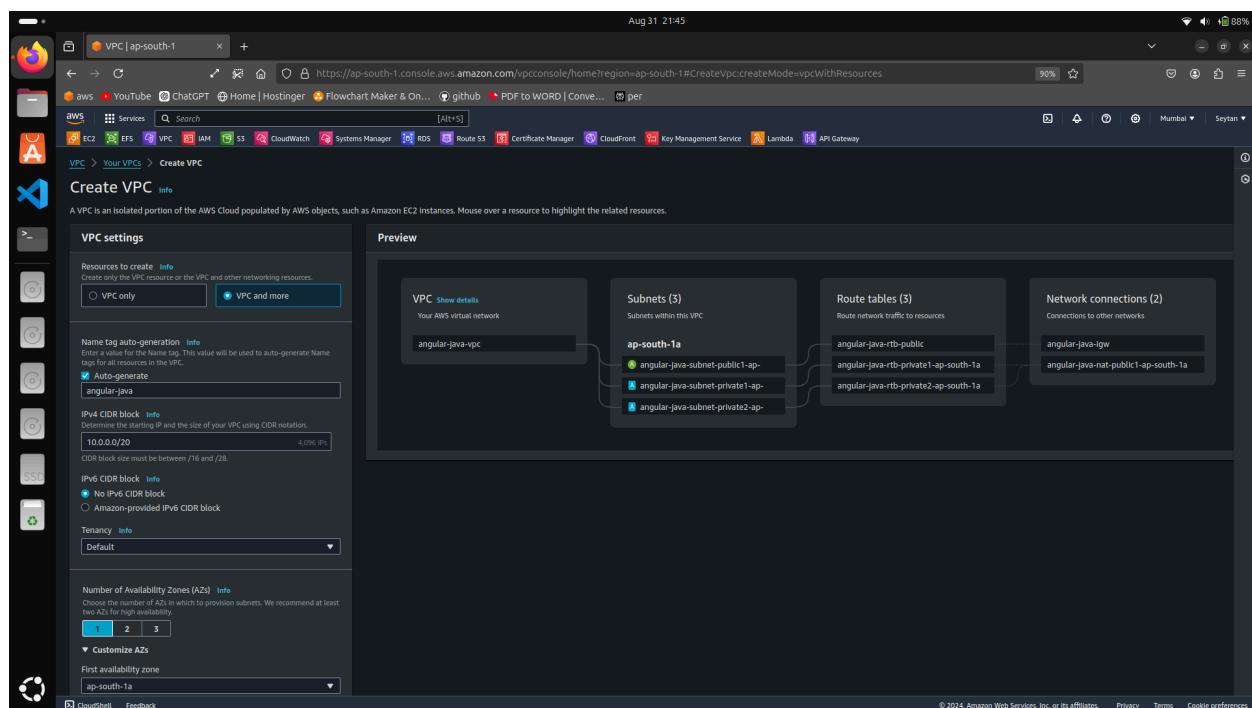
Diagrammatic Representation:

https://drive.google.com/file/d/1w9YR_SkHr1GC3JmxukGFPqMwSeWVHbR8/view?usp=sharing

Step 1: Create a VPC with 3 subnets (1 public and 2 private+ add nat to private subnets).

What is vpc ,subnet,nat gateway? →

1. A VPC (Virtual Private Cloud) is an isolated network within AWS where you can launch resources.
2. subnet is a segment of a VPC's IP address range where resources are placed.
3. A NAT Gateway allows instances in a private subnet to access the internet while keeping them secure.



Step 2: Now Launch 3 instances (frontend server, backend server, database server).

For the frontend server chose the Networking settings and edit it. Select the our newly created vpc and select the public subnet also make sure to enable the auto assing ip for the fronted server.

EC2 > Instances > Launch an instance

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
frontend

Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macos Ubuntu Windows Red Hat SUSE L

aws Mac ubuntu Microsoft RedHat SUSE

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0522ab6e1ddcc7055 (64-bit (x86)) / ami-0000791bad666add5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM) SSD General Purpose (SSD) Volume Type. Currently available from Canonical (Ubuntu 24.04 LTS)

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required
seytan_cloud

Create new key pair

▼ Network settings Info

VPC - required Info

vpc-0345beba67027169b (angular-java-vpc)
10.0.0.0/24

Create new subnet

Subnet Info

subnet-06353c7664aa5c7b9 angular-java-subnet-public1-ap-south-1a
VPC: vpc-0345beba67027169b Owner: 678204272547
Availability Zone: ap-south-1a Zone type: Availability Zone
IP addresses available: 250 CIDR: 10.0.0.0/24

Create new subnet

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

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

Create security group Select existing security group

Common security groups Info

Select security groups

default sg-0c497afdc217c2929 X
VPC: vpc-0345beba67027169b

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

Now create the Backend Server and select the appropriate vpc and private subnet 1 and disable the auto assing publi ip from network settings.

EC2 > Instances > Launch an instance

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: backend Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images

Recents Quick Start

- Amazon Linux
- macOS
- Ubuntu**
- Windows
- Red Hat
- SUSE L

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-052ab6e1ddc7055 (64-bit (x86)) / ami-0000791bad666add5 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type. Free tier eligible for 12 months after launch. [View details](#)

Key pair name - required
seytan_cloud Create new key pair

Network settings Info

VPC - required Info
vpc-0345beba67027169b (angular-java-vpc)
10.0.0.0/20 Create new subnet

Subnet Info
subnet-0d27ca14500fb0d07b angular-java-subnet-private1-ap-south-1a
VPC: vpc-0345beba67027169b Owner: 678204272547 Availability Zone: ap-south-1a Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.8.0/24

Auto-assign public IP Info
Disable Select existing security group

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 Compare security group rules

default sg-0c497afdc217c2929 X
VPC: vpc-0345beba67027169b

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

Advanced network configuration

Configure storage Info Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Now create the final database server, and perform the same instructions on this which were applied on backend while creating.

EC2 > Instances > Launch an instance

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
database Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE L Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-052ab6e1ddcc7055 (64-bit (x86)) / ami-0000791bad666add5 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM) EBS General Purpose (SSD) Volume Type - Free tier available for General Purpose (SSD) volumes.

Key pair name - required
seytan_cloud Create new key pair

Network settings Info

VPC - required Info
vpc-0345beba67027169b (angular-java-vpc)
10.0.0.0/20 Create new subnet

Subnet Info
subnet-05082caf72527cccd angular-java-subnet-private2-ap-south-1a
VPC: vpc-0345beba67027169b Owner: 678204272547 Availability Zone: ap-south-1a Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.9.0/24

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 Compare security group rules

Common security groups Info
Select security groups
default sg-0c497afdc217c2929 X
VPC: vpc-0345beba67027169b

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

Advanced network configuration

Configure storage Info Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Step 3: Whitelist the following ports in the security group of servers. (22,443,80,8080,3306)

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e3256988aa0ef2fb	All traffic	All	All	Custom	sg-0c497afdc217c2929
-	Custom TCP	TCP	80	Anywhere-IPv4	0.0.0.0/0
-	Custom TCP	TCP	8080	Anywhere-IPv4	0.0.0.0/0
-	Custom TCP	TCP	443	Anywhere-IPv4	0.0.0.0/0
-	Custom TCP	TCP	22	Anywhere-IPv4	0.0.0.0/0
-	Custom TCP	TCP	3306	Anywhere-IPv4	0.0.0.0/0

Step 4: Now as we are gonna connect to the private servers through bastion server i.e. from public servers ,we need the private key pair so we will transfer this key from our local machine to the frontend server using sftp .

What is sftp ? →

SFTP (Secure File Transfer Protocol) is a secure version of FTP (File Transfer Protocol) that uses SSH (Secure Shell) to encrypt data during transfer, ensuring secure file exchange over a network.

```
seytan@seytan-Inspiron-3501: $ sftp -i seyan.cloud.pem ubuntu@13.232.35.94
The authenticity of host '13.232.35.94' (13.232.35.94) can't be established.
ED25519 key fingerprint is SHA256:Dmnp0HDLtH65cVrCM6U/FS0qHrw3x0lUd3ECTc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.232.35.94' (ED25519) to the list of known hosts.
Connected to 13.232.35.94.
sftp> put /home/seytan/seyan.cloud.pem /home/ubuntu/seyan.cloud.pem
Uploading /home/seytan/seyan.cloud.pem to /home/ubuntu/seyan.cloud.pem
seytan.cloud.pem
sftp> _
```

Step 5: Now Connect to bastion server or jump server (frontend server) and connect to the backend and database server from it.

What is Bastion or jump servers ? →

Public servers that provide secure access to a private network are called bastion servers or jump servers.

Use the sudo hostnamectl set-hostname command to change the server names accordingly.

```
ubuntu@ip-10-0-0-9: ~
seytan@seytan-Inspiron-3501:~$ ssh -i seytan_cloud.pem ubuntu@13.232.3
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Sat Aug 31 16:22:06 UTC 2024

      System load:  0.0          Processes:           117
      Usage of /:   22.8% of 6.71GB  Users logged in:    0
      Memory usage: 5%            IPv4 address for enX0: 10.0.0.9
      Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-0-9:~$ sudo hostnamectl set-hostname frontend
```

```
ubuntu@frontend:~$ ssh -i seytan_cloud.pem ubuntu@10.0.8.252
The authenticity of host '10.0.8.252 (10.0.8.252)' can't be established.
ED25519 key fingerprint is SHA256:WwqdiuSq3GiftfaD/QC+SzUpEuNFMVSobIcWng/R9/
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.8.252' (ED25519) to the list of known hosts
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Aug 31 16:26:58 UTC 2024

System load: 0.02          Processes: 105
Usage of /: 22.8% of 6.71GB Users logged in: 0
Memory usage: 20%          IPv4 address for enX0: 10.0.8.252
Swap usage: 0%             

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-8-252:~$ sudo hostnamectl set-hostname backend_
```

```
ubuntu@frontend:~$ ssh -i seytan_cloud.pem ubuntu@10.0.9.178
The authenticity of host '10.0.9.178 (10.0.9.178)' can't be established.
ED25519 key fingerprint is SHA256:DlTp9HKfXZIu00GHkQxZqUChXg95Jxv+T0x0JFf9D
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.9.178' (ED25519) to the list of known hosts
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Aug 31 16:27:51 UTC 2024

System load: 0.08          Processes: 104
Usage of /: 22.8% of 6.71GB Users logged in: 0
Memory usage: 20%          IPv4 address for enX0: 10.0.9.178
Swap usage: 0%             

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

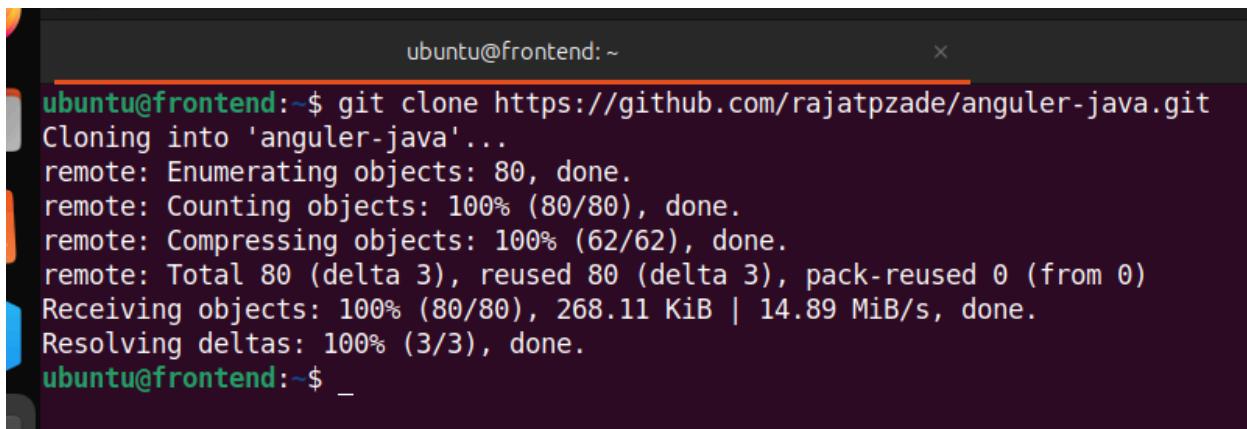
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

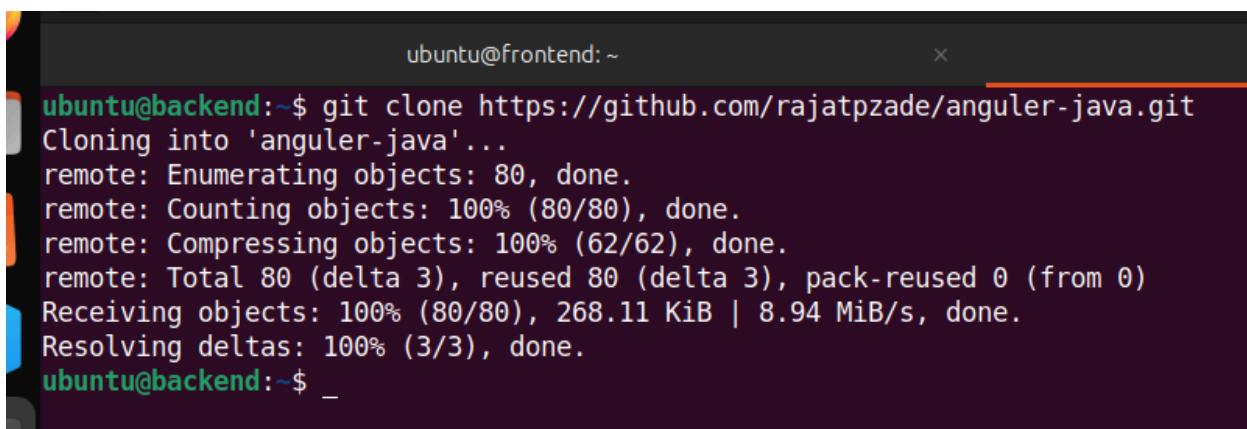
ubuntu@ip-10-0-9-178:~$ sudo hostnamectl set-hostname database
ubuntu@ip-10-0-9-178:~$ 
```

Backend and Database

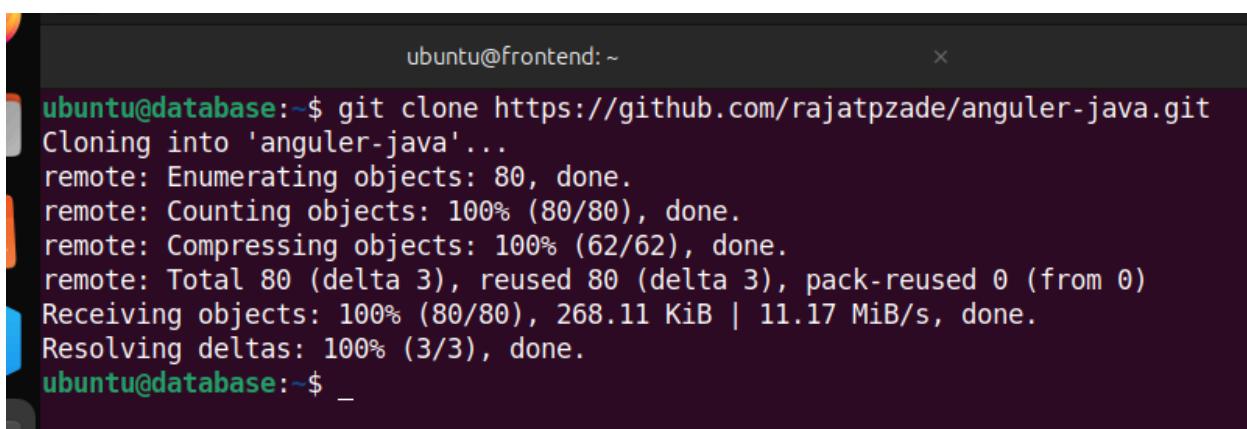
**Step 6: Now clone the github repository in all 3 servers.
Using the command
git clone <https://github.com/rajatpzade/angular-java.git>**



```
ubuntu@frontend:~$ git clone https://github.com/rajatpzade/angular-java.git
Cloning into 'angular-java'...
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 80 (delta 3), reused 80 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (80/80), 268.11 KiB | 14.89 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@frontend:~$ _
```



```
ubuntu@backend:~$ git clone https://github.com/rajatpzade/angular-java.git
Cloning into 'angular-java'...
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 80 (delta 3), reused 80 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (80/80), 268.11 KiB | 8.94 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@backend:~$ _
```



```
ubuntu@database:~$ git clone https://github.com/rajatpzade/angular-java.git
Cloning into 'angular-java'...
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 80 (delta 3), reused 80 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (80/80), 268.11 KiB | 11.17 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@database:~$ _
```

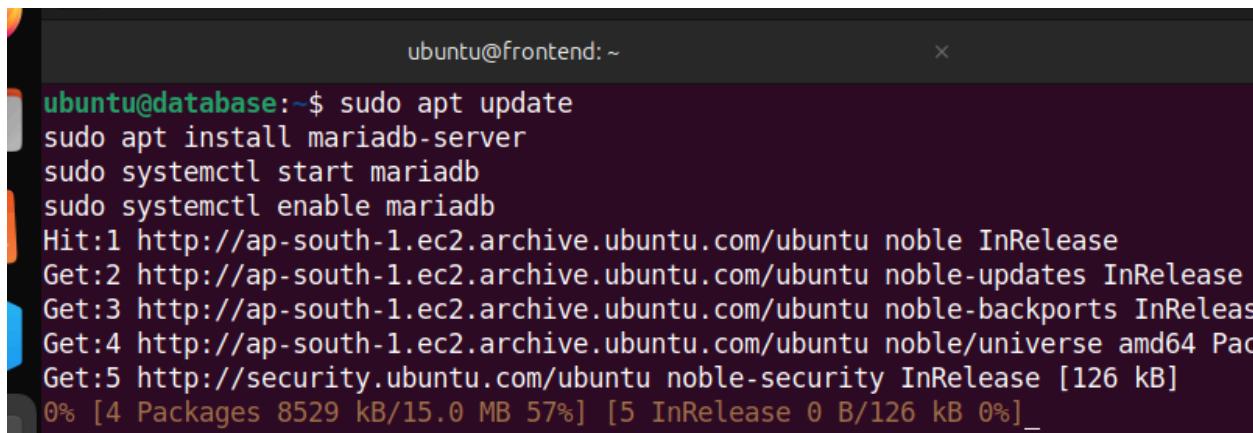
Step 7: Now go to the database server and run following commands →

sudo apt update

sudo apt install mariadb-server

sudo systemctl start mariadb-server

sudo systemctl enable mariadb-server



A screenshot of a terminal window titled "ubuntu@frontend: ~". The window contains the following command-line session:

```
ubuntu@database:~$ sudo apt update
sudo apt install mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
0% [4 Packages 8529 kB/15.0 MB 57%] [5 InRelease 0 B/126 kB 0%]
```

Step 8: Now Create a RDS Production Database named angular-java and choose mariadb as engine.

What is RDS ? →

RDS (Relational Database Service) is a managed database service provided by AWS that simplifies the setup, operation, and scaling of a relational database in the cloud. It supports multiple database engines, including MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB.

RDS > Create database

Create database

Choose a database creation method [Info](#)

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible)

Aurora (PostgreSQL Compatible)

MySQL

MariaDB

PostgreSQL

Oracle

Microsoft SQL Server

IBM Db2

Now in connectivity configuration select the connect to and EC2 compute resource option and select the database instance in it.

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.

Connectivity Info C

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.

EC2 instance Info C

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

Choose an EC2 instance ▲

Q

i-0f99fb92bb5416ed1
frontend

i-04a39311a6c68d074
backend

i-083cde47089d738df
database

Default VPC (vpc-09066077d7529c57f) ▼
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

i After a database is created, you can't change its VPC.

DB subnet group Info C

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.

Choose existing
Choose existing DB subnet group

Automatic setup
RDS creates a new subnet group for you or reuses an existing subnet group

Step 9: Now connect to the RDS database from database server using

sudo mysql -h rds-endpoint -u admin -p

Once connected execute the following commands:

CREATE DATABASE springbackend;

**GRANT ALL PRIVILEGES ON springbackend.* TO
'admin'@'backend-subnet_cidr' IDENTIFIED BY
'database_password'**

Once it is done do exit and run the next command from database server shell :

cd angular-java/

sudo mysql -h rds-endpoint -u admin -p springbackend < springbackend.sql

```
ubuntu@database: $ sudo mysql -h angular-java.c7kqi8k66m89.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 27
Server version: 10.11.8-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)]> CREATE DATABASE springbackend;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON springbackend.* TO 'admin'@'10.0.8.252' IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> exit
Bye
ubuntu@database: $ cd angular-java/
ubuntu@database:~/angular-java$ sudo mysql -h angular-java.c7kqi8k66m89.ap-south-1.rds.amazonaws.com -u admin -p springbackend < springbackend.sql
ubuntu@database:~/angular-java$ _
```

Edit the inbound rules for the rds security group and add the backend cidr for port 3306 allowing backend to make connection with the rds database .

The screenshot shows the AWS Management Console interface for managing security groups. The user is navigating through the EC2 > Security Groups > sg-0b6a681cf82ff992 - rds-ec2-1 > Edit inbound rules. The 'Inbound rules' table lists two rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-043111983482e6d82	MySQL/Aurora	TCP	3306	Custom	Rule to allow connections from EC2 instance sg-0b6a681cf82ff992 attached
-	Custom TCP	TCP	3306	Custom	Rule to allow connections from EC2 instance sg-08972a18951c2a920 attached

Step 10: Now go to the backend server and run the following commands →

sudo apt update && sudo apt install openjdk-8-jdk && sudo apt install maven -y

```
ubuntu@backend:~$ cd angular-java/spring-backend/
ubuntu@backend:~/angular-java/spring-backend$ sudo apt update && sudo apt install openjdk-8-jdk && sudo apt install maven -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
0% [4 Packages 6747 kB/15.0 MB 45%] [5 InRelease 14.2 kB/126 kB 11%]
```

Step 11 : Now go and edit the file application.properties present at the location

→

angular-java/springabackend/src/main/resources/application.properties

Add the rds endpoint in place localhost and username as rds username and password as rds password.

```
ubuntu@backend:~/angular-java/spring-backend$ sudo nano src/main/resources/application.properties
ubuntu@backend:~/angular-java/spring-backend$ _
```

```
GNU nano 7.2                                     src/main/resources/application.properties *
spring.datasource.url=jdbc:mysql://angular-java.c7kqi8k66m89.ap-south-1.rds.amazonaws.com:3306/springbackend?useSSL=false
spring.datasource.username=admin
spring.datasource.password=12345678_
spring.jpa.generate-ddl=true
```

Now build the backend using following command

→

mvn clean package -Dmaven.test.skip=true

```
ubuntu@backend:~/angular-java/spring-backend$ mvn clean package -Dmaven.test.skip=true
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.7.4/spring-boot-starter-parent-2.7.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-parent/2.7.4/spring-boot-starter-parent-2.7.4.pom (4.1 kB at 100 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.7.4/spring-boot-dependencies-2.7.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-dependencies/2.7.4/spring-boot-dependencies-2.7.4.pom (10 kB at 100 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/databricks/oss/java-driver-bom/4.14.1/java-driver-bom-4.14.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/databricks/oss/java-driver-bom/4.14.1/java-driver-bom-4.14.1.pom (4.1 kB at 100 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-bom/4.2.12/metrics-bom-4.2.12.pom
Downloaded from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-bom/4.2.12/metrics-bom-4.2.12.pom (6.9 kB at 286 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-parent/4.2.12/metrics-parent-4.2.12.pom
Downloaded from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-parent/4.2.12/metrics-parent-4.2.12.pom (20 kB at 100 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/groovy/groovy-bom/3.0.13/groovy-bom-3.0.13.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/groovy/groovy-bom/3.0.13/groovy-bom-3.0.13.pom (26 kB at 979 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-bom/13.0.11.Final/infinispan-bom-13.0.11.Final.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-bom/13.0.11.Final/infinispan-bom-13.0.11.Final.pom (10 kB at 100 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-build-configuration-parent/13.0.11.Final/infinispan-build-configuration-parent-13.0.11.Final.pom
```

Step 12 : Now go to the frontend server and navigate to the angular-java/angular-frontend directory and run the following commands.

sudo apt update

sudo apt install nodejs npm

```
ubuntu@frontend:~$ cd angular-java/angular-frontend/
ubuntu@frontend:~/angular-java/angular-frontend$ sudo apt update
sudo apt install nodejs npm
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [469 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [117 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [7768 B]
```

Now run the next commands :

sudo npm install -g @angular/cli@14.2.1

npm install

```
ubuntu@frontend:~/angular-java/angular-frontend$ sudo npm install -g @angular/cli@14.2.1
npm WARN deprecated npmlog@6.0.2: This package is no longer supported.
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use
a key value, which is much more comprehensive and powerful.
npm WARN deprecated @npmcli/move-file@0.1.1: This functionality has been moved to @npmcli/fs
npm WARN deprecated read-package-json@5.0.2: This package is no longer supported. Please use @
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated are-we-there-yet@3.0.1: This package is no longer supported.
npm WARN deprecated glob@8.1.0: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated gauge@4.0.4: This package is no longer supported.

added 212 packages in 11s

27 packages are looking for funding
  run `npm fund` for details
ubuntu@frontend:~/angular-java/angular-frontend$ npm install

added 918 packages, and audited 919 packages in 29s

124 packages are looking for funding
  run `npm fund` for details

23 vulnerabilities (6 moderate, 13 high, 4 critical)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
ubuntu@frontend:~/angular-java/angular-frontend$ _
```

Step 13: Now as we need to provide the ip of the backend server in the worker.service.ts file . But as the backend is running on private server/subnet it is not accessible over internet so we have to create a Network load balancer for the backend as well as frontend to access the frontend over the browser using nginx server.

First Create a target group for the backend and use tcp 8080 .

Facilitates routing to a single Lambda function.
Accessible to Application Load Balancers only.

Application Load Balancer
Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name
backend
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

TCP	8080
1-65535	

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

angular-java-vpc
vpc-0345beba67027169b
IPv4 VPC CIDR: 10.0.0.0/20

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Available instances (3)

Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID
I-083cde47089d759df	database	Running	default, ec2-rds-1	ap-south-1a	10.0.9.178	subnet-05082c
I-0439511a6c68d074	backend	Running	default	ap-south-1a	10.0.8.252	subnet-0127ca
I-0f9fb92bb5416ed1	frontend	Running	default	ap-south-1a	10.0.0.9	subnet-06353c

0 selected
Ports for the selected instances
Ports for routing traffic to the selected instances.

8080
1-65535 (separate multiple ports with commas)

Include as pending below

1 selection is now pending below. Include more or register targets when ready.

Review targets

Targets (1)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
I-0439511a6c68d074	backend	8080	Running	default	ap-south-1a	10.0.8.252	subnet-0d27ca14500fb07b	August 31, 2024, 21:48 (UTC+05:30)

The screenshot shows the 'Create Network Load Balancer' wizard. The first step, 'Basic configuration', is selected. It includes fields for 'Load balancer name' (set to 'backend'), 'Scheme' (set to 'Internet-facing'), 'Load balancer IP address type' (set to 'IPv4'), and 'Network mapping'. A note at the bottom states: 'The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.'

Add the protocol **tls** and port **443**, you have to add the certificate also .

The screenshot shows the 'Listener TLS:443' configuration. It includes fields for 'Protocol' (set to 'TLS'), 'Port' (set to '443'), 'Default action' (set to 'Forward to backend'), 'Security policy' (set to 'All security policies'), 'Policy name' (set to 'ELBSecurityPolicy-TLS13-1-2-2021-06 (recommended)'), and 'Default SSL/TLS server certificate' (set to 'seytan.in').

Now create a network load balancer for the frontend also.

Create a target group for the frontend add tcp and port 80.

Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.

TCP	80	1-65535
-----	----	---------

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

angular-java-vpc vpc-0345beb67027169b IPv4 VPC CIDR: 10.0.0.0/20
--

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

Health check path
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Available instances (3)

<input type="checkbox"/>	Instance ID	Name	State	Security groups	Zone
<input type="checkbox"/>	i-085cde47089d738df	database	Running	default, ec2-rds-1	ap-south-1a
<input type="checkbox"/>	i-04a39311a6c68d074	backend	Running	default	ap-south-1a
<input type="checkbox"/>	i-0f9fb92bb5416ed1	frontend	Running	default	ap-south-1a

Ports for the selected instances
Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below

1 selection is now pending below. Include more or register targets when ready.

Review targets

Targets (1)

<input type="checkbox"/>	Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID
<input type="checkbox"/>	i-0f9fb92bb5416ed1	frontend	80	Running	default	ap-south-1a	10.0.0.9	subnet-06353

1 pending

Now create load balancer for the frontend and add tls 443 and attach a ssl certificate from ACM.

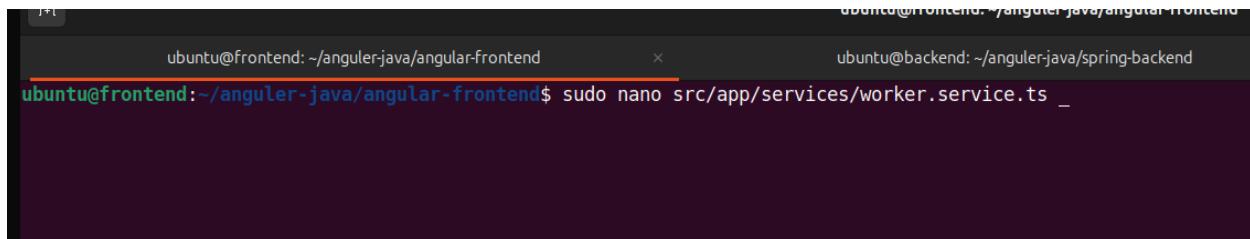
The screenshot shows the 'Create Network Load Balancer' wizard in the AWS Management Console. The 'Basic configuration' step is selected. Key fields include:

- Load balancer name:** frontend
- Scheme:** Internet-facing (selected)
- Load balancer IP address type:** IPv4 (selected)

The 'Network mapping' step is also visible, indicating the load balancer will exist and scale within the selected VPC.

The screenshot shows the 'Listener TLS:443' configuration page. The 'Protocol' is set to TLS and the 'Port' is 443. The 'Default action' is 'Forward to' target group 'frontend'. Under 'Secure listener settings', the 'Security policy' is set to 'All security policies' and the 'Policy name' is 'ELBSecurityPolicy-TLS13-1-2-2021-06 (recommended)'. The 'Default SSL/TLS server certificate' is set to 'From ACM' with the certificate 'seytan.in' selected. A note indicates the certificate is used if SNI protocol is not supported or no matching certificates are found.

Step 14: Now go to the worker.service.ts file and edit it and add the backend nlb dns in the place of localhost.



```
ubuntu@frontend:~/angular-java/angular-frontend
ubuntu@frontend:~/angular-java/angular-frontend$ sudo nano src/app/services/worker.service.ts
```

```
    providedIn: 'root'
})
export class WorkerService {
  private getUrl: string = "https://backend-85d635c285bb3274.elb.ap-south-1.amazonaws.com/api/v1/workers";
  constructor(private _httpClient: HttpClient) { }
  getWorkers(): Observable<Worker[]> {
    return this._httpClient.get<Worker[]>(this.getUrl).pipe(
```

Step 15: Now as we are using 2 different domain this will cause the cors issue .

What is CORS ? →CORS (Cross-Origin Resource Sharing) is a security feature implemented in web browsers that allows or restricts web pages from making requests to a domain different from the one that served the web page. It is used to enable secure cross-origin requests and data sharing between different domains

To solve this issue we have add the following in the backend application.properties file.

```
# CORS Configuration
spring.web.cors.allowed-origins=*
spring.web.cors.allowed-methods=GET,POST,PUT,DELETE
spring.web.cors.allowed-headers=*
```

Once it is done make sure add the dns names in the @CrossOrigin = dns-names in the controller file.

Note : Each time when we made any changes in the backend files make sure to run the following command .

mvn clean package -Dmaven.test.skip=true

Step 16: Run the following command to build the frontend .

ng build --configuration production

```
ubuntu@frontend:~/angular-java/angular-frontend$ ng build --configuration production
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files           | Names          | Raw Size | Estimated Transfer Size
main.03002de15067b11b.js     | main          | 237.78 kB | 61.81 kB
polyfills.b525ededa71d3b7f.js | polyfills    | 33.08 kB | 10.63 kB
runtime.e41le20b75d2e1de.js  | runtime      | 1.06 kB  | 607 bytes
styles.ef46db3751d8e999.css  | styles        | 0 bytes   | -
                                         | Initial Total | 271.93 kB | 73.03 kB

Build at: 2024-08-31T17:23:53.631Z - Hash: 0eae4a10deab2325 - Time: 9797ms
ubuntu@frontend:~/angular-java/angular-frontend$ _
```

This command will create dist directory inside of it there will be a angular-frontend directory which will contain the static files of our frontend.

ng build --configuration production → is an Angular CLI command that compiles the application in production mode, optimizing the build by enabling features like Ahead-of-Time (AOT) compilation, minification, and tree-shaking for better performance and smaller bundle sizes.

Step 17 : Install the nginx server on frontend ,

```
ubuntu@frontend:~/angular-java/angular-frontend$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
```

**and remove the default files in the home directory of nginx
and copy the static files in the dist/angular-frontend to the
nginx home directory.**

```
Build at: 2024-08-31T17:23:53.631Z - Hash: 0ae4a10deab2325 - Time: 9797ms
ubuntu@frontend:~/angular-java/angular-frontend$ sudo rm /var/www/html/*
ubuntu@frontend:~/angular-java/angular-frontend$ sudo cp dist/angular-frontend/* /var/www/html/
ubuntu@frontend:~/angular-java/angular-frontend$ _
```

Step 18 : Now run the next command in backend server

java -jar target/springbackend-v1.jar&

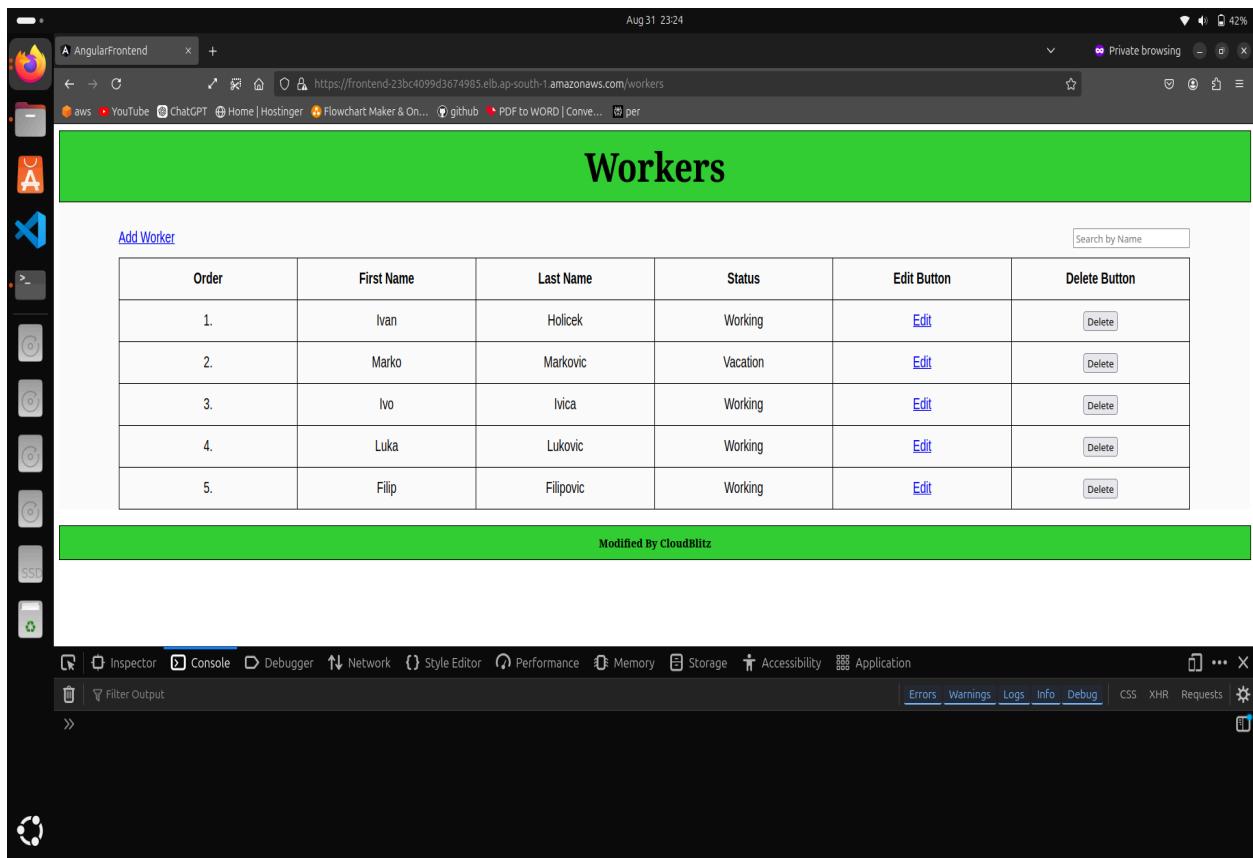
```
ubuntu@backend:~/angular-java/spring-backend$ java -jar target/spring-backend-v1.jar&
[1] 4821
ubuntu@backend:~/angular-java/spring-backend$ 

   .\n   \\\n   (( ))\n   \| |\n   ==|==|\n   :: Spring Boot ::\n           (v2.7.4)

2024-08-31 17:24:37.677 INFO 4821 --- [           main] c.e.s.SpringBackendApplication        : Starting SpringBackendApplication v1 using Java 1.8.0\n(/home/ubuntu/angular-java/spring-backend/target/spring-backend-v1.jar started by ubuntu in /home/ubuntu/angular-java/spring-backend)\n2024-08-31 17:24:37.685 INFO 4821 --- [           main] c.e.s.SpringBackendApplication        : No active profile set, falling back to 1 default profi\n2024-08-31 17:24:39.583 INFO 4821 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT r\n2024-08-31 17:24:39.688 INFO 4821 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 86 ms. Fou\n\n.\n2024-08-31 17:24:41.315 INFO 4821 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)\n2024-08-31 17:24:41.356 INFO 4821 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]\n2024-08-31 17:24:41.357 INFO 4821 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.65]\n2024-08-31 17:24:41.583 INFO 4821 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext\n2024-08-31 17:24:41.585 INFO 4821 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in\n2024-08-31 17:24:42.687 INFO 4821 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...\n2024-08-31 17:24:43.200 INFO 4821 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.\n2024-08-31 17:24:43.364 INFO 4821 --- [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: defau\n2024-08-31 17:24:43.544 INFO 4821 --- [           main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.11.Final\n2024-08-31 17:24:43.962 INFO 4821 --- [           main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}\n2024-08-31 17:24:44.275 INFO 4821 --- [           main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL5\n2024-08-31 17:24:45.415 INFO 4821 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibe\nplatform.internal.NoJtaPlatform]\n2024-08-31 17:24:45.435 INFO 4821 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit\n2024-08-31 17:24:46.369 WARN 4821 --- [           main] JpaBaseConfigurationsJpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, perfo\nermed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning\n2024-08-31 17:24:47.448 INFO 4821 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path \n2024-08-31 17:24:47.480 INFO 4821 --- [           main] c.e.s.SpringBackendApplication        : Started SpringBackendApplication in 10.972 seconds (JVM\n2024-08-31 17:24:47.943 INFO 4821 --- [nio-8080-exec-4] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'\n2024-08-31 17:24:47.945 INFO 4821 --- [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'\n2024-08-31 17:24:47.948 INFO 4821 --- [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
```

After this enter the frontend NLB dns in the browser and see the output

Output :



A screenshot of a web browser window titled "AngularFrontend". The URL in the address bar is <https://frontend-23bc4099d3674985.elb.ap-south-1.amazonaws.com/workers>. The page displays a table titled "Workers" with the following data:

Order	First Name	Last Name	Status	Edit Button	Delete Button
1.	Ivan	Holicek	Working	Edit	Delete
2.	Marko	Markovic	Vacation	Edit	Delete
3.	Ivo	Ivica	Working	Edit	Delete
4.	Luka	Lukovic	Working	Edit	Delete
5.	Filip	Filipovic	Working	Edit	Delete

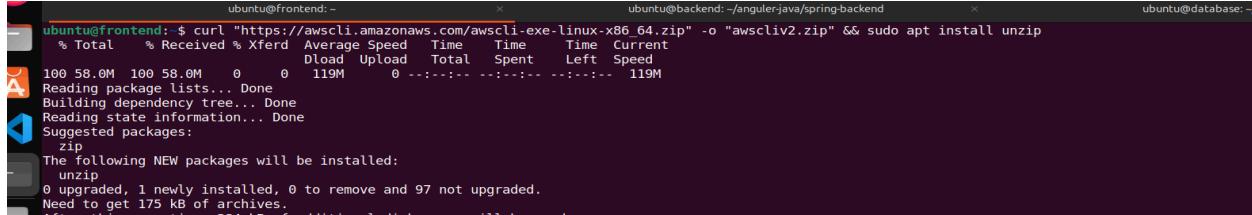
The browser interface includes a sidebar with icons for file operations, a search bar at the top right, and developer tools at the bottom. A green footer bar at the bottom of the page reads "Modified By CloudBlitz".

Step 19 : Now install the aws cli on frontend and attach s3 full permission role to the frontend server.

Command :

sudo apt install unzip && curl

**[“https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip”](https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip)
-o “awscliv2.zip”**



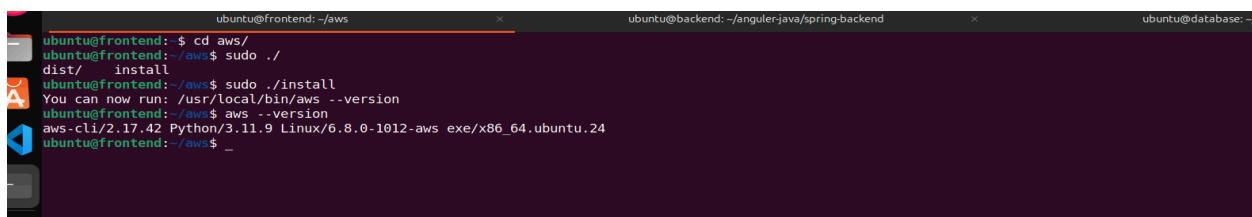
```
ubuntu@frontend:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" && sudo apt install unzip
% Total    % Received  % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
100 58.0M  100 58.0M    0     0  119M      0 --:--:-- 0:00:01  119M
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 97 not upgraded.
Need to get 175 kB of archives.
```

Unzip the file



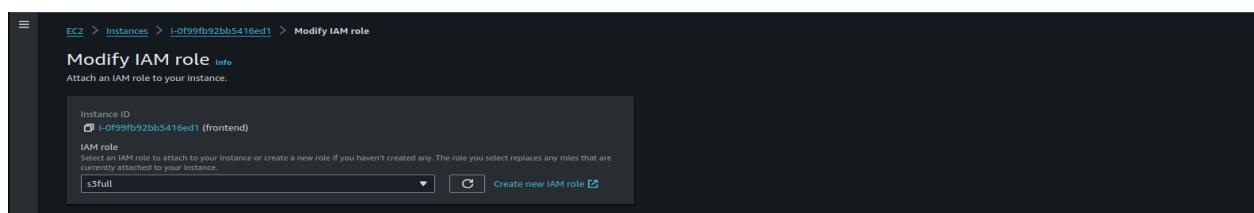
```
ubuntu@frontend:~$ unzip awscliv2.zip
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/README.md
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
  creating: aws/dist/awscli/
  creating: aws/dist/cryptography/
  creating: aws/dist/docutils/
  creating: aws/dist/lib-dynload/
  inflating: aws/dist/aws_completer
```

Execute the install file and verify aws installation

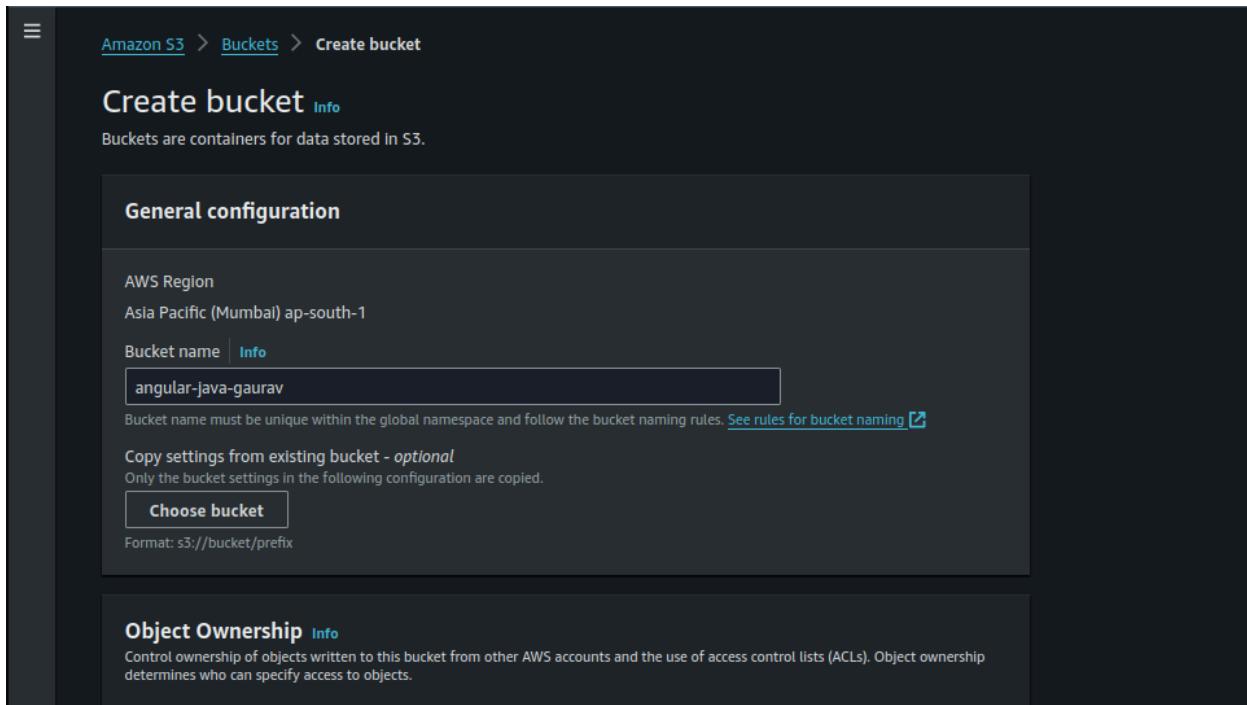


```
ubuntu@frontend:~$ cd aws/
ubuntu@frontend:~/aws$ sudo ./install
ubuntu@frontend:~/aws$ sudo ./install
You can now run: /usr/local/bin/aws --version
ubuntu@frontend:~/aws$ aws --version
aws-cli/2.17.42 Python/3.11.9 Linux/6.8.0-1012-aws exe/x86_64/ubuntu.24
ubuntu@frontend:~/aws$
```

Attach the S3 IAM role to the frontend server.



Step 20 : Create a s3 bucket .



Once bucket is created upload the necessary object files in it.

```
ubuntu@frontend:~$ aws s3 cp /home/ubuntu/angular-java/angular-frontend/dist/angular-frontend/ s3://angular-java-gaurav/ --recursive
upload: angular-java/angular-frontend/dist/angular-frontend/favicon.ico to s3://angular-java-gaurav/favicon.ico
upload: angular-java/angular-frontend/dist/angular-frontend/styles.ef46db3751d8e999.css to s3://angular-java-gaurav/styles.ef46db3751d8e999.css
upload: angular-java/angular-frontend/dist/angular-frontend/runtime.e411e20b75d2e1de.js to s3://angular-java-gaurav/runtime.e411e20b75d2e1de.js
upload: angular-java/angular-frontend/dist/angular-frontend/index.html to s3://angular-java-gaurav/index.html
upload: angular-java/angular-frontend/dist/angular-frontend/3rdpartylicenses.txt to s3://angular-java-gaurav/3rdpartylicenses.txt
upload: angular-java/angular-frontend/dist/angular-frontend/polyfills.b525ededa71d3b7f.js to s3://angular-java-gaurav/polyfills.b525ededa71d3b7f.js
upload: angular-java/angular-frontend/dist/angular-frontend/main.03002de15067b11b.js to s3://angular-java-gaurav/main.03002de15067b11b.js
ubuntu@frontend:~$
```

After adding the files go to the permission tab of bucket and make necessary changes for allowing anyone(publically access the objects in the bucket).

The screenshot shows the 'Edit Block public access (bucket settings)' page. At the top, there's a breadcrumb navigation: Amazon S3 > Buckets > angular-java-gaurav > Edit Block public access (bucket settings). The main section is titled 'Block public access (bucket settings)'. It contains a detailed description of what public access is and how it can be controlled. Below this, there are several checkboxes:

- Block all public access**: Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**: S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**: S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**: S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies.

The screenshot shows the 'Edit Object Ownership' page. The breadcrumb navigation is: Amazon S3 > Buckets > angular-java-gaurav > Edit Object Ownership. The main section is titled 'Object Ownership'. It explains that object ownership controls who can specify access to objects. There are two radio button options:

- ACLs disabled (recommended)**: All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
- ACLs enabled**: Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

A warning message in a box states: **⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.**

The 'Object Ownership' section has two radio button options:

- Bucket owner preferred**: If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.
- Object writer**: The object writer remains the object owner.

The screenshot shows the 'Access control list (ACL)' page. The breadcrumb navigation is: Amazon S3 > Buckets > angular-java-gaurav > Access control list (ACL). The main section is titled 'Access control list (ACL)'. It includes a note: 'Grant basic read/write permissions to other AWS accounts. Learn more' and a warning: 'The console displays combined access grants for duplicate grantees. To see the full list of ACLs, use the Amazon S3 REST API, AWS CLI, or AWS SDKs.' Another warning states: 'AWS doesn't recommend granting access to the Everyone grantee. Anyone in the world can access the objects in this bucket. Learn more'.

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: bd52c0f4c6ea494512da70e1abfa8cf8d6c01c03c3009b6127511a0773f0c12	List, Write	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	△ List	△ Read
Authenticated users group (anyone with an AWS account) Group: https://acs.amazonaws.com/groups/AuthenticatedUsers	-	-

After this make the objects public using acl from actions tab .

The screenshot shows the AWS S3 'Objects' list interface. A context menu is open over a selected file named 'index.html'. The menu includes options like 'Copy', 'Move', 'Initiate restore', 'Query with S3 Select', 'Edit actions', 'Rename object', 'Edit storage class', 'Edit server-side encryption', 'Edit metadata', 'Edit tags', and 'Make public using ACL'. The 'Make public using ACL' option is highlighted.

Now go to the properties tab of bucket and in the last enable the static web hosting and gave the index.html as serving page.

Step 21 : Now create a Distribution in the cloudfront with origin domain as the s3 bucket .

The screenshot shows the 'Create distribution' wizard in the AWS CloudFront console. The 'Origin' step is active. It requires an 'Origin domain' (set to 'angular-java-gaurav.s3-website.ap-south-1.amazonaws.com'), 'Protocol' (set to 'HTTP only'), and 'HTTP port' (set to '80'). An optional 'Origin path' field is empty. The 'Name' field is also empty. The overall background of the wizard is dark.

Also add ssl certificate to make our CDN DNS secured.

The screenshot shows the 'Settings' tab for a CloudFront distribution. Under 'Price class', 'Use all edge locations (best performance)' is selected. In the 'Alternate domain name (CNAME) - optional' section, there is a dropdown menu containing 'seytan.in (9ecca95e-c457-4352-b31d-4669fe351037)' and a 'Request certificate' button. Below this, a note about legacy client support and a checkbox for 'Enabled' are visible. In the 'Security policy' section, 'TLSv1.2_2021 (recommended)' is selected, along with other options like 'TLSv1.2_2019', 'TLSv1.2_2018', 'TLSv1.1_2016', and 'TLSv1_2016'.

Now once it is deployed paste the CDN DNS in the browser and see the Output

Output :

The screenshot shows a web browser window with the URL 'https://d266r4221bf6d.cloudfront.net/workers'. The page has a green header bar with the word 'Workers'. Below it is a table titled 'Add Worker' with a search bar. The table lists five workers with columns for Order, First Name, Last Name, Status, Edit Button, and Delete Button. The workers are: 1. Ivan Holicek (Working), 2. Marko Markovic (Vacation), 3. Ivo Ivica (Working), 4. Luka Lukovic (Working), and 5. Filip Filipovic (Working). At the bottom of the page is a green footer bar with the text 'Modified by CloudBlitz'.

Order	First Name	Last Name	Status	Edit Button	Delete Button
1.	Ivan	Holicek	Working	Edit	Delete
2.	Marko	Markovic	Vacation	Edit	Delete
3.	Ivo	Ivica	Working	Edit	Delete
4.	Luka	Lukovic	Working	Edit	Delete
5.	Filip	Filipovic	Working	Edit	Delete

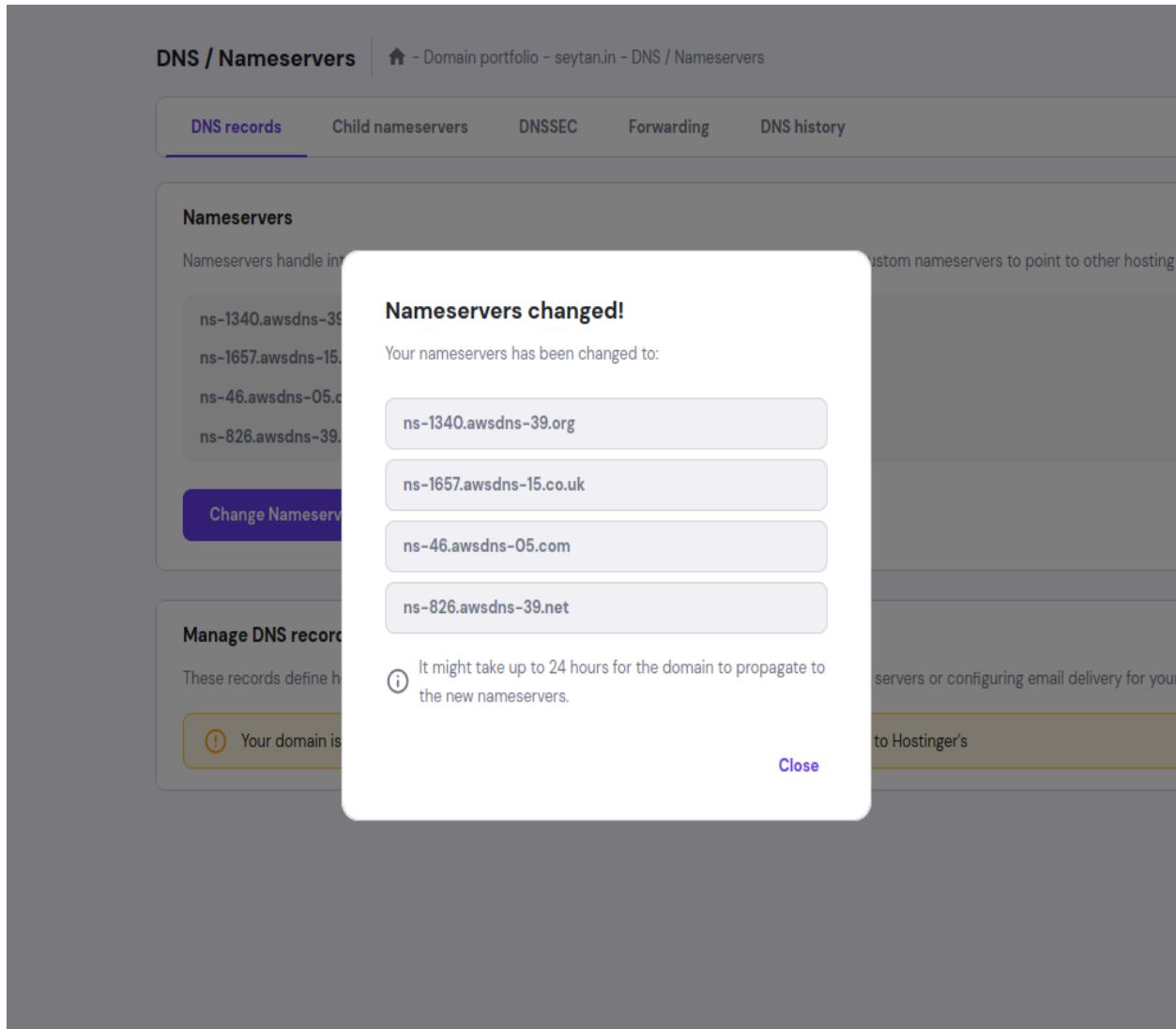
Step 22: Now Create a Hosted Zone in the route53 with your domain.

The screenshot shows the 'Create hosted zone' configuration page in the AWS Route 53 console. The 'Hosted zone configuration' section is displayed, containing fields for 'Domain name' (set to 'seytan.in'), 'Description - optional' (set to 'The hosted zone is used for...'), and 'Type' (set to 'Internet'). A note at the bottom indicates that the type indicates whether traffic is routed on the internet or in an Amazon VPC.

Now copy the ns-record and paste it in the 3rd party domain provider ns-records.

The screenshot shows the 'Edit hosted zone' details page for the 'seytan.in' domain. It displays the 'Record details' sidebar with information about an NS record named 'seytan.in'. The record has a value of 'ns-826.awsdns-39.net.' and a TTL of 172800 seconds. The main pane shows a table of records for the 'seytan.in' domain, including an SOA record and several NS records.

Type	Alias	Value/Route traffic to	TTL
SOA	-	ns-826.awsdns-39.net. ns-1340.awsdns-39.org. ns-1657.awsdns-15.co.uk. ns-46.awsdns-05.com.	172800
NS	-	ns-826.awsdns-39.net. awsd...	900



Ns-records changed in 3rd party domain provider .

Now Request a ssl certificate from ACM and add it to the domain.

AWS Certificate Manager > Certificates > Request certificate > Request public certificate

Request public certificate

Domain names
Provide one or more domain names for your certificate.

Fully qualified domain name | Info

Add another name to this certificate
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Validation method Info
Select a method for validating domain ownership.

DNS validation - recommended
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

Email validation
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

Key algorithm Info
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

RSA 2048
RSA is the most widely used key type.

ECDSA P 256
Equivalent to countermeasures to RSA 3072

Successfully requested certificate with ID 870c2003-78a7-41ec-9847-46d62b8259f5
A certificate request with a status of pending validation has been created. Further action is needed to complete the validation and activate the certificate.

AWS Certificate Manager > Certificates > [870c2003-78a7-41ec-9847-46d62b8259f5](#) > Create DNS records in Amazon Route 53

Create DNS records in Amazon Route 53 (1/1)

Search domains 1 match

Validation status = Pending validation X Validation status = Failed X

Is domain in Route 53? = Yes X Clear filters

< 1 >

Domain	Validation status	Is domain in Route 53?
seytan.in	Pending validation	Yes

Cancel Create records

Step 23: Now add a CNAME to the CDN Distribution which we created by editing it .

The screenshot shows the 'Edit settings' page for a CloudFront distribution. In the 'Alternate domain name (CNAME) - optional' section, the value 'seytan.in' is entered into a text input field. Below this, there is a note about legacy clients and a link to the bulk editor. In the 'Custom SSL certificate - optional' section, a dropdown menu shows 'seytan.in (9ecca95e-c457-4352-b31d-4669fe351037)' selected, with a checkbox for 'Request certificate' checked.

Once it is done wait for deployed state after that add an A-record with alias as the CDN DNS

The screenshot shows the 'Create record' page in Route 53. A new record is being created with the name 'subdomain' and type 'A'. The 'Alias' option is selected, and the target is set to 'Alias to CloudFront distribution' with the region 'US East (N. Virginia)'. The target IP address is listed as 'd266r4221lbf6d.cloudfront.net'. The 'Evaluate target health' setting is set to 'No'. There is also a search bar for target IP addresses.

**Step 24: Now enter your domain seytan.in
And see it is secured and the content is coming via CDN
DNS.**

Output :

The screenshot shows a browser window with the following details:

- Address Bar:** https://seytan.in/workers
- Page Title:** Site information for seytan.in
- Content Area:** A green header bar with the word "Workers". Below it is a table titled "Add Worker" with the following data:

Order	First Name	Last Name	Status	Edit Button	Delete Button
1.	Ivan	Holicek	Working	Edit	Delete
2.	Marko	Markovic	Vacation	Edit	Delete
3.	Ivo	Ivica	Working	Edit	Delete
4.	Luka	Lukovic	Working	Edit	Delete
5.	Elin	Filinovic	Working	Edit	Delete

- Network Tab:** Shows network activity for the domain seytan.in. One request is highlighted in blue:

Status	Method	Domain	File	Initiator	Type	Transferred	Size
304	GET	seytan.in	workers document		htm	cached	551
200	GET	seytan.in	runtime script		js	cached	0 B
200	GET	seytan.in	polyfills script		js	cached	0 B
200	GET	seytan.in	main.03 script		js	cached	0 B
200	GET	backend-8...	workers polyfills.b525...		js	660 B	373
200	GET	seytan.in	Favicon	Favicon Loader	x-icc	cached	948

- Headers Section:** Shows detailed headers for the highlighted request (304 GET to workers document). Some headers are expanded:

 - date: Sat, 31 Aug 2024 19:27:40 GMT
 - etag: "847ae570aae3f9965cb216c8531ee60"
 - last-modified: Sat, 31 Aug 2024 18:04:50 GMT
 - server: AmazonS3
 - via: 1.1 d50d717134ed031589d1b934a41d279a.cloudfront.net (CloudFront)
 - x-amz-cf-id: qv-ncGdE63BF94M_klF1NPRxe_BaqCyzvQw0U-GRkBjRz1KaMqThvA==
 - x-amz-cf-pop: BOM70-PS
 - x-cache: Error from cloudfront
 - X-Firefox-Spdy: h2

- Request Headers:** Shows the raw request headers sent by the browser.

Step 25: Database backup to s3 bucket automatically .
Now go to the database server and install aws cli and attach the s3 Full IAM role to the database server .

Refer this above step with step Number 19

Step 26 : Create a s3 bucket for database backup.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The process is divided into three main steps:

- General configuration:** Set the AWS Region to 'Asia Pacific (Mumbai) ap-south-1'. The Bucket name is 'bcakup-database-gaurav'. A note indicates the bucket name must be unique and follow naming rules. A 'Choose bucket' button is available for copying settings from existing buckets.
- Object Ownership:** The 'Object Ownership' section specifies that objects are owned by the account ('Bucket owner enforced'). It offers two options:
 - ACLs disabled (recommended):** All objects are owned by the account; access is controlled by policies.
 - ACLs enabled:** Objects can be owned by other accounts; access is controlled by ACLs.
- Block Public Access settings for this bucket:** A note states that public access is granted through ACLs, bucket policies, or access point policies. It recommends turning on 'Block all public access'.

Also enable the bucket versioning in this case .

The screenshot shows the 'Bucket Versioning' section of the AWS S3 Bucket Properties page. It includes a description of what bucket versioning is, a note about its availability, and a link to learn more. Below this, there is a radio button group for enabling or disabling bucket versioning, with 'Enable' selected. The 'Tags - optional (0)' section shows no tags associated with the bucket and has an 'Add tag' button. The 'Default encryption' section indicates server-side encryption is automatically applied to new objects and lists three encryption type options: SSE-S3 (selected), SSE-KMS, and DSSE-KMS. The 'Bucket Key' section notes that using an S3 Bucket Key for SSE-KMS reduces costs but is not supported for DSSE-KMS, with a link to learn more.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

Enable

Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable

Enable

Step 27:

**Once it is done now we have to close the nat gateway from this private subnets backend and database .
Go to the route tables of this 2 subnet and remove them from the subnet association explicitly section .**

Step 28 : As Now we have removed the Nat gateway we have to add the s3 vpc endpoint in this vpc pointing to the private subnets so we can connect to the s3 without nat gateway ,internet.

What is s3 vpc endpoint ?

→An S3 VPC endpoint allows you to connect your VPC directly to Amazon S3 without using an internet gateway, NAT device, VPN connection, or AWS Direct Connect. This enhances security by ensuring that S3 traffic stays within the AWS network.

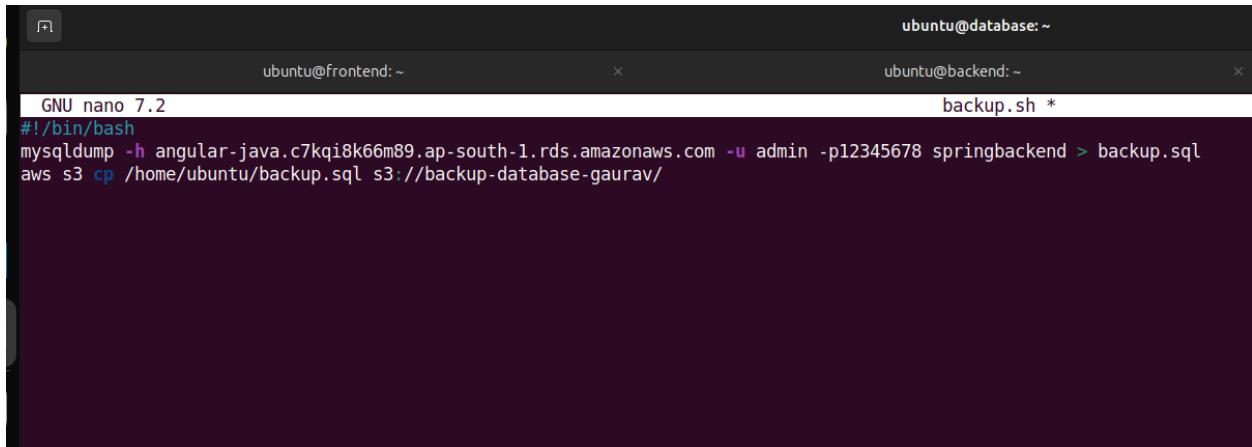
The screenshot shows the AWS VPC service interface with the 'Endpoints' section selected. A new endpoint is being created, specifically a VPC endpoint for the AWS service 'Amazon S3'. The 'Name tag' field contains 'my-endpoint-01'. Under 'Service category', 'AWS services' is selected. In the 'Services' list, 'com.amazonaws.ap-south-1.s3' is chosen. For the 'VPC', 'vpc-0345beba67027169b' is selected. The overall process is titled 'Create endpoint'.

Add the private subnets in this .

The screenshot shows the 'Route tables' configuration step. It lists four route tables: 'RDS-Pvt-rt' (Main, 3 subnets), 'angular-java-rtb-private2-ap-south-1a' (No, -), 'angular-java-rtb-private1-ap-south-1a' (Yes, 2 subnets), and 'angular-java-rtb-public' (No, subnet-06355c7684aa5c7b9). A note at the bottom states: 'When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.' There are also sections for 'Policy' and 'Full access'.

Step 29 : Now create a backup.sh file and add the followings lines to it .

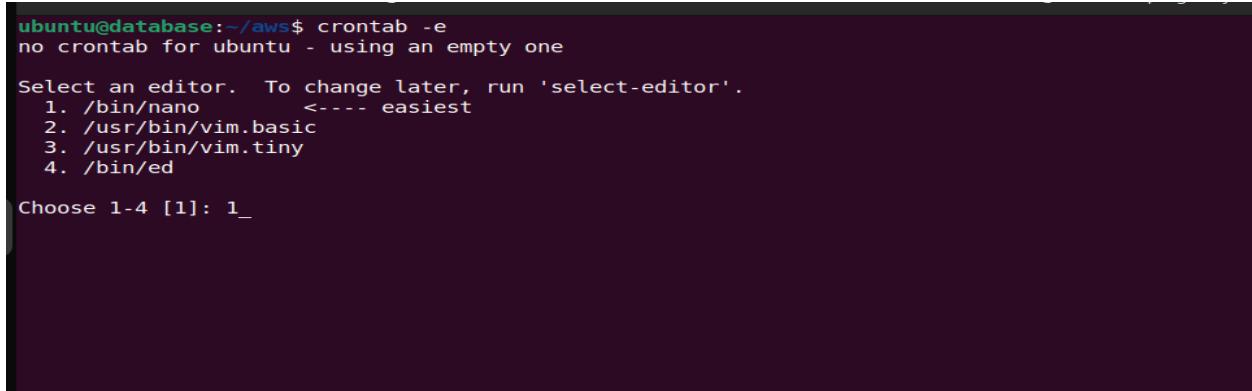
It will create a backup of database and copy it to the backup-database-gaurav bucket.



The screenshot shows a terminal window with two tabs: "ubuntu@frontend: ~" and "ubuntu@backend: ~". The "ubuntu@backend: ~" tab contains the following command:

```
GNU nano 7.2
#!/bin/bash
mysqldump -h angular-java.c7kqi8k66m89.ap-south-1.rds.amazonaws.com -u admin -p12345678 springbackend > backup.sql
aws s3 cp /home/ubuntu/backup.sql s3://backup-database-gaurav/
```

As we want to make this process automatically we will use crontab for this.

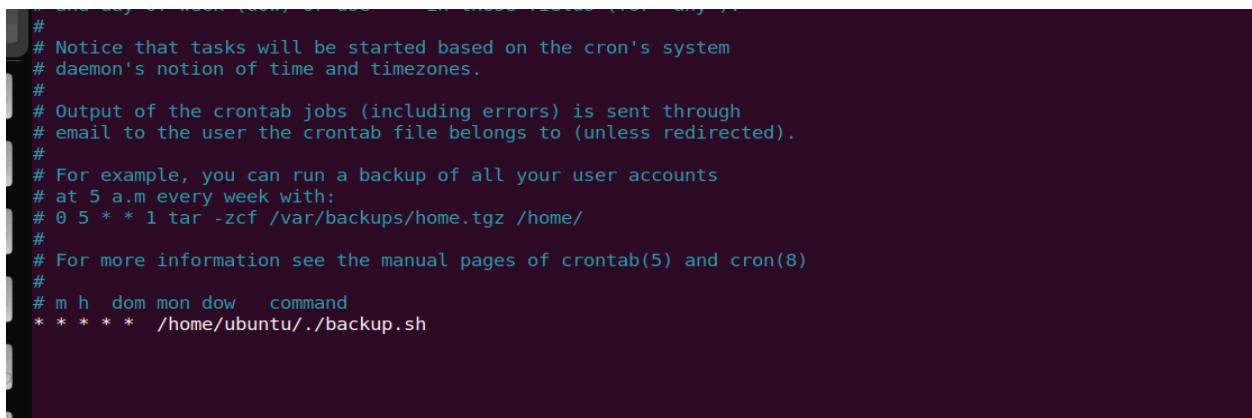


The screenshot shows a terminal window where the user is selecting an editor to edit the crontab. The options listed are:

- 1. /bin/nano <---- easiest
- 2. /usr/bin/vim.basic
- 3. /usr/bin/vim.tiny
- 4. /bin/ed

The user has chosen option 1 (nano) and is prompted to enter the editor command:

```
Choose 1-4 [1]: 1_
```



The screenshot shows the contents of a crontab file. It includes comments explaining the cron syntax and a specific command to run a backup script every minute:

```
# This way, cron (the daemon that runs cron jobs) can invoke the shell (you may ...).
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /home/ubuntu/.backup.sh
```

It will execute the task every 1 minute .

Output :

As we can see we had enabled the versioning so we can easily identify the first backup and latest backup of our database.

The screenshot shows the AWS S3 console interface. The URL in the browser is <https://ap-south-1.console.aws.amazon.com/s3/buckets/backup-database-gaurav?region=ap-south-1&bucketType=general&tab=objects&showversions=true>. The AWS navigation bar includes services like EC2, EFS, VPC, IAM, S3, CloudWatch, Systems Manager, RDS, Route 53, Certificate Manager, CloudFront, Key Management Service, Lambda, and API Gateway. The main content area shows the 'Objects' tab for the 'backup-database-gaurav' bucket. There are two objects listed:

Name	Type	Version ID	Last modified	Size	Storage class
backup.sql	sql	eRTCLj8a4uj2bWUjimRPEXV96XyHxJ	September 1, 2024, 01:46:02 (UTC+05:30)	2.2 KB	Standard
sigSFSPRzVADBSq6yflrHx0YnB526V	sql	slg5F5PRzVADBSq6yflrHx0YnB526V	September 1, 2024, 01:45:02 (UTC+05:30)	2.2 KB	Standard

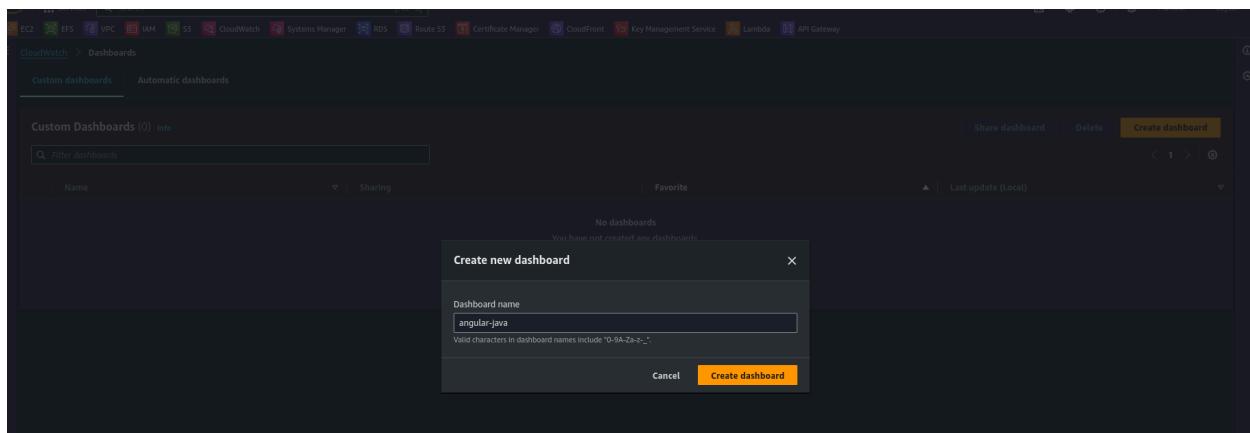
An 'Actions' dropdown menu is open over the second object ('sigSFSPRzVADBSq6yflrHx0YnB526V'). The menu items include Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, and Upload. A 'Show versions' link is also visible above the object list.

Step 30:

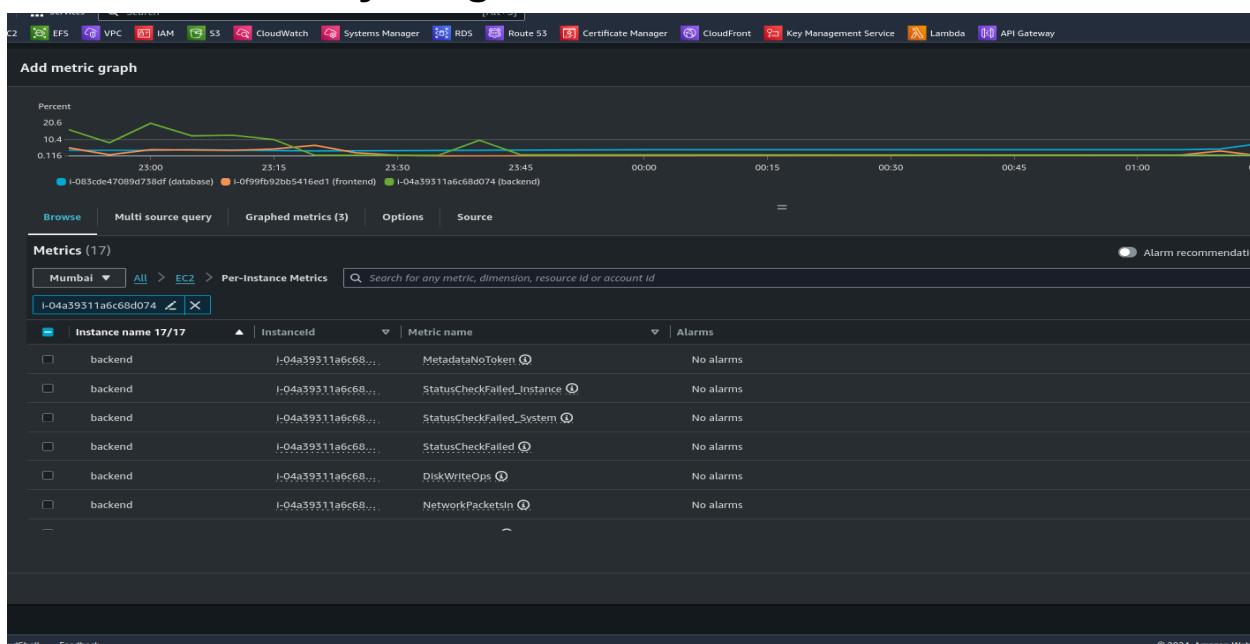
Now Create a Dashboard for monitoring our 3-tier application in Cloudwatch .

What is cloudwatch ?

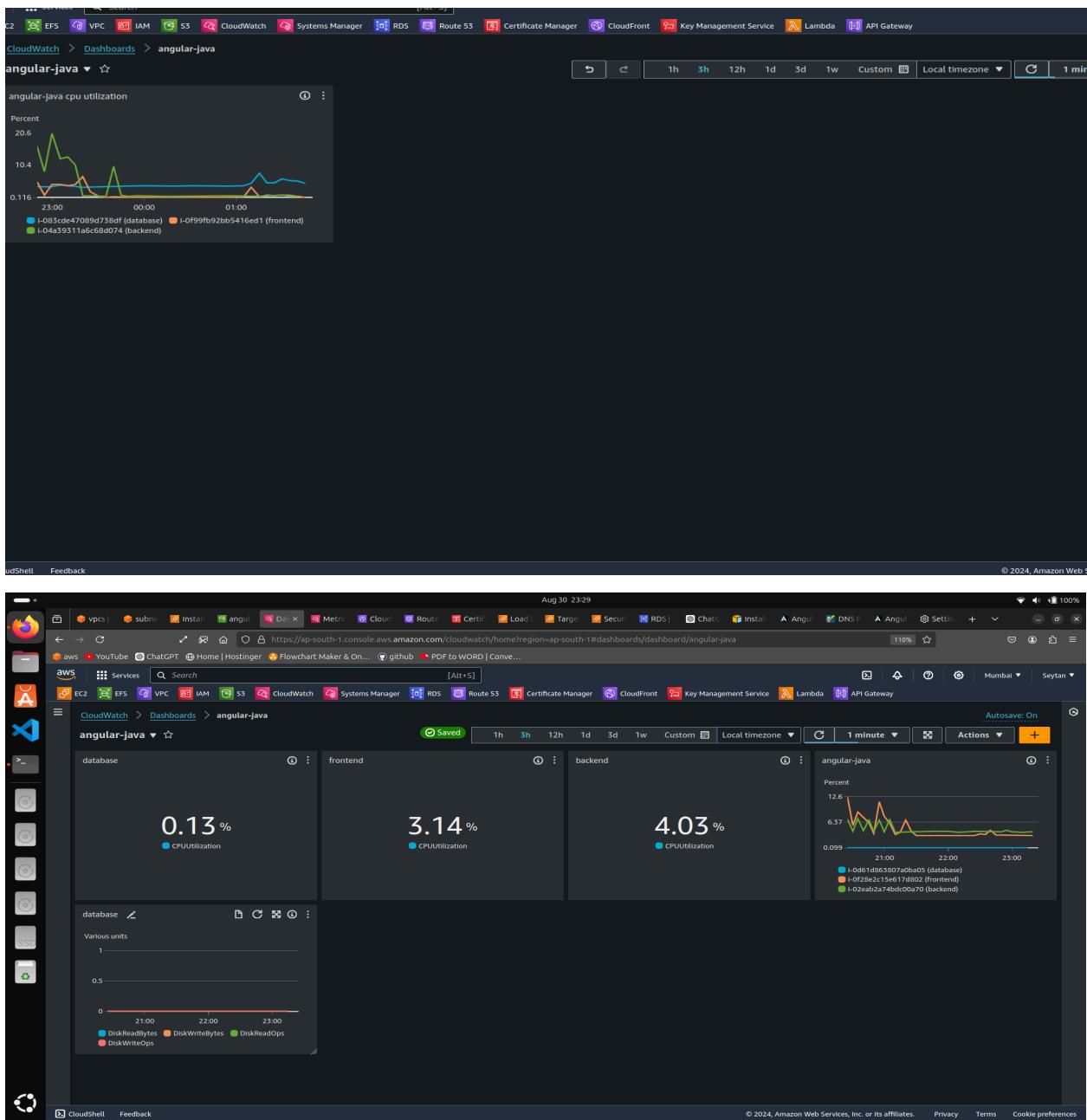
→CloudWatch is an AWS service that monitors and logs performance metrics and operational data from AWS resources and applications. It provides real-time insights, enabling you to collect, access, and analyze logs, set alarms, and automatically react to changes in your environment.



Add the necessary widgets.



Output:



Conclusion :

This deployment ensures a secure and scalable environment for your 3-tier application, leveraging AWS services effectively. By utilizing Route 53 and AWS CDN, you'll achieve optimal performance and reliability for both frontend and backend components.