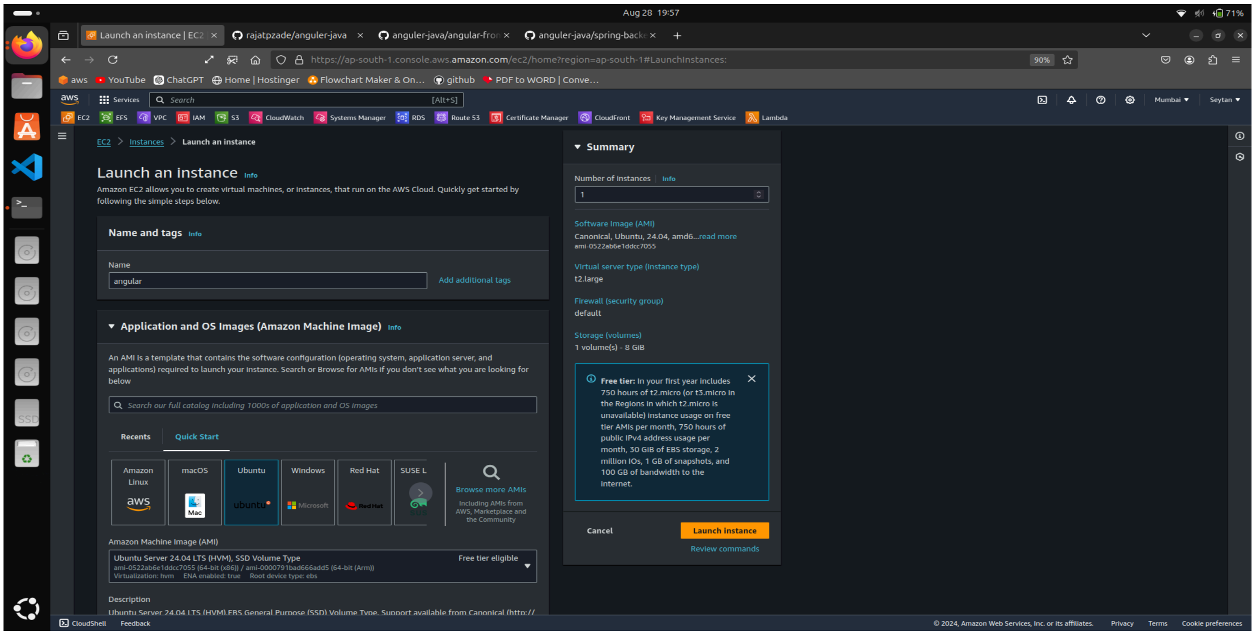
Task : Angular-Java App Deployment on EC2 Instance.

Step 1:Create an EC2 Instance and get its connection by using following command

→ ssh -i private_key.pem ubutnu@public_ip



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: angular

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

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0522ab6e1ddcc7055 (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. Support available from Canonical (http://

Summary

Number of instances Info: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0522ab6e1ddcc7055

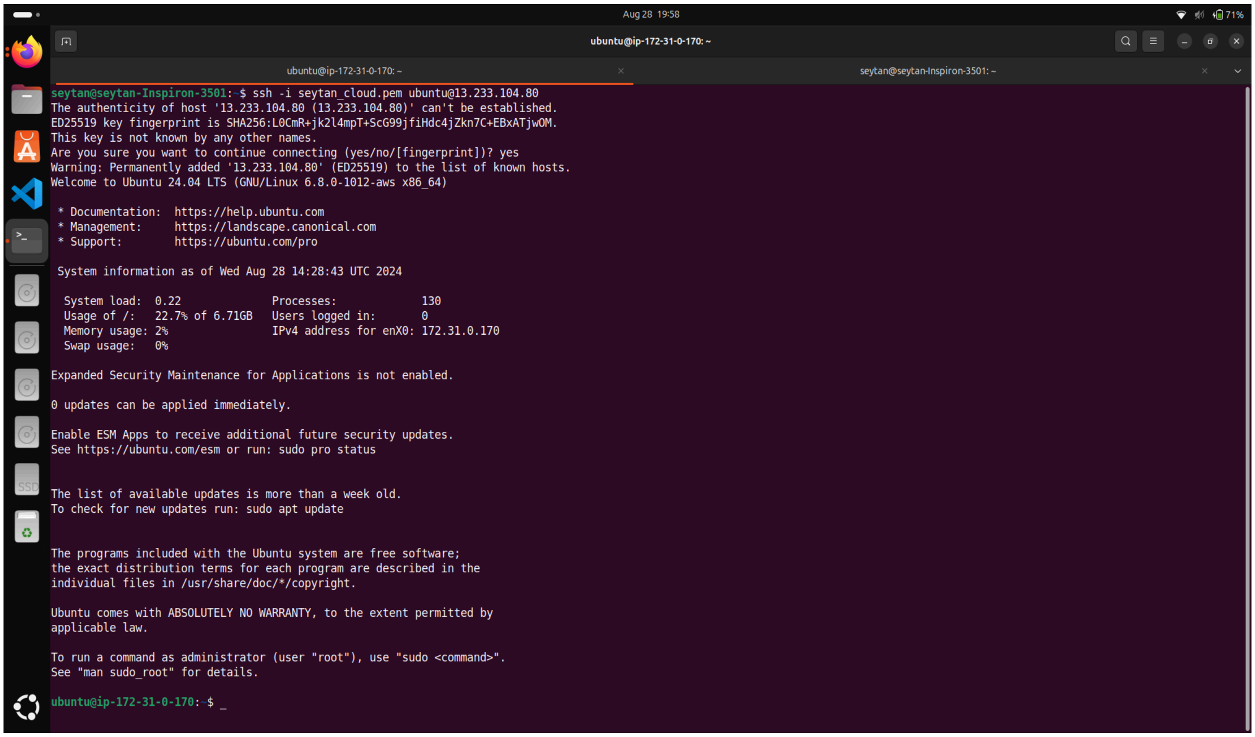
Virtual server type (instance type)
t2.large

Firewall (security group)
default

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

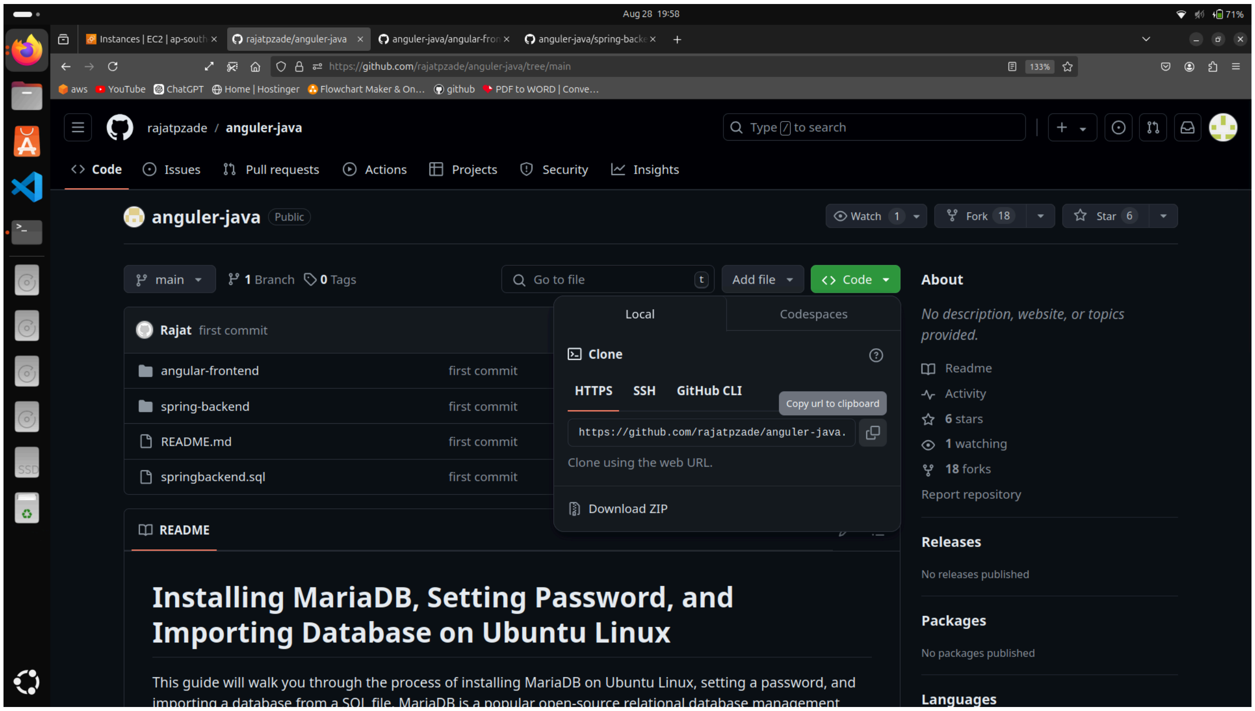

Aug 28 19:58
71%
ubuntu@ip-172-31-0-170: ~
ubuntu@ip-172-31-0-170: ~
seytan@seytan-Inspiron-3501: ~

```
seytan@seytan-Inspiron-3501:~$ ssh -i seytan_cloud.pem ubuntu@13.233.104.80
The authenticity of host '13.233.104.80 (13.233.104.80)' can't be established.
ED25519 key fingerprint is SHA256:L0CmR+jk2l4mpT+ScG99jfiHdc4jZkn7C+EBxATjwOM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.233.104.80' (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 Wed Aug 28 14:28:43 UTC 2024

  System load:  0.22              Processes:             130
  Usage of /:   22.7% of 6.71GB   Users logged in:       0
  Memory usage: 2%                IPv4 address for enX0: 172.31.0.170
  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-172-31-0-170:~$ _
```

Step 2:

Now Clone the git repository on ec2 .

git clone https://github.com/rajatpzade/anguler-java.git



Rajat first commit	
angular-frontend	first commit
spring-backend	first commit
README.md	first commit
springbackend.sql	first commit

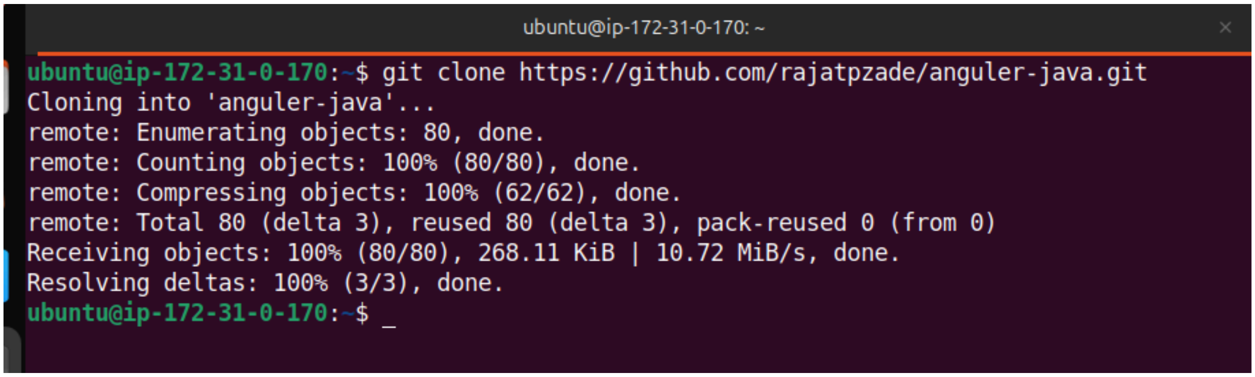
```
https://github.com/rajatpzade/anguler-java.
```

Clone using the web URL.

No description, website, or topics provided.

Installing MariaDB, Setting Password, and Importing Database on Ubuntu Linux

This guide will walk you through the process of installing MariaDB on Ubuntu Linux, setting a password, and importing a database from a SQL file. MariaDB is a popular open-source relational database management



```
ubuntu@ip-172-31-0-170:~$ git clone https://github.com/rajatpzade/anguler-java.git
Cloning into 'anguler-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 | 10.72 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@ip-172-31-0-170:~$ _
```

Step 3: Update the system and install mariadb database .

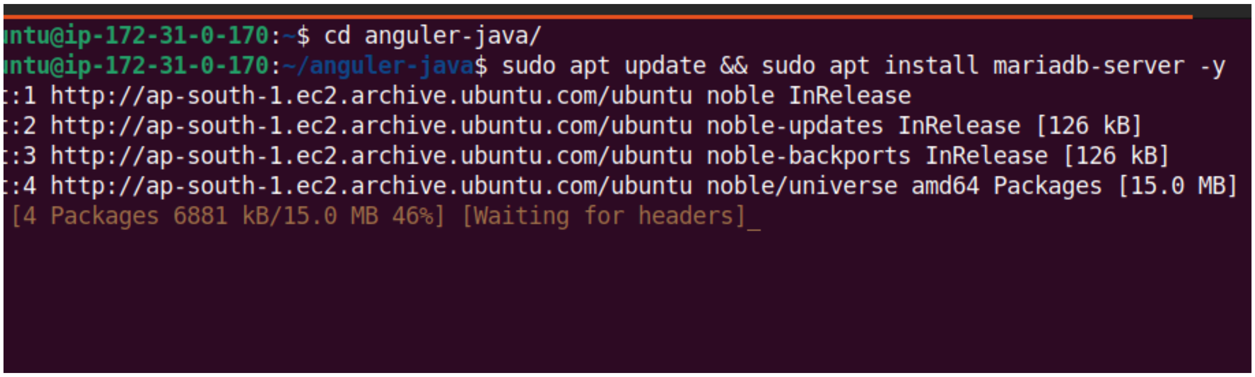
Commands:

sudo apt update

sudo apt install mariadb-server

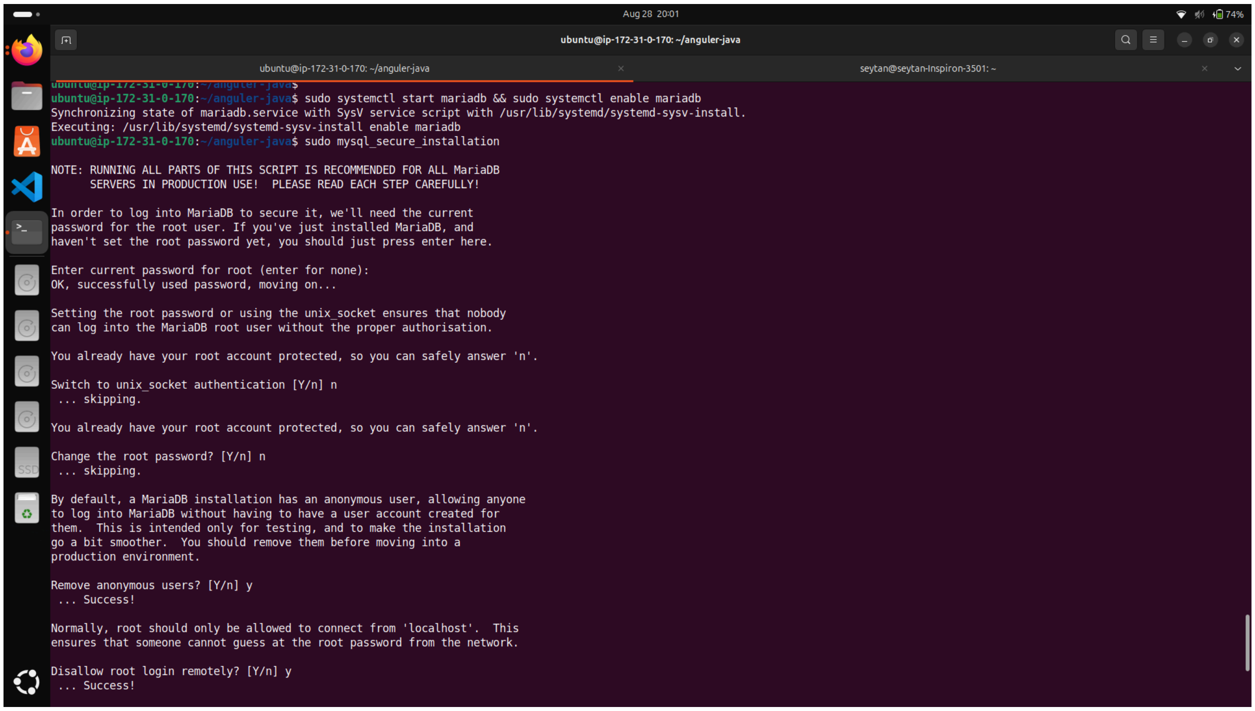
sudo systemctl start mariadb

sudo systemctl enable mariadb



```
ntu@ip-172-31-0-170:~$ cd anguler-java/
ntu@ip-172-31-0-170:~/anguler-java$ sudo apt update && sudo apt install mariadb-server -y
:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
 [4 Packages 6881 kB/15.0 MB 46%] [Waiting for headers]_
```

Now use the command → sudo mysql_secure_installation

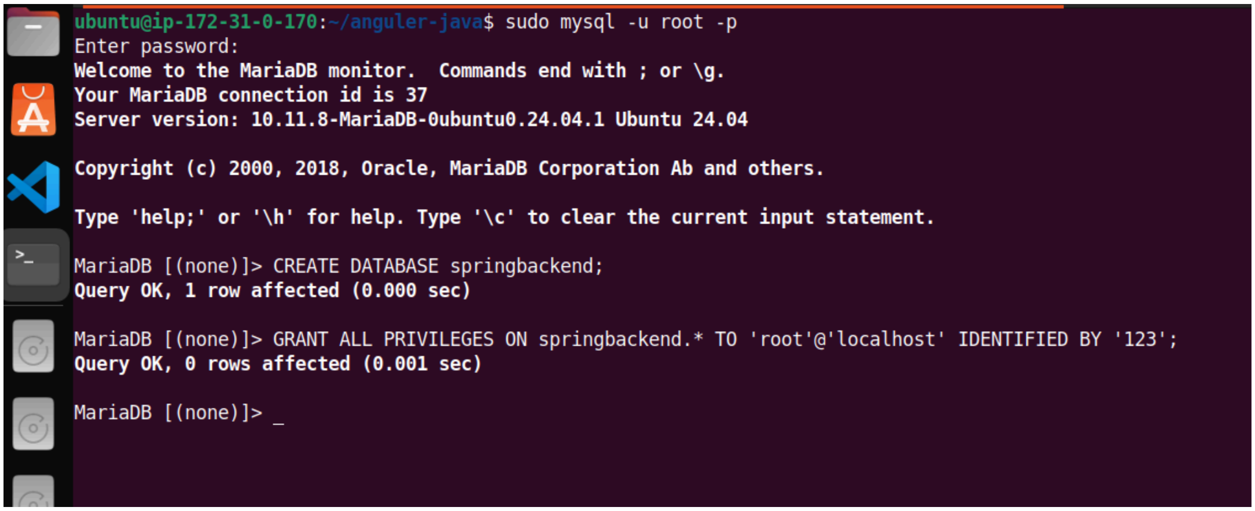


Aug 28 20:01
74%
ubuntu@ip-172-31-0-170: ~/anguler-java
ubuntu@ip-172-31-0-170: ~/anguler-java
seytan@seytan-Inspiron-3501: ~
SSD

```
ubuntu@ip-172-31-0-170:~/anguler-java$
ubuntu@ip-172-31-0-170:~/anguler-java$ sudo systemctl start mariadb && sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mariadb
ubuntu@ip-172-31-0-170:~/anguler-java$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!
```

Step 4: Now login to database and use following commands.
Login command: sudo mysql -u root -p

Database commands:
CREATE DATABASE springbackend;
GRANT ALL PRIVILEGES ON springbackend.* TO 'root'@'localhost' IDENTIFIED BY ‘123’;

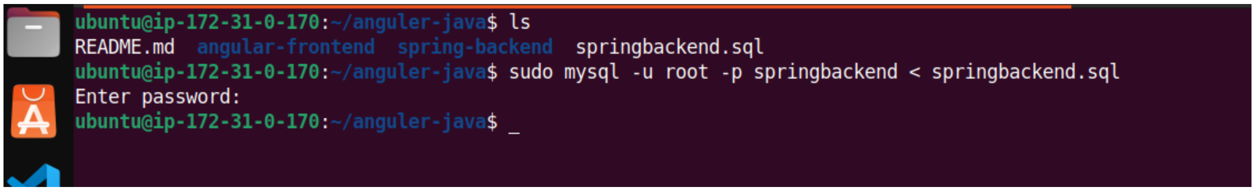


```
ubuntu@ip-172-31-0-170:~/anguler-java$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.11.8-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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 (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON springbackend.* TO 'root'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> _
```

Now , exit .

Then import the database from angular-java directory to our database.

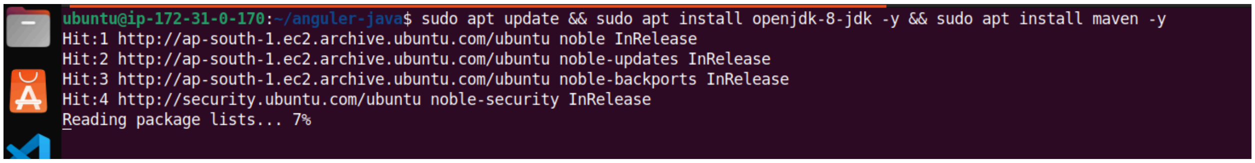
Use command :
sudo mysql -u root -p springbackend < springbackend.sql



```
ubuntu@ip-172-31-0-170:~/anguler-java$ ls
README.md  angular-frontend  spring-backend  springbackend.sql
ubuntu@ip-172-31-0-170:~/anguler-java$ sudo mysql -u root -p springbackend < springbackend.sql
Enter password:
ubuntu@ip-172-31-0-170:~/anguler-java$ _
```

Step 5: **Now go to springbackend folder and follow the next instructions.**
Use command :

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

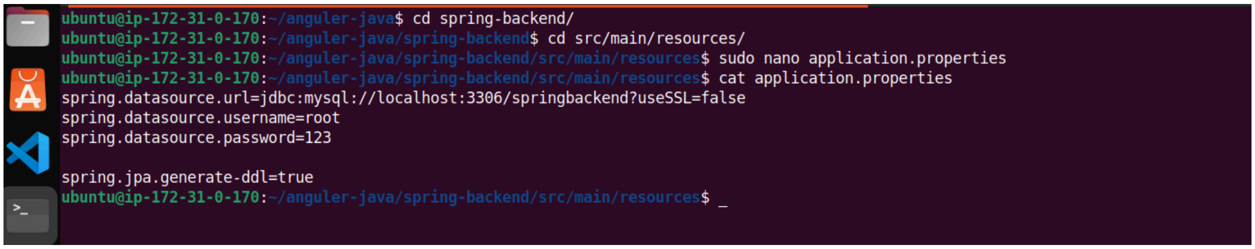


```
ubuntu@ip-172-31-0-170:~/anguler-java$ sudo apt update && sudo apt install openjdk-8-jdk -y && sudo apt install maven -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... 7%
```

Step 6:

Now go to /angular-java/springbackend/src/main/resources/

And edit the application.properties there set the username to root and password to what you set during mariadb installation.

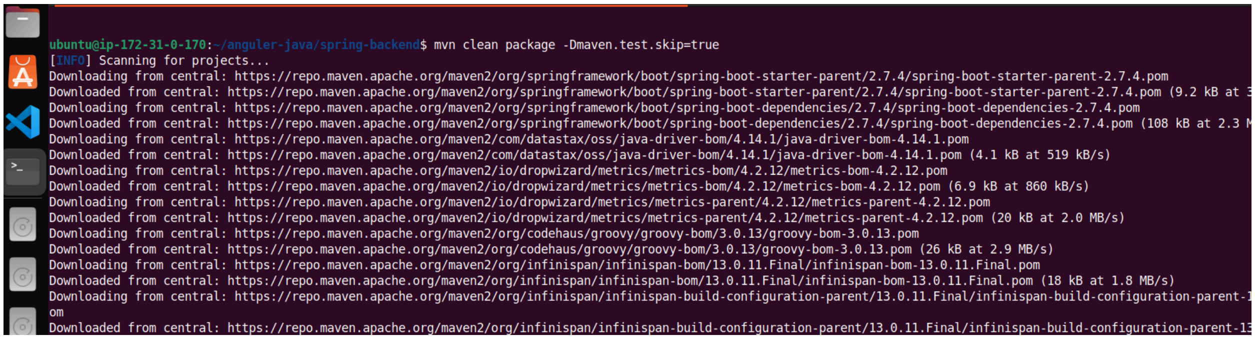


```
ubuntu@ip-172-31-0-170:~/anguler-java$ cd spring-backend/
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend$ cd src/main/resources/
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend/src/main/resources$ sudo nano application.properties
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend/src/main/resources$ cat application.properties
spring.datasource.url=jdbc:mysql://localhost:3306/springbackend?useSSL=false
spring.datasource.username=root
spring.datasource.password=123

spring.jpa.generate-ddl=true
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend/src/main/resources$ _
```

Step 7: Now go back to springbackend directory and run the following command .

→

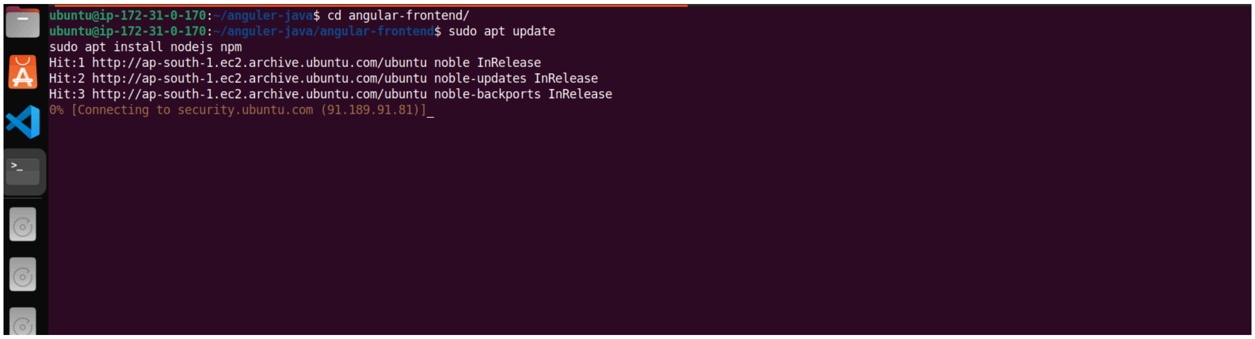
mvn clean package -Dmaven.test.skip=true



```
ubuntu@ip-172-31-0-170:~/anguler-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 (9.2 kB at 3
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 (108 kB at 2.3 M
Downloading from central: https://repo.maven.apache.org/maven2/com/datastax/oss/java-driver-bom/4.14.1/java-driver-bom-4.14.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/datastax/oss/java-driver-bom/4.14.1/java-driver-bom-4.14.1.pom (4.1 kB at 519 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 860 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 2.0 MB/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 2.9 MB/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 (18 kB at 1.8 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-build-configuration-parent/13.0.11.Final/infinispan-build-configuration-parent-1
om
Downloaded from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-build-configuration-parent/13.0.11.Final/infinispan-build-configuration-parent-13
```

Step 8: Now go to the angular-frontend directory and run the following command.

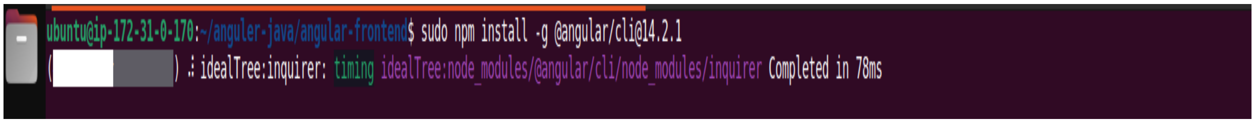
sudo apt update
sudo apt install nodejs npm



```
ubuntu@ip-172-31-0-170:~/anguler-java$ cd angular-frontend/
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ sudo apt update
sudo apt install nodejs npm
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
0% [Connecting to security.ubuntu.com (91.189.91.81)]_
```

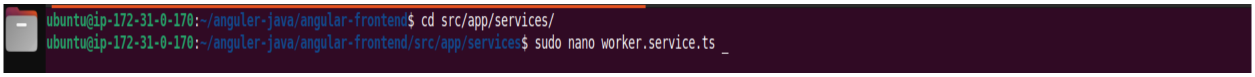
Step 9: Now install angular cli globally by the command:

sudo npm install -g @angular/cli@14.2.1



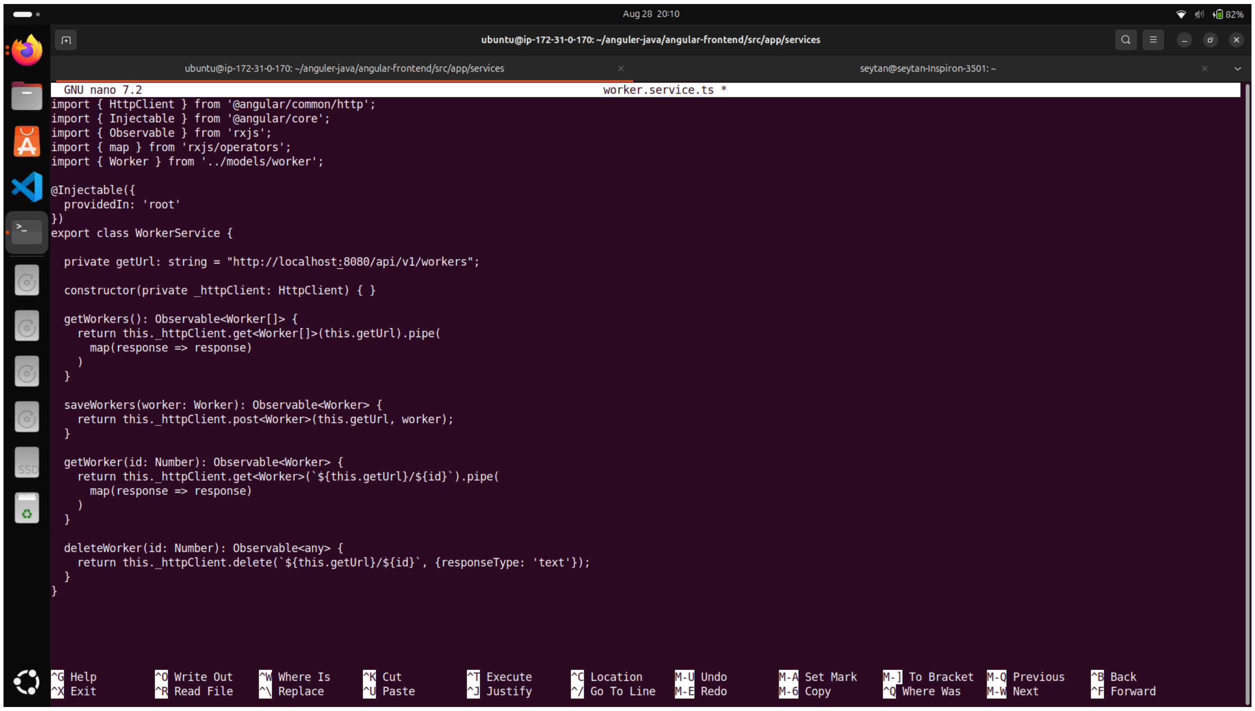
```
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ sudo npm install -g @angular/cli@14.2.1
(          ) ⠼ idealTree:inquirer: timing idealTree:node_modules/@angular/cli/node_modules/inquirer Completed in 78ms
```

Step 10:Now go to the src/app/services directory and edit the worker.service.ts file .



```
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ cd src/app/services/
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend/src/app/services$ sudo nano worker.service.ts _
```

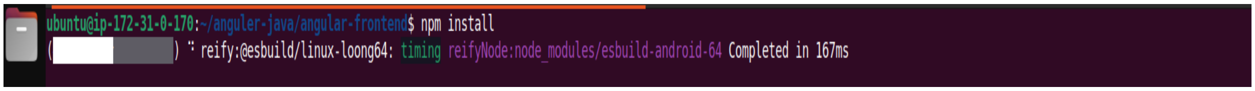
Replace the localhost in the file with public ip of instance.



Aug 28 20:10
82%
ubuntu@ip-172-31-0-170: ~/anguler-java/angular-frontend/src/app/services
ubuntu@ip-172-31-0-170: ~/anguler-java/angular-frontend/src/app/services
seytan@seytan-Inspiron-3501: ~

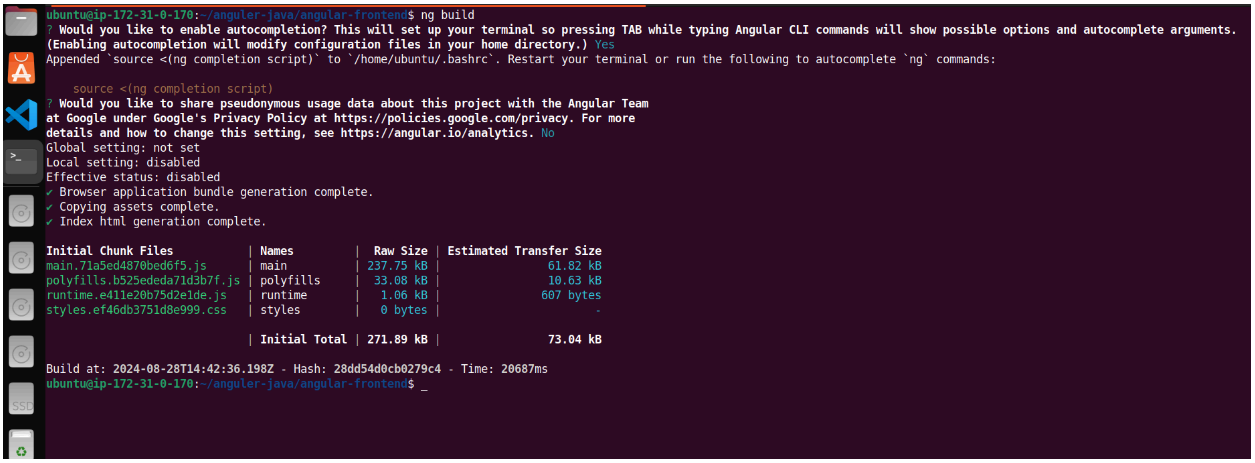
```
  GNU nano 7.2                                      worker.service.ts *
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
import { Worker } from '../models/worker';

@Injectable({
  providedIn: 'root'
})
export class WorkerService {

  private getUrl: string = "http://localhost:8080/api/v1/workers";

  constructor(private _httpClient: HttpClient) { }

  getWorkers(): Observable<Worker[]> {
    return this._httpClient.get<Worker[]>(this.getUrl).pipe(
      map(response => response)
    )
  }

  saveWorkers(worker: Worker): Observable<Worker> {
    return this._httpClient.post<Worker>(this.getUrl, worker);
  }

  getWorker(id: Number): Observable<Worker> {
    return this._httpClient.get<Worker>(`${this.getUrl}/${id}`).pipe(
      map(response => response)
    )
  }

  deleteWorker(id: Number): Observable<any> {
    return this._httpClient.delete(`${this.getUrl}/${id}`, {responseType: 'text'});
  }
}

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous  ^B Back
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was  M-W Next      ^F Forward
```

Step 11: Now go back to angular-frontend and run the npm install command.



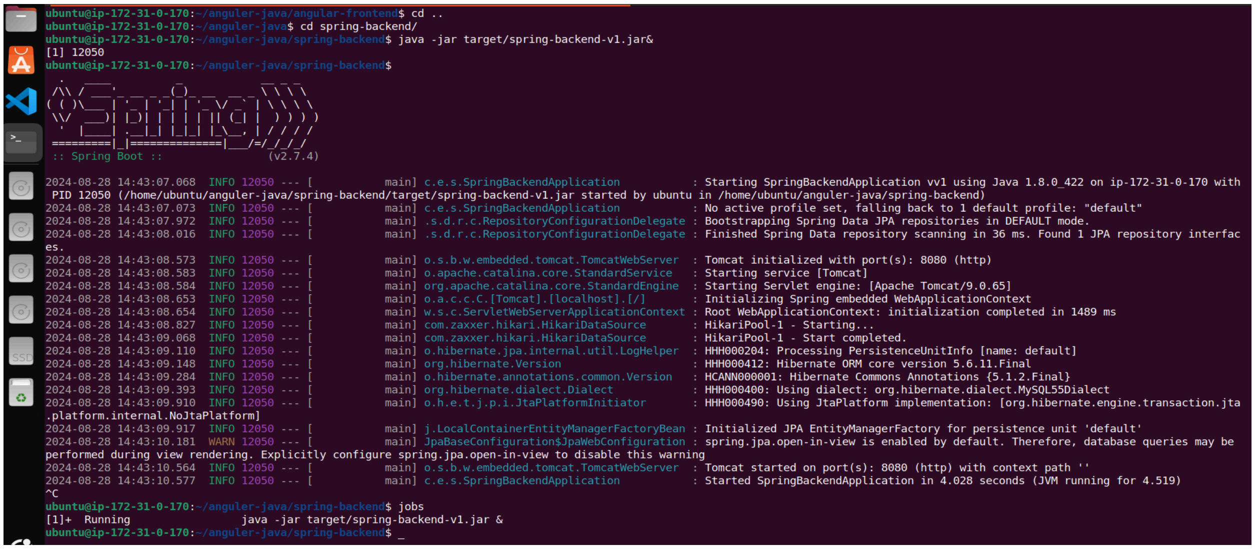
```
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ npm install
(█████████░░░░░░░░░) ⠏ reify:@esbuild/linux-loong64: timing reifyNode:node_modules/esbuild-android-64 Completed in 167ms
```

Step 12: Now run the ng build command it will create a dist/angular-frontend in the directory of angular-java/angular-frontend



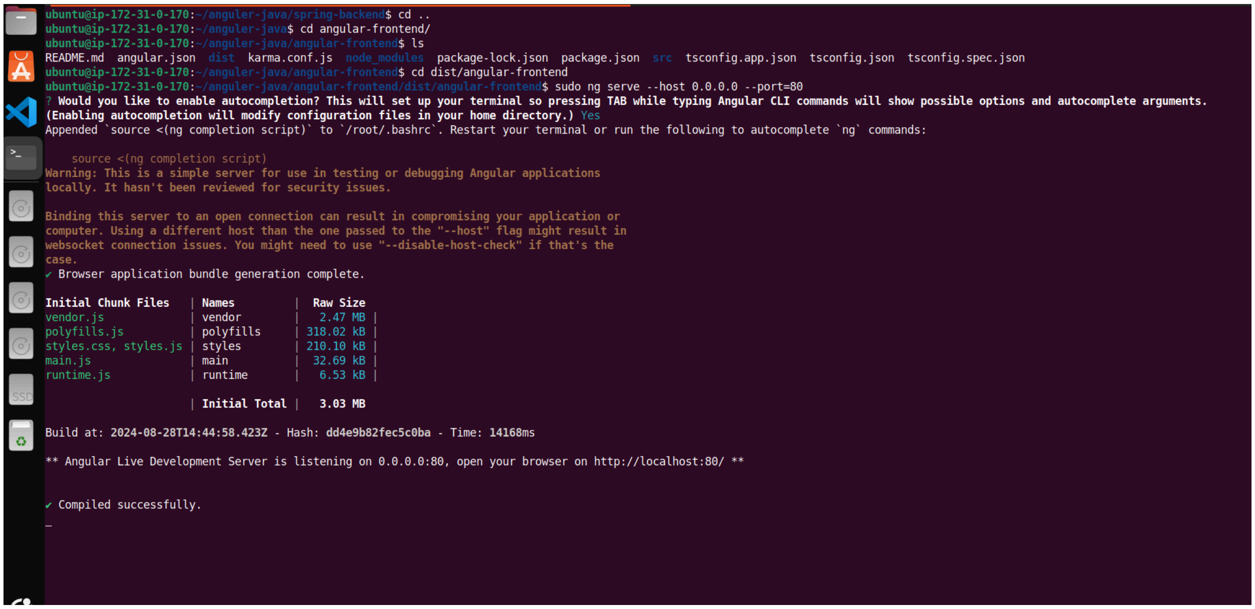
```
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ ng build
? Would you like to enable autocompletion? This will set up your terminal so pressing TAB while typing Angular CLI commands will show possible options and autocomplete arguments.
(Enabling autocompletion will modify configuration files in your home directory.) Yes
Appended `source <(ng completion script)` to `/home/ubuntu/.bashrc`. Restart your terminal or run the following to autocomplete `ng` commands:

    source <(ng completion script)
? Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.io/analytics. No
Global setting: not set
Local setting: disabled
Effective status: disabled
✔ Browser application bundle generation complete.
✔ Copying assets complete.
✔ Index html generation complete.

Initial Chunk Files           | Names         |  Raw Size | Estimated Transfer Size
main.71a5ed4870bed6f5.js      | main          | 237.75 kB |                61.82 kB
polyfills.b525ededa71d3b7f.js | polyfills     |  33.08 kB |                10.63 kB
runtime.e411e20b75d2e1de.js   | runtime       |   1.06 kB |               607 bytes
styles.ef46db3751d8e999.css   | styles        |   0 bytes |                       -

                              | Initial Total | 271.89 kB |                73.04 kB

Build at: 2024-08-28T14:42:36.198Z - Hash: 28dd54d0cb0279c4 - Time: 20687ms
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ _
```

Step 13: Now go back to spring-backend directory and run the following command in background.
java -jar target/spring-backend-v1.jar &



```
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ cd ..
ubuntu@ip-172-31-0-170:~/anguler-java$ cd spring-backend/
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend$ java -jar target/spring-backend-v1.jar&
[1] 12050
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend$
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.4)

2024-08-28 14:43:07.068  INFO 12050 --- [           main] c.e.s.SpringBackendApplication           : Starting SpringBackendApplication vv1 using Java 1.8.0_422 on ip-172-31-0-170 with
 PID 12050 (/home/ubuntu/anguler-java/spring-backend/target/spring-backend-v1.jar started by ubuntu in /home/ubuntu/anguler-java/spring-backend)
2024-08-28 14:43:07.073  INFO 12050 --- [           main] c.e.s.SpringBackendApplication           : No active profile set, falling back to 1 default profile: "default"
2024-08-28 14:43:07.972  INFO 12050 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-08-28 14:43:08.016  INFO 12050 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 36 ms. Found 1 JPA repository interfac
es.
2024-08-28 14:43:08.573  INFO 12050 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2024-08-28 14:43:08.583  INFO 12050 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-08-28 14:43:08.584  INFO 12050 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.65]
2024-08-28 14:43:08.653  INFO 12050 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-08-28 14:43:08.654  INFO 12050 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1489 ms
2024-08-28 14:43:08.827  INFO 12050 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-08-28 14:43:09.068  INFO 12050 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-08-28 14:43:09.110  INFO 12050 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-08-28 14:43:09.148  INFO 12050 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.6.11.Final
2024-08-28 14:43:09.284  INFO 12050 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2024-08-28 14:43:09.393  INFO 12050 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL55Dialect
2024-08-28 14:43:09.910  INFO 12050 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta
.platform.internal.NoJtaPlatform]
2024-08-28 14:43:09.917  INFO 12050 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-08-28 14:43:10.181  WARN 12050 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be
performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2024-08-28 14:43:10.564  INFO 12050 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2024-08-28 14:43:10.577  INFO 12050 --- [           main] c.e.s.SpringBackendApplication           : Started SpringBackendApplication in 4.028 seconds (JVM running for 4.519)
^C
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend$ jobs
[1]+  Running                 java -jar target/spring-backend-v1.jar &
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend$ _
```

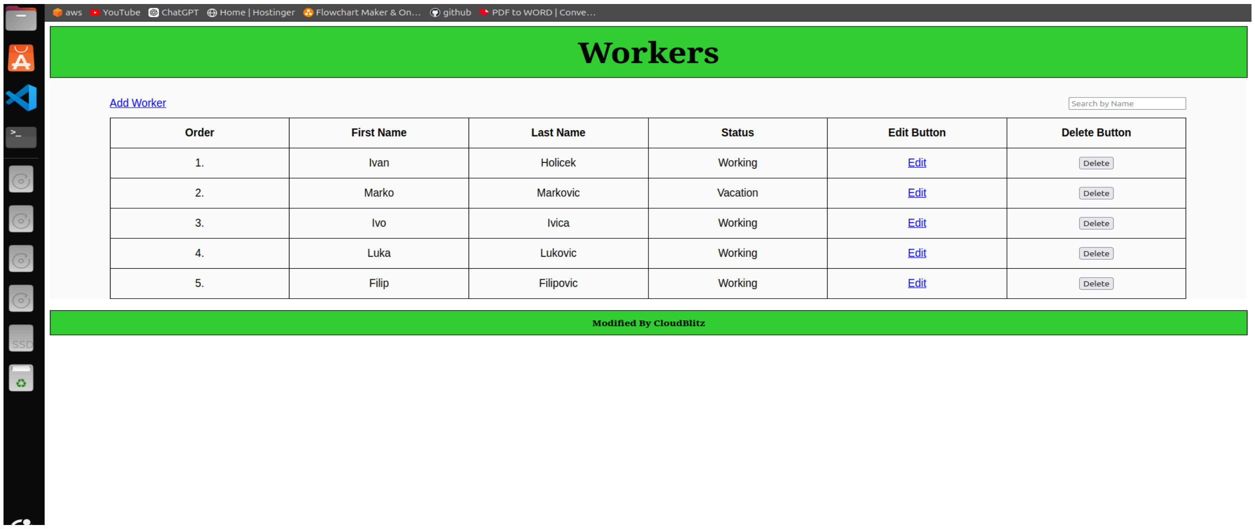
Step 14: After this go to the newly created directory dist/angular-frontend in the angular-java/angular-frontend And run the sudo ng serve --host 0.0.0.0 --port=80 command.



```
ubuntu@ip-172-31-0-170:~/anguler-java/spring-backend$ cd ..
ubuntu@ip-172-31-0-170:~/anguler-java$ cd angular-frontend/
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ ls
README.md  angular.json  dist  karma.conf.js  node_modules  package-lock.json  package.json  src  tsconfig.app.json  tsconfig.json  tsconfig.spec.json
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend$ cd dist/angular-frontend
ubuntu@ip-172-31-0-170:~/anguler-java/angular-frontend/dist/angular-frontend$ sudo ng serve --host 0.0.0.0 --port=80
? Would you like to enable autocompletion? This will set up your terminal so pressing TAB while typing Angular CLI commands will show possible options and autocomplete arguments.
(Enabling autocompletion will modify configuration files in your home directory.) Yes
Appended `source <(ng completion script)` to `/root/.bashrc`. Restart your terminal or run the following to autocomplete `ng` commands:

    source <(ng completion script)
Warning: This is a simple server for use in testing or debugging Angular applications
locally. It hasn't been reviewed for security issues.

Binding this server to an open connection can result in compromising your application or
computer. Using a different host than the one passed to the "--host" flag might result in
websocket connection issues. You might need to use "--disable-host-check" if that's the
case.
✓ Browser application bundle generation complete.

Initial Chunk Files    | Names         |  Raw Size
vendor.js              | vendor        |   2.47 MB |
polyfills.js           | polyfills     | 318.02 kB |
styles.css, styles.js  | styles        | 210.10 kB |
main.js                | main          |  32.69 kB |
runtime.js             | runtime       |   6.53 kB |

                       | Initial Total |   3.03 MB

Build at: 2024-08-28T14:44:58.423Z - Hash: dd4e9b82fec5c0ba - Time: 14168ms

** Angular Live Development Server is listening on 0.0.0.0:80, open your browser on http://localhost:80/ **


✓ Compiled successfully.
_
```

Step 15 : Now You can paste the public ip in the browser and see the output.

Output:



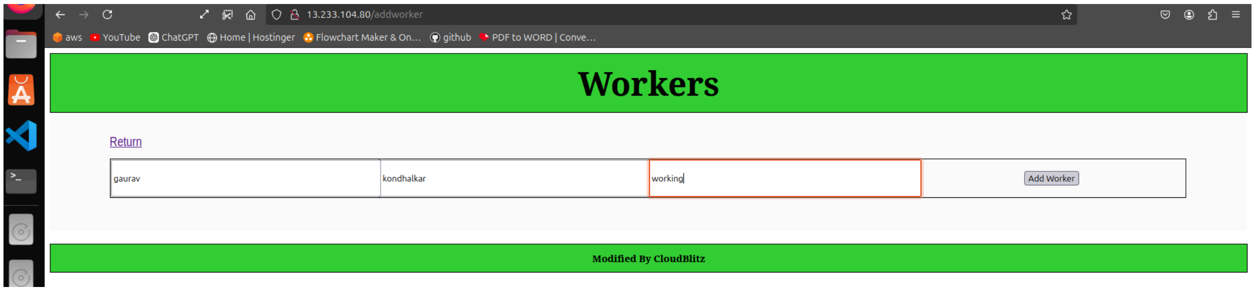
Workers

Add Worker

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

Modified By CloudBlitz

You can add worker and see the result as bellow.

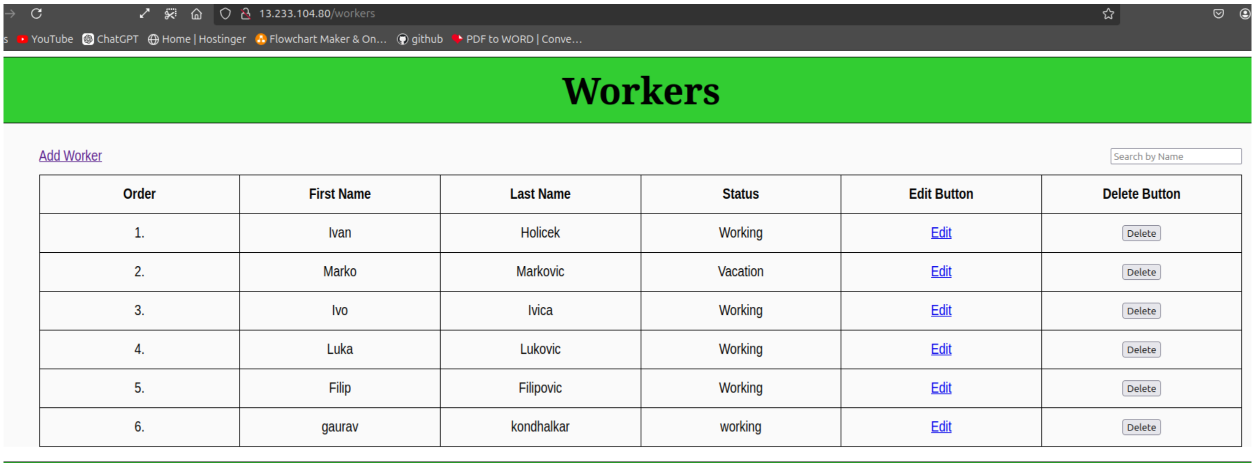


Workers

Return

gaurav	kondhalkar	working	Add Worker

Modified By CloudBlitz



Workers

Add Worker

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
6.	gaurav	kondhalkar	working	Edit	Delete