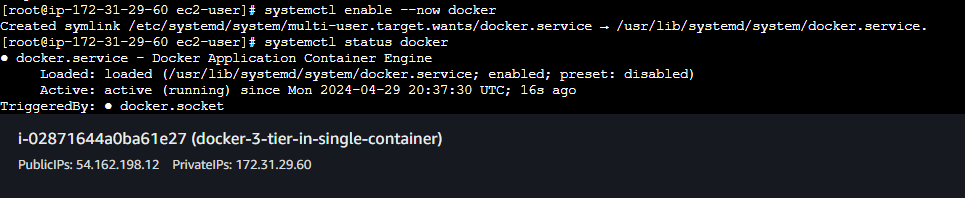
**Docker 3 tier Project**

**Single Container Manually**

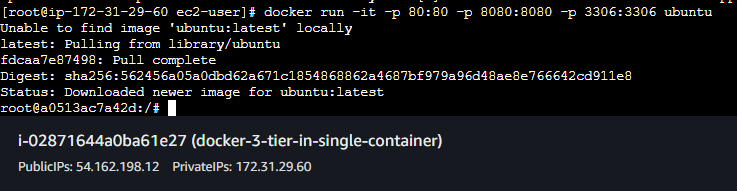
* Launch an instance with default AMI.



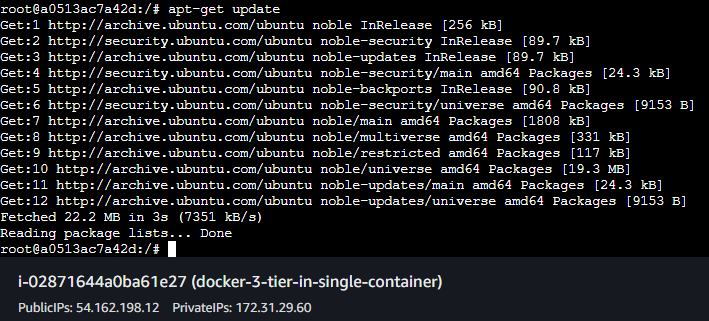
* Login using root user.
* Hit “**sudo su**”.
* yum update
* Now we need to install docker.
* Hit command “**yum install docker -y**”.
* Now when the docker is installed we need to start and enable the service.
* Hit command “**systemctl enable –now docker**” this will start and enable the docker.
* We can check the status of the docker using command “**systemctl status docker**”.



* Now for the container we will be using the ubuntu image.
* Hit command “**docker run -it -p 80:80 -p 8080:8080 -p 3306:3306 ubuntu**” this command will find the ubuntu image locally and if not present will pull the image from the dockerhub and will expose the ports specified and will enter the container.
* We are directly entering the container because if we run the docker image the container exits as no process is running in the container.
* And if we try to start it later then also it exits.
* Using the above command we will directly enter the container.



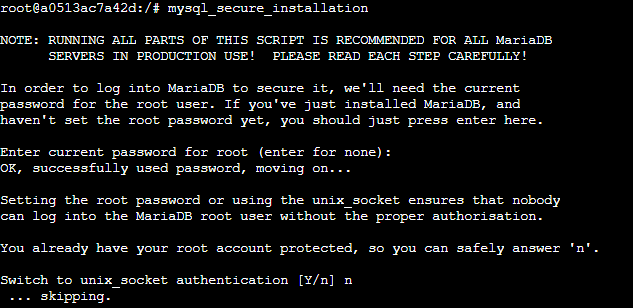
* Now we need to update and upgrade the packages.
* Hit command “**apt-get update**” to update.



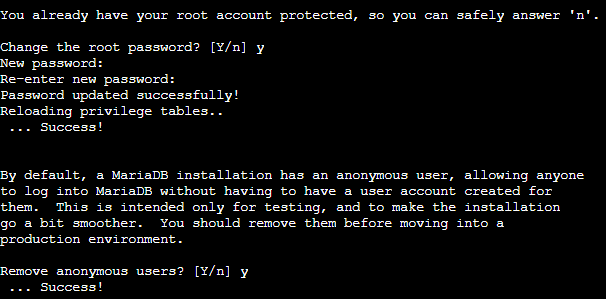
* Now we need to install the mariadb-server for our database.
* Hit command “**apt-get install mariadb-server -y**” to install the mariadb server.
* Now the mariadb-server is installed we need to start the service.
* Here systemctl will not work so we have to use the service command.
* Hit command “**service mariadb start**” to start the mariadb-service.



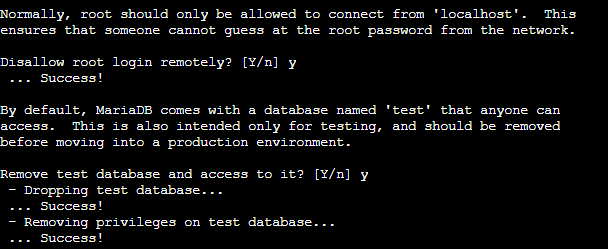
* Now we have started the mariadb service.
* We need to create a database for that we need to login but we don’t know the password.
* To set the password we need a secure installation of mysql.
* Hit command “**mysql\_secure\_installation**” to install mysql securely this will ask for the password simply hit enter.
* Now it will ask switch to unix\_socket authentication hit n.



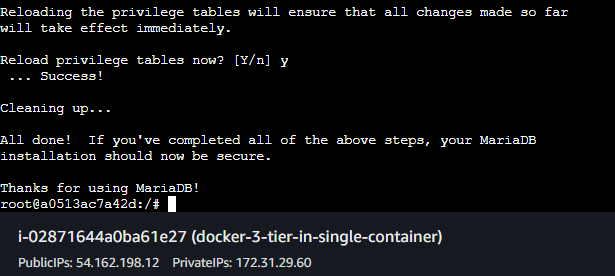
* Now it will ask if you want to change the root password hit y and enter the password you want.
* Also it will ask to remove anonymous users hit y.



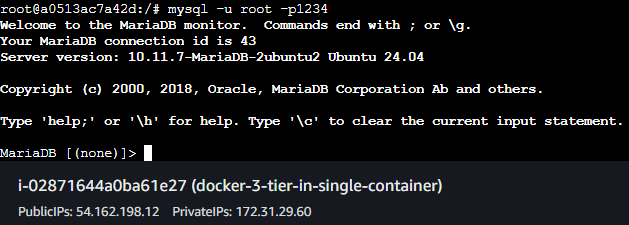
* Now it will prompt Disallow root login remotely hit y.
* Also, it will ask to remove the test database and access to it here also hit y.



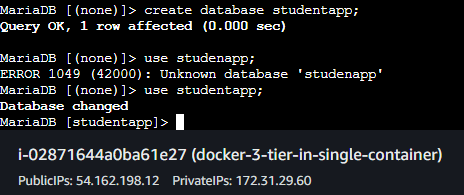
* Now it will ask to reload privilege tables now hit y.
* All done, now we can login in the mysql using password and create database and tables.



* Hit command “**mysql -u root -p1234**” to login in the mysql.
* Here in the above command I have set the password as 1234.



* Now we need to create database;
* Hit command “**create database studentapp**;” to create a database named studentapp.
* Now we have to create table inside studentapp so we have to use it.
* Hit command “**use studentapp**;” to use the database.



* Now we need a schema for a table i.e structure of the table.
* Hit the command below to create a table.

**CREATE TABLE if not exists students(student\_id INT NOT NULL AUTO\_INCREMENT,**

**student\_name VARCHAR(100) NOT NULL,**

**student\_addr VARCHAR(100) NOT NULL,**

**student\_age VARCHAR(3) NOT NULL,**

**student\_qual VARCHAR(20) NOT NULL,**

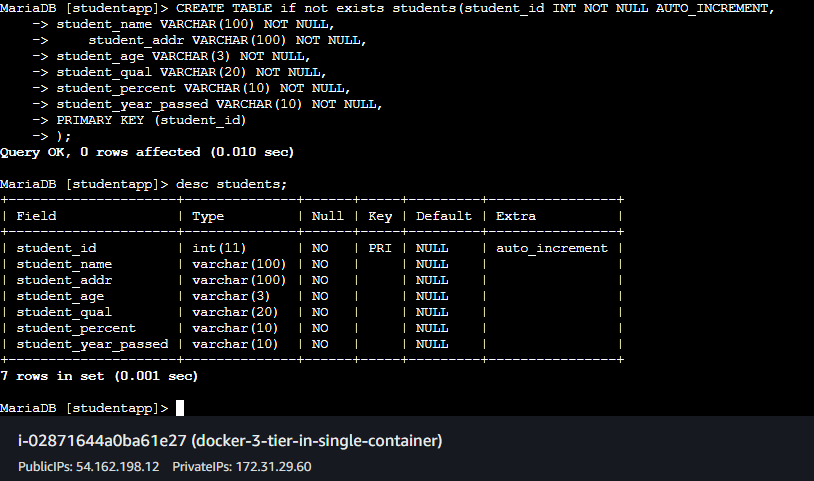
**student\_percent VARCHAR(10) NOT NULL,**

**student\_year\_passed VARCHAR(10) NOT NULL,**

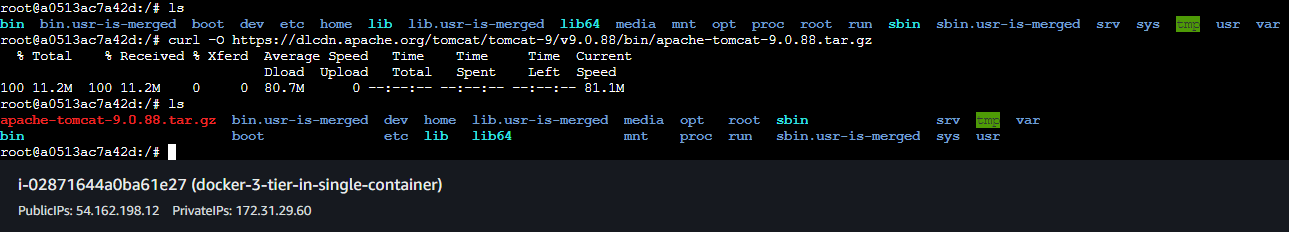
**PRIMARY KEY (student\_id)**

**);**

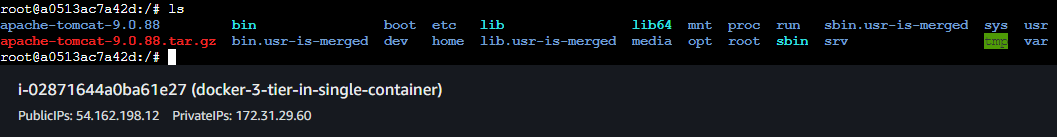
* After creating a table we can view it by using command “**desc students**;”
* This command will show the structure of the table.



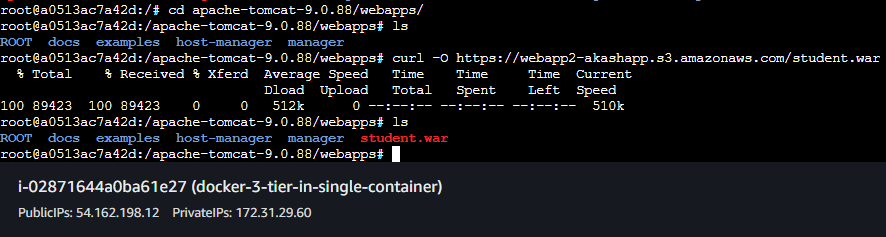
* Now our work with the database is done.
* Exit from the database simply type exit and hit enter to exit.
* Now we need tomcat.
* We can download the tomcat using curl command.
* But we have to first download the curl command.
* Hit command “**apt-get install curl -y**” to download the curl command.
* Hit command “**curl -O** [**https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.88/bin/apache-tomcat-9.0.88.tar.gz**](https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.88/bin/apache-tomcat-9.0.88.tar.gz)” this will download the tomcat tar file.



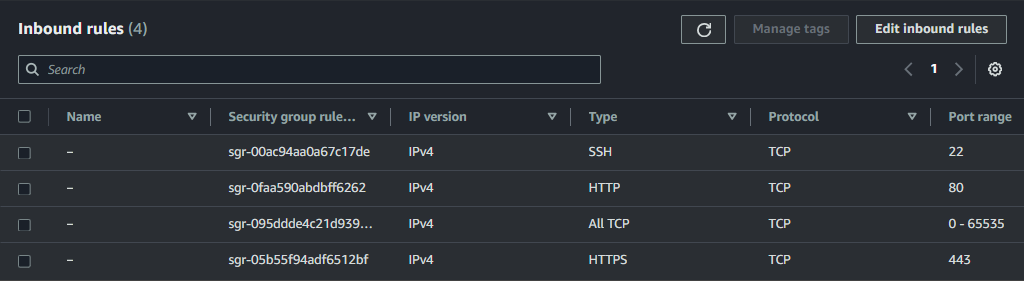
* Tomcat downloaded.
* Now we have to extract the tar file.
* Hit command “**tar -xvf apache-tomcat-9.0.88.tar.gz**” this will extract the tar file.
* Our tar file is extracted.



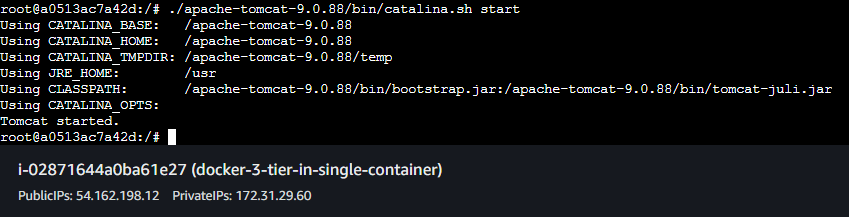
* Now go inside the webapps using cd command.
* Now we need our application here.
* I have already added the application in the S3 bucket we will use the link to download the application.
* Hit command “**curl -O** [**https://webapp2-akashapp.s3.amazonaws.com/student.war**](https://webapp2-akashapp.s3.amazonaws.com/student.war)” this will download the student.war file.



* Now we need to modify the security group of the instance and Add the inbound rule here we will Add Rule for ALL TCP.



* Now come back in the instance and change the directory to /.
* Now we need to start the Catalina.sh and for that we need a java environmet.
* Hit command “**apt-get install openjdk-8-jdk -y**” to install java environment.
* Now our environment is ready we can start the tomcat service which is Catalina.sh.
* Hit command **“./apache-tomcat-9.0.88/bin/catalina.sh start**” to start the service.
* Tomcat started successfully.



* Now copy the instance public IP and hit it in the incognito mode with port 8080/student because the name of our application is student.

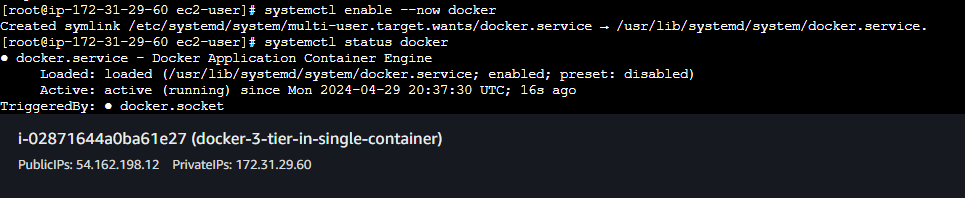
**Docker 3 tier Project**

**Single Container Manually**

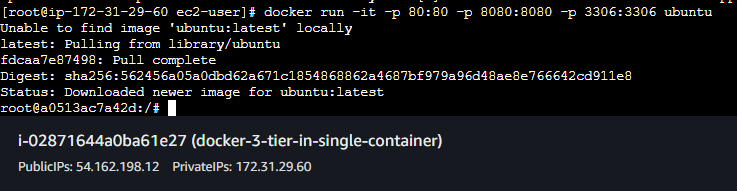
* Launch an instance with default AMI.



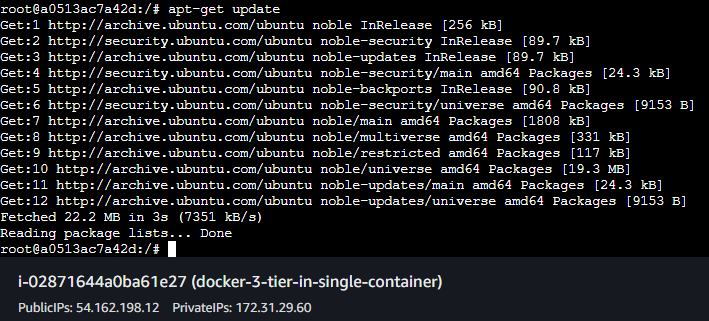
* Login using root user.
* Hit “**sudo su**”.
* Now we need to install docker.
* Hit command “**yum install docker -y**”.
* Now when the docker is installed we need to start and enable the service.
* Hit command “**systemctl enable –now docker**” this will start and enable the docker.
* We can check the status of the docker using command “**systemctl status docker**”.



* Now for the container we will be using the ubuntu image.
* Hit command “**docker run -it -p 80:80 -p 8080:8080 -p 3306:3306 ubuntu**” this command will find the ubuntu image locally and if not present will pull the image from the dockerhub and will expose the ports specified and will enter the container.
* We are directly entering the container because if we run the docker image the container exits as no process is running in the container.
* And if we try to start it later then also it exits.
* Using the above command we will directly enter the container.



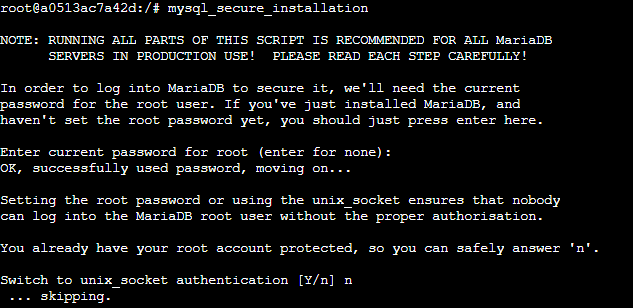
* Now we need to update and upgrade the packages.
* Hit command “**apt-get update**” to update.



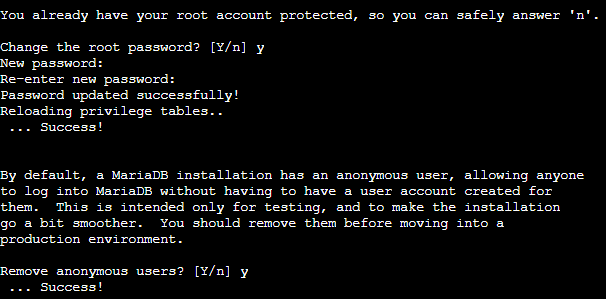
* Now we need to install the mariadb-server for our database.
* Hit command “**apt-get install mariadb-server -y**” to install the mariadb server.
* Now the mariadb-server is installed we need to start the service.
* Here systemctl will not work so we have to use the service command.
* Hit command “**service mariadb start**” to start the mariadb-service.



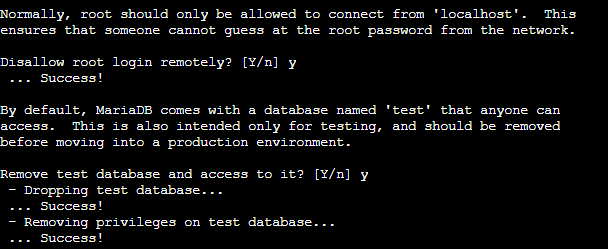
* Now we have started the mariadb service.
* We need to create a database for that we need to login but we don’t know the password.
* To set the password we need a secure installation of mysql.
* Hit command “**mysql\_secure\_installation**” to install mysql securely this will ask for the password simply hit enter.
* Now it will ask switch to unix\_socket authentication hit n.



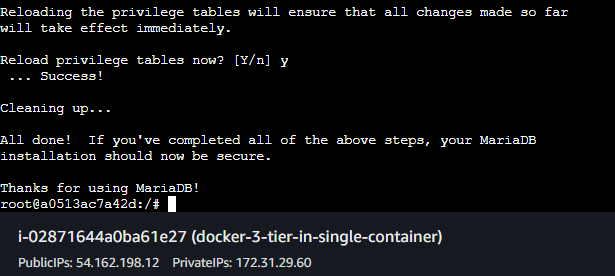
* Now it will ask if you want to change the root password hit y and enter the password you want.
* Also it will ask to remove anonymous users hit y.



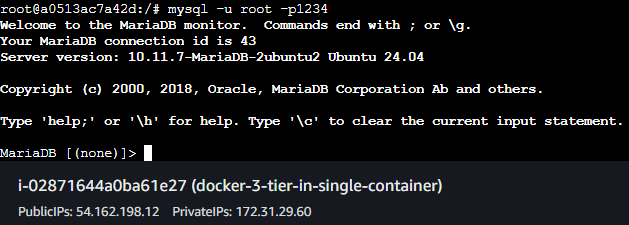
* Now it will prompt Disallow root login remotely hit y.
* Also, it will ask to remove the test database and access to it here also hit y.



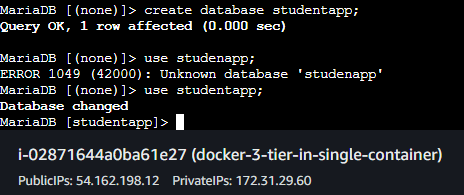
* Now it will ask to reload privilege tables now hit y.
* All done, now we can login in the mysql using password and create database and tables.



* Hit command “**mysql -u root -p1234**” to login in the mysql.
* Here in the above command I have set the password as 1234.



* Now we need to create database;
* Hit command “**create database studentapp**;” to create a database named studentapp.
* Now we have to create table inside studentapp so we have to use it.
* Hit command “**use studentapp**;” to use the database.



* Now we need a schema for a table i.e structure of the table.
* Hit the command below to create a table.

**CREATE TABLE if not exists students(student\_id INT NOT NULL AUTO\_INCREMENT,**

**student\_name VARCHAR(100) NOT NULL,**

**student\_addr VARCHAR(100) NOT NULL,**

**student\_age VARCHAR(3) NOT NULL,**

**student\_qual VARCHAR(20) NOT NULL,**

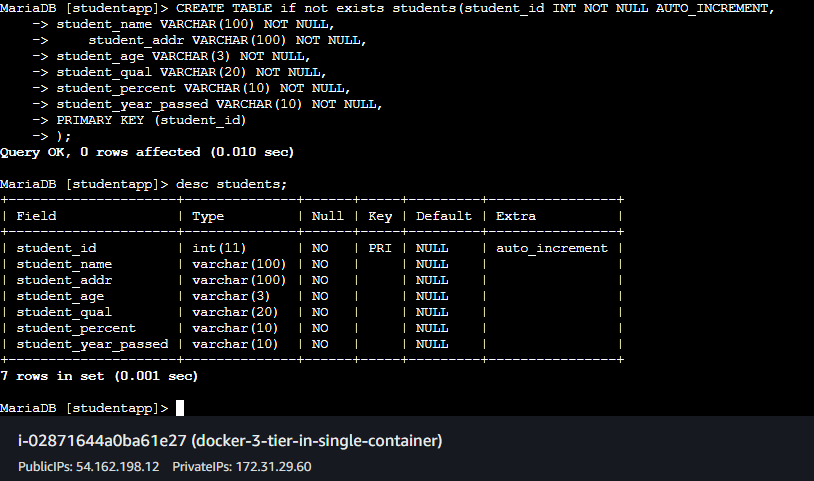
**student\_percent VARCHAR(10) NOT NULL,**

**student\_year\_passed VARCHAR(10) NOT NULL,**

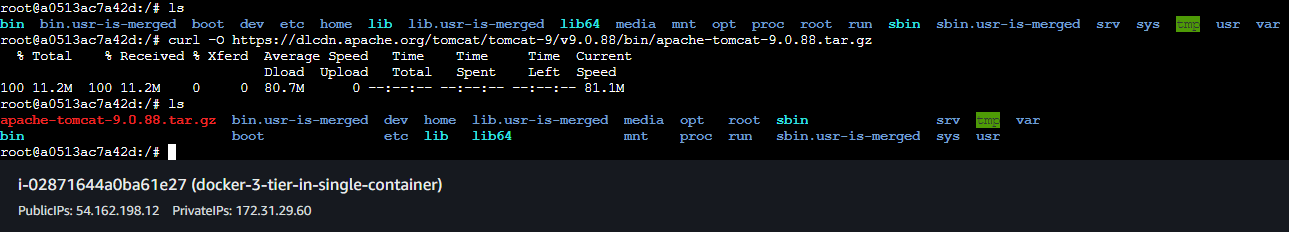
**PRIMARY KEY (student\_id)**

**);**

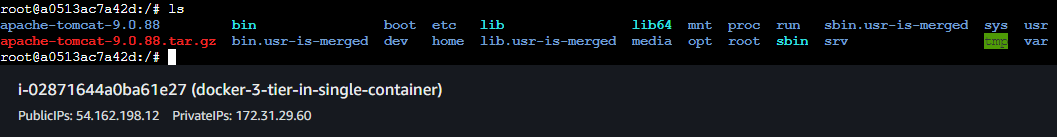
* After creating a table we can view it by using command “**desc students**;”
* This command will show the structure of the table.



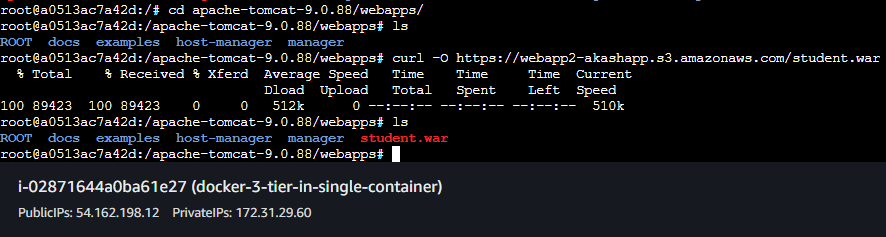
* Now our work with the database is done.
* Exit from the database simply type exit and hit enter to exit.
* Now we need tomcat.
* We can download the tomcat using curl command.
* But we have to first download the curl command.
* Hit command “**apt-get install curl -y**” to download the curl command.
* Hit command “**curl -O** [**https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.88/bin/apache-tomcat-9.0.88.tar.gz**](https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.88/bin/apache-tomcat-9.0.88.tar.gz)” this will download the tomcat tar file.



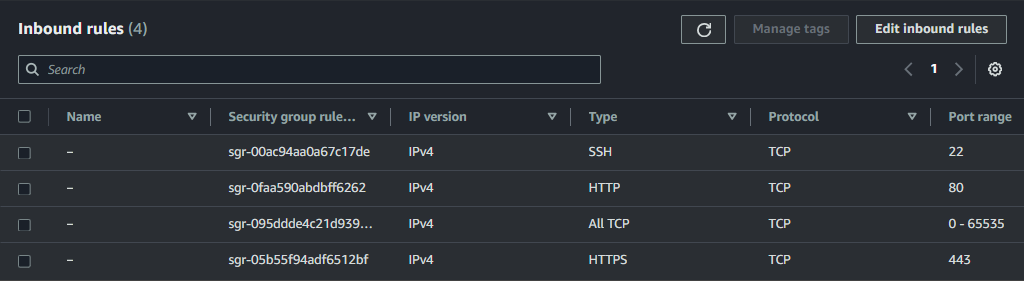
* Tomcat downloaded.
* Now we have to extract the tar file.
* Hit command “**tar -xvf apache-tomcat-9.0.88.tar.gz**” this will extract the tar file.
* Our tar file is extracted.



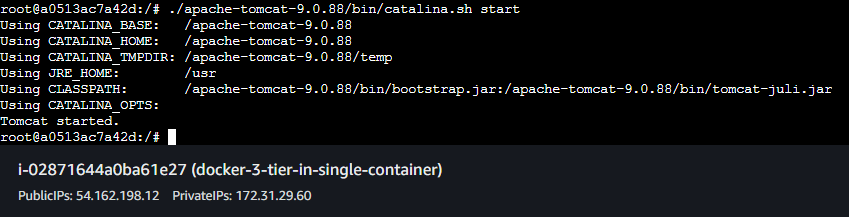
* Now go inside the webapps using cd command.
* Now we need our application here.
* I have already added the application in the S3 bucket we will use the link to download the application.
* Hit command “**curl -O** [**https://webapp2-akashapp.s3.amazonaws.com/student.war**](https://webapp2-akashapp.s3.amazonaws.com/student.war)” this will download the student.war file.



* Now we need to modify the security group of the instance and Add the inbound rule here we will Add Rule for ALL TCP.



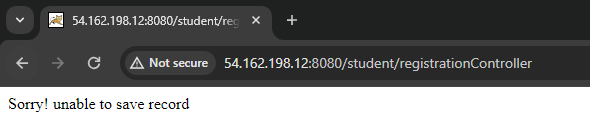
* Now come back in the instance and change the directory to /.
* Now we need to start the Catalina.sh and for that we need a java environmet.
* Hit command “**apt-get install openjdk-8-jdk -y**” to install java environment.
* Now our environment is ready we can start the tomcat service which is Catalina.sh.
* Hit command **“./apache-tomcat-9.0.88/bin/catalina.sh start**” to start the service.
* Tomcat started successfully.



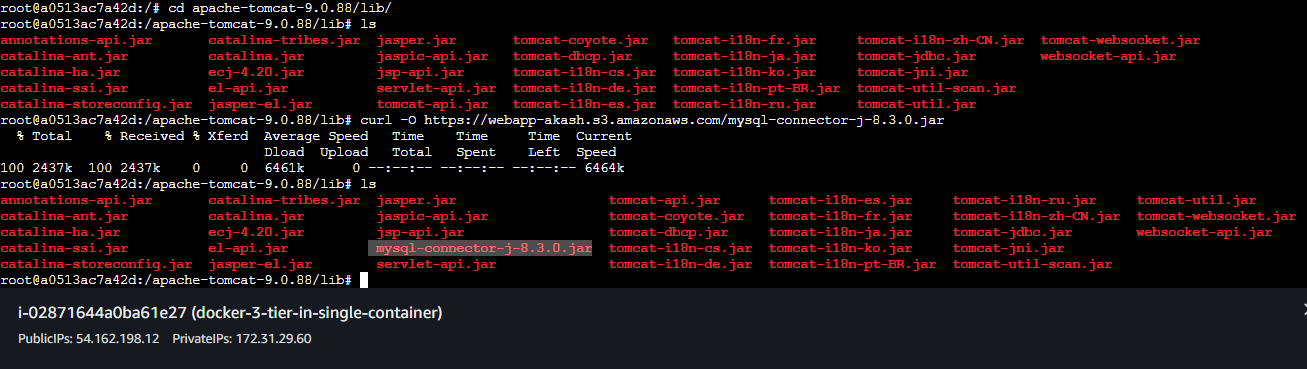
* Now copy the instance public IP and hit it in the incognito mode with port 8080/student because the name of our application is student.



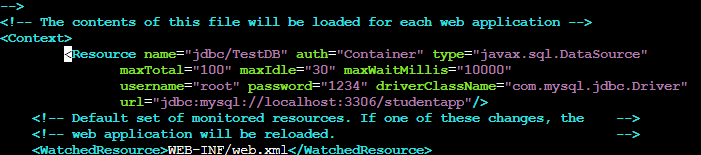
* Our page is visible but if we fill the data and hit register button then the data will not be saved.



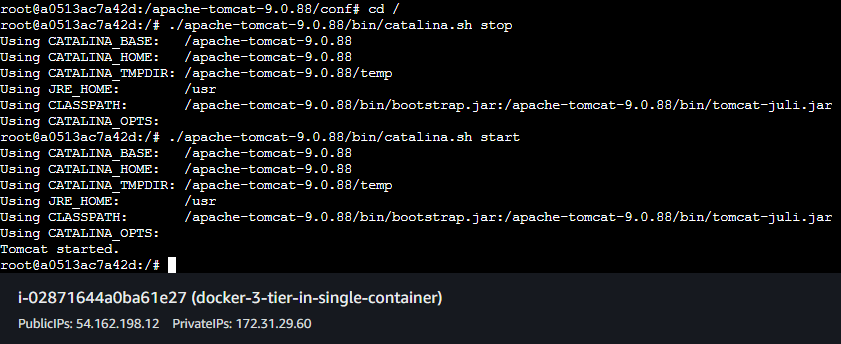
* Now we need to connect the database with the application for that we need mysql-connector.
* Browse to the lib directory in tomcat.
* Hit command “**curl -O <https://webapp-akash.s3.amazonaws.com/mysql-connector-j-8.3.0.jar>**” to download the mysql-connector file in the lib directory.



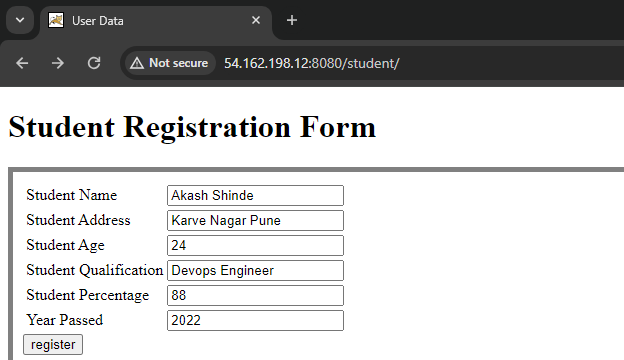
* We also need to add the configuration in conf directory in context.xml file.
* Browse to conf directory.
* Now we need editor to open the file and modify it.
* We will install the vim editor.
* Hit command “**apt install vim**” to install the vim editor.
* Hit command “**vim context.xml**” to open the file in vim editor.
* Add the configuration shown in screenshot.

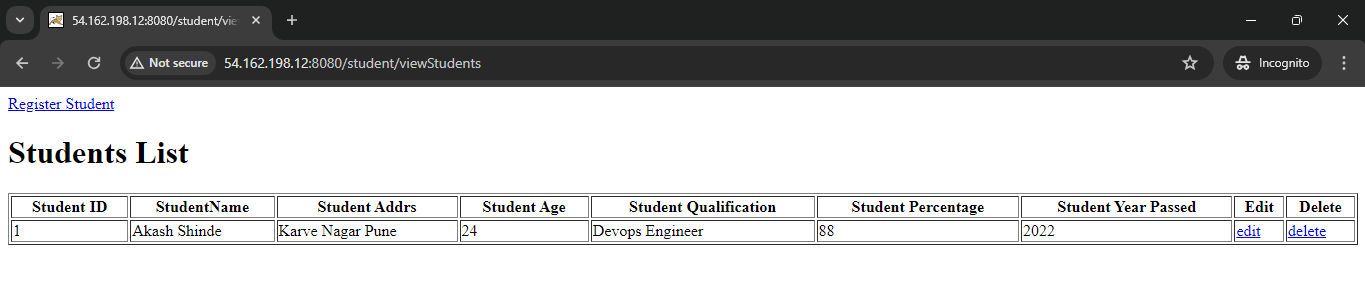


* Now save the file and exit.
* Now we need to restart the tomcat service but there is no option to restart.
* So we need to stop the service and then start it again.
* Browse back to / directory by using **cd /.**
* Hit command **“./apache-tomcat-9.0.88/bin/catalina.sh stop**” to stop the service.
* Hit command **“./apache-tomcat-9.0.88/bin/catalina.sh start**” to start the service.

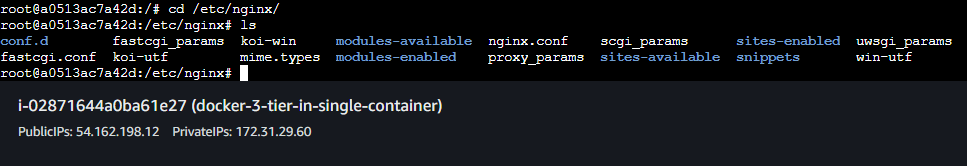


* Now reload the incognito mode in which you have opened the webapp.
* Now try to fill the data and hit register.





* Our data is saving successfully in the database.
* Now we need to add proxy.
* Hit command “**apt-get install nginx -y**” to install nginx for our proxy.
* After installing nginx we need to browse **to /etc/nginx directory**.



* Here we need to add the configuration in the nginx.conf.
* Hit command “**vim nginx.conf**” to open the file in the vim editor.
* Add the configuration given below in http tag.

**server {**

**listen 80;**

**server\_name localhost;**

**location / {**

**proxy\_pass http://localhost:8080; # Forward requests to your application running on Tomcat**

**proxy\_set\_header Host $host;**

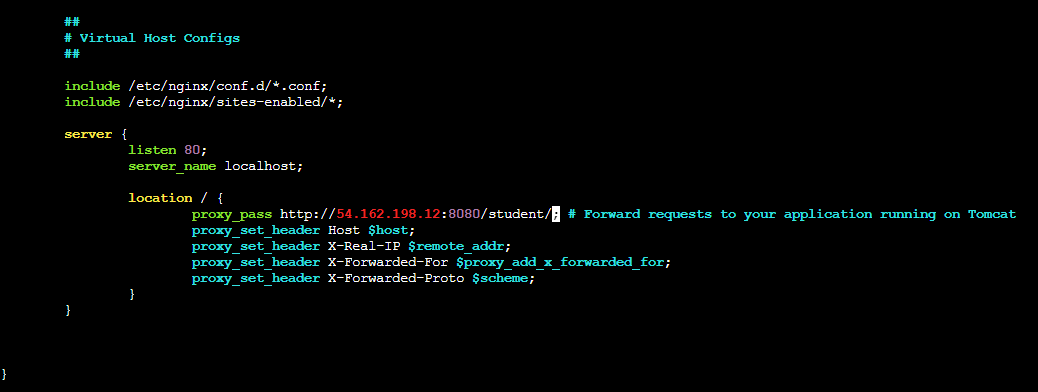
**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

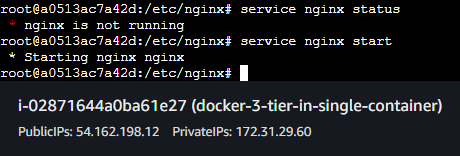
**proxy\_set\_header X-Forwarded-Proto $scheme;**

**}**

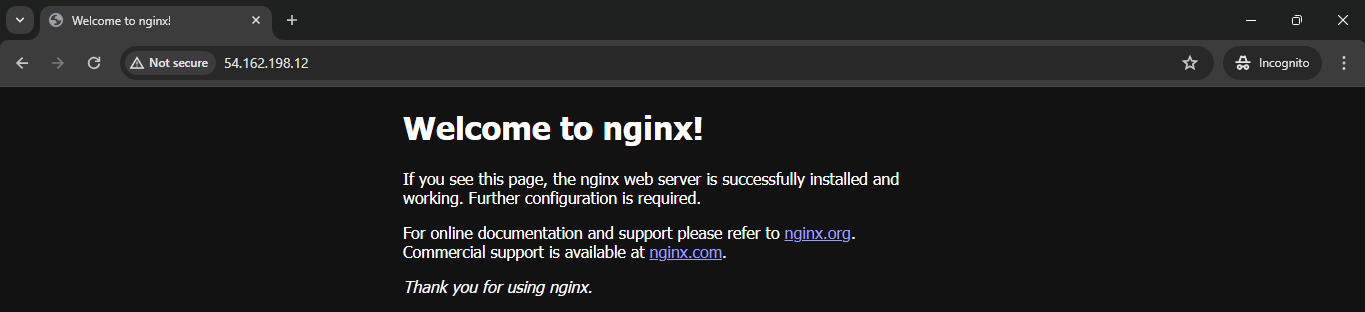
**}**



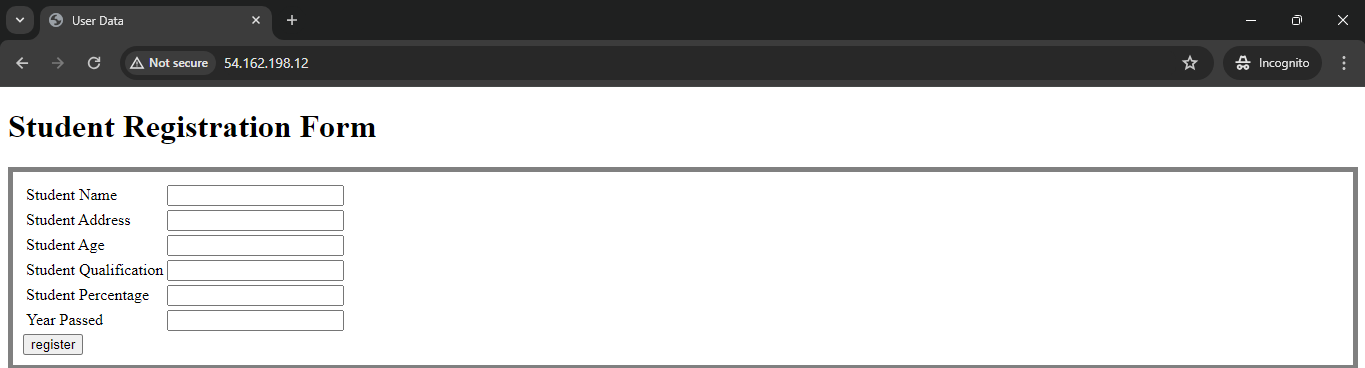
* Save the file and exit.
* Now we need to start the nginx service.
* Hit command “**service nginx start**”



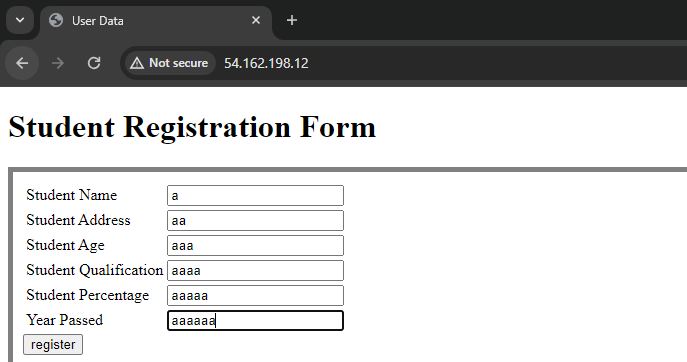
* Nginx service started.
* Now hit the IP of the instance in Incognito mode in browser.
* We are seeing the nginx page instead of tomcat page.



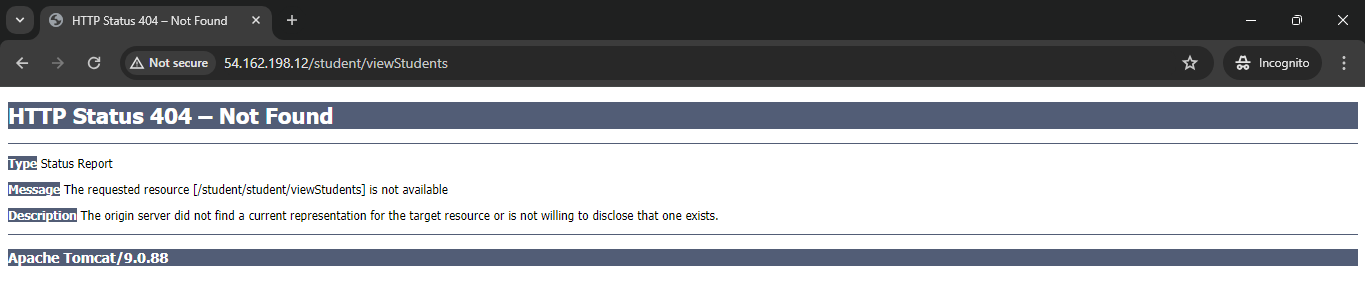
* We need to change the configuration.
* Hit command “**nginx.conf**” to modify the file.
* Remove the line “include **/etc/nginx/sites-enabled/\*;”** and then save the file and exit.
* How try to refresh the page.
* Our page is visible.



* Now try to fill the data and hit register button.



* Hit the register button.



* We got an error now again open the nginx.conf file.
* Remove all the configurations from the file and add the configuration given below.

**user nginx;**

**worker\_processes 1;**

**error\_log /var/log/nginx/error.log warn;**

**pid /var/run/nginx.pid;**

**events {**

**worker\_connections 1024;**

**}**

**http {**

**include /etc/nginx/mime.types;**

**default\_type application/octet-stream;**

**log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '**

**'$status $body\_bytes\_sent "$http\_referer" '**

**'"$http\_user\_agent" "$http\_x\_forwarded\_for"';**

**access\_log /var/log/nginx/access.log main;**

**sendfile on;**

**tcp\_nopush on;**

**tcp\_nodelay on;**

**keepalive\_timeout 65;**

**types\_hash\_max\_size 2048;**

**include /etc/nginx/conf.d/\*.conf;**

**server {**

**listen 80;**

**server\_name localhost;**

**location / {**

**proxy\_pass http://Instance-IP:8080/student/;**

**}**

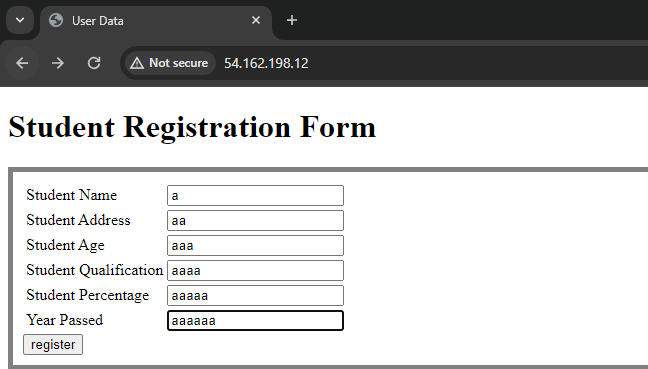
**}**

**}**

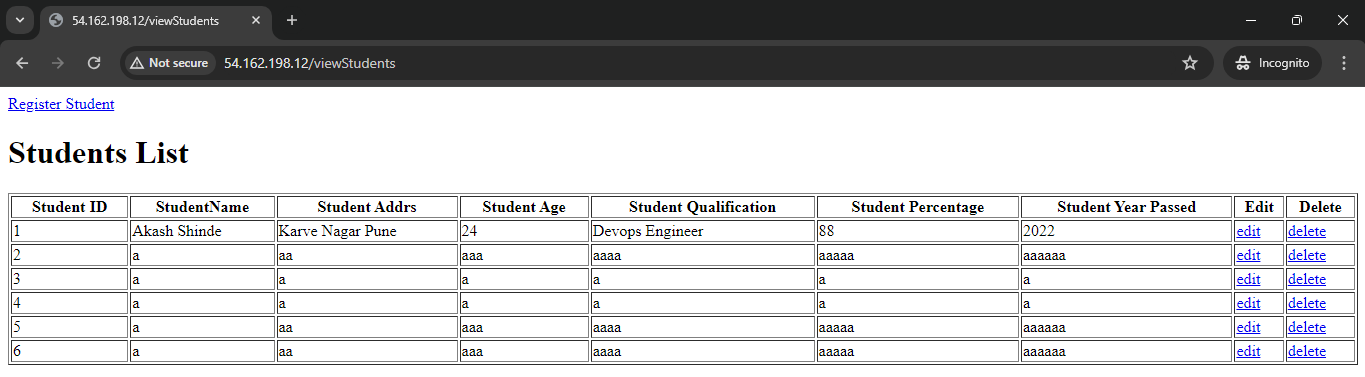
* Save the file and exit.
* Restart the nginx service.
* Hit command “**service nginx restart**” to restart the service.
* Refresh the page.
* Our page is visible



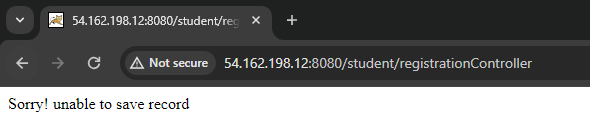
* Fill up the data.



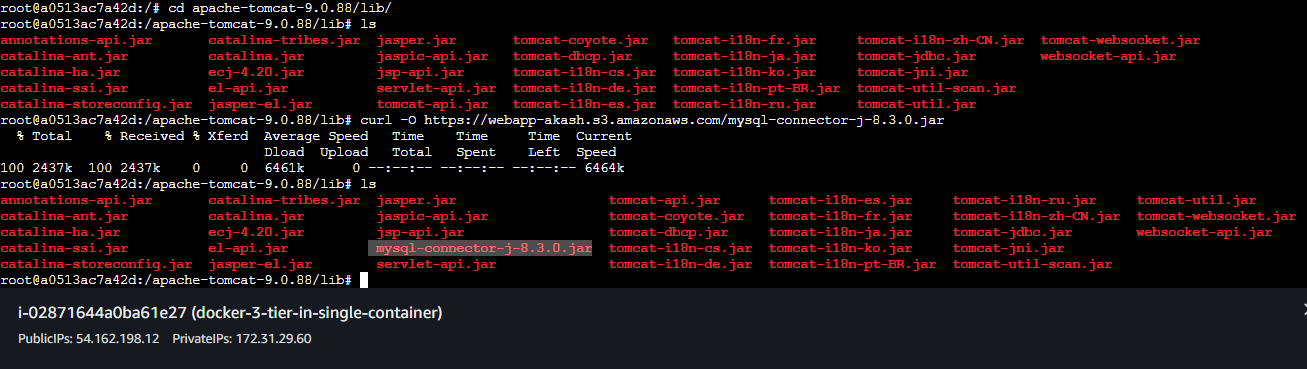
* Click register.



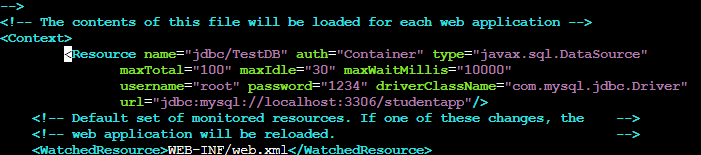
* Our data is successfully added in the database.
* Our page is visible but if we fill the data and hit register button then the data will not be saved.



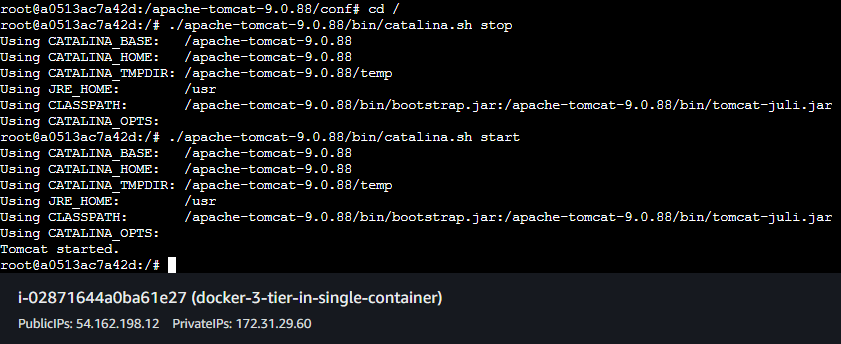
* Now we need to connect the database with the application for that we need mysql-connector.
* Browse to the lib directory in tomcat.
* Hit command “**curl -O** [**https://webapp-akash.s3.amazonaws.com/mysql-connector-j-8.3.0.jar**](https://webapp-akash.s3.amazonaws.com/mysql-connector-j-8.3.0.jar)” to download the mysql-connector file in the lib directory.



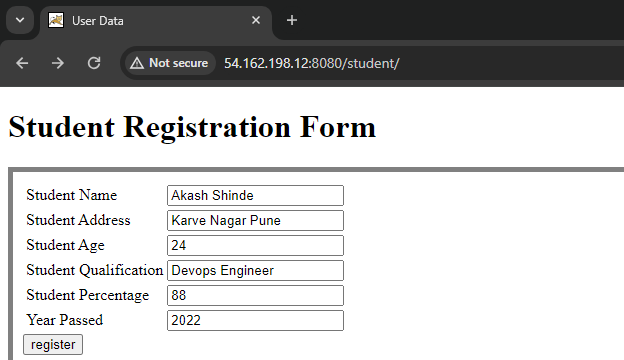
* We also need to add the configuration in conf directory in context.xml file.
* Browse to conf directory.
* Now we need editor to open the file and modify it.
* We will install the vim editor.
* Hit command “**apt install vim**” to install the vim editor.
* Hit command “**vim context.xml**” to open the file in vim editor.
* Add the configuration shown in screenshot.

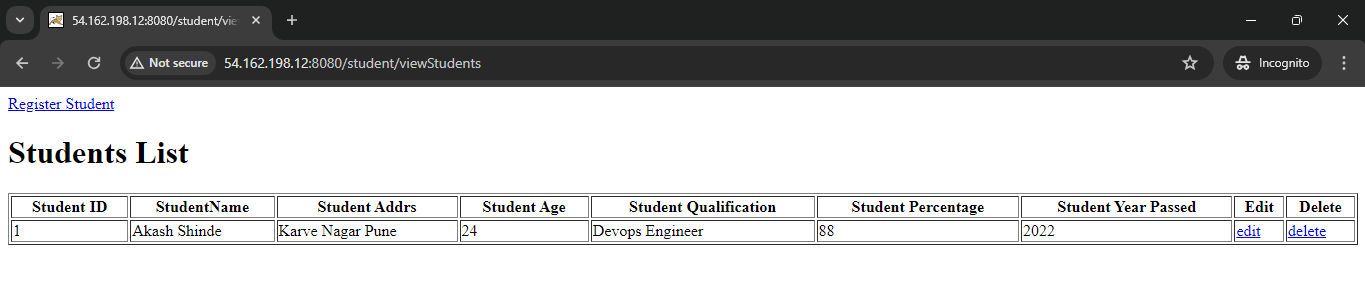


* Now save the file and exit.
* Now we need to restart the tomcat service but there is no option to restart.
* So we need to stop the service and then start it again.
* Browse back to / directory by using **cd /.**
* Hit command **“./apache-tomcat-9.0.88/bin/catalina.sh stop**” to stop the service.
* Hit command **“./apache-tomcat-9.0.88/bin/catalina.sh start**” to start the service.

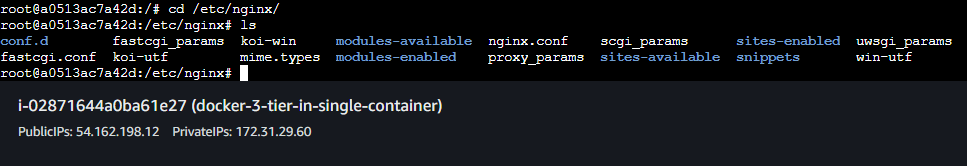


* Now reload the incognito mode in which you have opened the webapp.
* Now try to fill the data and hit register.





* Our data is saving successfully in the database.
* Now we need to add proxy.
* Hit command “**apt-get install nginx -y**” to install nginx for our proxy.
* After installing nginx we need to browse **to /etc/nginx directory**.



* Here we need to add the configuration in the nginx.conf.
* Hit command “**vim nginx.conf**” to open the file in the vim editor.
* Add the configuration given below in http tag.

**server {**

**listen 80;**

**server\_name localhost;**

**location / {**

**proxy\_pass http://localhost:8080; # Forward requests to your application running on Tomcat**

**proxy\_set\_header Host $host;**

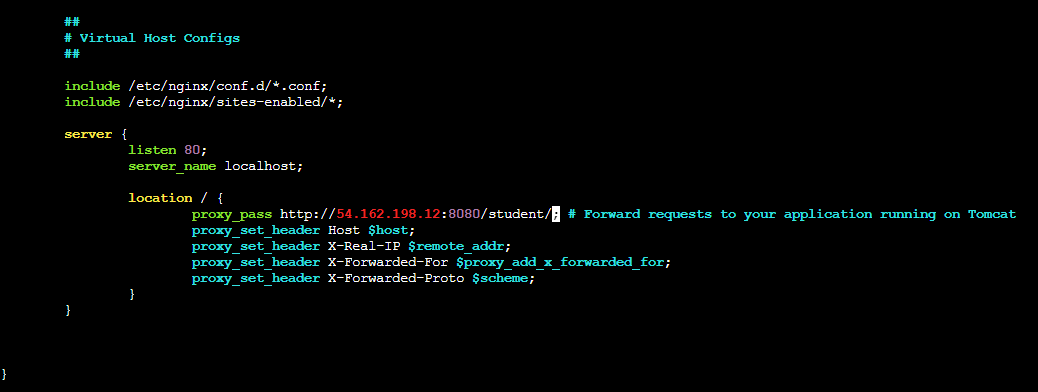
**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

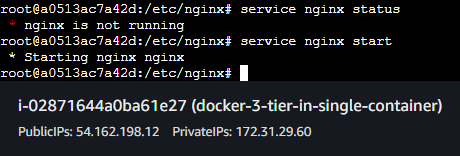
**proxy\_set\_header X-Forwarded-Proto $scheme;**

**}**

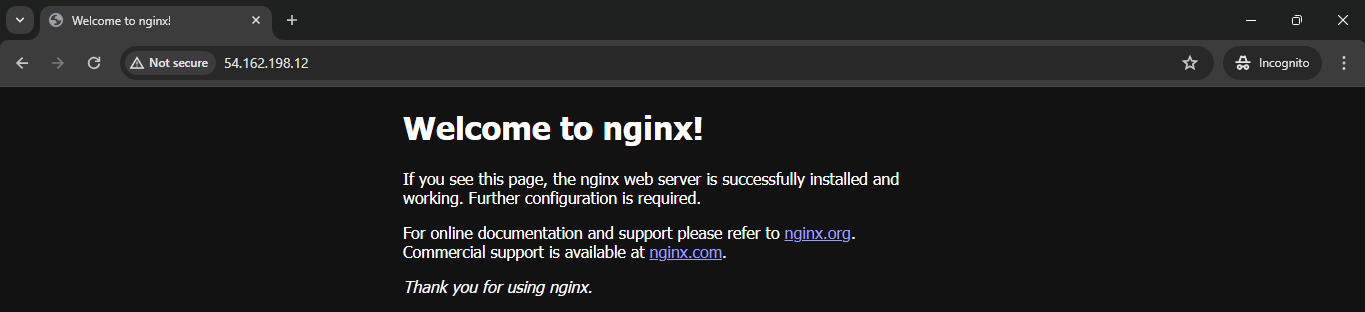
**}**



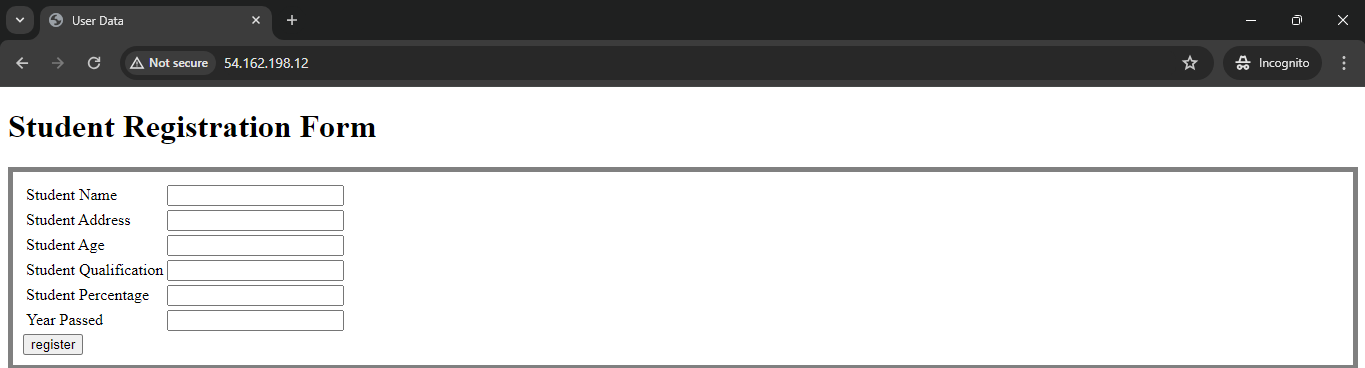
* Save the file and exit.
* Now we need to start the nginx service.
* Hit command “**service nginx start**”



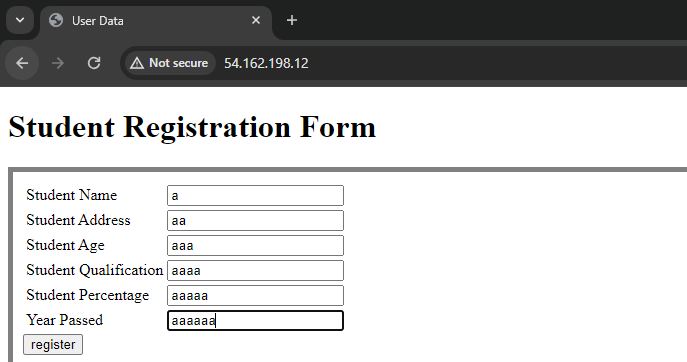
* Nginx service started.
* Now hit the IP of the instance in Incognito mode in browser.
* We are seeing the nginx page instead of tomcat page.



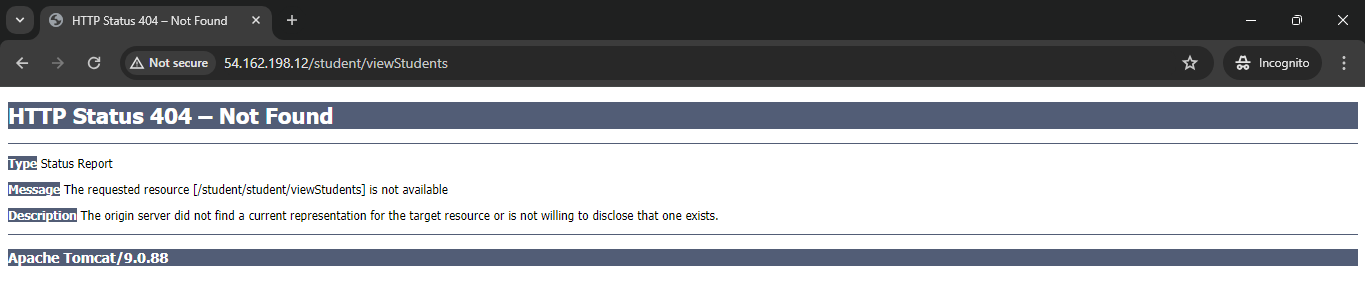
* We need to change the configuration.
* Hit command “**nginx.conf**” to modify the file.
* Remove the line “include **/etc/nginx/sites-enabled/\*;”** and then save the file and exit.
* How try to refresh the page.
* Our page is visible.



* Now try to fill the data and hit register button.



* Hit the register button.



* We got an error now again open the nginx.conf file.
* Remove all the configurations from the file and add the configuration given below.

**user nginx;**

**worker\_processes 1;**

**error\_log /var/log/nginx/error.log warn;**

**pid /var/run/nginx.pid;**

**events {**

**worker\_connections 1024;**

**}**

**http {**

**include /etc/nginx/mime.types;**

**default\_type application/octet-stream;**

**log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '**

**'$status $body\_bytes\_sent "$http\_referer" '**

**'"$http\_user\_agent" "$http\_x\_forwarded\_for"';**

**access\_log /var/log/nginx/access.log main;**

**sendfile on;**

**tcp\_nopush on;**

**tcp\_nodelay on;**

**keepalive\_timeout 65;**

**types\_hash\_max\_size 2048;**

**include /etc/nginx/conf.d/\*.conf;**

**server {**

**listen 80;**

**server\_name localhost;**

**location / {**

**proxy\_pass http://Instance-IP:8080/student/;**

**}**

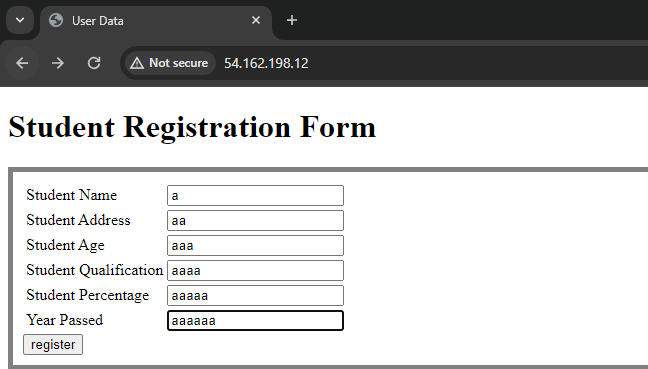
**}**

**}**

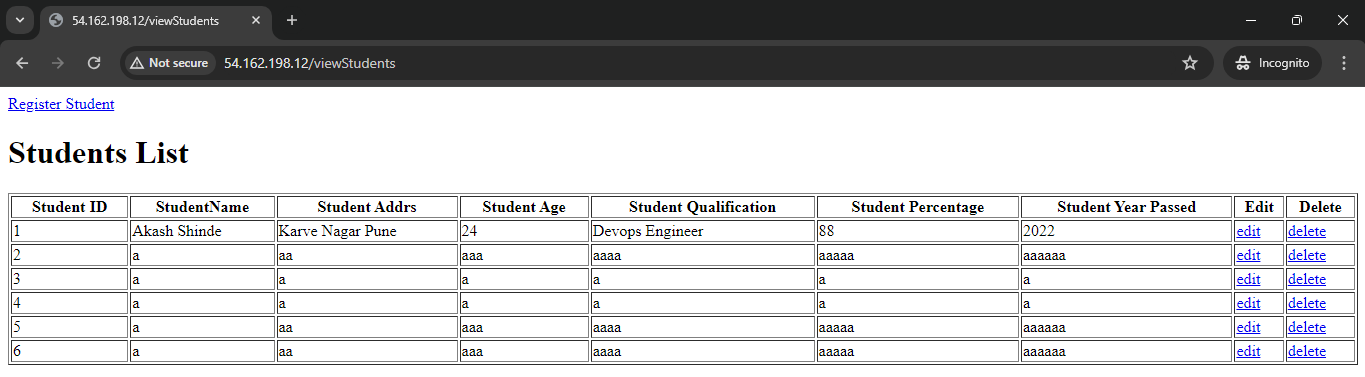
* Save the file and exit.
* Restart the nginx service.
* Hit command “**service nginx restart**” to restart the service.
* Refresh the page.
* Our page is visible



* Fill up the data.



* Click register.



* Our data is successfully added in the database.

**Dockerfile that automates the setup described:**

# Use Ubuntu as base image

FROM ubuntu:latest

# Set non-interactive mode during installation

ARG DEBIAN\_FRONTEND=noninteractive

# Update and install necessary packages

RUN apt-get update && apt-get install -y \

curl \

vim \

mariadb-server \

openjdk-8-jdk \

nginx \

&& apt-get clean \

&& rm -rf /var/lib/apt/lists/\*

# Install Docker

RUN apt-get update && apt-get install -y \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common \

&& curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add - \

&& add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" \

&& apt-get update && apt-get install -y docker-ce docker-ce-cli containerd.io \

&& apt-get clean \

&& rm -rf /var/lib/apt/lists/\*

# Expose ports

EXPOSE 80 8080 3306

# Create a directory for MySQL scripts

RUN mkdir -p /docker-entrypoint-initdb.d

# Copy MySQL scripts into the container

COPY mysql\_scripts/ /docker-entrypoint-initdb.d/

# Copy nginx configuration file

COPY nginx.conf /etc/nginx/nginx.conf

# Copy Tomcat and MySQL connectors

COPY apache-tomcat-9.0.88 /apache-tomcat-9.0.88

COPY mysql-connector-j-8.3.0.jar /apache-tomcat-9.0.88/lib/

# Start services

CMD service mysql start && /apache-tomcat-9.0.88/bin/catalina.sh run && nginx -g "daemon off;"

You'll need to organize your files accordingly:

1. Put your MySQL scripts into a directory named mysql\_scripts.
2. Place your nginx.conf, apache-tomcat-9.0.88 directory, and mysql-connector-j-8.3.0.jar file in the same directory as your Dockerfile.

This Dockerfile automates the installation and configuration process you've described, allowing you to build a Docker image that sets up the entire environment. Make sure to replace placeholders like Instance-IP with the appropriate values for your setup.