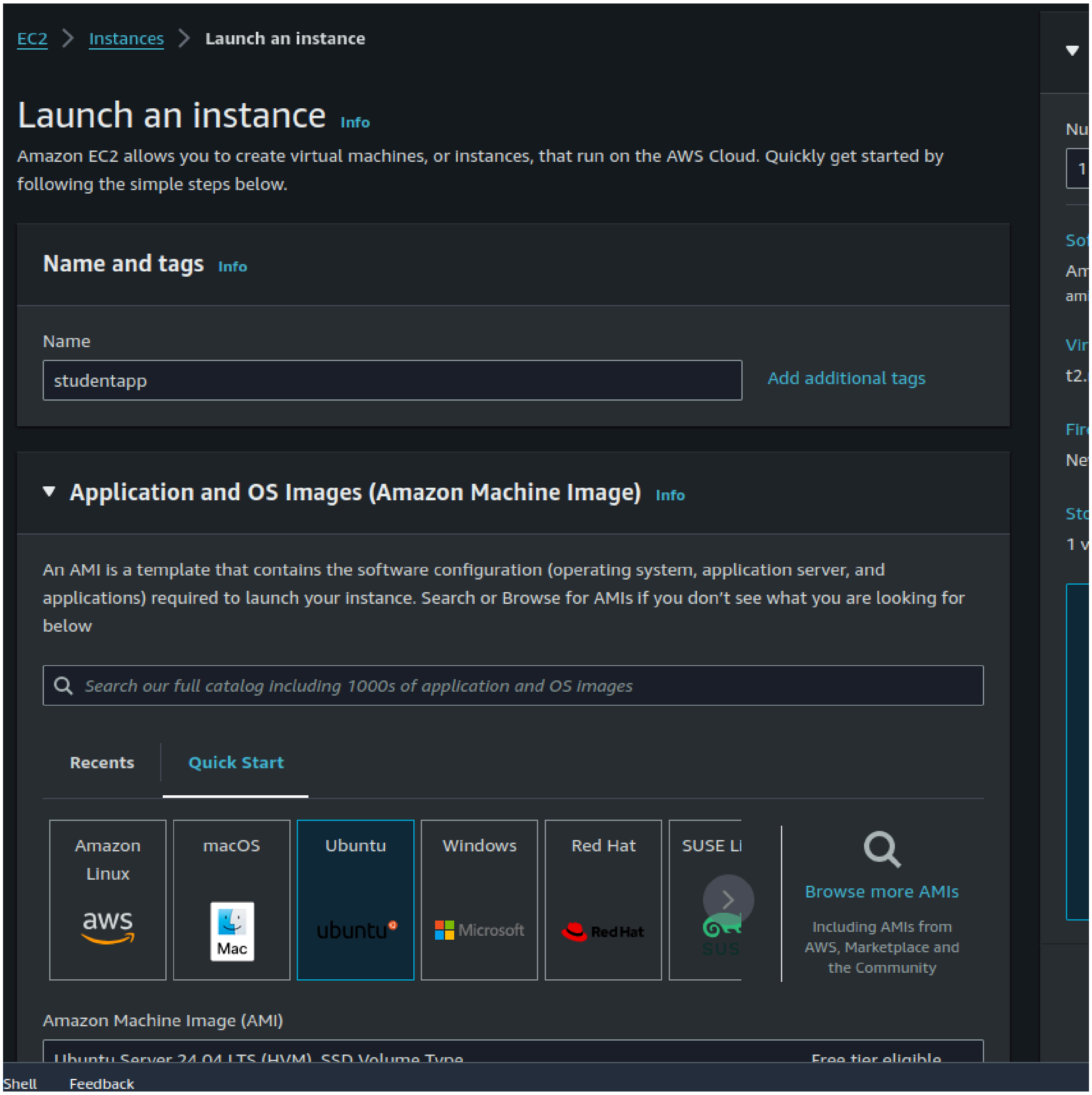
Task : Student-App-Containerization.

Step 1: Create an ec2 instance and launched it .



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: studentapp

Add additional tags

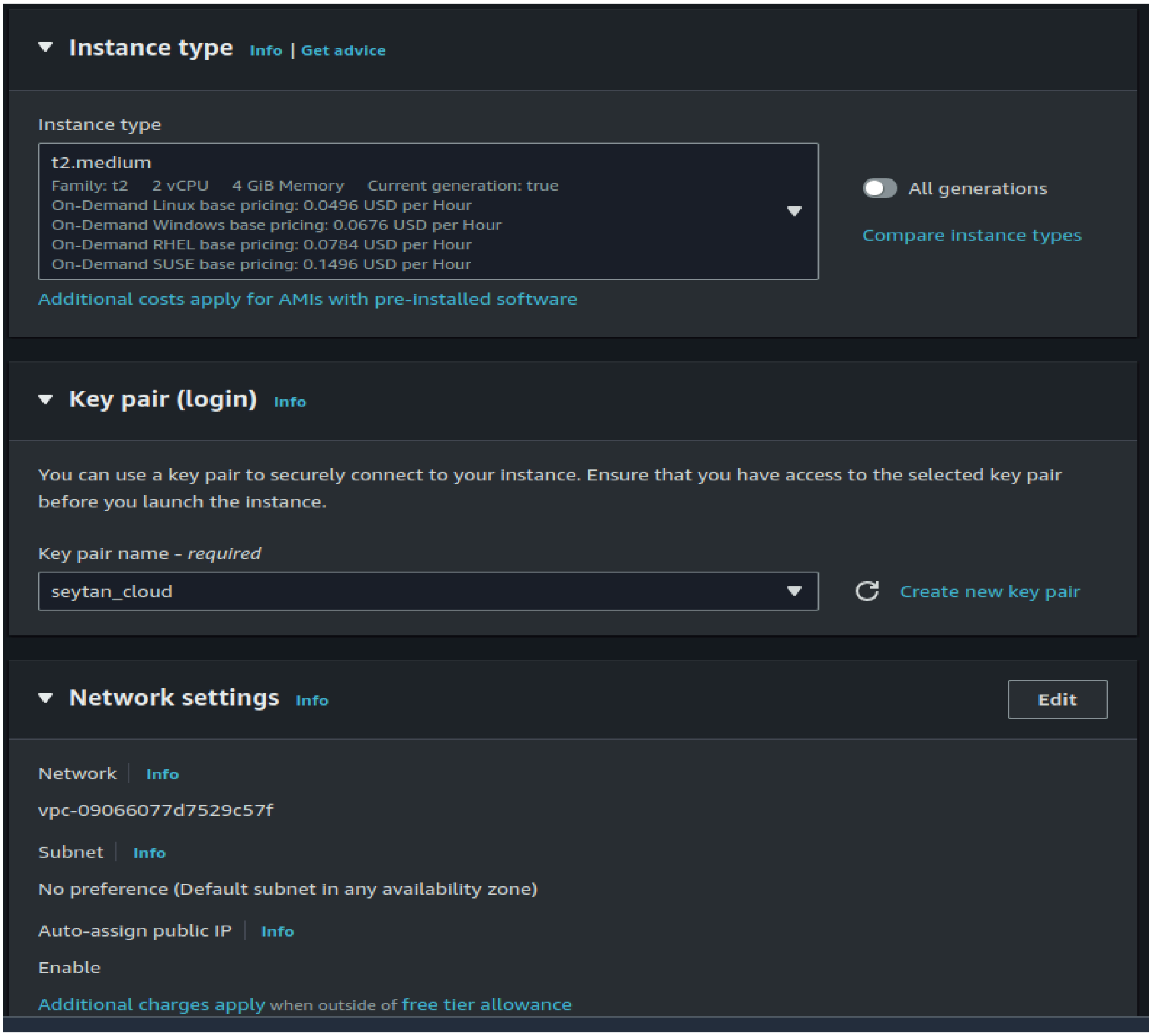
Application and OS Images (Amazon Machine Image) Info

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



Instance type

t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0496 USD per Hour
On-Demand Windows base pricing: 0.0676 USD per Hour
On-Demand RHEL base pricing: 0.0784 USD per Hour
On-Demand SUSE base pricing: 0.1496 USD per Hour

Additional costs apply for AMIs with pre-installed software

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

Network | Info

vpc-09066077d7529c57f

Subnet | Info

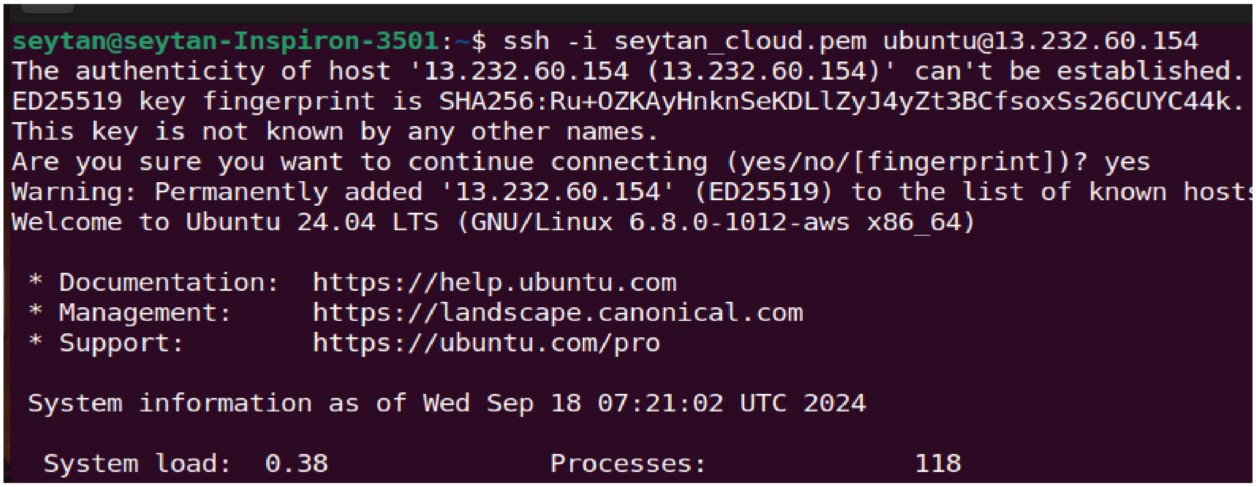
No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

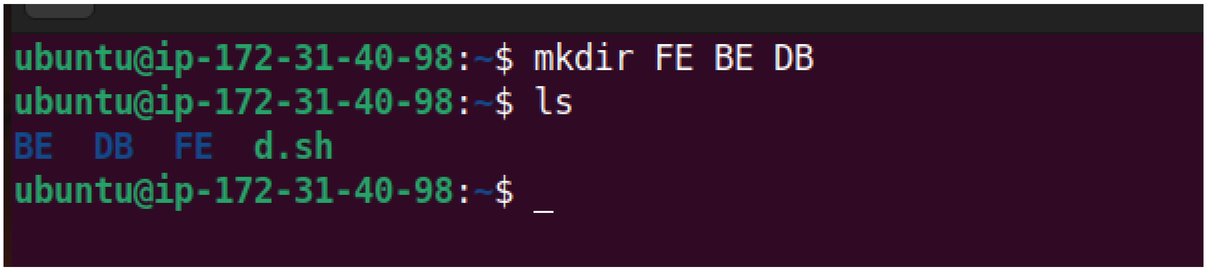
Additional charges apply when outside of free tier allowance

Step 2 : Take its ssh.



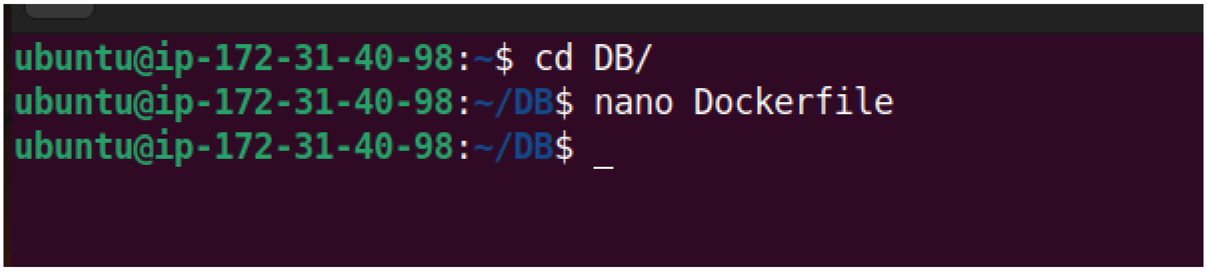
```
seytan@seytan-Inspiron-3501:~$ ssh -i seytan_cloud.pem ubuntu@13.232.60.154
The authenticity of host '13.232.60.154 (13.232.60.154)' can't be established.
ED25519 key fingerprint is SHA256:Ru+OZKAyHnknSeKDLlZyJ4yZt3BCfsoxSs26CUYC44k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.232.60.154' (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 Sep 18 07:21:02 UTC 2024

  System load:  0.38              Processes:             118
```

Step 3: Create 3 directories named FE ,BE ,DB

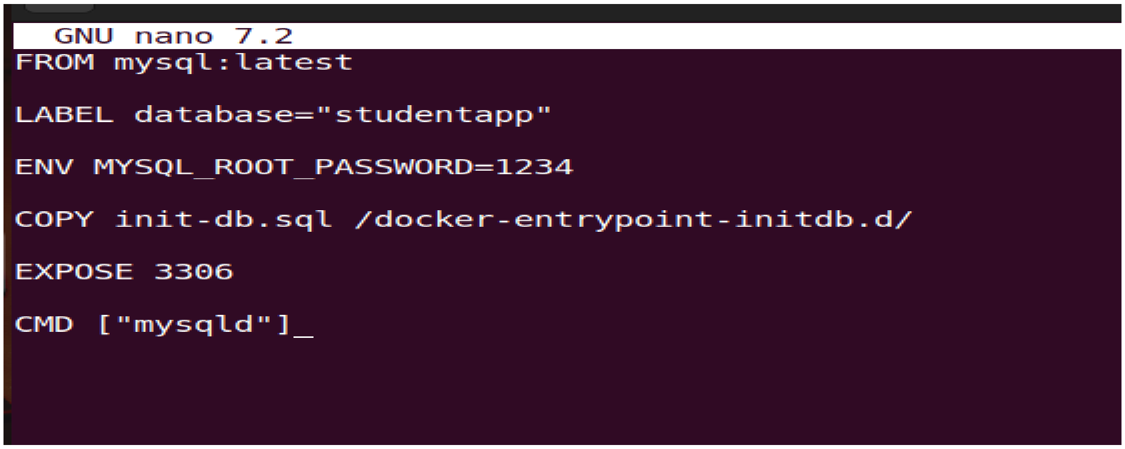


```
ubuntu@ip-172-31-40-98:~$ mkdir FE BE DB
ubuntu@ip-172-31-40-98:~$ ls
BE  DB  FE  d.sh
ubuntu@ip-172-31-40-98:~$ _
```

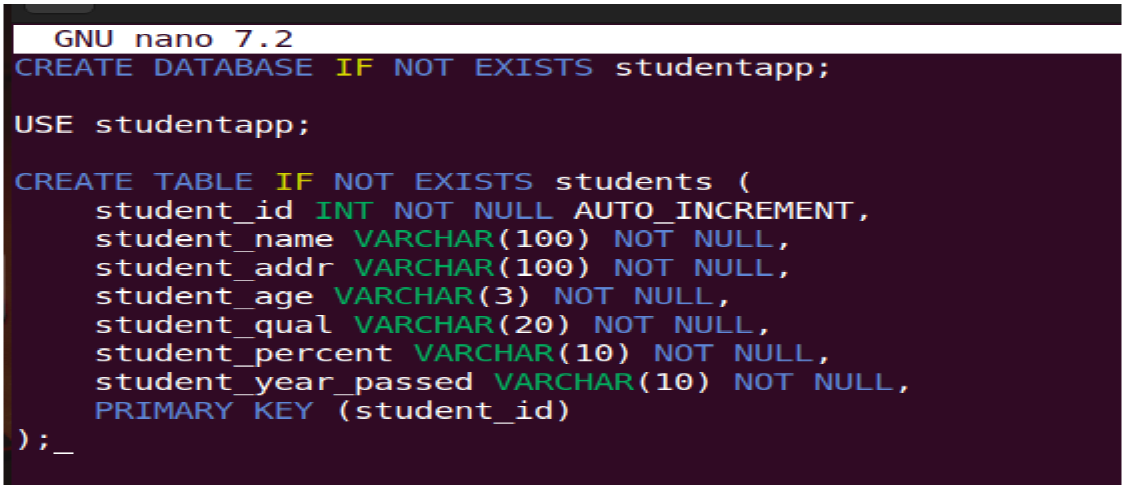
Step 4: Go to DB directory and create a Dockerfile and init-db.sql file .



```
ubuntu@ip-172-31-40-98:~$ cd DB/
ubuntu@ip-172-31-40-98:~/DB$ nano Dockerfile
ubuntu@ip-172-31-40-98:~/DB$ _
```

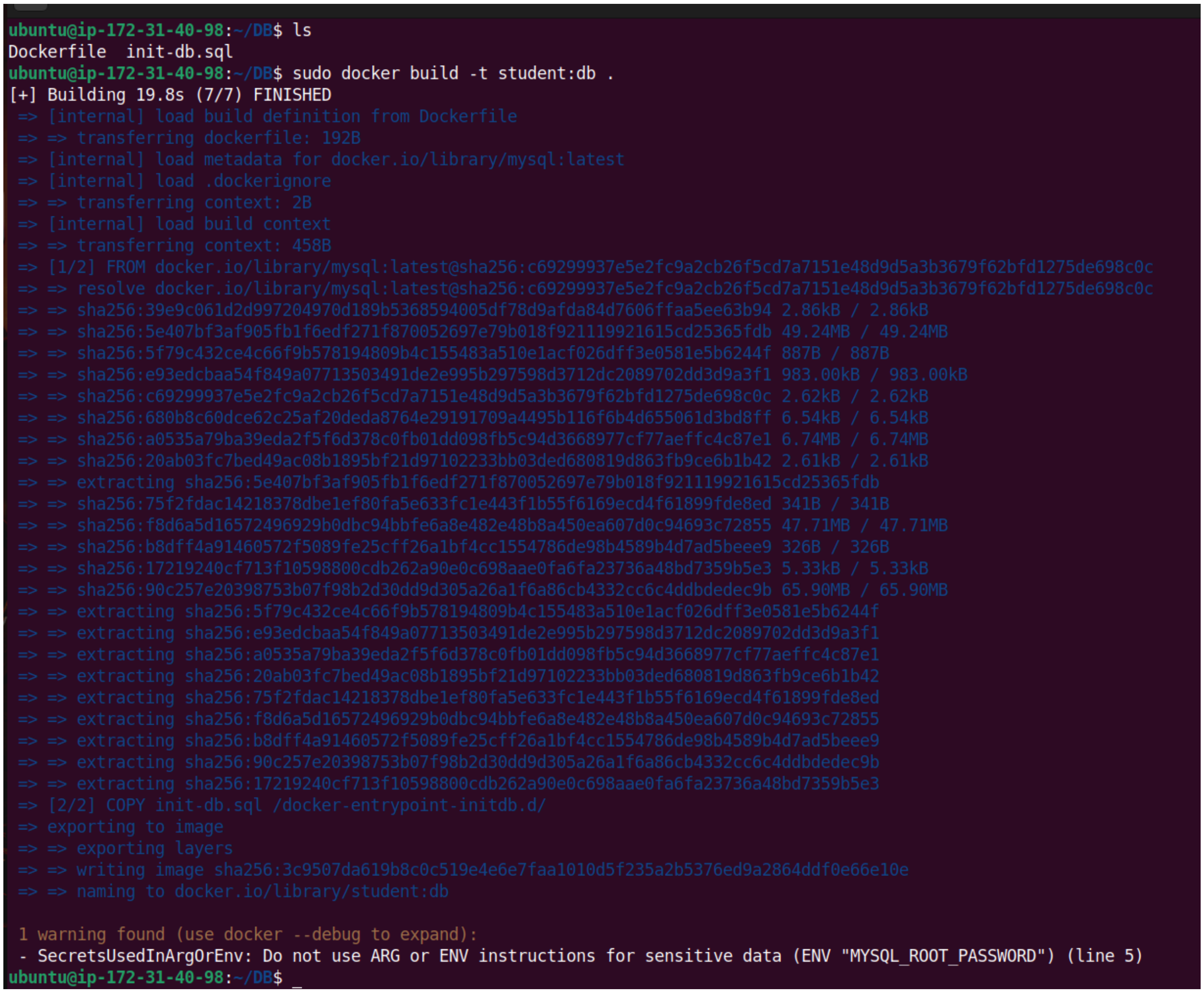


```
  GNU nano 7.2
FROM mysql:latest

LABEL database="studentapp"

ENV MYSQL_ROOT_PASSWORD=1234

COPY init-db.sql /docker-entrypoint-initdb.d/

EXPOSE 3306

CMD ["mysqld"]_
```



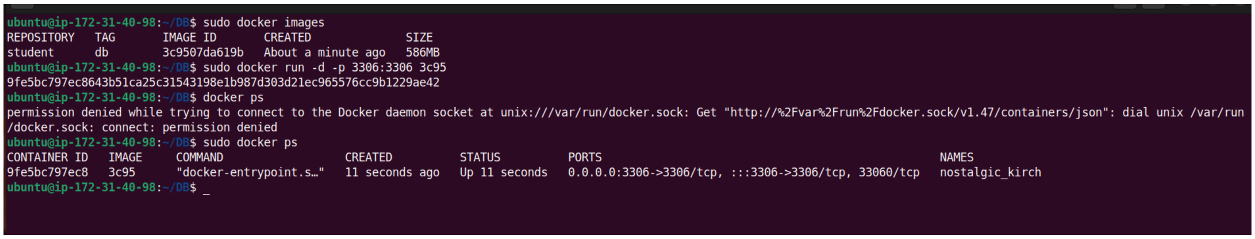
```
  GNU nano 7.2
CREATE DATABASE IF NOT EXISTS studentapp;

USE studentapp;

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)
);_
```

Step 5: build the images using docker build -t name:tag command .



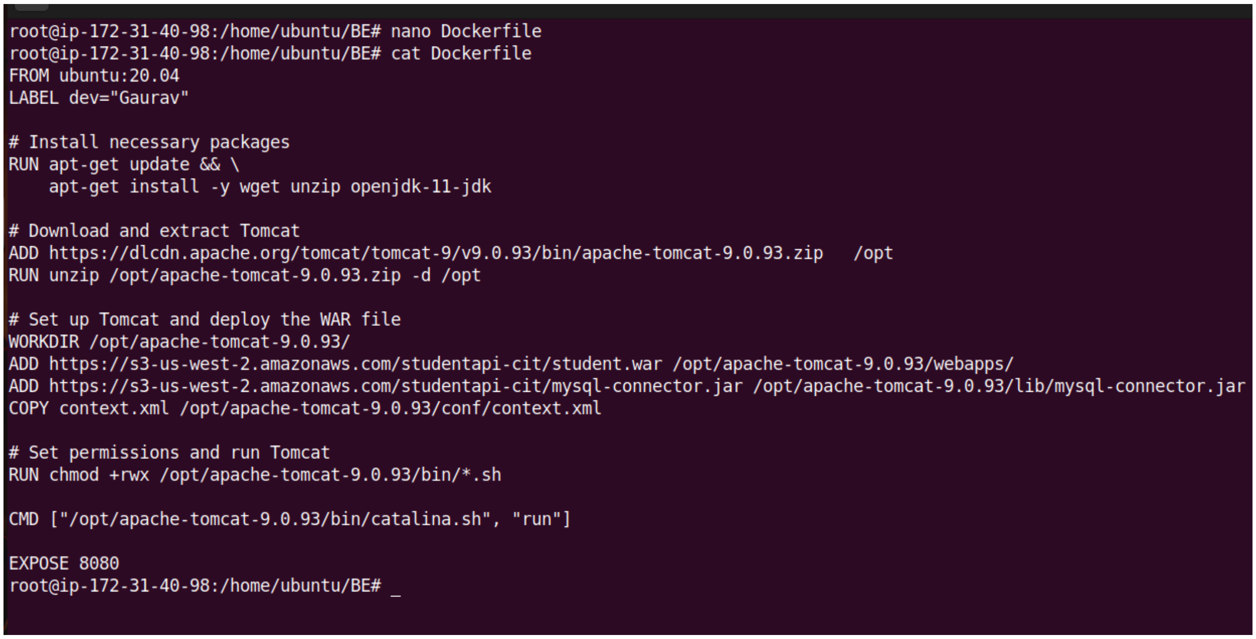
```
ubuntu@ip-172-31-40-98:~/DB$ ls
Dockerfile  init-db.sql
ubuntu@ip-172-31-40-98:~/DB$ sudo docker build -t student:db .
[+] Building 19.8s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 192B
 => [internal] load metadata for docker.io/library/mysql:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build context
 => => transferring context: 458B
 => [1/2] FROM docker.io/library/mysql:latest@sha256:c69299937e5e2fc9a2cb26f5cd7a7151e48d9d5a3b3679f62bfd1275de698c0c
 => => resolve docker.io/library/mysql:latest@sha256:c69299937e5e2fc9a2cb26f5cd7a7151e48d9d5a3b3679f62bfd1275de698c0c
 => => sha256:39e9c061d2d997204970d189b5368594005df78d9afda84d7606ffaa5ee63b94 2.86kB / 2.86kB
 => => sha256:5e407bf3af905fb1f6edf271f870052697e79b018f921119921615cd25365fdb 49.24MB / 49.24MB
 => => sha256:5f79c432ce4c66f9b578194809b4c155483a510e1acf026dff3e0581e5b6244f 887B / 887B
 => => sha256:e93edcbaa54f849a07713503491de2e995b297598d3712dc2089702dd3d9a3f1 983.00kB / 983.00kB
 => => sha256:c69299937e5e2fc9a2cb26f5cd7a7151e48d9d5a3b3679f62bfd1275de698c0c 2.62kB / 2.62kB
 => => sha256:680b8c60dce62c25af20deda8764e29191709a4495b116f6b4d655061d3bd8ff 6.54kB / 6.54kB
 => => sha256:a0535a79ba39eda2f5f6d378c0fb01dd098fb5c94d3668977cf77aeffc4c87e1 6.74MB / 6.74MB
 => => sha256:20ab03fc7bed49ac08b1895bf21d97102233bb03ded680819d863fb9ce6b1b42 2.61kB / 2.61kB
 => => extracting sha256:5e407bf3af905fb1f6edf271f870052697e79b018f921119921615cd25365fdb
 => => sha256:75f2fdac14218378dbe1ef80fa5e633fc1e443f1b55f6169ecd4f61899fde8ed 341B / 341B
 => => sha256:f8d6a5d16572496929b0dbc94bbfe6a8e482e48b8a450ea607d0c94693c72855 47.71MB / 47.71MB
 => => sha256:b8dff4a91460572f5089fe25cff26a1bf4cc1554786de98b4589b4d7ad5beee9 326B / 326B
 => => sha256:17219240cf713f10598800cdb262a90e0c698aae0fa6fa23736a48bd7359b5e3 5.33kB / 5.33kB
 => => sha256:90c257e20398753b07f98b2d30dd9d305a26a1f6a86cb4332cc6c4ddbdedec9b 65.90MB / 65.90MB
 => => extracting sha256:5f79c432ce4c66f9b578194809b4c155483a510e1acf026dff3e0581e5b6244f
 => => extracting sha256:e93edcbaa54f849a07713503491de2e995b297598d3712dc2089702dd3d9a3f1
 => => extracting sha256:a0535a79ba39eda2f5f6d378c0fb01dd098fb5c94d3668977cf77aeffc4c87e1
 => => extracting sha256:20ab03fc7bed49ac08b1895bf21d97102233bb03ded680819d863fb9ce6b1b42
 => => extracting sha256:75f2fdac14218378dbe1ef80fa5e633fc1e443f1b55f6169ecd4f61899fde8ed
 => => extracting sha256:f8d6a5d16572496929b0dbc94bbfe6a8e482e48b8a450ea607d0c94693c72855
 => => extracting sha256:b8dff4a91460572f5089fe25cff26a1bf4cc1554786de98b4589b4d7ad5beee9
 => => extracting sha256:90c257e20398753b07f98b2d30dd9d305a26a1f6a86cb4332cc6c4ddbdedec9b
 => => extracting sha256:17219240cf713f10598800cdb262a90e0c698aae0fa6fa23736a48bd7359b5e3
 => [2/2] COPY init-db.sql /docker-entrypoint-initdb.d/
 => exporting to image
 => => exporting layers
 => => writing image sha256:3c9507da619b8c0c519e4e6e7faa1010d5f235a2b5376ed9a2864ddf0e66e10e
 => => naming to docker.io/library/student:db

 1 warning found (use docker --debug to expand):
 - SecretsUsedInArgOrEnv: Do not use ARG or ENV instructions for sensitive data (ENV "MYSQL_ROOT_PASSWORD") (line 5)
ubuntu@ip-172-31-40-98:~/DB$ _
```

After this check the images and then run the container .

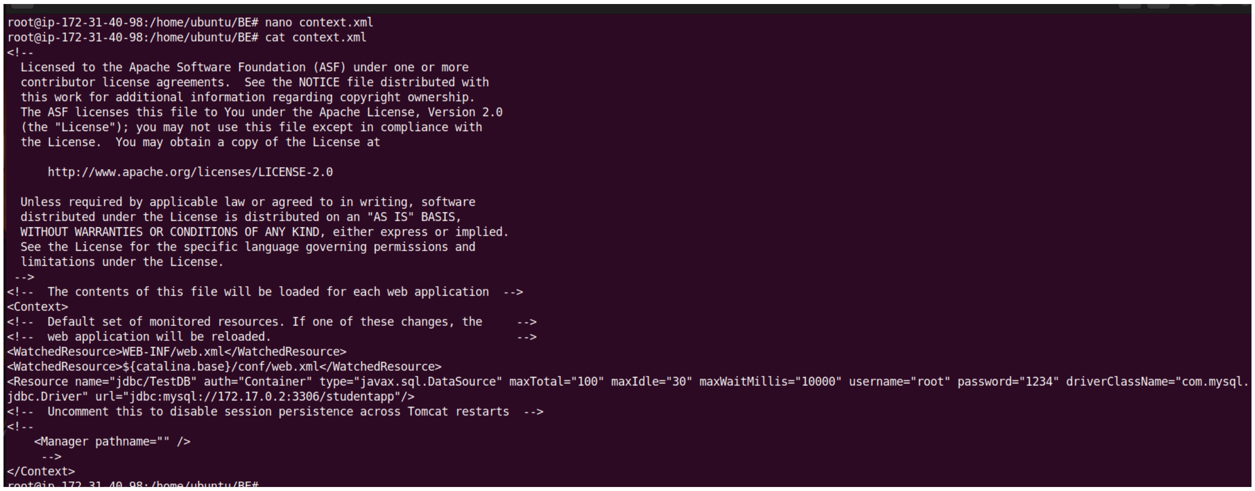


```
ubuntu@ip-172-31-40-98:~/DB$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
student      db        3c9507da619b   About a minute ago   586MB
ubuntu@ip-172-31-40-98:~/DB$ sudo docker run -d -p 3306:3306 3c95
9fe5bc797ec8643b51ca25c31543198e1b987d303d21ec965576cc9b1229ae42
ubuntu@ip-172-31-40-98:~/DB$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.47/containers/json": dial unix /var/run
/docker.sock: connect: permission denied
ubuntu@ip-172-31-40-98:~/DB$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                                  NAMES
9fe5bc797ec8   3c95      "docker-entrypoint.s…"   11 seconds ago   Up 11 seconds   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   nostalgic_kirch
ubuntu@ip-172-31-40-98:~/DB$ _
```

Step 6: Now go to the Backend BE directory and create a 2 files Dockerfile and context.xml.

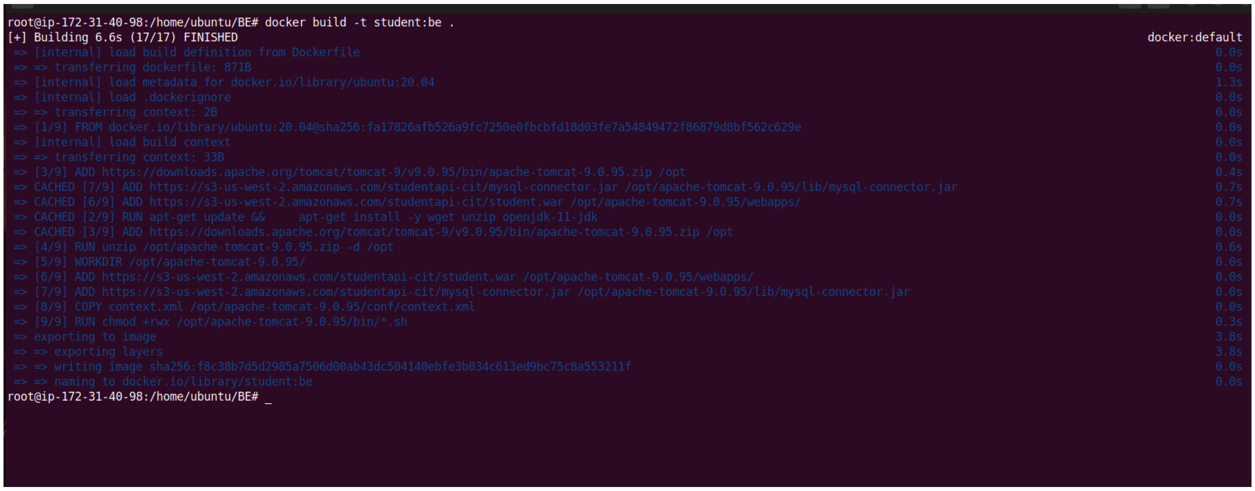


```
root@ip-172-31-40-98:/home/ubuntu/BE# nano Dockerfile
root@ip-172-31-40-98:/home/ubuntu/BE# cat Dockerfile
FROM ubuntu:20.04
LABEL dev="Gaurav"

# Install necessary packages
RUN apt-get update && \
    apt-get install -y wget unzip openjdk-11-jdk

# Download and extract Tomcat
ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.zip   /opt
RUN unzip /opt/apache-tomcat-9.0.93.zip -d /opt

# Set up Tomcat and deploy the WAR file
WORKDIR /opt/apache-tomcat-9.0.93/
ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war /opt/apache-tomcat-9.0.93/webapps/
ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar /opt/apache-tomcat-9.0.93/lib/mysql-connector.jar
COPY context.xml /opt/apache-tomcat-9.0.93/conf/context.xml

# Set permissions and run Tomcat
RUN chmod +rwx /opt/apache-tomcat-9.0.93/bin/*.sh

CMD ["/opt/apache-tomcat-9.0.93/bin/catalina.sh", "run"]

EXPOSE 8080
root@ip-172-31-40-98:/home/ubuntu/BE# _
```

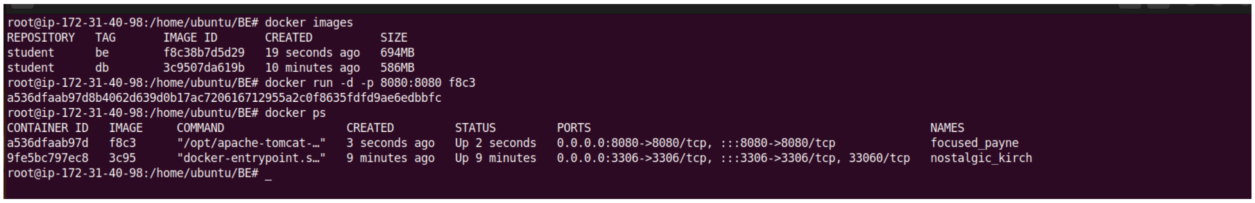


```
root@ip-172-31-40-98:/home/ubuntu/BE# nano context.xml
root@ip-172-31-40-98:/home/ubuntu/BE# cat context.xml
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
 -->
<!--  The contents of this file will be loaded for each web application  -->
<Context>
<!--  Default set of monitored resources. If one of these changes, the     -->
<!--  web application will be reloaded.                                    -->
<WatchedResource>WEB-INF/web.xml</WatchedResource>
<WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
<Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource" maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="root" password="1234" driverClassName="com.mysql.
jdbc.Driver" url="jdbc:mysql://172.17.0.2:3306/studentapp"/>
<!--  Uncomment this to disable session persistence across Tomcat restarts  -->
<!--
    <Manager pathname="" />
     -->
</Context>
root@ip-172-31-40-98:/home/ubuntu/BE#
```

Step 6: Now build the image for Backend and run the container.

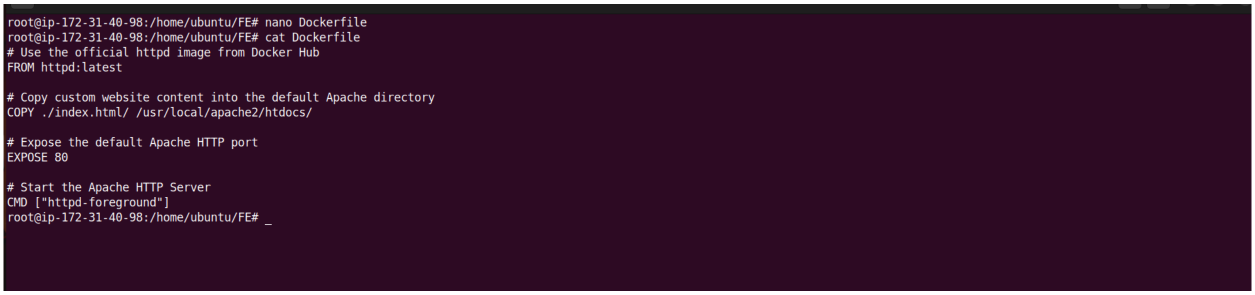


```
root@ip-172-31-40-98:/home/ubuntu/BE# docker build -t student:be .
[+] Building 6.6s (17/17) FINISHED                                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                                            0.0s
 => => transferring dockerfile: 871B                                                                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                                                                 1.3s
 => [internal] load .dockerignore                                                                                                               0.0s
 => => transferring context: 2B                                                                                                                 0.0s
 => [1/9] FROM docker.io/library/ubuntu:20.04@sha256:fa17826afb526a9fc7250e0fbcbfd18d03fe7a54849472f86879d8bf562c629e                         0.0s
 => [internal] load build context                                                                                                               0.0s
 => => transferring context: 33B                                                                                                                0.0s
 => [3/9] ADD https://downloads.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.zip /opt                                           0.4s
 => CACHED [7/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar /opt/apache-tomcat-9.0.95/lib/mysql-connector.jar   0.7s
 => CACHED [6/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war /opt/apache-tomcat-9.0.95/webapps/                           0.7s
 => CACHED [2/9] RUN apt-get update &&     apt-get install -y wget unzip openjdk-11-jdk                                                       0.0s
 => CACHED [3/9] ADD https://downloads.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.zip /opt                                    0.0s
 => [4/9] RUN unzip /opt/apache-tomcat-9.0.95.zip -d /opt                                                                                       0.6s
 => [5/9] WORKDIR /opt/apache-tomcat-9.0.95/                                                                                                    0.0s
 => [6/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war /opt/apache-tomcat-9.0.95/webapps/                                  0.0s
 => [7/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar /opt/apache-tomcat-9.0.95/lib/mysql-connector.jar          0.0s
 => [8/9] COPY context.xml /opt/apache-tomcat-9.0.95/conf/context.xml                                                                           0.0s
 => [9/9] RUN chmod +rwx /opt/apache-tomcat-9.0.95/bin/*.sh                                                                                     0.3s
 => exporting to image                                                                                                                          3.8s
 => => exporting layers                                                                                                                         3.8s
 => => writing image sha256:f8c38b7d5d2985a7506d00ab43dc504140ebfe3b034c613ed9bc75c8a553211f                                                    0.0s
 => => naming to docker.io/library/student:be                                                                                                   0.0s
root@ip-172-31-40-98:/home/ubuntu/BE# _
```

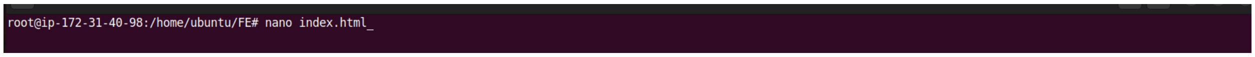


```
root@ip-172-31-40-98:/home/ubuntu/BE# docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
student      be        f8c38b7d5d29   19 seconds ago   694MB
student      db        3c9507da619b   10 minutes ago   586MB
root@ip-172-31-40-98:/home/ubuntu/BE# docker run -d -p 8080:8080 f8c3
a536dfaab97d8b4062d639d0b17ac720616712955a2c0f8635fdfd9ae6edbbfc
root@ip-172-31-40-98:/home/ubuntu/BE# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                                  NAMES
a536dfaab97d   f8c3      "/opt/apache-tomcat-…"   3 seconds ago    Up 2 seconds    0.0.0.0:8080->8080/tcp, :::8080->8080/tcp              focused_payne
9fe5bc797ec8   3c95      "docker-entrypoint.s…"   9 minutes ago    Up 9 minutes    0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   nostalgic_kirch
root@ip-172-31-40-98:/home/ubuntu/BE# _
```

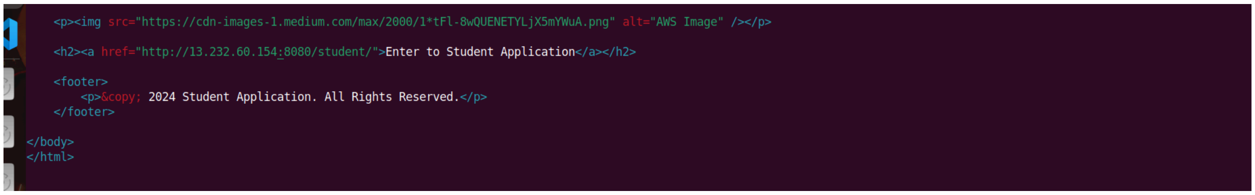
Step 7: Now go to the Frontedn folder and create Dockerfile and an index.html.



```
root@ip-172-31-40-98:/home/ubuntu/FE# nano Dockerfile
root@ip-172-31-40-98:/home/ubuntu/FE# cat Dockerfile
# Use the official httpd image from Docker Hub
FROM httpd:latest

# Copy custom website content into the default Apache directory
COPY ./index.html/ /usr/local/apache2/htdocs/

# Expose the default Apache HTTP port
EXPOSE 80

# Start the Apache HTTP Server
CMD ["httpd-foreground"]
root@ip-172-31-40-98:/home/ubuntu/FE# _
```

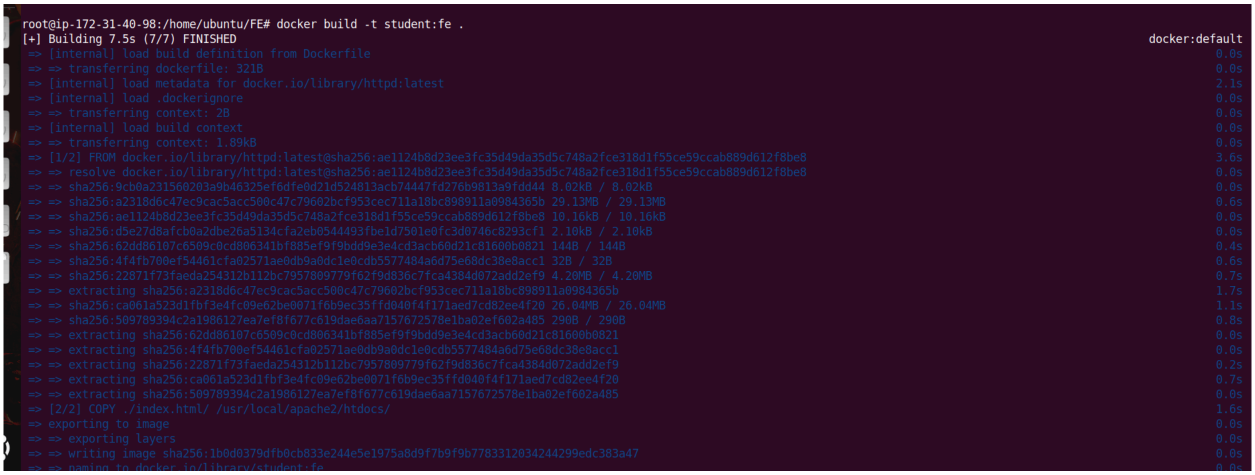


```
root@ip-172-31-40-98:/home/ubuntu/FE# nano index.html_
```



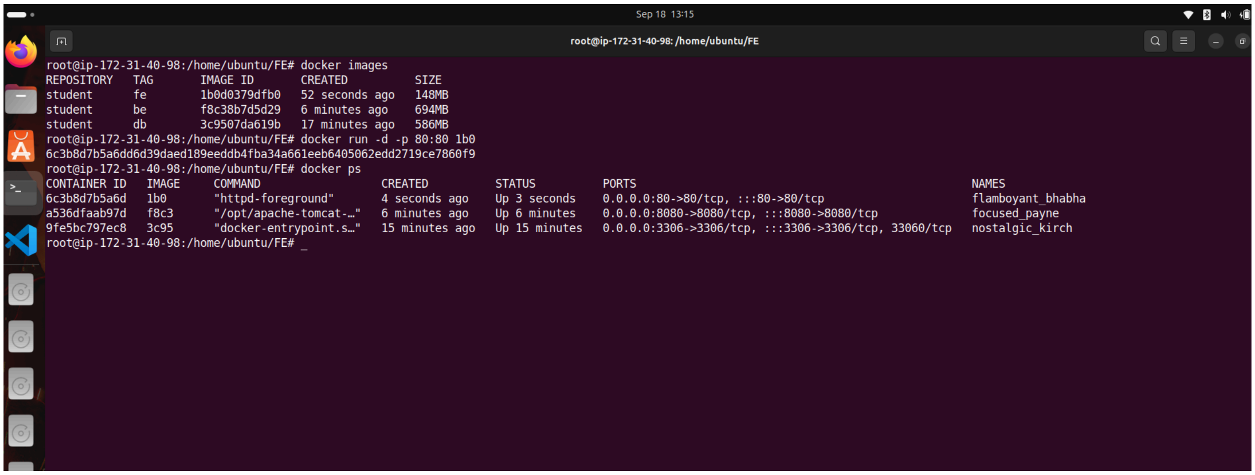
```
    <p><img src="https://cdn-images-1.medium.com/max/2000/1*tFl-8wQUENETYLjX5mYWuA.png" alt="AWS Image" /></p>

    <h2><a href="http://13.232.60.154:8080/student/">Enter to Student Application</a></h2>

    <footer>
        <p>&copy; 2024 Student Application. All Rights Reserved.</p>
    </footer>

</body>
</html>
```

Step 8: Now build the image for frontend and run the container.



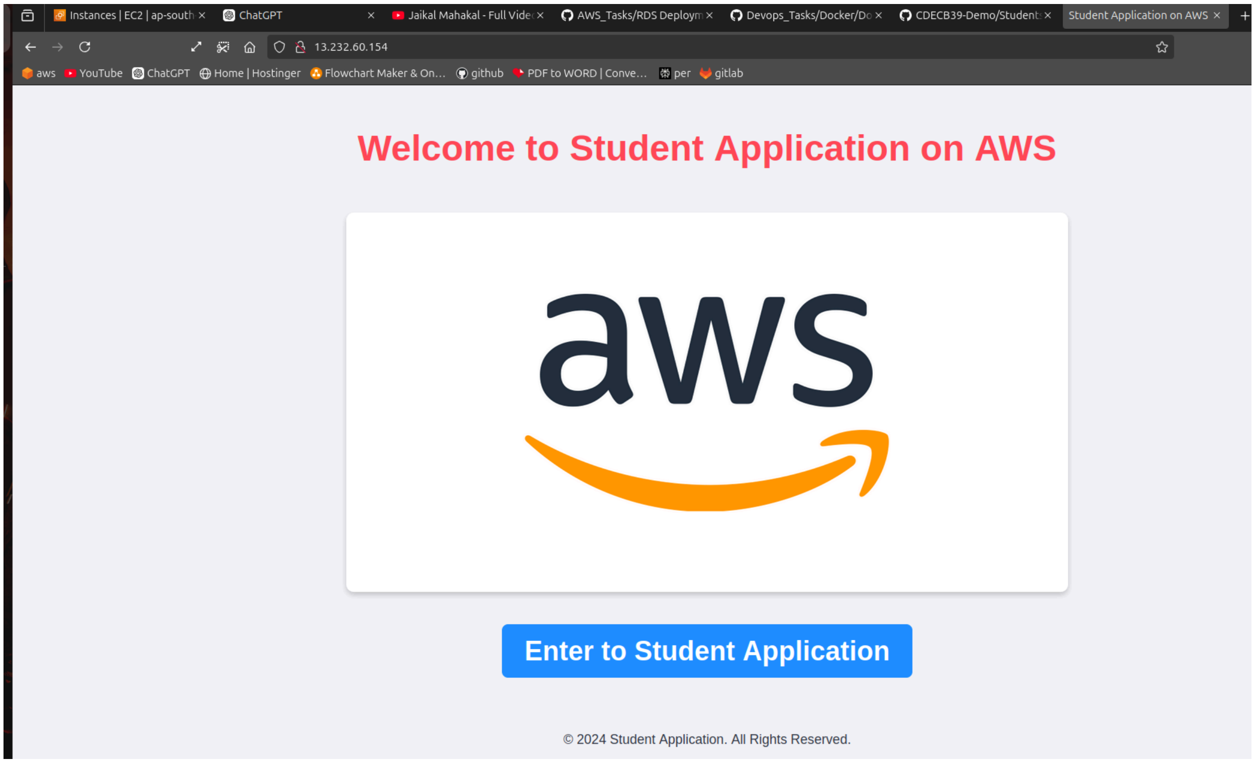
```
root@ip-172-31-40-98:/home/ubuntu/FE# docker build -t student:fe .
[+] Building 7.5s (7/7) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 321B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/httpd:latest                                                               2.1s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 1.89kB                                                                                           0.0s
 => [1/2] FROM docker.io/library/httpd:latest@sha256:ae1124b8d23ee3fc35d49da35d5c748a2fce318d1f55ce59ccab889d612f8be8         3.6s
 => => resolve docker.io/library/httpd:latest@sha256:ae1124b8d23ee3fc35d49da35d5c748a2fce318d1f55ce59ccab889d612f8be8         0.0s
 => => sha256:9cb0a231560203a9b46325ef6dfe0d21d524813acb74447fd276b9813a9fdd44 8.02kB / 8.02kB                                0.0s
 => => sha256:a2318d6c47ec9cac5acc500c47c79602bcf953cec711a18bc898911a0984365b 29.13MB / 29.13MB                              0.6s
 => => sha256:ae1124b8d23ee3fc35d49da35d5c748a2fce318d1f55ce59ccab889d612f8be8 10.16kB / 10.16kB                              0.0s
 => => sha256:d5e27d8afcb0a2dbe26a5134cfa2eb0544493fbe1d7501e0fc3d0746c8293cf1 2.10kB / 2.10kB                                0.0s
 => => sha256:62dd86107c6509c0cd806341bf885ef9f9bdd9e3e4cd3acb60d21c81600b0821 144B / 144B                                    0.4s
 => => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B / 32B                                      0.6s
 => => sha256:22871f73faeda254312b112bc7957809779f62f9d836c7fca4384d072add2ef9 4.20MB / 4.20MB                                0.7s
 => => extracting sha256:a2318d6c47ec9cac5acc500c47c79602bcf953cec711a18bc898911a0984365b                                     1.7s
 => => sha256:ca061a523d1fbf3e4fc09e62be0071f6b9ec35ffd040f4f171aed7cd82ee4f20 26.04MB / 26.04MB                              1.1s
 => => sha256:509789394c2a1986127ea7ef8f677c619dae6aa7157672578e1ba02ef602a485 290B / 290B                                    0.8s
 => => extracting sha256:62dd86107c6509c0cd806341bf885ef9f9bdd9e3e4cd3acb60d21c81600b0821                                     0.0s
 => => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1                                     0.0s
 => => extracting sha256:22871f73faeda254312b112bc7957809779f62f9d836c7fca4384d072add2ef9                                     0.2s
 => => extracting sha256:ca061a523d1fbf3e4fc09e62be0071f6b9ec35ffd040f4f171aed7cd82ee4f20                                     0.7s
 => => extracting sha256:509789394c2a1986127ea7ef8f677c619dae6aa7157672578e1ba02ef602a485                                     0.0s
 => [2/2] COPY ./index.html/ /usr/local/apache2/htdocs/                                                                       1.6s
 => exporting to image                                                                                                        0.0s
 => => exporting layers                                                                                                       0.0s
 => => writing image sha256:1b0d0379dfb0cb833e244e5e1975a8d9f7b9f9b7783312034244299edc383a47                                  0.0s
 => => naming to docker.io/library/student:fe                                                                                 0.0s
```

Check the images and container are running properly.



```
root@ip-172-31-40-98:/home/ubuntu/FE# docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
student      fe        1b0d0379dfb0   52 seconds ago   148MB
student      be        f8c38b7d5d29   6 minutes ago    694MB
student      db        3c9507da619b   17 minutes ago   586MB
root@ip-172-31-40-98:/home/ubuntu/FE# docker run -d -p 80:80 1b0
6c3b8d7b5a6dd6d39daed189eeddb4fba34a661eeb6405062edd2719ce7860f9
root@ip-172-31-40-98:/home/ubuntu/FE# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                                   NAMES
6c3b8d7b5a6d   1b0       "httpd-foreground"       4 seconds ago    Up 3 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp                       flamboyant_bhabha
a536dfaab97d   f8c3      "/opt/apache-tomcat-…"   6 minutes ago    Up 6 minutes    0.0.0.0:8080->8080/tcp, :::8080->8080/tcp               focused_payne
9fe5bc797ec8   3c95      "docker-entrypoint.s…"   15 minutes ago   Up 15 minutes   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp    nostalgic_kirch
root@ip-172-31-40-98:/home/ubuntu/FE# _
```

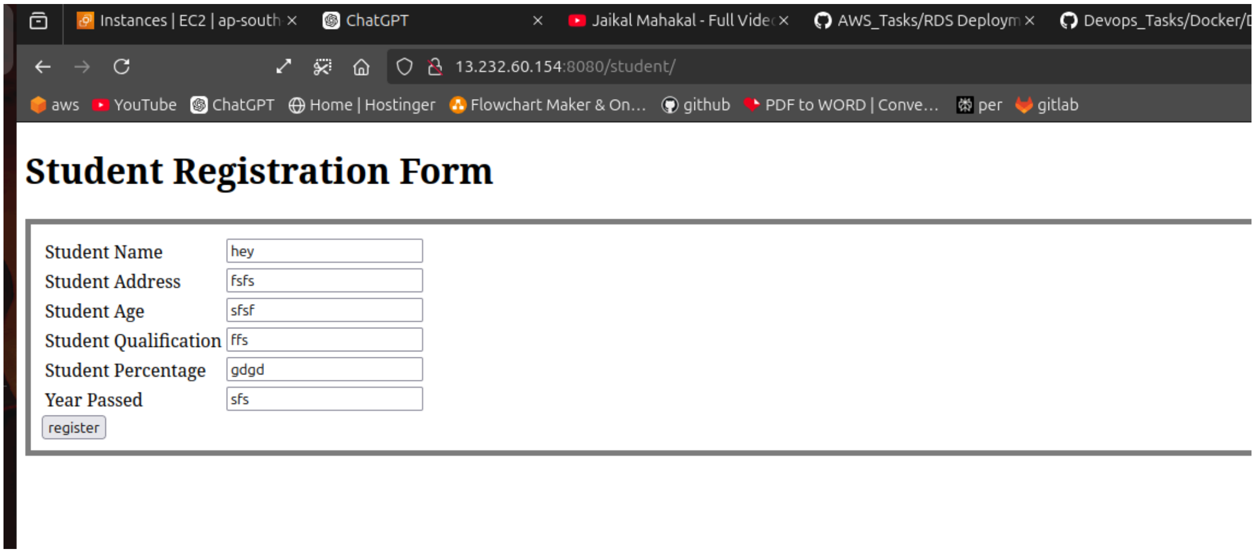
Output :



Welcome to Student Application on AWS

Enter to Student Application

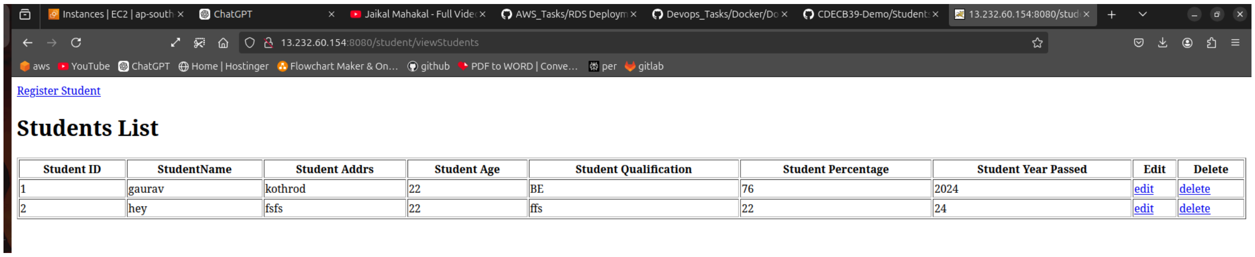
© 2024 Student Application. All Rights Reserved.



Student Registration Form

Student Name	hey
Student Address	fsfs
Student Age	sfsf
Student Qualification	ffs
Student Percentage	gdgd
Year Passed	sfs

register



Register Student

Students List

Student ID	StudentName	Student Addrs	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	gaurav	kothrod	22	BE	76	2024	edit	delete
2	hey	fsfs	22	ffs	22	24	edit	delete

Conclusion : This task enhances reliability, scalability, and ease of management by encapsulating the app with all its dependencies, allowing seamless transitions across different stages and platforms.