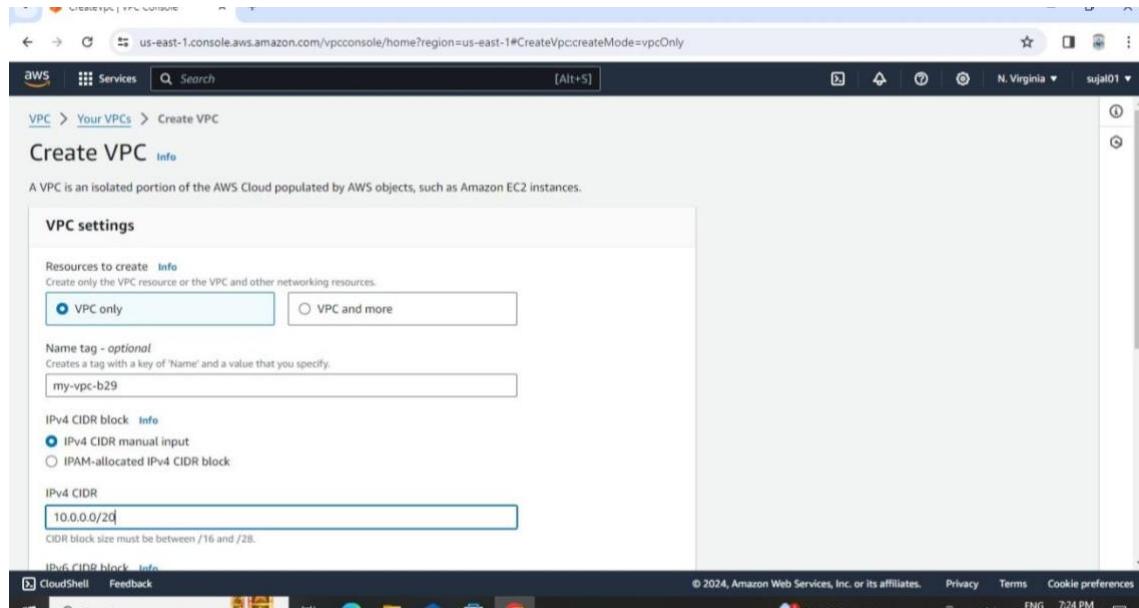
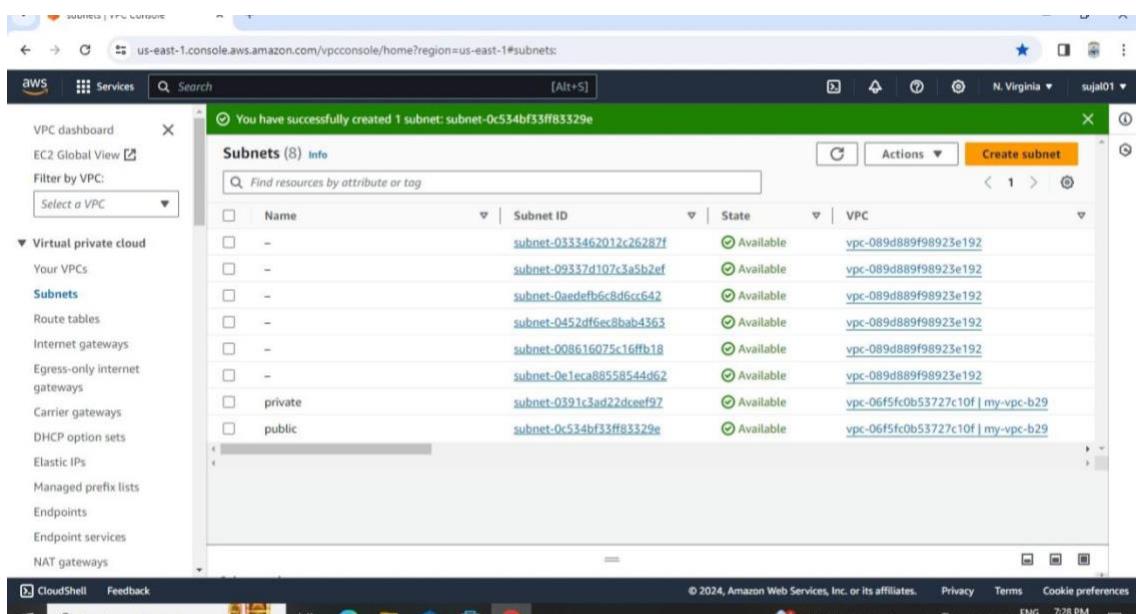


AWS Project

- 1) Login to aws using registered user credentials. Go to the VPC dashboard and create a new VPC. We have used '10.0.0.0/20' as IPv4 CIDR, while creating VPC.



- 2) Go to subnets dashboard and create 2 new subnets of the new VPC. Private subnet has IPv4 CIDR block '10.0.1.0/25' configuration and public subnet has IPv4 CIDR block '10.0.2.0/25' configuration. Create both subnets in different availability zones.



- 3) Go to internet gateways and create a new internet gateway. Click on actions and attach this internet gateway to our new VPC.

The screenshot shows the AWS VPC Internet Gateways console. On the left, there's a navigation sidebar with options like 'Virtual private cloud', 'Internet gateways', and 'Egress-only internet gateways'. The main area displays a single internet gateway with the following details:

- Internet gateway ID:** igw-04fcf9764826598b5
- State:** Detached
- VPC ID:** -
- Owner:** 637423656162
- Tags:** Name: my-igw-b29

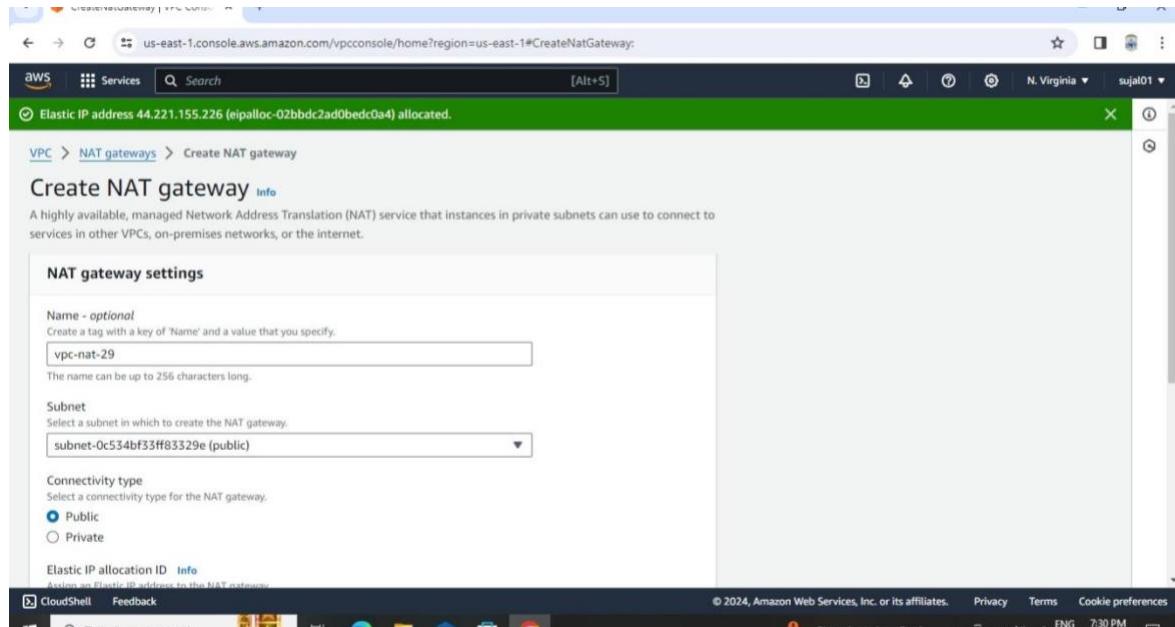
- 4) Go to routing tables dashboard and open the new routing table that was created by default when new VPC was created. Click on edit routes and add the new internet gateway to the route table.

The screenshot shows the AWS Route Tables console. The URL indicates we're editing the routes for a specific route table. The 'Edit routes' page has the following structure:

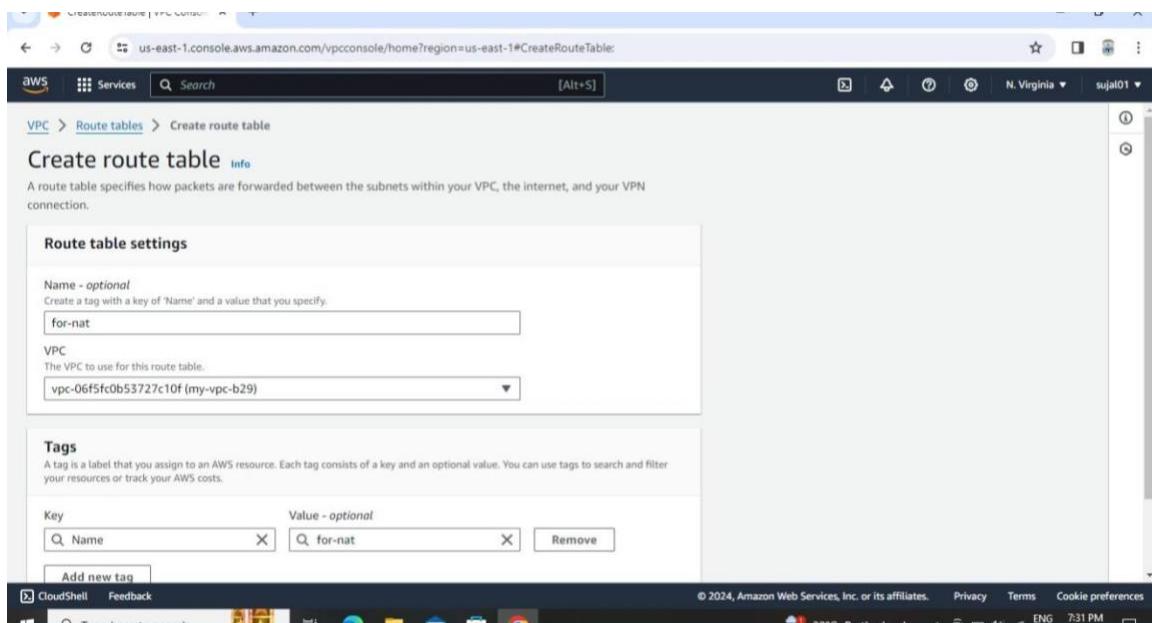
Destination	Target	Status	Propagated
10.0.0.0/20	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-04fcf9764826598b5		<input type="button" value="Remove"/>

At the bottom, there are buttons for 'Add route', 'Cancel', 'Preview', and a prominent orange 'Save changes' button.

5) Go to NAT gateway dashboard and create a NAT gateway. While creating the gateway choose public subnet from subnet drop-down.



6) Go to route tables dashboard and create a new route table (for-nat). Click on edit routes.



7) Add new NAT Gateway route and click save changes.

The screenshot shows the 'Edit routes' page for a specific route table. A new route is being added:

Destination	Target	Status	Propagated
10.0.0.0/20	local	Active	No
Q_ 0.0.0.0/0	NAT Gateway	-	No
	Q_ nat-0296cc8f651c17748		

Buttons at the bottom include 'Add route', 'Cancel', 'Preview', and a highlighted 'Save changes' button.

8) Go to subnets dashboard and open private subnets. Under route tables click on edit subnet association and change the route table to for-nat. Click on save changes.

The screenshot shows the 'Edit route table association' page for a specific subnet. The route table ID is set to 'rtb-0b6579727a509ae31 (for-nat)'.

Subnet route table settings

Subnet ID	subnet-0391c3ad22dceef97
Route table ID	rtb-0b6579727a509ae31 (for-nat)

Routes (2)

Destination	Target
10.0.0.0/20	local
0.0.0.0/0	nat-0296cc8f651c17748

Buttons at the bottom include 'Cancel' and a highlighted 'Save' button.

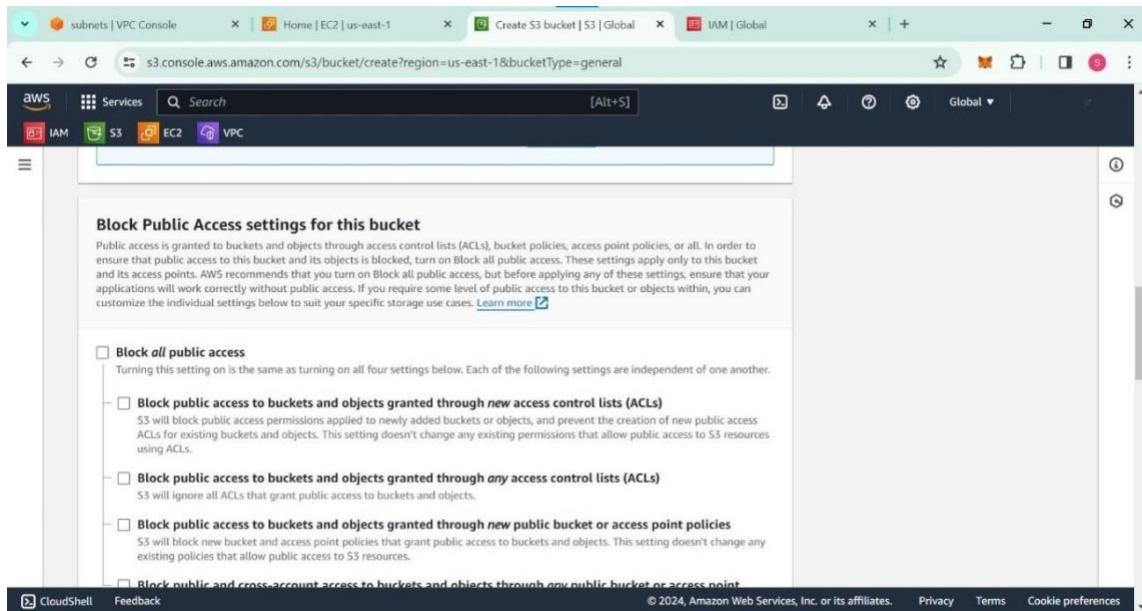
9) Go to S3 service dashboard and click on create bucket. Enter a unique bucknet name.

The screenshot shows the 'Create bucket' page in the AWS S3 console. The 'General configuration' section is visible, featuring an 'AWS Region' dropdown set to 'US East (N. Virginia) us-east-1'. Under 'Bucket type', the 'General purpose' option is selected, with a note explaining it's recommended for most use cases. A second option, 'Directory - New', is also listed. The 'Bucket name' field contains 'project-bucket-b29'. Below the name field is a note about uniqueness and naming rules, with a link to 'See rules for bucket naming'. At the bottom of the configuration section, there's a note about 'Copy settings from existing bucket - optional'.

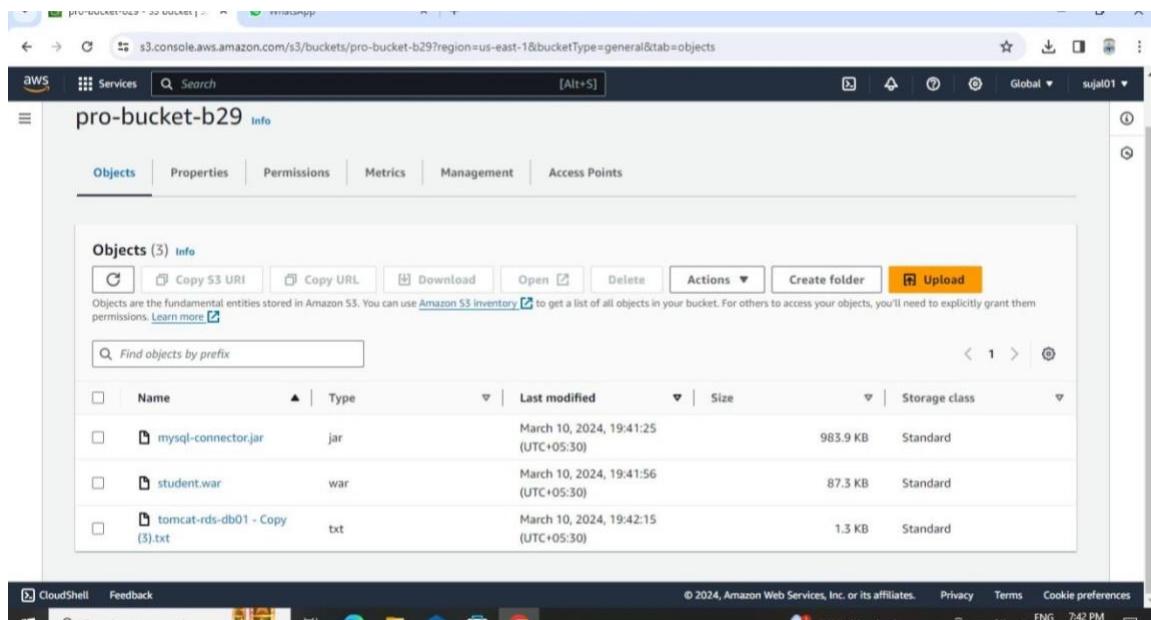
10) Choose ACLs enabled option.

The screenshot shows the 'Object Ownership' configuration page. It features two main options: 'ACLs disabled (recommended)' and 'ACLs enabled'. The 'ACLs enabled' option is selected, with a note explaining that objects can be owned by other AWS accounts and accessed via ACLs. A warning message in a yellow box advises against disabling ACLs unless individual object access control is needed. Below this, the 'Object Ownership' section includes 'Bucket owner preferred' (selected) and 'Object writer' options. A note at the bottom states that if object ownership is enforced, a bucket policy must be specified.

11) Untick the block all public access box. Click on create bucket.



12) Upload the required configuration files in the bucket.



13) Select all uploaded objects in the bucket and click on actions, then click on make public using ACL.

The screenshot shows the AWS S3 console with the URL s3.console.aws.amazon.com/s3/buckets/pro-bucket-b29/object/edit_public_read_access?region=us-east-1&bucketType=general&showVersions=false. The page title is "Amazon S3 > Buckets > pro-bucket-b29 > Make public". The main content area is titled "Make public" with a sub-section "Specified objects". A yellow warning box states: "When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects." Below this, a table lists three objects:

Name	Type	Last modified	Size
mysql-connector.jar	jar	March 10, 2024, 19:41:25 (UTC+05:30)	983.9 KB
student.war	war	March 10, 2024, 19:41:56 (UTC+05:30)	87.3 KB
tomcat-rds-db01 - Copy (3).txt	txt	March 10, 2024, 19:42:15 (UTC+05:30)	1.3 KB

At the bottom right of the dialog are "Cancel" and "Make public" buttons. The status bar at the bottom of the browser window shows "CloudShell Feedback" and the date "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 7:42 PM".

14) Now open the IAM service dashboard and click on create role. Choose the EC2 service under use case and click on next.

The screenshot shows the AWS IAM service dashboard with the URL us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create. The page title is "Service or use case". A dropdown menu is open, showing "EC2" selected. Below the dropdown, a section titled "Choose a use case for the specified service." lists several options, each with a description:

- EC2**
Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role**
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- EC2 - Spot Fleet Auto Scaling**
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- EC2 - Spot Fleet Tagging**
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- EC2 - Spot Instances**
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- EC2 - Spot Fleet**
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- EC2 - Scheduled Instances**
Allows EC2 Scheduled Instances to manage instances on your behalf.

At the bottom right of the dialog are "Cancel" and "Next" buttons. The status bar at the bottom of the browser window shows "CloudShell Feedback" and the date "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 7:44 PM".

15) In the permissions policy section choose the 'AmazonS3FullAccess' and click on next. Enter a valid role name and click on create role.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 2: Add permissions'. A search bar at the top right contains 's3'. Below it is a table titled 'Permissions policies (1/917) Info' with a filter 'Filter by Type' set to 'All types'. The table lists several AWS managed policies, with 'AmazonS3FullAccess' selected (indicated by a checked checkbox). Other policies listed include AmazonDMSRedshiftFullAccess, AmazonS3ObjectLambdaFullAccess, AmazonS3OutpostFullAccess, AmazonS3OutpostReadFullAccess, AmazonS3ReadOnlyFullAccess, and AWSRarkininService.

16) 'project-role' has been successfully created.

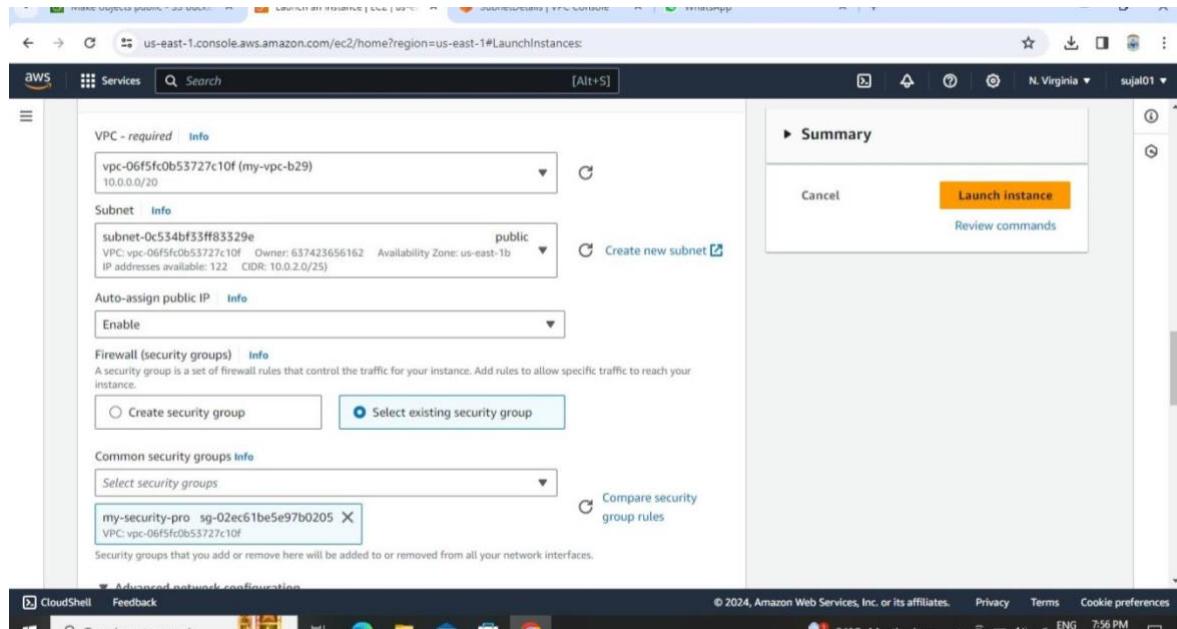
The screenshot shows the 'Roles' page in the AWS IAM console. A green banner at the top center says 'Role s3full-pro created.' On the left, a sidebar shows 'Identity and Access Management (IAM)' with 'Access management' expanded, showing 'User groups', 'Users', and 'Roles'. Under 'Roles', 's3full-pro' is listed. The main pane shows a table of roles with columns 'Role name', 'Trusted entities', and 'Last activity'. The newly created 's3full-pro' role is listed, along with other roles like 's3full-access' and various temporary roles starting with 's3crr_'. The status bar at the bottom right shows 'ENG 7:45 PM'.

17) Go to EC2 dashboard and from left navigation bar select Security group. Create a security group with a valid name (here launch-wizard-7) and add all the following rules as inbound rules: SSH, HTTP, MYSQL/Aurora, ALL ICMP and Custom TCP with port range as 8080. Click on create security group.

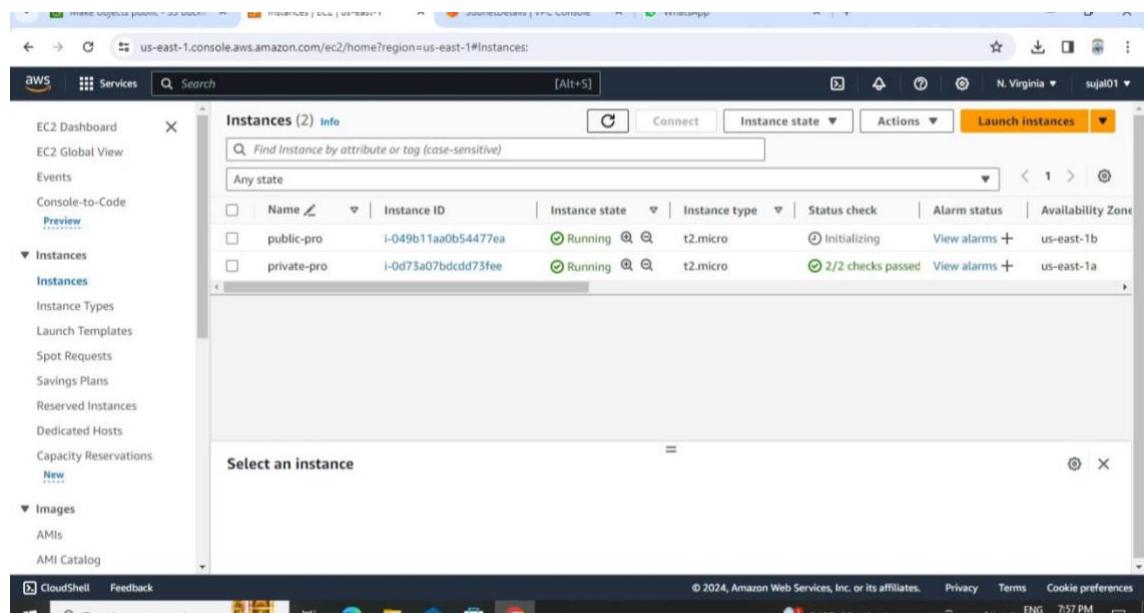
Security group rule	IP version	Type	Protocol	Port range
sgr-0556dc69da698bc...	IPv4	Custom TCP	TCP	8080
sgr-0c1ade22238bf2a2b	IPv4	MYSQL/Aurora	TCP	3306
sgr-025b054edfb2628cb	IPv4	SSH	TCP	22
sgr-0dbda9e8f1aa5c33b	IPv4	All ICMP - IPv4	ICMP	All
sgr-078cb9189ad4275...	IPv4	HTTP	TCP	80

18) Go to EC2 instance dashboard and click on launch instance. Enter a valid instance name (project_private) and select a key-pair. Click edit in network settings and choose the newly created VPC. For private instance select private subnet. Add the existing security group 'launch-wizard-7' as security group. Click on launch instance.

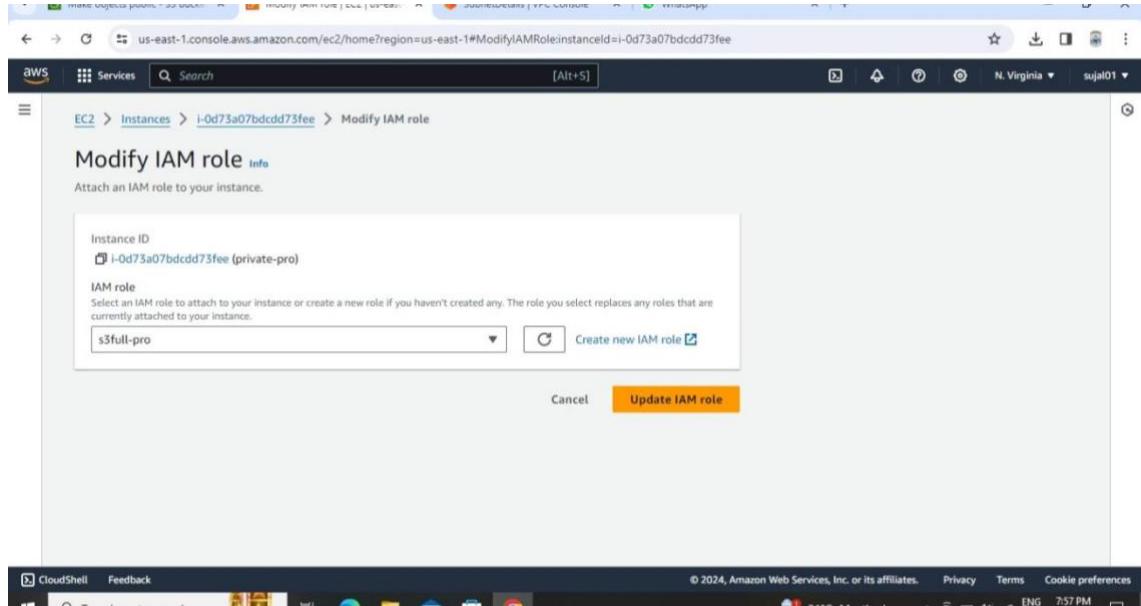
19) Again click on launch instance to create a second instance. Enter a valid instance name (project_public) and select a key-pair. Click edit in network settings and choose the newly created VPC. For public instance select public subnet. Add SSH and ICMP rules in security group and click on launch instance. Add the existing security group 'launch-wizard-7' as security group. Click on launch instance.



20) The two instances have been successfully created.



21) Select the private instance. Click on actions, security group and then Modify IAM role. Select the newly created role (project-role) and click on update IAM role.



22) Open cmd in the downloads folder where the private key is present. Run the following command to connect to the public instance:

- ssh -i instance_key.pem ec2-user@public_ip_public_instance

Use command '**vim newkey.pem**' to create a file. Open the key in cmd in notepad and copy the key. Paste the key in our 'newkey.pem' file and save the file.

```
ec2-user@ip-10-0-2-90:~----  
-----BEGIN RSA PRIVATE KEY-----  
MIIEpQTBAAKCAQEAjQoipOKl9Ng1jNB3BwZmPGy678cyPLqYYDREVuYXhZPMCJ  
dhFoxdfDUu4z9PuSF0tF/Vi81eu0sgsZ3zIjpzb75EMGe+A+zjezqlagdpC+rA  
7MFe/usdHk27du3G4swamf119cV3Xo+nkFd0/P0d4/yhHRUIC3HDXA+mfXldt  
Kqa3Qz/J0riqhZ1Xn76amfTaTm6nFzixvUrgtFk327Lxn9HzqsFrKnuzip7FsV  
CROkPMvcmRKHDEVLBW008Uj34ULXq59eYx3UzkjH+k1S1KL52hf5kFQ81ig60Ncu  
RASxetSP0w/BpuR1az8vrrrYuQgbhKPx3QaXTDQA0BaotBAFTrtRnXja831JL  
NbZ5Xxgk2cc/187qNn3esryEDPHV2C6JeLzKTQASRuvawskuSk2p4inThee  
ElVi1cAU5k7eShriw053nywBld+1YFB0Q5QutcvOr17Qzlwq79Unh+2A7Ju5  
QH0k8/as5a/wp1PoFT39v9Bj;IghmVvNyce71d2zapwTxosMEMBrqt5dmzH4J  
bTt7x1y4nTH0k2Pd+x5FzBuOnIUVGe+Af14rUbzAwj5pLnxxkwY9Th/Vy7vI5  
Hfj506kn7WE8z5Q5hohNw1+QsH1pFD62mfD7g1t/4rsrct7vwLxxjYmwc1RntT0  
dtUDENEcg7YeAv:j=44w5At7y5O#/_i111ikQs99jQur/HVh643C8L25T1QH7YZ84V  
Pp7N2ktDB1msnB/u/bcettUrystJnkXqemw/ed0HfD8N31j08gpkP0dhInb  
zyBxx+51pm+1sqNwMsJ2/71bcEWkoJzdp21tEgjy-bmTRnb+rcGyAvca8  
eSHVxPd2z17y/yya6v7QPpJXympY1D9fAT580w7EHMlopQ0Ef/1lnScdEU  
UAV1n37yv6gaClbvT31VE1NeGY1074+c0IB/0/z2Gfj1iu188+ilkyteq31Pd  
0nsi3*+axTRBuAltdsX7AGVTPVa/os9vBOCF24+cgYEA1v1q5071225+4npADX  
dASUE64L3OMUr/x/nwxNsVEX4Lr78a4l,q/6PrhnzoJ6s/Z8V4hs41HrSK0Vz2w  
snBs38y+f61LFxG4PdHde-1/4vqof83bPX92<xAhcLafuk79kmXLqsa1C8  
4Vciy14eyuq+07abj5N9nx/ECgyEAotZmXmWzN3zjzxzAn2xuaedpWn16GkLmngK  
sk3+0SkH1g3l3IAhNIV117pTVOcygCn10QhgNgCED16p+FAluNFdh831mPl+Y  
erFT1uy51BH70w1701ay14n/gv0CC4qZjKrQgsSB0jE4Mxi9X7tgtX163qH21v  
jb-B36UcgYEaj1290P61+ht+Bw64F/GvD2rz11hCYMvJy7N7sLB7kuM1L99h  
RP2uz08CJArgrNmynurGrYOgkgy4QcQdy63qTbUSQdbd9jWcLmg5Kt4+F9z  
Vs2teok/J1i03d7egI0mOnr/oJjzh2ANr/NP8AQ80hI8nMenetNQKc  
-----END RSA PRIVATE KEY-----  
-- INSERT --
```

23) Once the key is copied, use the command '**chmod 600 newkey.pem**' to change permissions of the file. Then run the following command to take remote access of private instance on our public instance:

- ssh -i newkey.pem ec2-user@private_ip_private_instance

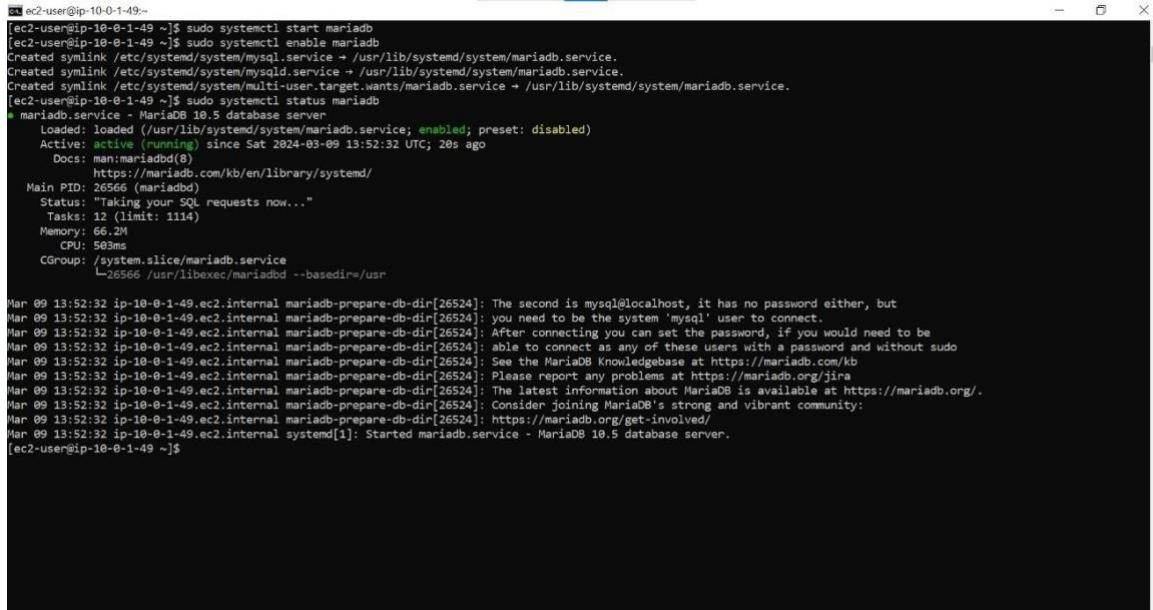
24) Now, to create a database download the mariadb package using the command:

- sudo yum install mariadb105-server -y

```
[ec2-user@ip-10-0-1-49 ~]$ [ec2-user@ip-10-0-1-49 ~]$ sudo yum install mariadb105-server -y Last metadata expiration check: 0:05:10 ago on Sat Mar  9 13:46:23 2024. Dependencies resolved. Package           Architecture      Version       Repository  Size ===== Installing: mariadb105-server          x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   11 M mariadb-connector-c             x86_64        3.1.13-1.amzn2023.0.3     amazonlinux   196 k mariadb-connector-c-config      noarch       3.1.13-1.amzn2023.0.3     amazonlinux   9.2 k mariadb105                  x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   1.6 M mariadb105-common              x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   38 k mariadb105-ermsg                x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   214 k mysql-selinux                 noarch       1.8.4-2.amzn2023.0.3      amazonlinux   36 k perl-B                       x86_64        1.88-477.amzn2023.0.6     amazonlinux   179 k perl-DBD-MariaDB               x86_64        1.22-1.amzn2023.0.4      amazonlinux   153 k perl-DBI                      x86_64        1.643-7.amzn2023.0.3     amazonlinux   700 k perl-Data-Dumper                x86_64        2.174-460.amzn2023.0.2    amazonlinux   55 k perl-File-Copy                 noarch       2.34-477.amzn2023.0.6     amazonlinux   29 k perl-FileHandle                noarch       2.01-477.amzn2023.0.6     amazonlinux   16 k perl-Math-BigInt               noarch       1:1.9998-18.458.amzn2023.0.2 amazonlinux   189 k perl-Math-Complex               noarch       1.59-477.amzn2023.0.6     amazonlinux   47 k perl-Sys-Hostname               x86_64        1.23-477.amzn2023.0.6     amazonlinux   18 k perl-base                     noarch       2.27-477.amzn2023.0.6     amazonlinux   17 k Installing weak dependencies: mariadb105-backup                x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   6.3 M mariadb105-cracklib-password-check x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   16 k mariadb105-gssapi-server          x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   18 k mariadb105-server-utils          x86_64        3:10.5.23-1.amzn2023.0.1    amazonlinux   216 k Transaction Summary ===== Install 21 Packages Total download size: 20 M Installed size: 117 M Downloading Packages: (1/21): mariadb105-cracklib-password-check-10.5.23-1.amzn2023.0.1.x86_64.rpm 212 kB/s | 16 kB  00:00 (2/21): perl-B-1.88-477.amzn2023.0.6.x86_64.rpm 6.5 MB/s | 179 kB  00:00
```

25) Start and enable the service once the package is downloaded.

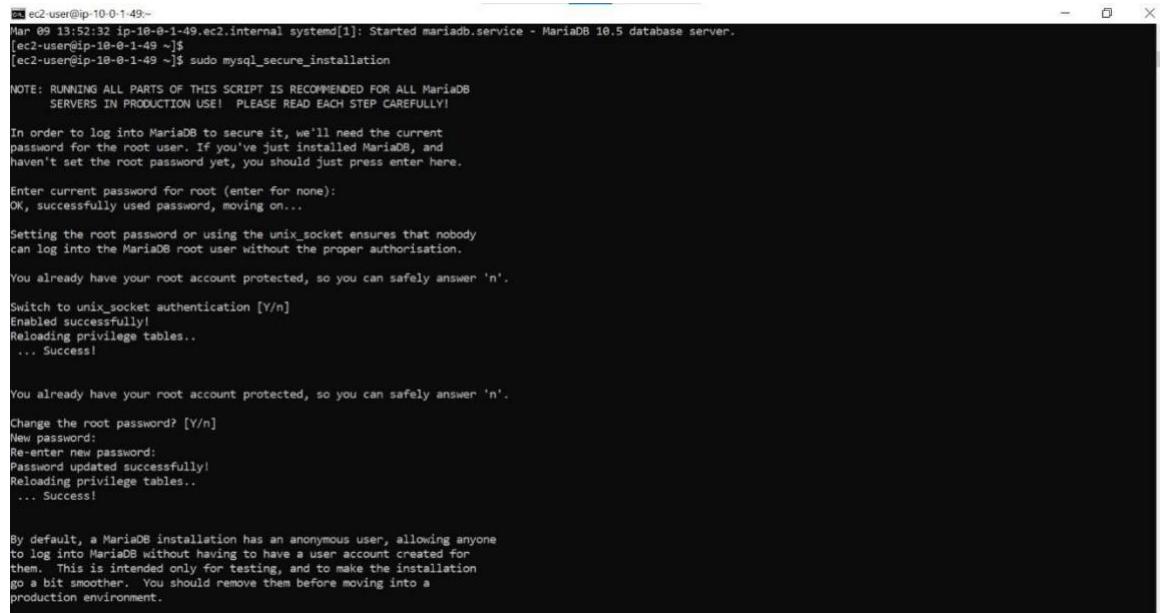
- sudo systemctl start mariadb
- sudo systemctl enable mariadb



```
[ec2-user@ip-10-0-1-49:~]$ sudo systemctl start mariadb
[ec2-user@ip-10-0-1-49:~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-10-0-1-49:~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
    Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
      Active: active (running) since Sat 2024-03-09 13:52:32 UTC; 20s ago
        Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
     Main PID: 26566 (mariadb)
       Status: "Taking your SQL requests now..."
          Tasks: 12 (limit: 1114)
         Memory: 66.2M
            CPU: 503ms
           CGroup: /system.slice/mariadb.service
                   └─26566 /usr/libexec/mariadb --basedir=/usr

Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: The second is mysql@localhost, it has no password either, but
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: you need to be the system 'mysql' user to connect.
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: After connecting you can set the password, if you would need to be
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: able to connect as any of these users with a password and without sudo
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: See the MariaDB Knowledgebase at https://mariadb.com/kb
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: Please report any problems at https://mariadb.org/jira
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: The latest information about MariaDB is available at https://mariadb.org/.
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: Consider joining MariaDB's strong and vibrant community;
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal mariadb-prepare-db-dir[26524]: https://mariadb.org/get-involved/
Mar 09 13:52:32 ip-10-0-1-49.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.
[ec2-user@ip-10-0-1-49:~]$
```

26) Using the ‘sudo mysql_secure_installation’ command change the root user password to ‘1’.



```
[ec2-user@ip-10-0-1-49:~]
[ec2-user@ip-10-0-1-49:~]$ 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]
Enabled successfully!
Reloading privilege tables..
... Success!

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

Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

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.
```

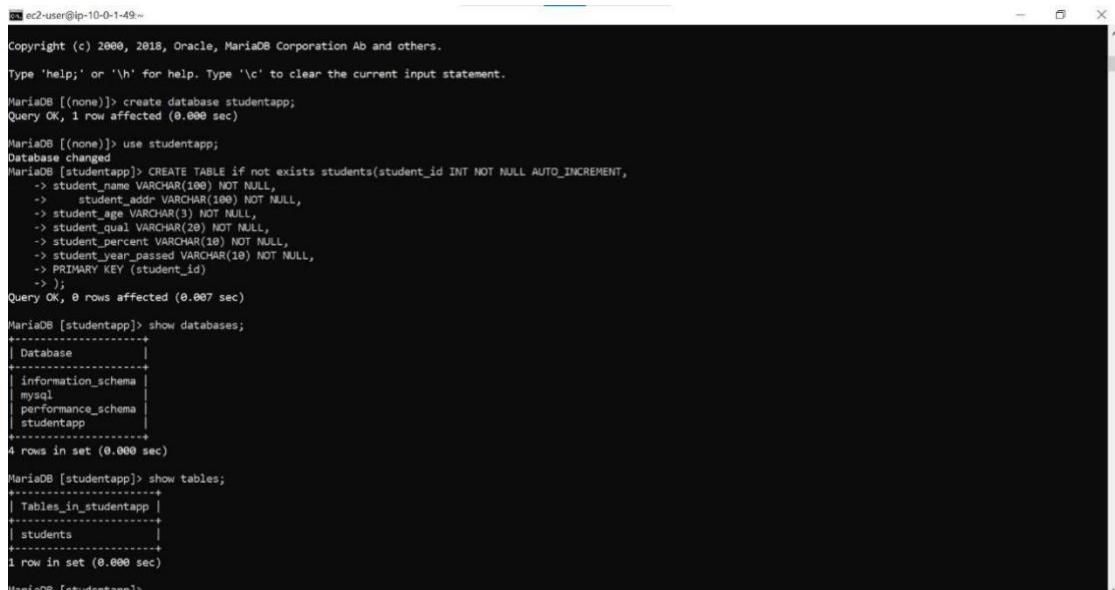
27) Login to mariadb server using the ‘mysql -u root -p1’ command. Create a database named studentapp and a students table in it to store the data. Refer to the code bellow:

```
create database
```

```
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));
```



```
ec2-user@ip-10-0-1-49:~  
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 studentapp;  
Query OK, 1 row affected (0.000 sec)  
MariaDB [(none)]> use studentapp;  
Database changed  
MariaDB [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)  
-> );  
Query OK, 0 rows affected (0.007 sec)  
MariaDB [studentapp]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| studentapp |  
+-----+  
4 rows in set (0.000 sec)  
MariaDB [studentapp]> show tables;  
+-----+  
| Tables_in_studentapp |  
+-----+  
| students |  
+-----+  
1 row in set (0.000 sec)  
MariaDB [studentapp]> _
```

28) We need to download apache tomcat package to host the webpage, use command:

- curl -O download_link_from_browser

A zip file will be downloaded. Unzip the file using the ‘unzip’ command.

```
[ec2-user@ip-10-0-1-49 ~]$ curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.86/bin/apache-tomcat-9.0.86.zip
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100 11.7M  100 11.7M    0     0  10.6M  0:00:01  0:00:01  --:-- 10.6M
[ec2-user@ip-10-0-1-49 ~]$ unzip apache-tomcat-9.0.86.zip
```

29) Next we need to download the files from bucket in the instance. Use command '**aws s3 ls**' to check if there is connection between ec2 instance and s3 bucket. Use the following command to download file from bucket into the terminal:

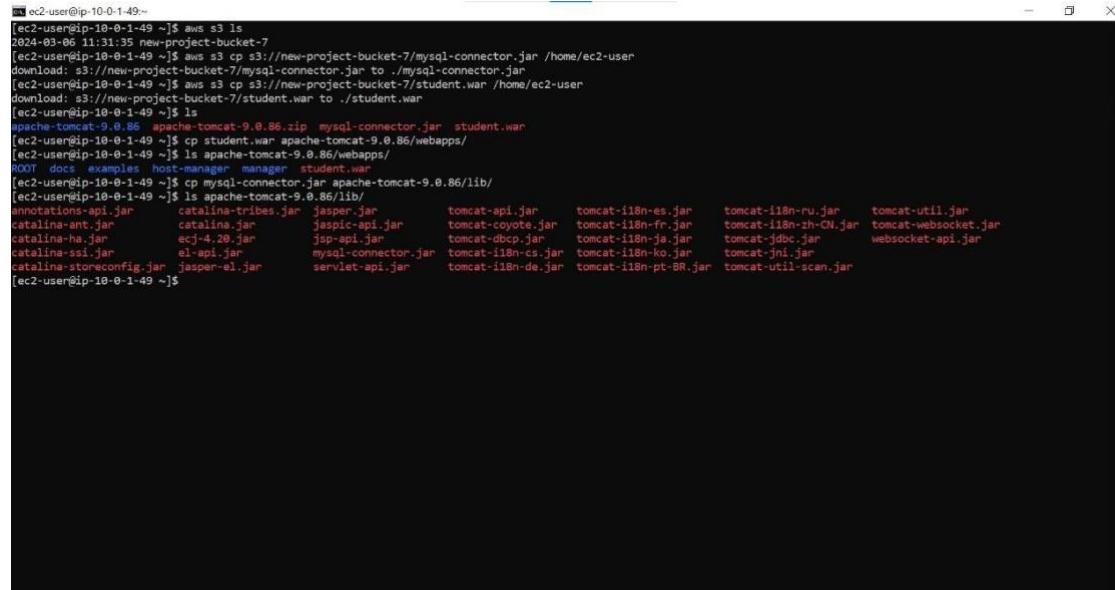
- aws s3 cp S3_URI /home/ec2-user

Once the files are downloaded, copy web page file in webapps directory and connector file in lib directory using the following commands:

- cp student.war apache-tomcat-9.0.86/webapps/

- cp mysql-connector.jar apache-tomcat-9.0.86/lib/

Check is files have been copied using the '**ls**' command.



```
[ec2-user@ip-10-0-1-49 ~]$ aws s3 ls
2024-03-06 11:31:35 new-project-bucket-7
[ec2-user@ip-10-0-1-49 ~]$ aws s3 cp s3://new-project-bucket-7/mysql-connector.jar /home/ec2-user
download: s3://new-project-bucket-7/mysql-connector.jar to ./mysql-connector.jar
[ec2-user@ip-10-0-1-49 ~]$ aws s3 cp s3://new-project-bucket-7/student.war /home/ec2-user
download: s3://new-project-bucket-7/student.war to ./student.war
[ec2-user@ip-10-0-1-49 ~]$ ls
apache-tomcat-9.0.86 apache-tomcat-9.0.86.zip mysql-connector.jar student.war
[ec2-user@ip-10-0-1-49 ~]$ cp student.war apache-tomcat-9.0.86/webapps/
[ec2-user@ip-10-0-1-49 ~]$ ls apache-tomcat-9.0.86/webapps/
ROOT doc examples hostmanager manager student.war
[ec2-user@ip-10-0-1-49 ~]$ cd apache-tomcat-9.0.86/lib/
[ec2-user@ip-10-0-1-49 ~]$ ls apache-tomcat-9.0.86/lib/
annotation-api.jar catalina-tribes.jar jasper.jar tomcat-api.jar tomcat-i18n-es.jar tomcat-i18n-ru.jar tomcat-util.jar
catalinelistener.jar catalina-jmx.jar jasper-api.jar tomcat-coyote.jar tomcat-i18n-fr.jar tomcat-i18n-zh-CN.jar tomcat-websocket.jar
catalina-ha.jar ecj-4.20.jar jsp-api.jar tomcat-dbcp.jar tomcat-i18n-ja.jar tomcat-jdbc.jar websocket-api.jar
catalina-sel.jar el-api.jar mysql-connector.jar tomcat-i18n-cs.jar tomcat-i18n-ko.jar tomcat-jni.jar
catalina-storeconfig.jar jasper-el.jar servlet-api.jar tomcat-i18n-de.jar tomcat-i18n-pi-BR.jar tomcat-util-scan.jar
[ec2-user@ip-10-0-1-49 ~]$
```

30) Using '**sudo vim apache-tomcat-9.0.86/conf/context.xml**' open the xml file and enter the following code on line 20 :

```
<Resource name="jdbc/TestDB" auth="Container"
    type="javax.sql.DataSource" maxTotal="100"
    maxIdle="30" maxWaitMillis="10000"
    username="root" password="1"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/studentapp"/>
```

```
cc2-user@ip-10-0-1-49:~
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Licensed to the Apache Software Foundation (ASF) under one or more
4 contributor license agreements. See the NOTICE file distributed with
5 this work for additional information regarding copyright ownership.
6 The ASF licenses this file to you under the Apache License, Version 2.0
7 (the "License"); you may not use this file except in compliance with
8 the License. You may obtain a copy of the License at
9
10   http://www.apache.org/licenses/LICENSE-2.0
11
12 Unless required by applicable law or agreed to in writing, software
13 distributed under the License is distributed on an "AS IS" BASIS,
14 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 See the License for the specific language governing permissions and
16 limitations under the License.
17 -->
18 <!-- The contents of this file will be loaded for each web application -->
19 <Context>
20   <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
21     maxTotal="100" maxIdle="50" maxWaitMillis="10000"
22     username="root" password="1" driverClassName="com.mysql.jdbc.Driver"
23     url="jdbc:mysql://localhost:3306/studentapp" />
24   <!-- Default set of monitored resources. If one of these changes, the -->
25   <!-- web application will be reloaded. -->
26   <WatchedResource>WEB-INF/web.xml</WatchedResource>
27   <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
28   <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
29
30   <!-- Uncomment this to disable session persistence across Tomcat restarts -->
31   <!--
32   <Manager pathname="" />
33   -->
34 </Context>
```

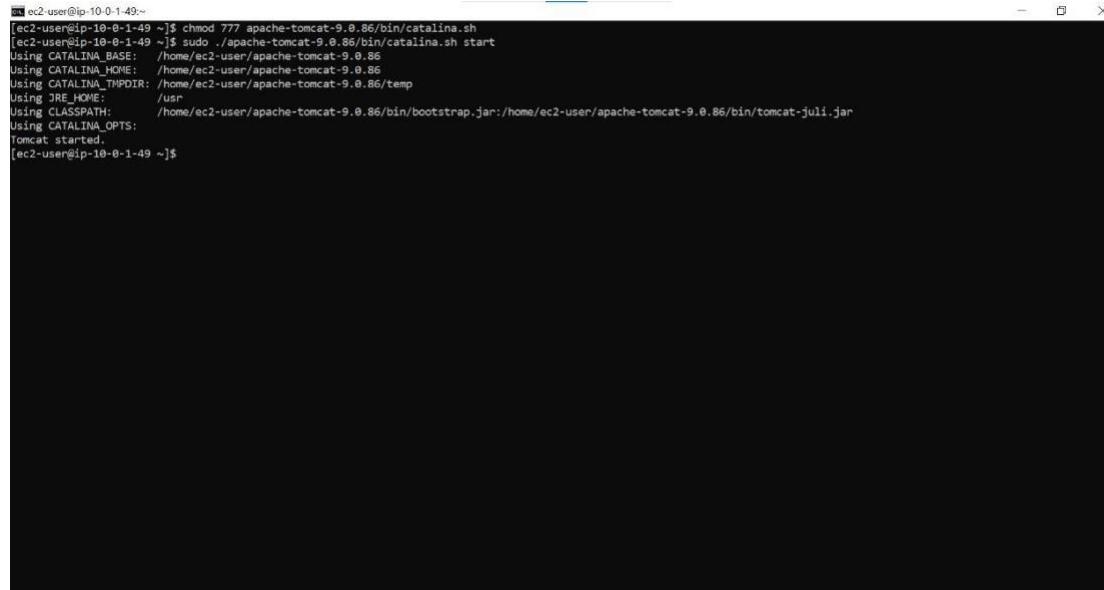
31) Download java package '**sudo yum install java -y**'. We need this package to start the apache tomcat server.

```
[ec2-user@ip-10-0-1-49 ~]$ sudo vim apache-tomcat-9.0.86/conf/context.xml
[ec2-user@ip-10-0-1-49 ~]$ 
[ec2-user@ip-10-0-1-49 ~]$ sudo yum install java -y
Last metadata expiration check: 0:17:23 ago on Sat Mar  9 13:46:23 2024.
Dependencies resolved.
=====
Package          Architecture Version      Repository   Size
=====
Installing:
java-21-amazon-corretto x86_64       1:21.0.2+14-1.amzn2023.1 amazonlinux 213 k
Installing dependencies:
  alselib           x86_64       1.2.7.2-1.amzn2023.0.2 amazonlinux 504 k
  cairo             x86_64       1.17.6-2.amzn2023.0.1 amazonlinux 684 k
  dejavu-sans-fonts noarch        2.37-16.amzn2023.0.2 amazonlinux 1.3 M
  dejavu-sans-mono-fonts noarch        2.37-16.amzn2023.0.2 amazonlinux 467 k
  dejavu-serif-fonts noarch        2.37-16.amzn2023.0.2 amazonlinux 1.0 M
  fontconfig         x86_64       2.13.94-2.amzn2023.0.2 amazonlinux 273 k
  fonts-filesystem  noarch        1:2.0.5-12.amzn2023.0.2 amazonlinux 9.5 k
  freetype           x86_64       2.13.8-2.amzn2023.0.1 amazonlinux 422 k
  giflib             x86_64       5.2.1-9.amzn2023.0.1 amazonlinux 49 k
  google-noto-fonts-common noarch        30201200.2.amzn2023.0.2 amazonlinux 15 k
  google-noto-sans-vf-fonts noarch        30201200.2.amzn2023.0.2 amazonlinux 493 k
  graphite2          x86_64       1.3.14-7.amzn2023.0.2 amazonlinux 97 k
  harfbuzz            x86_64       7.0.0-2.amzn2023.0.1 amazonlinux 868 k
Java-21-amazon-corretto-headless x86_64       1:21.0.2+14-1.amzn2023.1 amazonlinux 97 M
javapackages-filesystem noarch        6.0.0-7.amzn2023.0.6 amazonlinux 12 k
langpacks-core-font-en noarch        3.0-21.amzn2023.0.4 amazonlinux 16 k
libICE              x86_64       1.0.10-6.amzn2023.0.2 amazonlinux 73 k
libSM               x86_64       1.2.3-8.amzn2023.0.2 amazonlinux 42 k
libX11              x86_64       1.7.2-3.amzn2023.0.4 amazonlinux 657 k
libX11-common        noarch        1.7.2-3.amzn2023.0.4 amazonlinux 152 k
libXau              x86_64       1.0.9-6.amzn2023.0.2 amazonlinux 31 k
libXext              x86_64       1.3.4-6.amzn2023.0.2 amazonlinux 41 k
libXi               x86_64       1.7.10-6.amzn2023.0.2 amazonlinux 46 k
libXinerama          x86_64       1.1.4-8.amzn2023.0.2 amazonlinux 15 k
libXrandr             x86_64       1.5.2-6.amzn2023.0.2 amazonlinux 28 k
libXrender            x86_64       0.9.10-14.amzn2023.0.2 amazonlinux 28 k
libXt               x86_64       1.2.0-4.amzn2023.0.2 amazonlinux 181 k
libXtst              x86_64       1.2.3-14.amzn2023.0.2 amazonlinux 21 k
libbrotli            x86_64       1.0.9-4.amzn2023.0.2 amazonlinux 315 k
libjpeg-turbo         x86_64       2.1.4-2.amzn2023.0.5 amazonlinux 196 k
```

32) Use the command '**chmod 777 apache-tomcat-**

9.0.86/bin/catalina.sh' to change permissions of the catalina file, then start the tomcat server:

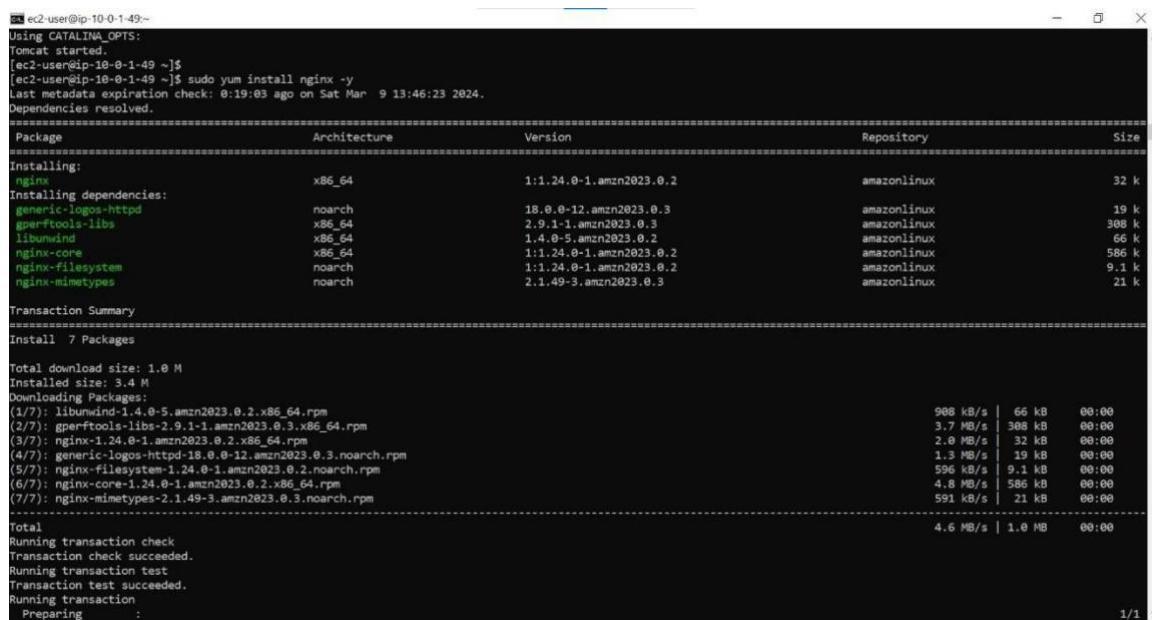
- sudo ./apache-tomcat-9.0.86/bin/catalina.sh start



```
[ec2-user@ip-10-0-1-49 ~]$ chmod 777 apache-tomcat-9.0.86/bin/catalina.sh
[ec2-user@ip-10-0-1-49 ~]$ sudo ./apache-tomcat-9.0.86/bin/catalina.sh start
Using CATALINA_BASE: /home/ec2-user/apache-tomcat-9.0.86
Using CATALINA_HOME: /home/ec2-user/apache-tomcat-9.0.86
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.86/temp
Using JRE_HOME: /usr
Using CLASSPATH:
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-10-0-1-49 ~]$
```

33) Now we need the nginx package to pass the incoming request from port 80 to port 8080.

- sudo yum install nginx -y



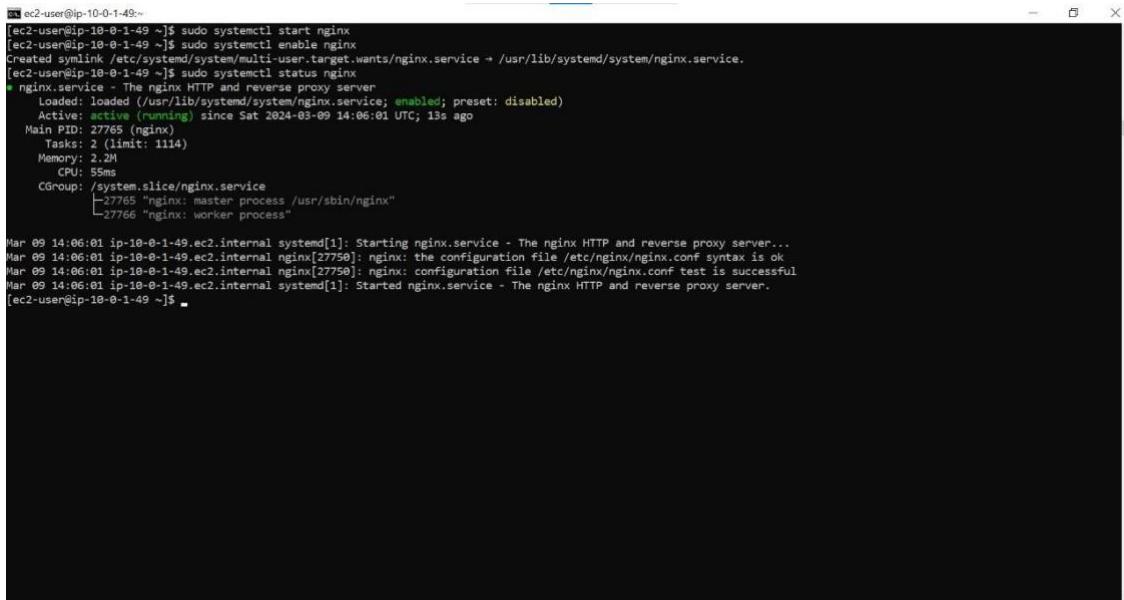
```
[ec2-user@ip-10-0-1-49 ~]
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-10-0-1-49 ~]$ [ec2-user@ip-10-0-1-49 ~]$ sudo yum install nginx -y
Last metadata expiration check: 0:19:03 ago on Sat Mar  9 13:46:23 2024.
Dependencies resolved.
=====
Transaction Summary
=====
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm           908 kB/s | 66 kB   00:00
(2/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm      3.7 MB/s | 308 kB   00:00
(3/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm             2.0 MB/s | 32 kB   00:00
(4/7): generic-logos-https-18.0.0-12.amzn2023.0.3.noarch.rpm 1.3 MB/s | 19 kB   00:00
(5/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm     596 kB/s | 9.1 kB   00:00
(6/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm          4.8 MB/s | 586 kB   00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm     591 kB/s | 21 kB   00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 4.6 MB/s | 1.0 MB   00:00
  1/1
```

34) Start and enable the nginx service:

- **sudo systemctl start nginx**

- **sudo systemctl enable nginx**



```
[ec2-user@ip-10-0-1-49 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-1-49 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-03-09 14:06:01 UTC; 15s ago
     Main PID: 27765 (nginx)
       Tasks: 2 (limit: 1114)
      Memory: 2.2M
        CPU: 55ms
       CGroup: /system.slice/nginx.service
           ├─27765 "nginx: master process /usr/sbin/nginx"
           ├─27766 "nginx: worker process"

Mar 09 14:06:01 ip-10-0-1-49.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Mar 09 14:06:01 ip-10-0-1-49.ec2.internal nginx[27765]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Mar 09 14:06:01 ip-10-0-1-49.ec2.internal nginx[27765]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Mar 09 14:06:01 ip-10-0-1-49.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-1-49 ~]$
```

35) Open the nginx configuration file using '**sudo vim**

/etc/nginx/nginx.conf command and enter the following code on line 45:

```
location / {
    proxy_pass http://localhost:8080/student/;
}
```

Save the file and restart the nginx service. (**sudo systemctl restart nginx**)

```
ec2-user@ip-10-0-1-49:~
```

```
16
17 http {
18     log_format main '$remote_addr - $remote_user [$time_local] "$request" '
19             '$status $body_bytes_sent "'.$http_referer"
20             '"$http_user_agent" "$http_x_forwarded_for"';
21
22     access_log /var/log/nginx/access.log main;
23
24     sendfile      on;
25     tcp_nopush   on;
26     keepalive_timeout 65;
27     types_hash_max_size 4096;
28
29     include       /etc/nginx/mime.types;
30     default_type application/octet-stream;
31
32     # Load modular configuration files from the /etc/nginx/conf.d directory.
33     # See http://nginx.org/en/docs/ngx_core_module.html#include
34     # for more information.
35     include /etc/nginx/conf.d/*.conf;
36
37     server {
38         listen       80;
39         listen       [::]:80;
40         server_name  ~^;
41         root        /usr/share/nginx/html;
42
43         # Load configuration files for the default server block.
44         include /etc/nginx/default.d/*.conf;
45         location / {
46             proxy_pass http://localhost:8080/student/;
47
48             error_page 404 /404.html;
49             location = /404.html {
50             }
51
52             error_page 500 502 503 504 /50x.html;
53             location = /50x.html {
54             }
55         }
56     }
57 }
58
59 :set nu
```

47,2-9

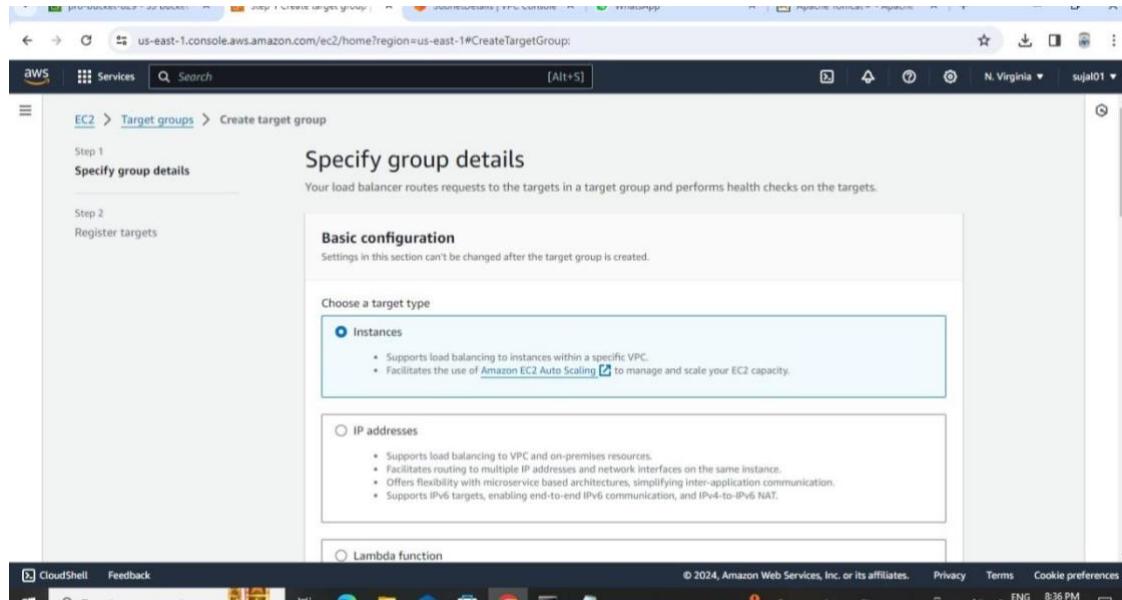
33%

36) Now run the ‘curl localhost’ command, you will get an output that is the code of our student registration page. This output is a indication that we have completed all the configurations correctly.

```
ec2-user@ip-10-0-1-49:~$ sudo systemctl restart nginx
[ec2-user@ip-10-0-1-49 ~]$ curl localhost
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>User Data</title>
</head>
<style>
div.ex {
    text-align: right width:300px;
    padding: 10px;
    border: 5px solid grey;
    margin: 0px
}
</style>
<body>
<h1>Student Registration Form</h1>
<div class="ex">
    <form action="registrationController" method="post">
        <table style="width: 50%">
            <tr>
                <td>Student Name</td>
                <td><input type="text" name="fullname"/></td>
            </tr>
            <tr>
                <td>Student Address</td>
                <td><input type="text" name="address"/></td>
            </tr>
            <tr>
                <td>Student Age</td>
                <td><input type="text" name="age"/></td>
            </tr>
            <tr>
                <td>Student Qualification</td>
                <td><input type="text" name="qual"/></td>
            </tr>
            <tr>
                <td>Student Percentage</td>
                <td><input type="text" name="percent"/></td>
            </tr>
        </table>
    </form>
</div>

```

37) In the aws console, go to target groups dashboard under EC2 service and click on create target group. Enter a valid target group name (project-tg) and select our custom VPC in the VPC drop-down. Click on next.



38) Add only the private instance as target and click on create target group.

The screenshot shows the 'Create target group' wizard in the AWS EC2 console. The current step is 'Specify group details'. A table titled 'Available instances (1/2)' lists two instances:

Instance ID	Name	State	Security groups
i-049b1aa0b54477ea	public-pro	Running	my-security-pro
i-0d73a07bdcd73fee	private-pro	Running	my-security-pro

The instance 'i-0d73a07bdcd73fee' is selected, indicated by a checked checkbox. Below the table, it says '1 selected'. Under 'Ports for the selected instances', the port '80' is listed.

39) Now go to load balancers dashboard and click on create load balancer. Under Application Load Balancer click on create. Enter a valid name (sample-lb) and choose scheme as 'internet-facing'.

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS EC2 console. The current step is 'Basic configuration'. The 'Load balancer name' field contains 'project-lb'. The 'Scheme' section shows 'Internet-facing' is selected, with a note that it routes requests from clients over the internet to targets. Other options like 'Internal' are also shown.

40) Select our custom VPC and check both the availability zones under mappings.

The screenshot shows the 'Network mapping' step of the Create Load Balancer Wizard. In the 'VPC Info' section, 'my-vpc-b29' is selected. In the 'Mappings Info' section, 'us-east-1a (use1-az6)' is checked, and a warning message states: 'The selected subnet does not have a route to an internet gateway. This means that your load balancer will not receive internet traffic. You can proceed with this selection; however, for internet traffic to reach your load balancer, you must update the subnet's route table in the VPC console.' The subnet listed is 'subnet-0391c3ad22dceef97'.

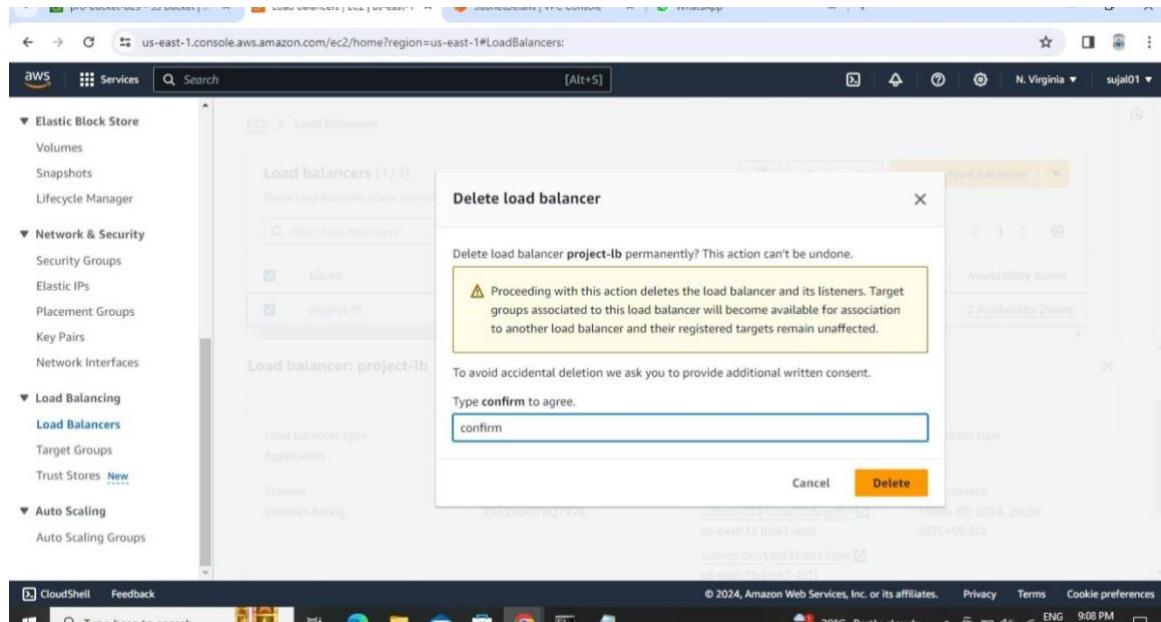
41) Under security group choose the group that we created (launch-wizaard-7) and add the newly created target group (project-tg) in listeners and routing section. Click on create load balancer.

The screenshot shows the 'Security groups' step of the Create Load Balancer Wizard. A security group 'my-security-pro' is selected. In the 'Listeners and routing' section, a new listener 'HTTP:80' is being configured. The protocol is set to 'HTTP' and the port is '80'. The default action is 'Forward to' the target group 'project-tg', which is defined as 'Target type: Instance, IPv4'. A link 'Create target group' is visible below the configuration.

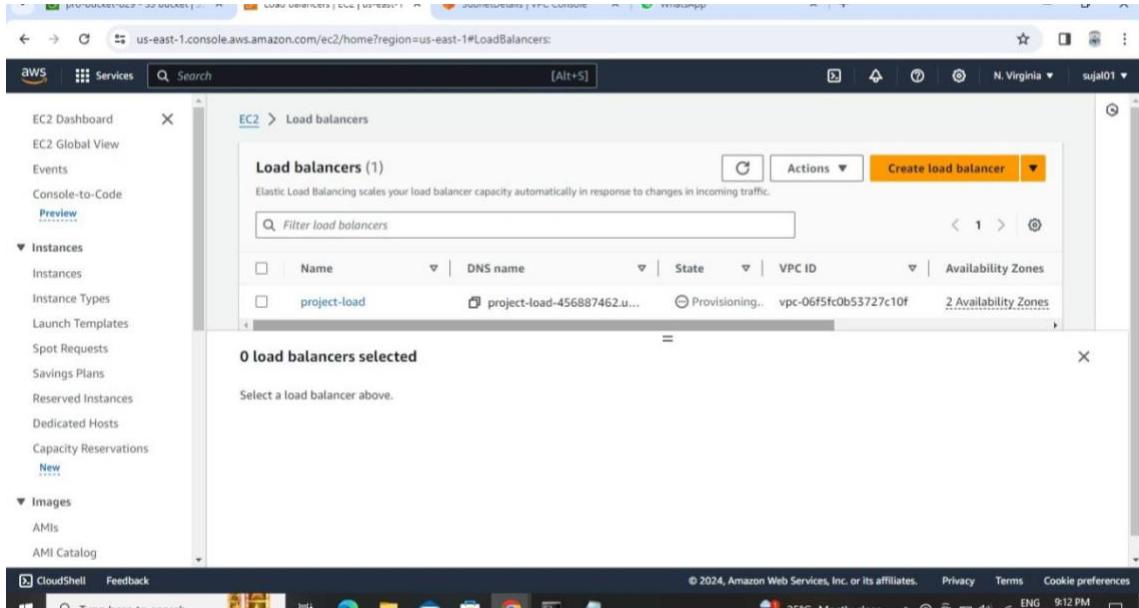
42) Once the load balancer is created and goes into active state, copy the dns name and paste it in a new incognito window. You will see that the web page loads successfully and you can also register the users here.

Student ID	StudentName	Student Addrs	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	vaibhav	dhagat	89	7th	25	1999	edit	delete

43) On the load balancers dashboard delete the newly created load balancer (sample-lb). We only needed the load balancer for testing purpose.

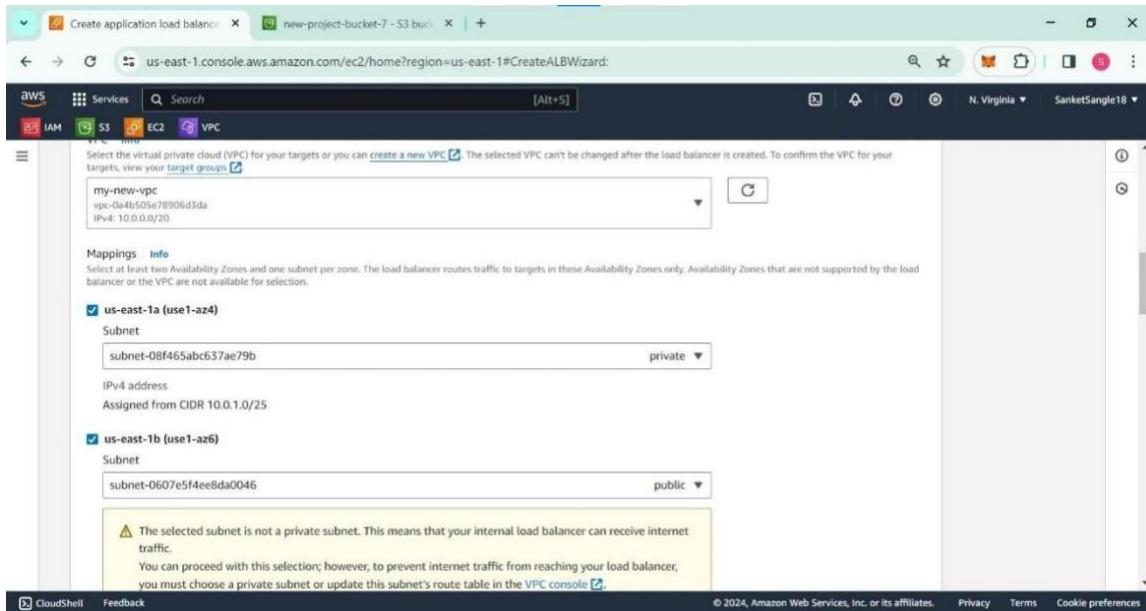


44) Again, on load balancers dashboard click on create load balancer. Under Application Load Balancer click on create. Enter a valid name (project-lb) and this time choose scheme as ‘internal’.



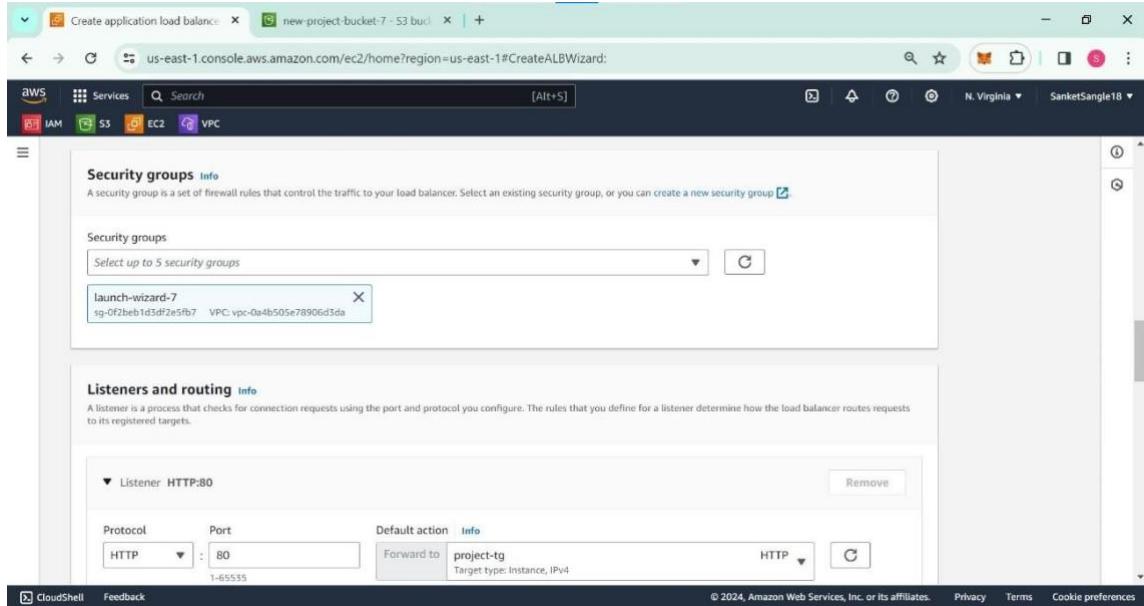
The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a sidebar with 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Console-to-Code Preview', 'Instances' (selected), 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMIs', and 'AMI Catalog'. The main content area is titled 'Load balancers (1)' and shows a table with one row for 'project-load'. The table columns are Name, DNS name, State, VPC ID, and Availability Zones. The 'Availability Zones' column shows '2 Availability Zones'. Below the table, a message says '0 load balancers selected' and 'Select a load balancer above.' At the top right, there are 'Actions' and 'Create load balancer' buttons. The status bar at the bottom indicates it's from 2024, and the user is in the N. Virginia region.

45) Select our custom VPC and check both the availability zones under mappings.

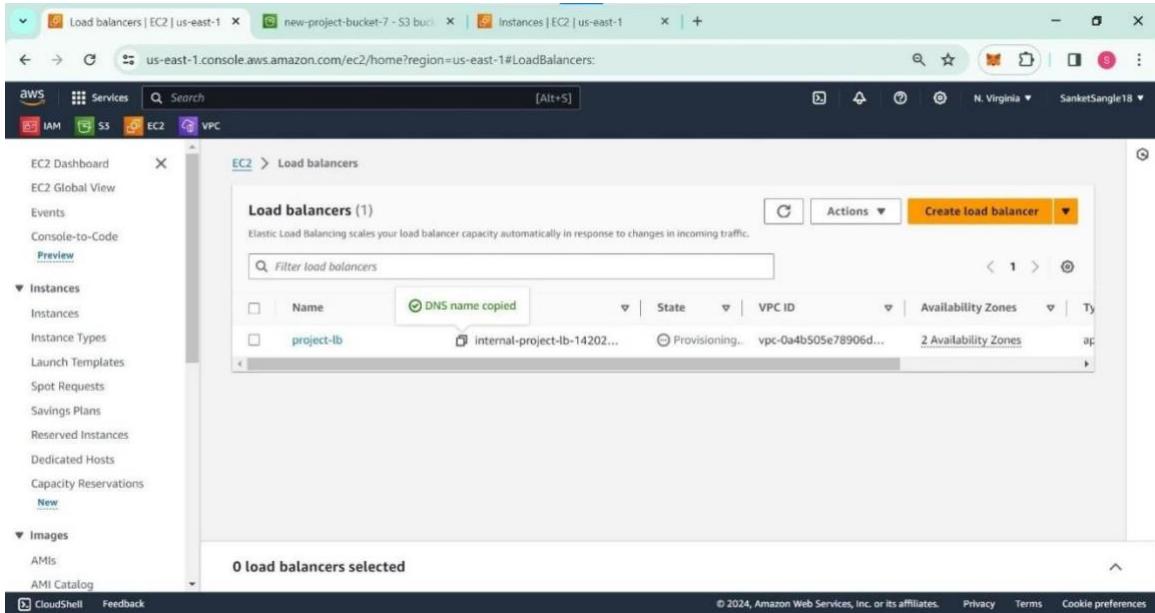


The screenshot shows the 'Create application load balance' wizard on the 'Mappings' step. The top navigation bar includes tabs for 'Create application load balance', 'new-project-bucket-7 - S3 bucket', and '+'. The sidebar shows 'Services' like IAM, S3, EC2 (selected), and VPC. The main content area has a heading 'Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups.' A dropdown menu shows 'my-new-vpc' with details: 'vpc-0a4b505e78906d3da' and 'IPv4: 10.0.0.0/20'. Below this, the 'Mappings' section lists two target groups: 'us-east-1a (use1-az4)' and 'us-east-1b (use1-az6)'. Each group has a 'Subnet' dropdown. 'us-east-1a' is set to 'private' and 'us-east-1b' is set to 'public'. A warning message in a yellow box states: '⚠ The selected subnet is not a private subnet. This means that your internal load balancer can receive internet traffic. You can proceed with this selection; however, to prevent internet traffic from reaching your load balancer, you must choose a private subnet or update this subnet's route table in the VPC console.' The status bar at the bottom indicates it's from 2024, and the user is in the N. Virginia region.

46) Under security group choose the group that we created (launch-wizaard-7) and add the newly created target group (project-tg) in listeners and routing section. Click on create load balancer.



47) Our load balancer is successfully created. Copy the dns name of load balancer.



48) Open cmd in the downloads folder where the private key is present.
Run the following command to connect to the public instance:

- **ssh -i instance_key.pem ec2-user@public_ip_public_instance**

Download the nginx package: **sudo yum install nginx -y**

```
ec2-user@ip-10-0-2-90:~$ sudo yum install nginx -y
Last metadata expiration check: 0:34:27 ago on Sat Mar  9 13:47:34 2024.
Dependencies resolved.
=====
Package          Architecture Version      Repository   Size
=====
Installing:
nginx           x86_64      1:1.24.0-1.amzn2023.0.2      amazonlinux 32 k
Installing dependencies:
generic-logos-httd noarch      18.0.0-12.amzn2023.0.3      amazonlinux 19 k
gperftools-libs x86_64      2.9.1-1.amzn2023.0.3      amazonlinux 308 k
libunwind        x86_64      1.4.0-5.amzn2023.0.2      amazonlinux 66 k
nginx-core       x86_64      1:1.24.0-1.amzn2023.0.2      amazonlinux 586 k
nginx-filesystem noarch      1:1.24.0-1.amzn2023.0.2      amazonlinux 9.1 k
nginx-mimetypes noarch      2.1.49-3.amzn2023.0.3      amazonlinux 21 k
Transaction Summary
=====
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm      7.4 MB/s | 586 kB  00:00
(2/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm    3.1 MB/s | 308 kB  00:00
(3/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm            1.4 MB/s | 32 kB  00:00
(4/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm         635 kB/s | 66 kB  00:00
(5/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm  700 kB/s | 9.1 kB  00:00
(6/7): generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch.rpm 99 kB/s | 19 kB  00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm   1.3 MB/s | 21 kB  00:00
Total                                         5.6 MB/s | 1.0 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
    Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
  Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
  Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
                                                               1/1
                                                               1/7
                                                               1/7
                                                               2/7
```

49) Start and enable the nginx service:

- **sudo systemctl start nginx**

- **sudo systemctl enable nginx**

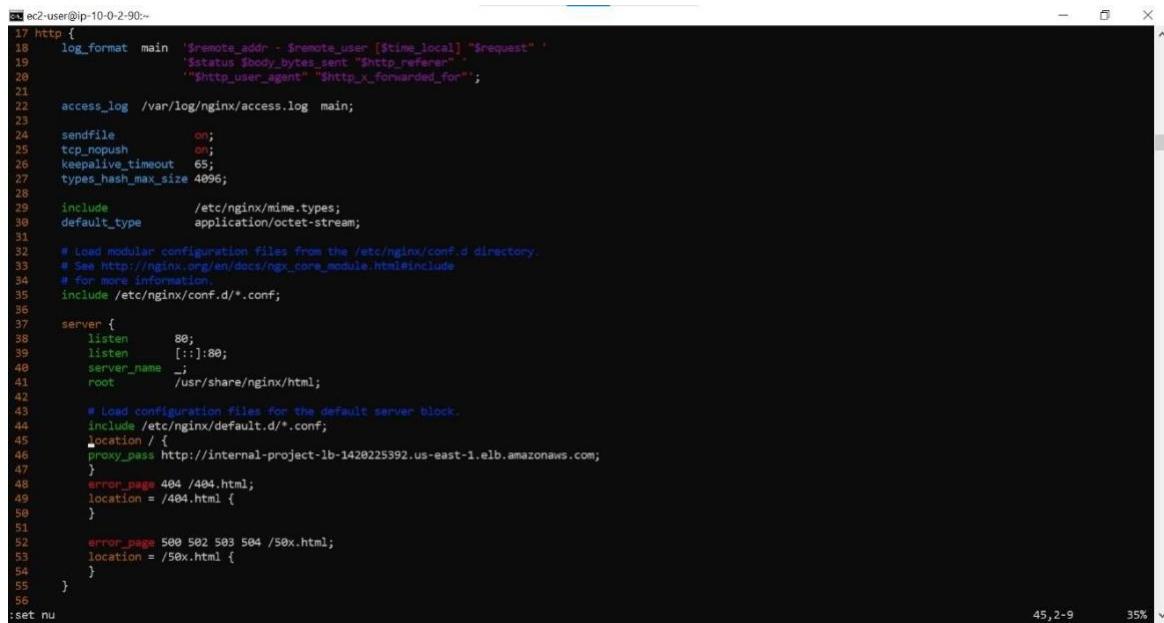
```
ec2-user@ip-10-0-2-90:~$ sudo systemctl start nginx
[ec2-user@ip-10-0-2-90 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-2-90 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-03-09 14:22:32 UTC; 15s ago
     Main PID: 26252 (nginx)
        Tasks: 2 (limit: 1114)
       Memory: 2.2M
          CPU: 56ms
         CGroup: /system.slice/nginx.service
             └─26252 "nginx: master process /usr/sbin/nginx"
                  ├─26253 "nginx: worker process"

Mar 09 14:22:32 ip-10-0-2-90.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Mar 09 14:22:32 ip-10-0-2-90.ec2.internal nginx[26258]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Mar 09 14:22:32 ip-10-0-2-90.ec2.internal nginx[26258]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Mar 09 14:22:32 ip-10-0-2-90.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-2-90 ~]$
```

50) Open the nginx configuration file using ‘**sudo vim /etc/nginx/nginx.conf**’ command and enter the following code on line 45:

```
location / {  
    proxy_pass http://dns_of_load_balancer;  
}
```

Save the file and restart the nginx service. (**sudo systemctl restart nginx**)



```
17 http {  
18     log_format main '$remote_addr - $remote_user [$time_local] "$request" '  
19             '$status $body_bytes_sent "$http_referer" '  
20             '"$http_user_agent" "$http_x_forwarded_for"';  
21  
22     access_log /var/log/nginx/access.log main;  
23  
24     sendfile      on;  
25     tcp_nopush    on;  
26     keepalive_timeout 65;  
27     types_hash_max_size 4096;  
28  
29     include       /etc/nginx/mime.types;  
30     default_type  application/octet-stream;  
31  
32     # Load modular configuration files from the /etc/nginx/conf.d directory.  
33     # See http://nginx.org/en/docs/ngx_core_module.html#include  
34     # for more information.  
35     include /etc/nginx/conf.d/*.conf;  
36  
37     server {  
38         listen      80;  
39         listen      [::]:80;  
40         server_name ;  
41         root        /usr/share/nginx/html;  
42  
43         # Load configuration files for the default server block.  
44         include /etc/nginx/default.d/*.conf;  
45         location / {  
46             proxy_pass http://internal-project-lb-1420225392.us-east-1.elb.amazonaws.com;  
47         }  
48         error_page 404 /404.html;  
49         location = /404.html {  
50         }  
51  
52         error_page 500 502 503 504 /50x.html;  
53         location = /50x.html {  
54         }  
55     }  
56 }
```

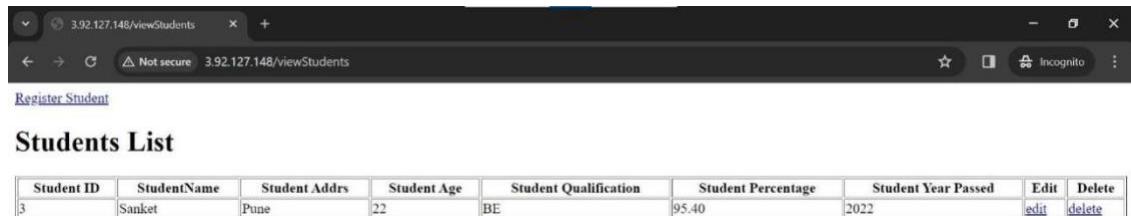
51) In chrome browser open new incognito window and paste the public ip address of public instance (project_public). Enter the student data and click on register.



Student Registration Form

Student Name	Sanket
Student Address	Pune
Student Age	22
Student Qualification	BE
Student Percentage	95.40
Year Passed	2022
<input type="button" value="register"/>	

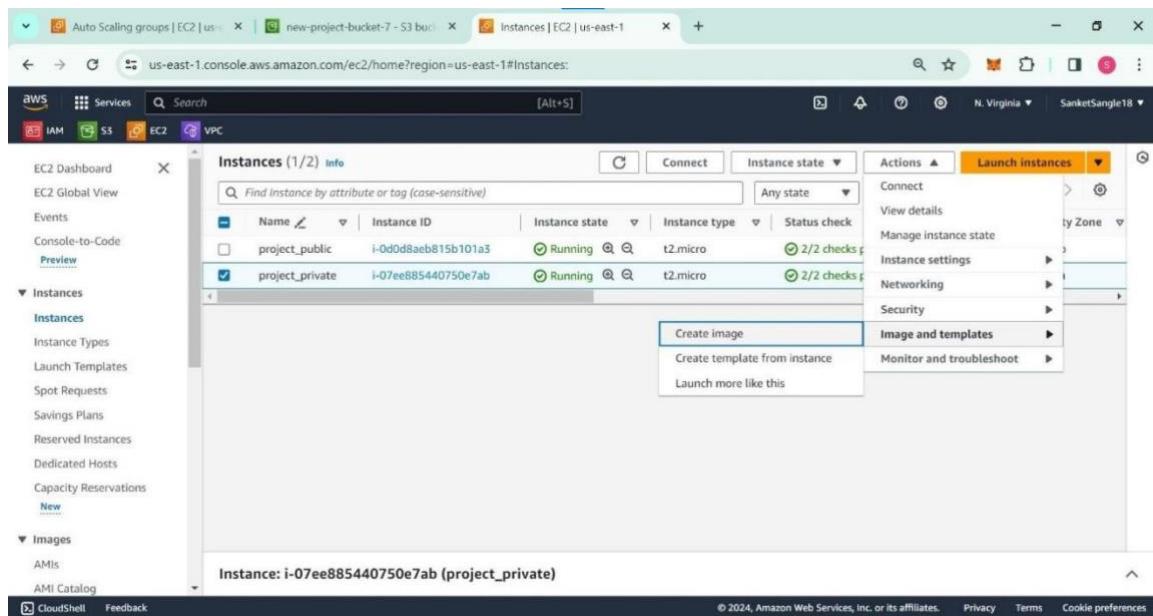
52) The student data has been successfully registered.



A screenshot of a web browser window titled "3.92.127.148/viewStudents". The page displays a table titled "Students List" with one row of data. The columns are: Student ID, StudentName, Student Addr, Student Age, Student Qualification, Student Percentage, Student Year Passed, Edit, and Delete. The data row is: 3, Sanket, Pune, 22, BE, 95.40, 2022, edit, delete.

Student ID	StudentName	Student Addr	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
3	Sanket	Pune	22	BE	95.40	2022	edit	delete

53) On instances dashboard select the private instance and click on actions then image and templates and click on create image.



54) In AMIs dashboard you can see that the image has been successfully created.

The screenshot shows the AWS EC2 Images (AMIs) dashboard. The left sidebar includes links for Instances, Images (selected), AMIs, and Elastic Block Store. The main content area displays a table titled 'Amazon Machine Images (AMIs) (1) Info'. The table has columns for Name, AMI name, AMI ID, Source, and Owner. One row is listed: 'Name' is 'project-ami', 'AMI name' is 'project-ami', 'AMI ID' is 'ami-05d2d0cf42e5971cd', 'Source' is '975049921144/project-ami', and 'Owner' is '975049921144'. Below the table, a modal window titled 'Select an AMI' is open, showing the same information. The bottom right corner of the screen shows the AWS footer with copyright information and links for Privacy, Terms, and Cookie preferences.

55) Go to launch templates dashboard under the EC2 service and click on create launch template. Enter a valid name (project-template) and add some description.

The screenshot shows the 'Create launch template' wizard. The left sidebar lists 'Launch template name and description', 'Template version description', 'Auto Scaling guidance', 'Template tags', and 'Source template'. The main content area has sections for 'Summary' (Software Image (AMI), Virtual server type (instance type), Firewall (security group), Storage (volumes)), 'Free tier' (info about 750 hours of t2.micro usage), and 'Actions' (Cancel, Create launch template). The 'Create launch template' button is highlighted with a yellow background. The bottom right corner shows the AWS footer.

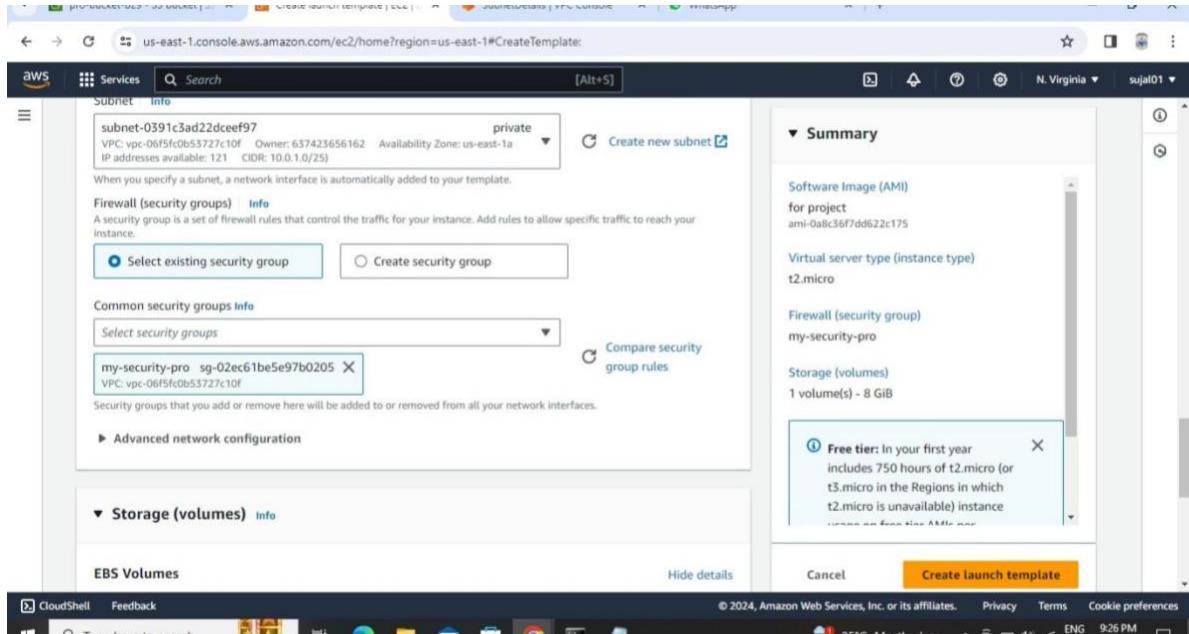
56) Select the newly created AMI under My AMIs section.

The screenshot shows the AWS EC2 console with the 'My AMIs' tab selected. A new AMI named 'project-ami' is listed, showing its creation date (2024-03-09T14:54:36.000Z), virtualization type (hvm), ENA enabled status (true), and root device type (ebs). Below the AMI list, there's a 'Description' field containing the placeholder 'for project'. On the right side, there's a 'Summary' panel with sections for Software Image (AMI), Virtual server type (instance type), Firewall (security group), and Storage (volumes). A tooltip for the 'Free tier' is visible, stating: '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'. At the bottom right, there are 'Cancel' and 'Create launch template' buttons, with the latter being highlighted.

57) Select instance type as 't2.micro' and the key pair same as that of created instances.

The screenshot shows the 'Create launch template' wizard. In the 'Instance type' section, 't2.micro' is selected from a dropdown menu. The dropdown also lists other options like 'All generations' and 'Compare instance types'. A tooltip for the 'Free tier' is visible, stating: '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'. At the bottom right, there are 'Cancel' and 'Create launch template' buttons, with the latter being highlighted.

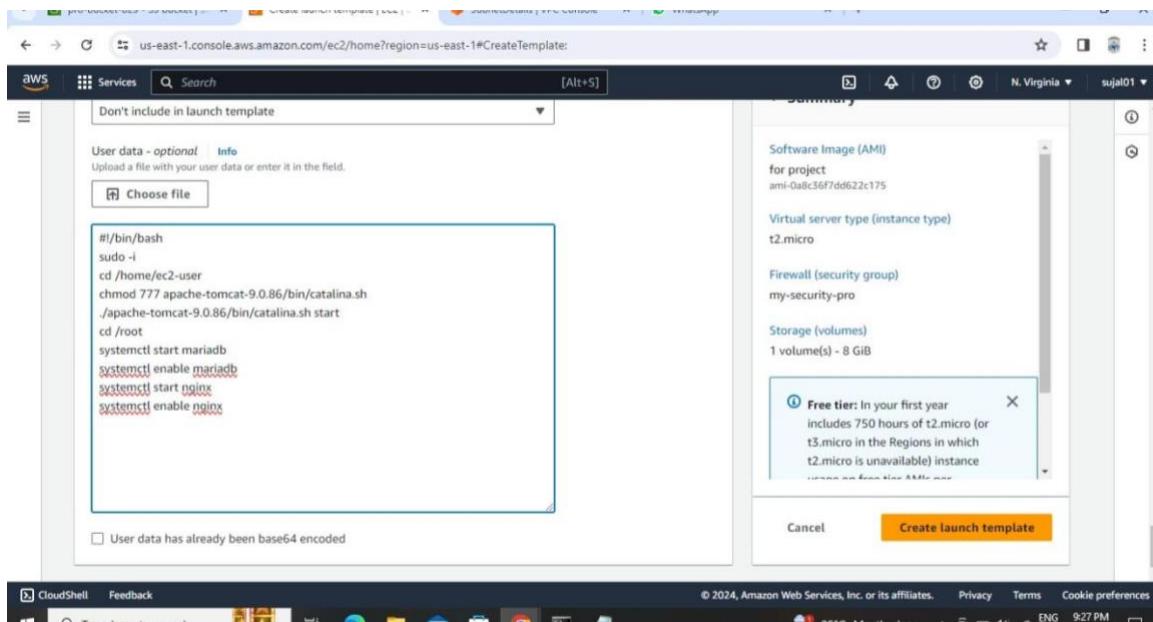
58) In network settings select the custom VPC and the private subnet. Choose the created security group (launch-wizard-7).



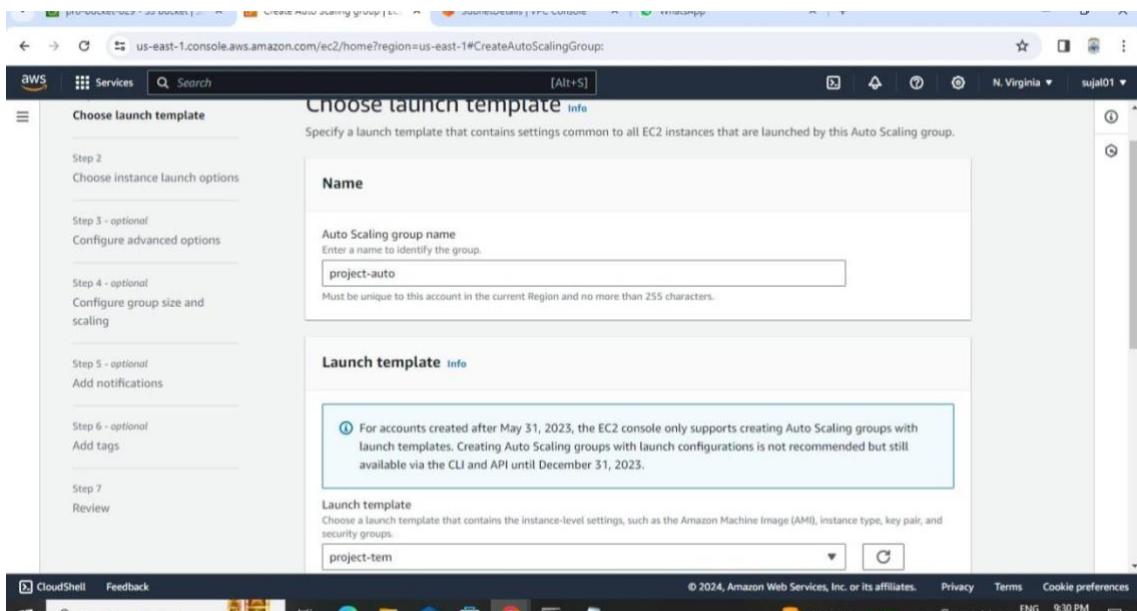
59) Click on advanced settings and scroll down to user data section and add the following script:

```
#!/bin/bash
sudo -i
cd /home/ec2-user
chmod 777 apache-tomcat-9.0.86/bin/catalina.sh
./apache-tomcat-9.0.86/bin/catalina.sh start
cd /root
systemctl start mariadb
systemctl enable mariadb
systemctl start nginx
systemctl enable nginx
```

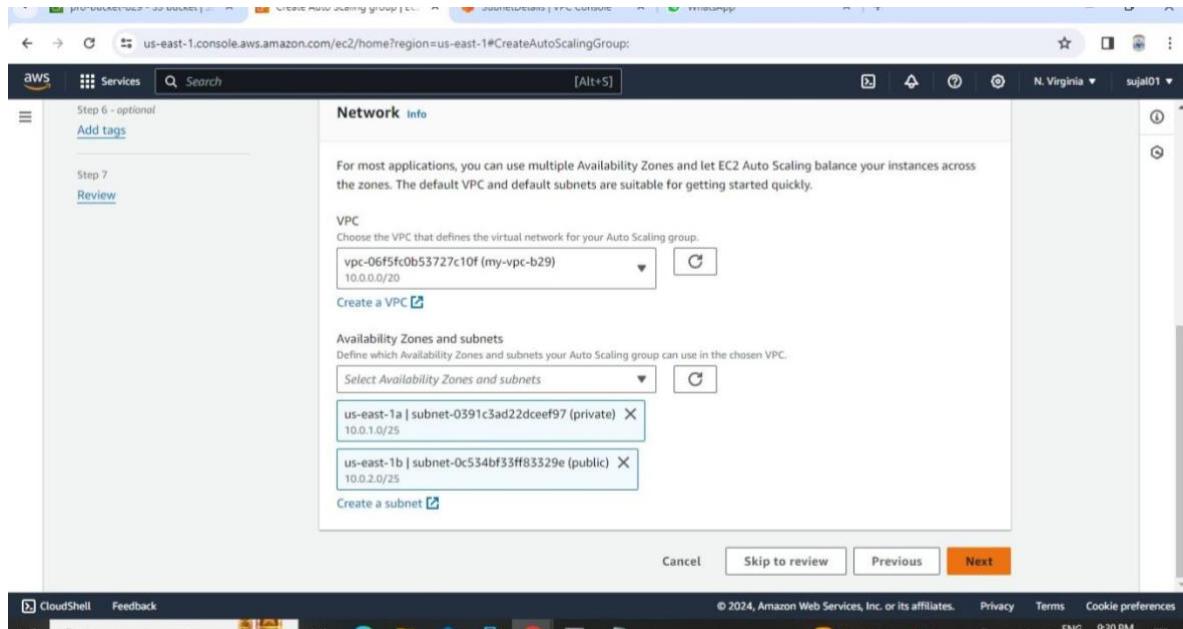
Click on create launch template.



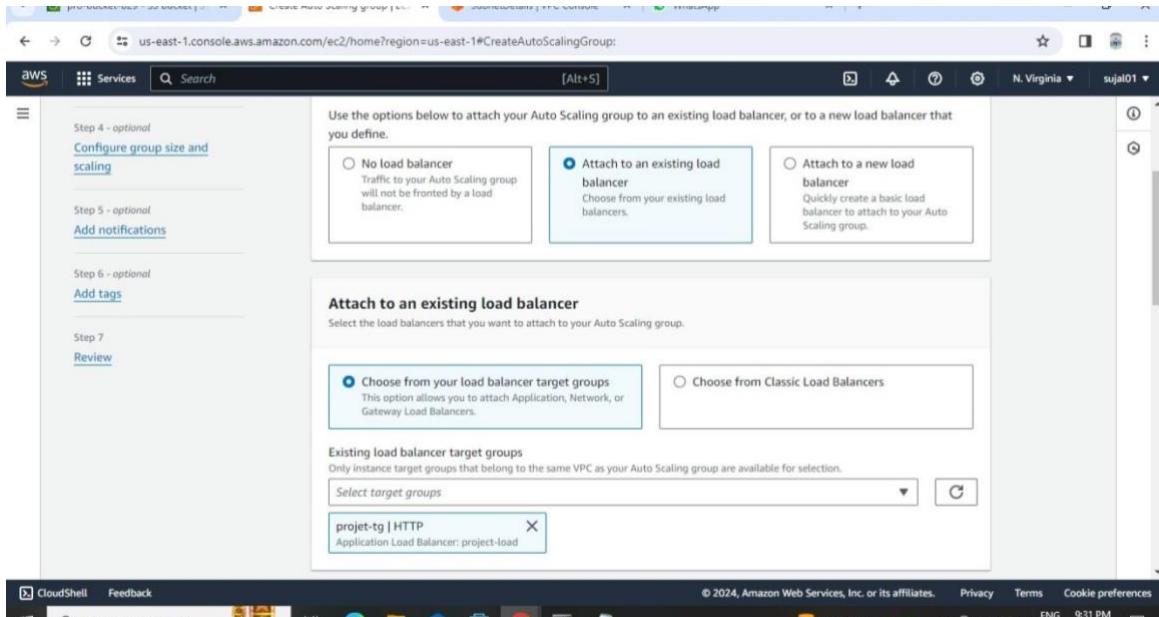
60) Go to Auto Scaling Group dashboard and click on create auto scaling group. Enter a valid name (project-asg) and select the newly created launch template (project-template). Enter data in other fields if required.



61) Select the custom VPC and choose both the availability zones. Click on next.



62) Choose the attach to an existing load balancer option and select the created target group (project-tg) under existing load balancer target groups.



63) Enter the desired capacity, min desired capacity and max desired capacity as per requirement.

Desired capacity: 3

Min desired capacity: 1

Max desired capacity: 5

No scaling policies (selected)

64) Choose the target tracking scaling policy option and fill other fields as per requirement. Review and click on create auto scaling group.

Target tracking scaling policy (selected)

Scaling policy name: project-scaling

Metric type: Average CPU utilization

Target value: 50

Instance warmup: 10 seconds

Disable scale in to create only a scale-out policy:

65) The auto scaling group has been successfully created.

The screenshot shows the AWS EC2 Auto Scaling Groups page. At the top, there is a search bar and a 'Create Auto Scaling group' button. Below the search bar is a table with one row, showing the details of the 'project-auto' group. The table columns include Name, Launch template/configuration, Instances, Status, Desired capacity, Min, and Max. The status for 'project-auto' is listed as 'Updating capacity...'. At the bottom of the page, it says '0 Auto Scaling groups selected'.

66) The new instances have been automatically created by the scaling group.

The screenshot shows the AWS EC2 Instances page. A green banner at the top states: 'Currently creating AMI ami-0a8c36f7dd622c175 from instance i-0d73a07bdcd73fee. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.' Below the banner is a table listing five instances. The table columns are Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. All instances are listed as 'Running'. The availability column shows 'us-east-1b' for four instances and 'us-east-1a' for one. At the bottom of the page, there is a modal window titled 'Select an instance'.

67) In new incognito window paste the public ip of public instance (project_public). The web page will be loaded successfully. Enter the student data and click on register.

Student Data

Not secure 44.200.196.29

Incognito

Name: daya
Address: gokuldham
Age: 56
Qualification: 7vi fail
Percentage: 36
Sed: 2000

Type here to search

25°C Partly cloudy

ENG IN 9:36

68) Keep refreshing the page to see traffic being balanced by the load balancer.

A screenshot of a web browser window titled "44.200.196.29/viewStudents". The address bar shows the URL "44.200.196.29/viewStudents" and indicates "Not secure". Below the title bar, there are navigation buttons (back, forward, search) and a tab labeled "Incognito (5)". A link "Register Student" is visible at the top left of the page. The main content is titled "Students List" and displays a table with the following data:

Student ID	StudentName	Student Addr	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	vaibhav	dhangat	89	7th	25	1999	edit	delete
2	prathm	aundh	aws	none	35	1967	edit	delete
3	daya	gokuldham	56	7vi fail	36	2000	edit	delete

