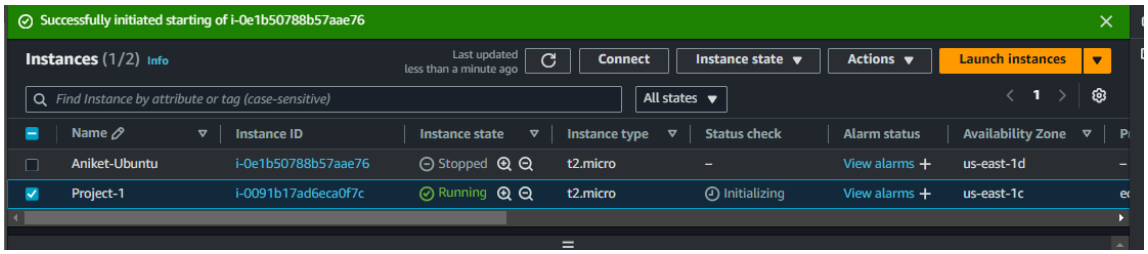
Problem Statement:

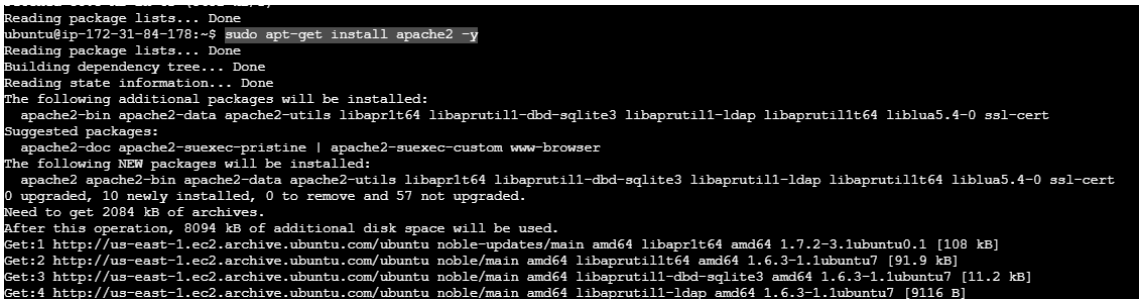
Company ABC wants to move their product to AWS. They have the following things set up right now:

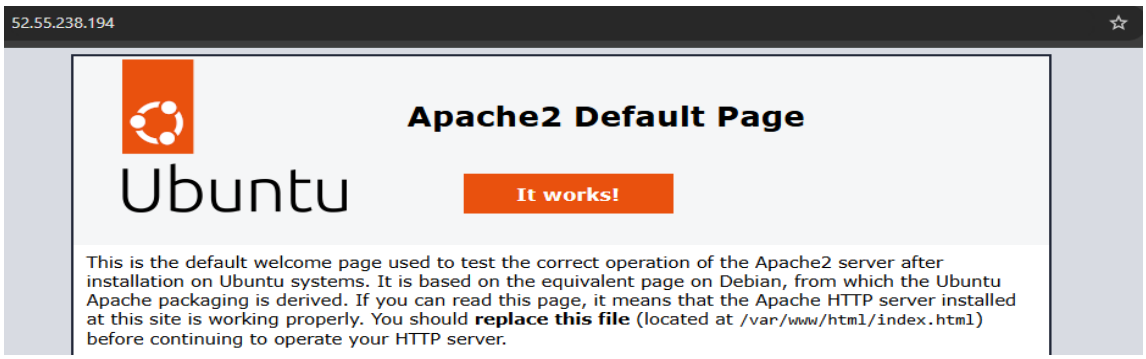
1. MySQL DB

2. Website (PHP) The company wants high availability on this product, therefore wants Auto Scaling to be enabled on this website

Steps→ Launch an Ubuntu instance > Update the machine and install any webserver in it







Now delete the index.html file of the web page

```
ubuntu@ip-172-31-84-178:~$ cd /var/www/html
ubuntu@ip-172-31-84-178:/var/www/html$ ls
index.html
ubuntu@ip-172-31-84-178:/var/www/html$ sudo rm index.html
ubuntu@ip-172-31-84-178:/var/www/html$
```

i-0091b17ad6eca0f7c (Project-1)

PublicIPs: 52.55.238.194 PrivateIPs: 172.31.84.178

Now copy your code from local machine to this instance > install the unzip package and unzip that copied file

```
C:\Users\Admin\Downloads>scp -r -i N.virginia-keypair.pem code.zip ubuntu@52.55.238.194:/home/ubuntu
The authenticity of host '52.55.238.194 (52.55.238.194)' can't be established.
ED25519 key fingerprint is SHA256:GQv83r2/esbcy8lfzjVhFg2LUOKTuwL3A/mvgicTGTI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '52.55.238.194' (ED25519) to the list of known hosts.
code.zip                                                                      100%  769KB 492.2KB/s   00:01

C:\Users\Admin\Downloads>
```

```
ubuntu@ip-172-31-84-178:/var/www/html$ cd
ubuntu@ip-172-31-84-178:~$ ls
code.zip
ubuntu@ip-172-31-84-178:~$
```

i-0091b17ad6eca0f7c (Project-1)

PublicIPs: 52.55.238.194 PrivateIPs: 172.31.84.178

```
ubuntu@ip-172-31-84-178:~$ sudo apt-get install unzip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 57 not upgraded.
Need to get 174 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (10.4 MB/s)
Selecting previously unselected package unzip.
(Reading database ... 68559 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4.1_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
```

i-0091b17ad6eca0f7c (Project-1)

PublicIPs: 52.55.238.194 PrivateIPs: 172.31.84.178

```
ubuntu@ip-172-31-84-178:~$ sudo unzip code.zip
Archive:  code.zip
   creating: 1243/images/
  inflating: 1243/images/1.png
  inflating: 1243/images/2.png
  inflating: 1243/index.php
ubuntu@ip-172-31-84-178:~$ ls
1243  code.zip
ubuntu@ip-172-31-84-178:~$
```

i-0091b17ad6eca0f7c (Project-1)

PublicIPs: 52.55.238.194 PrivateIPs: 172.31.84.178

```
ubuntu@ip-172-31-84-178:~$ cd 1243
ubuntu@ip-172-31-84-178:~/1243$ ls
images  index.php
ubuntu@ip-172-31-84-178:~/1243$
```

i-0091b17ad6eca0f7c (Project-1)

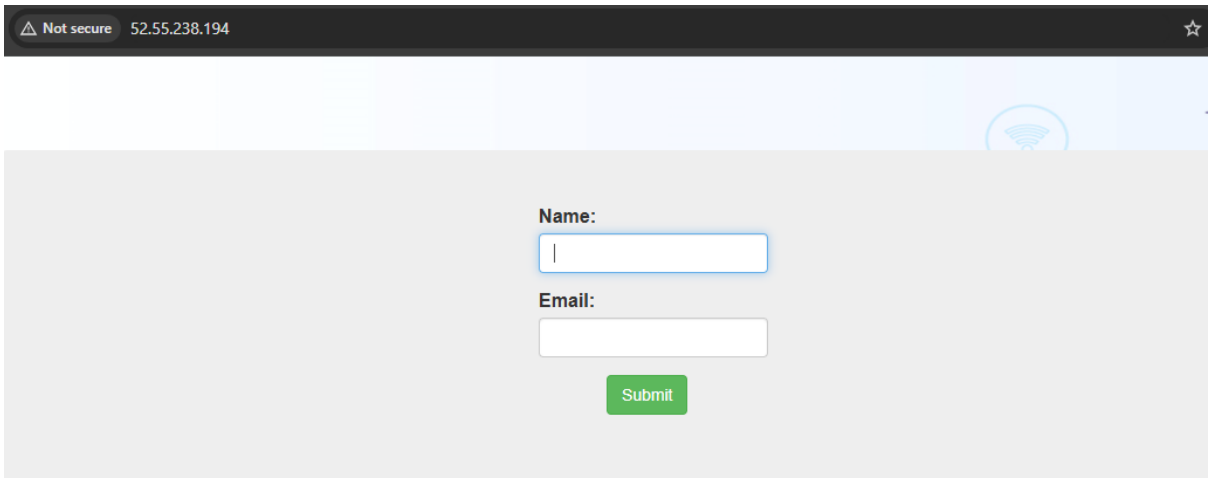
PublicIPs: 52.55.238.194 PrivateIPs: 172.31.84.178

Now move these files to the html location

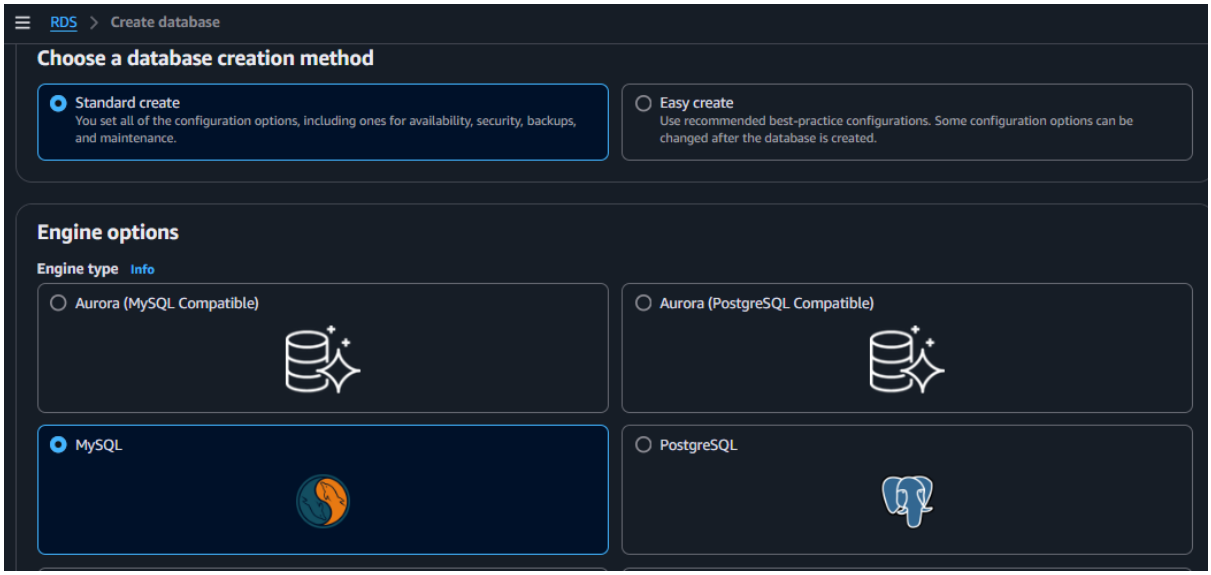
```
ubuntu@ip-172-31-84-178:~/1243$ sudo mv * /var/www/html
ubuntu@ip-172-31-84-178:~/1243$ cd /var/www/html
ubuntu@ip-172-31-84-178:/var/www/html$ ls
images  index.php
ubuntu@ip-172-31-84-178:/var/www/html$
```

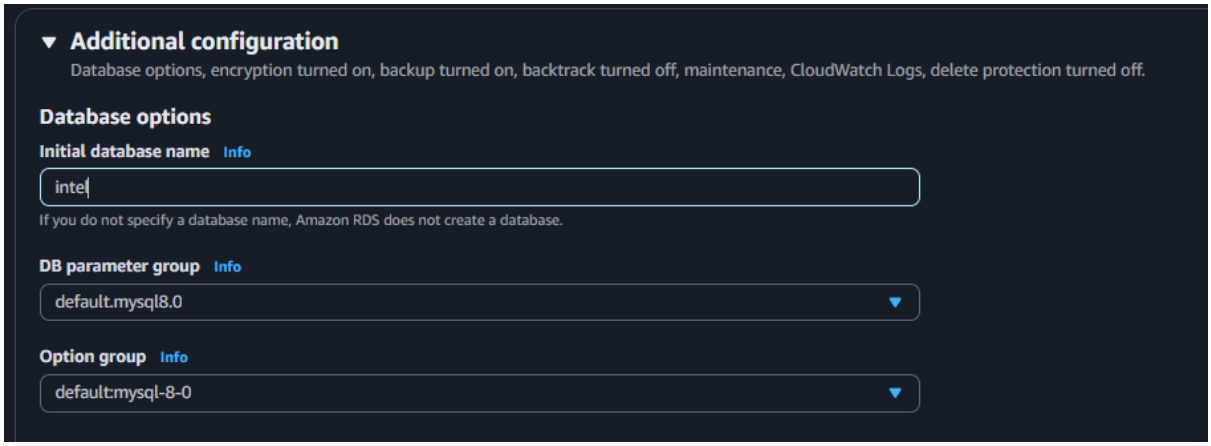
i-0091b17ad6eca0f7c (Project-1)

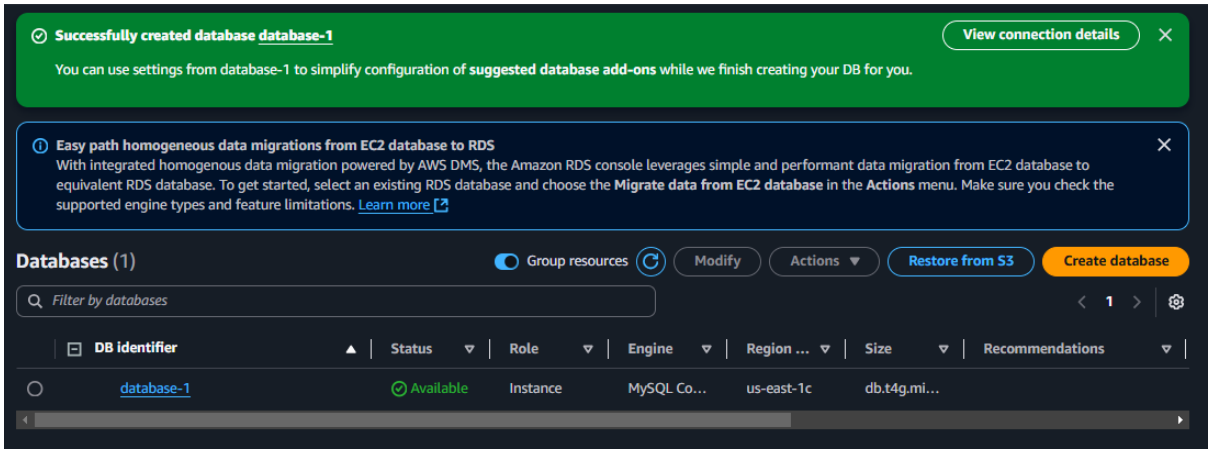
PublicIPs: 52.55.238.194 PrivateIPs: 172.31.84.178



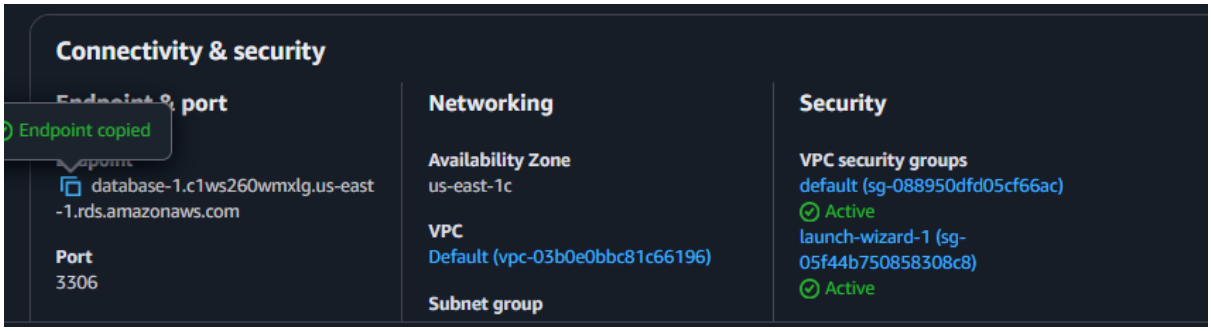
Now create the My SQL RDS database







Now copy the endpoint and paste that in the index.php file and change the username and pass



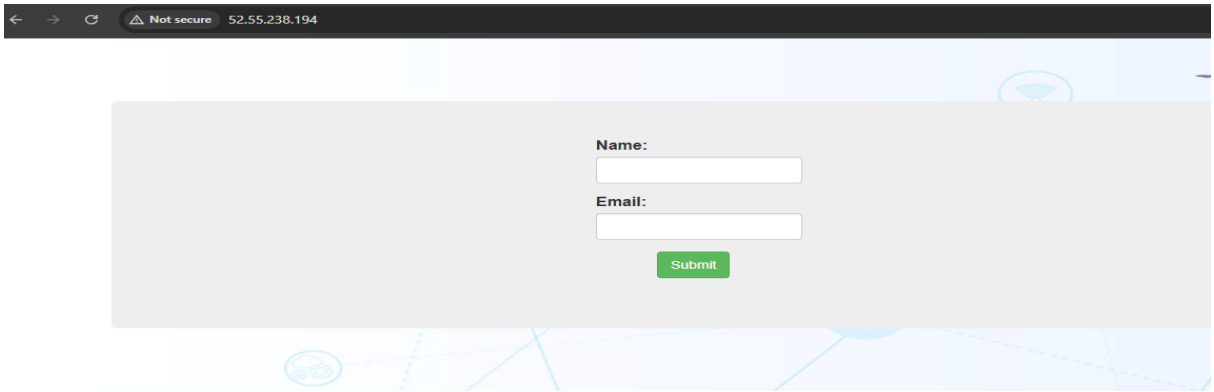

```php
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "database-1.c1ws260wmxlg.us-east-1.rds.amazonaws.com";
$username = "admin";
$password = "Aniket1234";
$db = "intel";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);
```

Add the ondrej repository > install the php 5.6 version to resolve the error

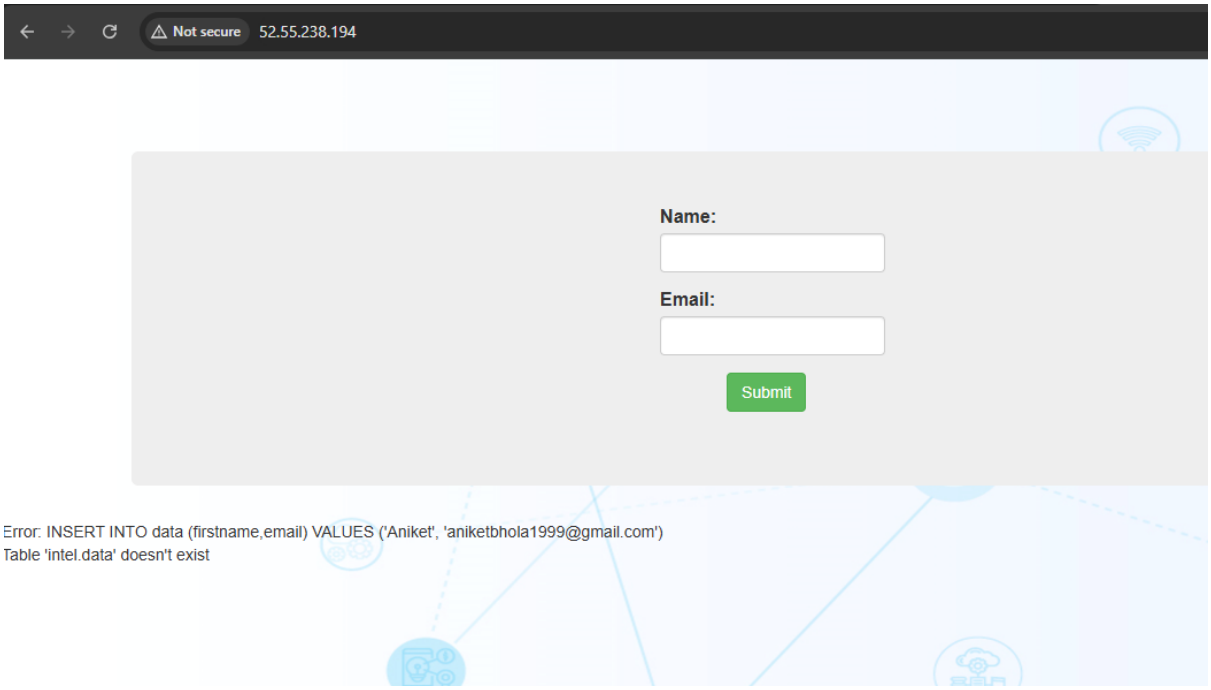
```
ubuntu@ip-172-31-84-178:/var/www/html$ sudo add-apt-repository -y ppa:ondrej/php
PPA publishes dbgsym, you may need to include 'main/debug' component
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ondrej/php/ubuntu/
Suites: noble
Components: main
'
Description:
Co-installable PHP versions: PHP 5.6, PHP 7.x, PHP 8.x and most requested extensions are included. Only Supported
re provided.

Debian oldstable and stable packages are provided as well: https://deb.sury.org/#debian-dpa
```

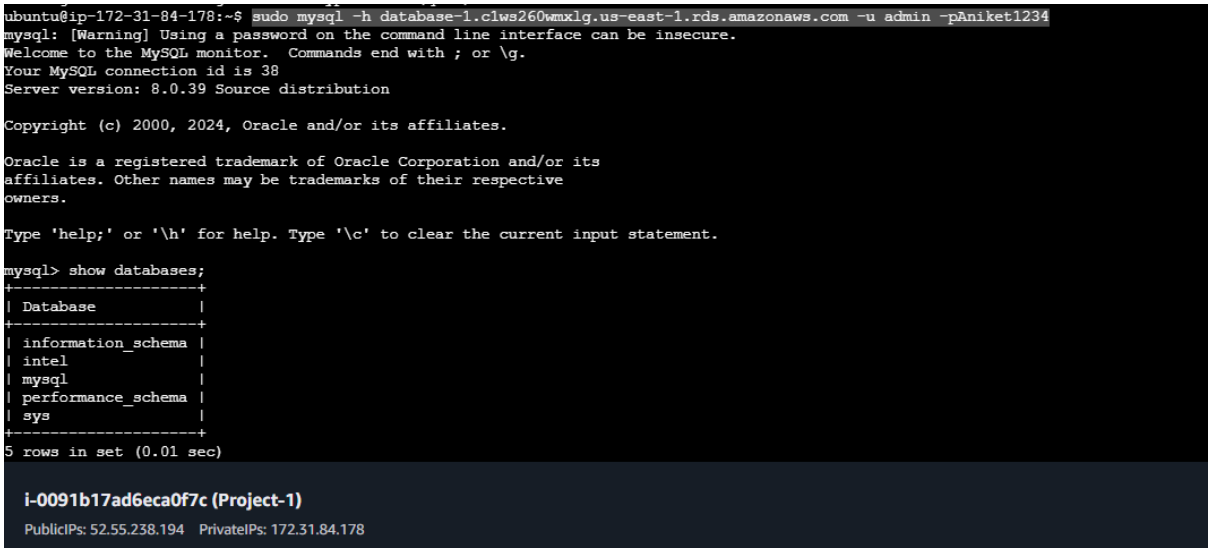
```
ubuntu@ip-172-31-84-178:~$ sudo apt-get install php5.6 mysql-client php5.6-mysqli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php5.6-mysql' instead of 'php5.6-mysqli'
The following additional packages will be installed:
  debsuryorg-archive-keyring libapache2-mod-php5.6 libpcre3 mysql-client-8.0 mysql-client-core-8.0 mysql-common
  php5.6-opcache php5.6-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
```

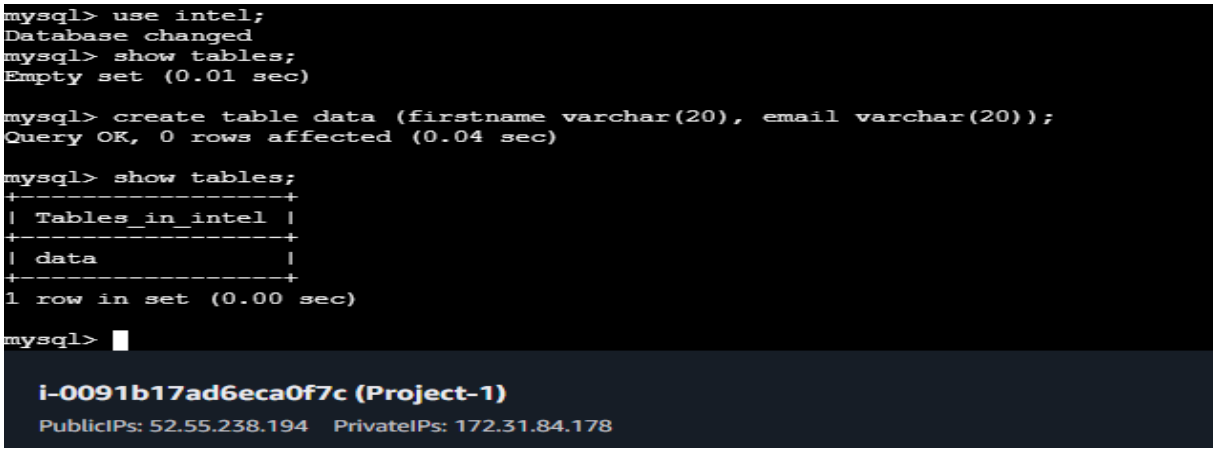


Now after adding the data Im getting this error:

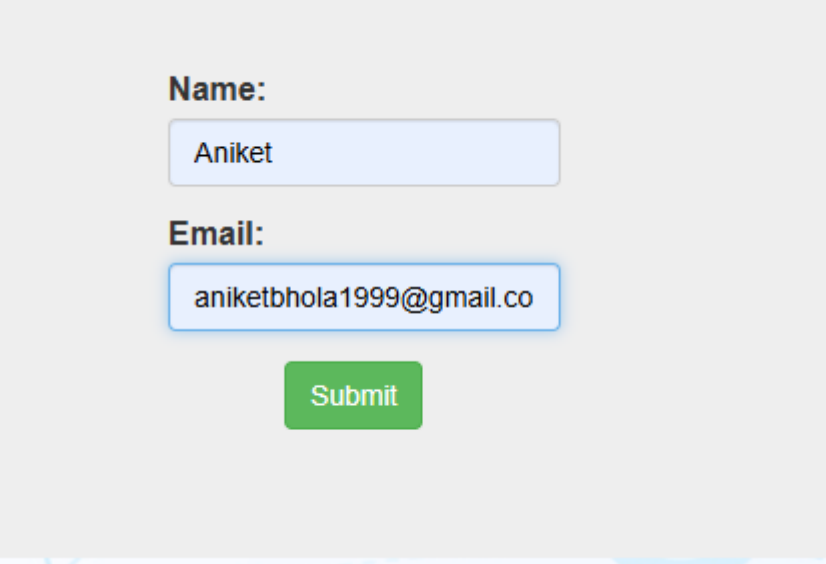


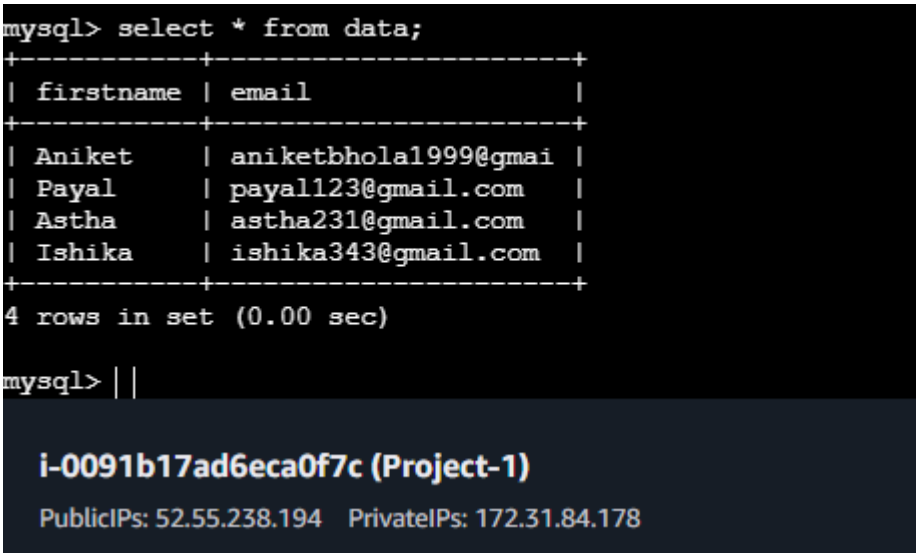
To resolve the error we need to create the table inside the database and add the data again





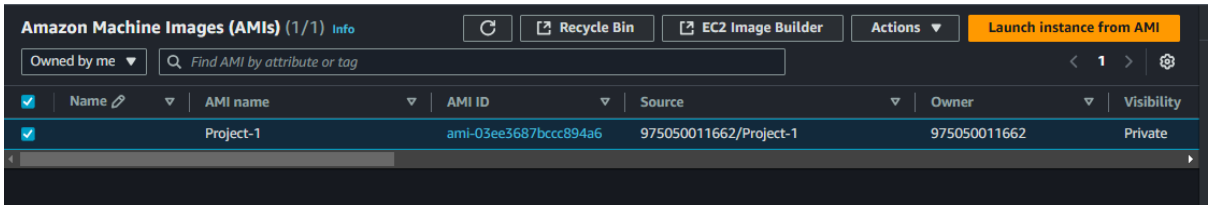
Now add the sample data

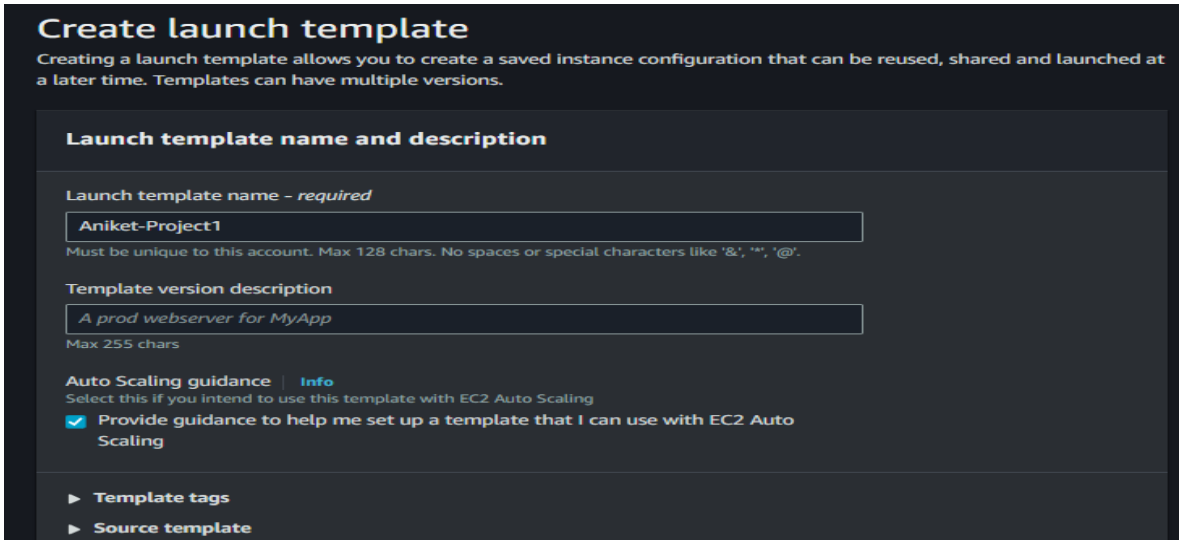


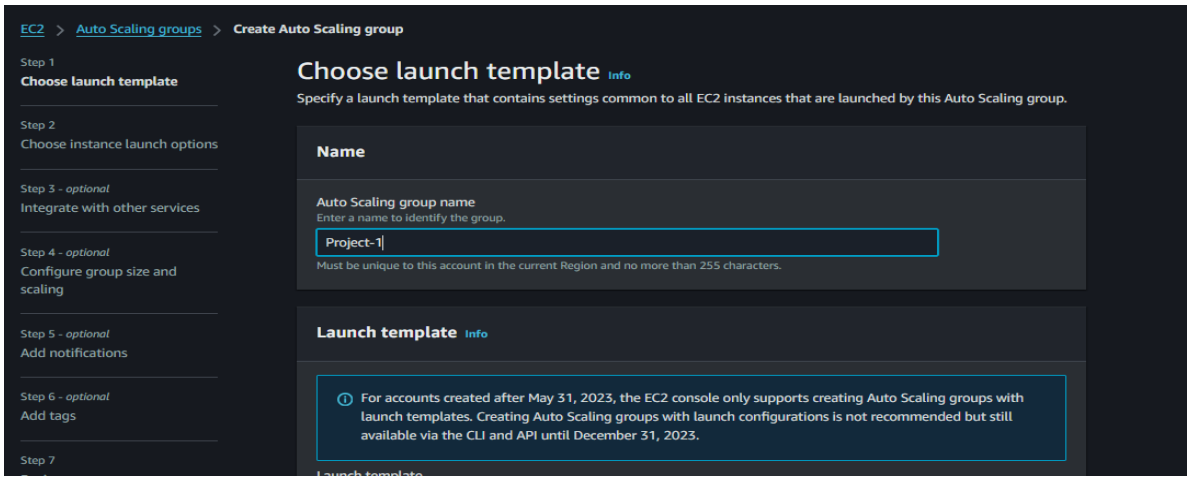


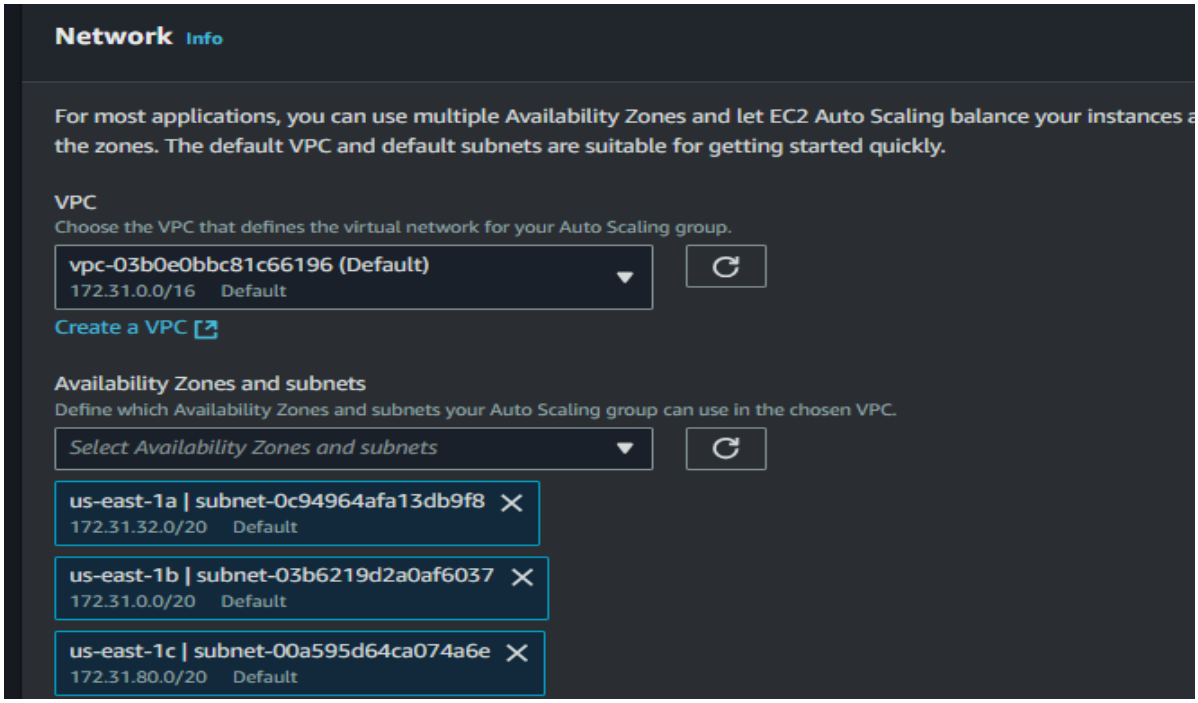
For High availability:

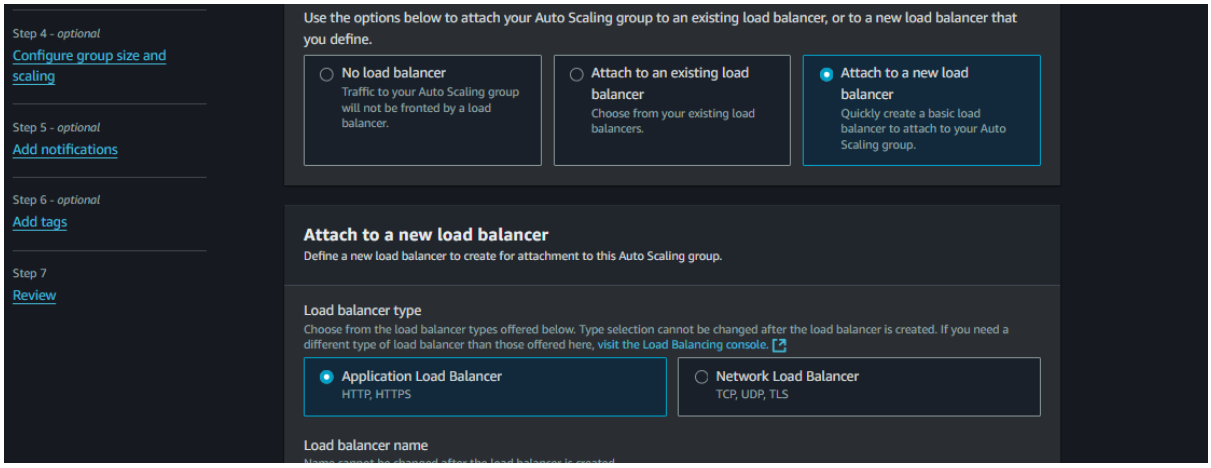
Create the AMI of the instance > Now create the ASG and ALB > copy the DNS of the ALB and paste that in the browser

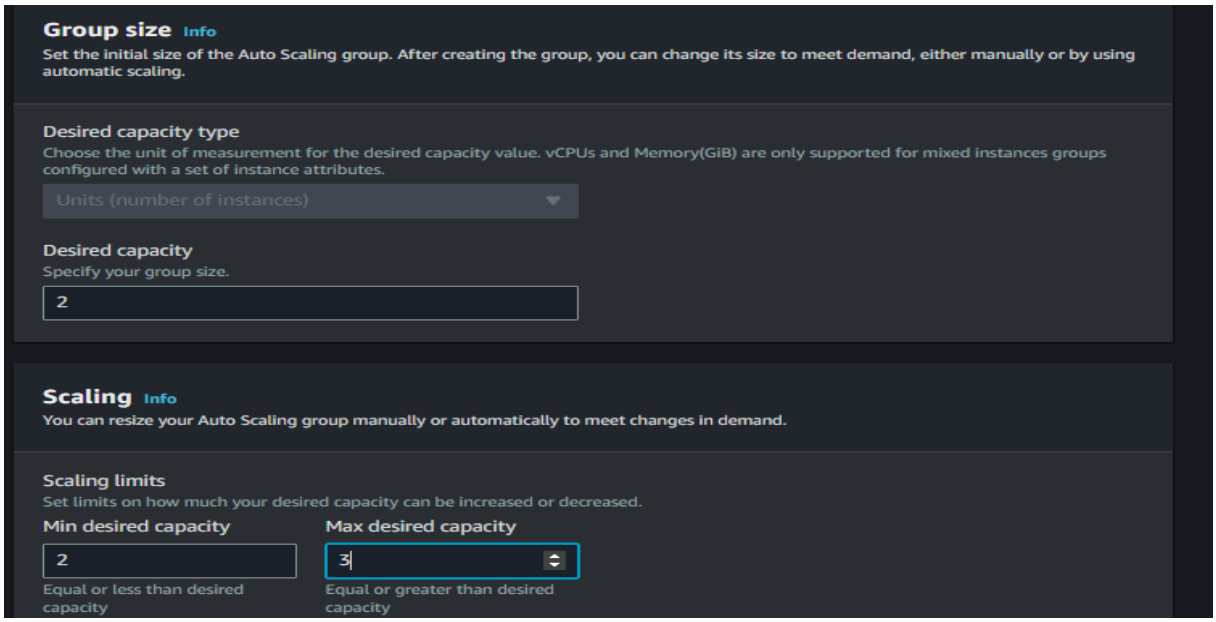


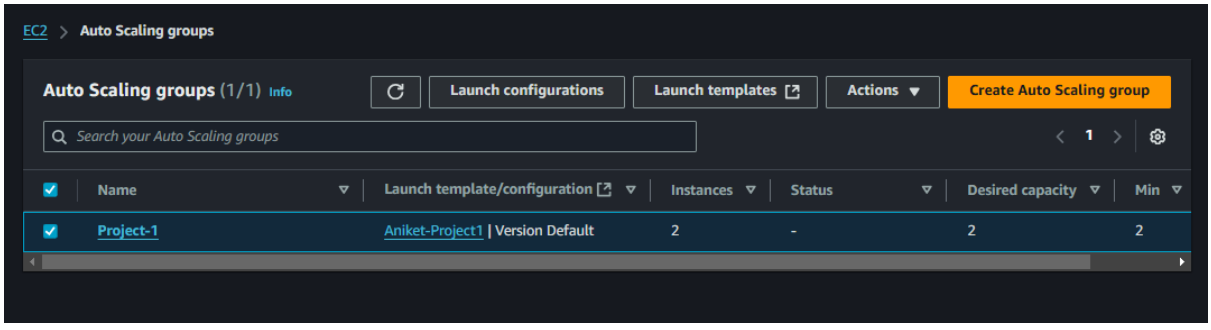

Create launch template
Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.
Launch template name and description
Launch template name – required
Aniket-Project1
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.
Template version description
A prod webserver for MyApp
Max 255 chars
Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
Provide guidance to help me set up a template that I can use with EC2 Auto Scaling
Template tags
Source template

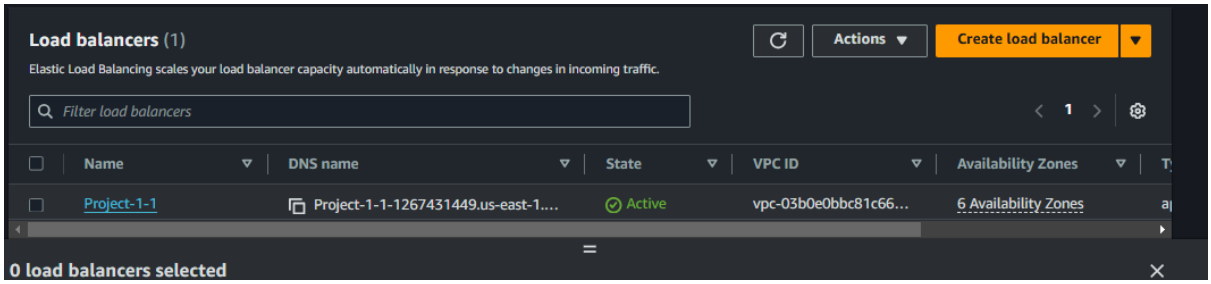

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.
Name
Auto Scaling group name
Enter a name to identify the group.
Project-1
Must be unique to this account in the current Region and no more than 255 characters.
Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

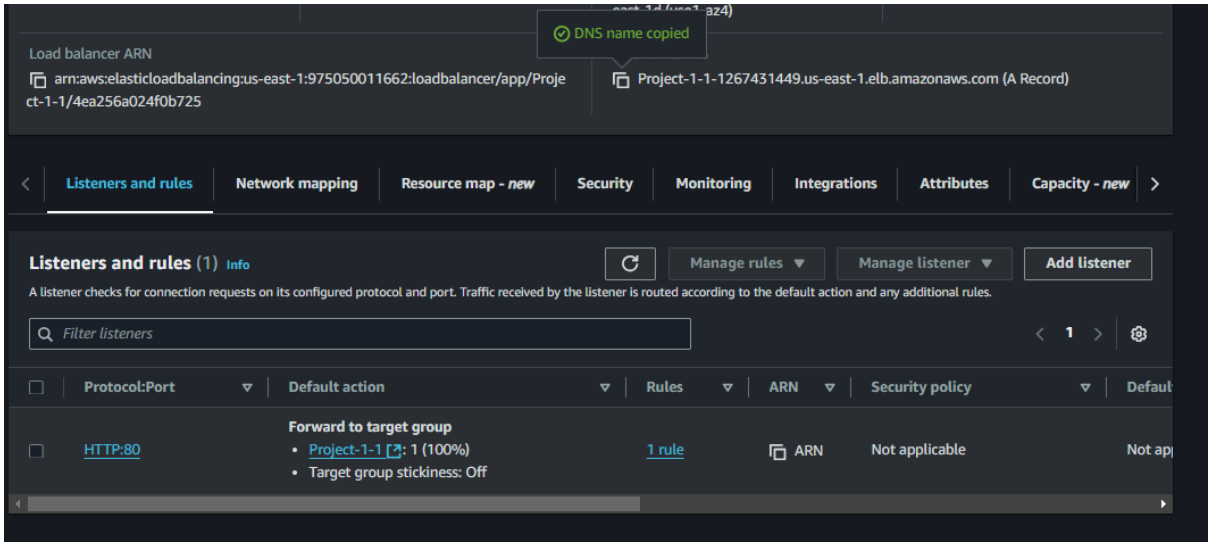

Network Info
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances a
the zones. The default VPC and default subnets are suitable for getting started quickly.
VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-03b0e0bbc81c66196 (Default)
172.31.0.0/16 Default
Create a VPC
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
us-east-1a | subnet-0c94964afa13db9f8
172.31.32.0/20 Default
us-east-1b | subnet-03b6219d2a0af6037
172.31.0.0/20 Default
us-east-1c | subnet-00a595d64ca074a6e
172.31.80.0/20 Default

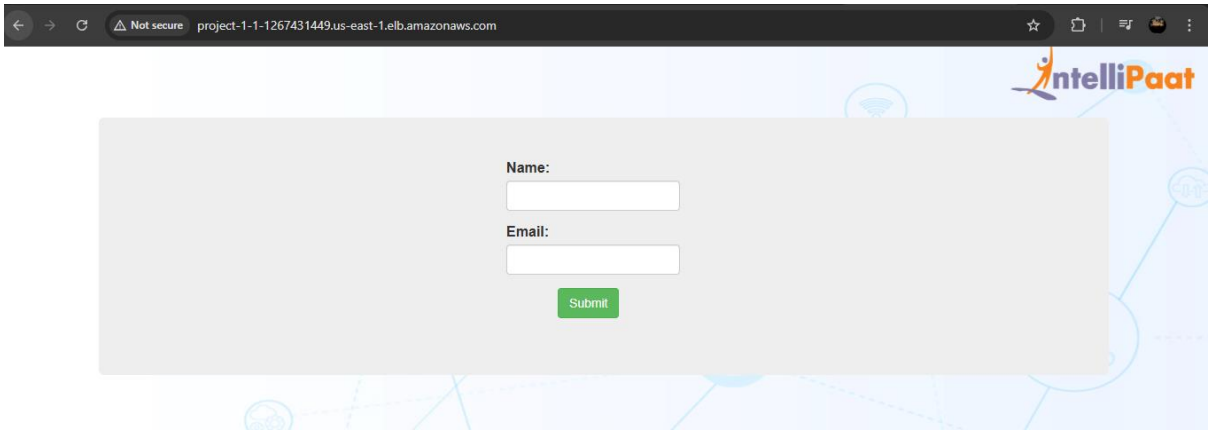

Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
Attach to an existing load balancer
Choose from your existing load balancers.
Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.
Attach to a new load balancer
Define a new load balancer to create for attachment to this Auto Scaling group.
Load balancer type
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.
Application Load Balancer
HTTP, HTTPS
Network Load Balancer
TCP, UDP, TLS
Load balancer name


Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.
Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)
Desired capacity
Specify your group size.
2
Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.
Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
2
Equal or less than desired capacity
Max desired capacity
3
Equal or greater than desired capacity


EC2 > Auto Scaling groups
Auto Scaling groups (1/1) Info
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
1
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Project-1
Aniket-Project1 | Version Default
2
-
2
2


Load balancers (1)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Filter load balancers
1
Name
DNS name
State
VPC ID
Availability Zones
Project-1-1
Project-1-1-1267431449.us-east-1....
Active
vpc-03b0e0bbc81c66...
6 Availability Zones
0 load balancers selected


DNS name copied
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:975050011662:loadbalancer/app/Project-1-1/4ea256a024f0b725
Project-1-1-1267431449.us-east-1.elb.amazonaws.com (A Record)
Listeners and rules
Network mapping
Resource map - new
Security
Monitoring
Integrations
Attributes
Capacity - new
Listeners and rules (1) Info
Manage rules
Manage listener
Add listener
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
Filter listeners
Protocol:Port
Default action
Rules
ARN
Security policy
HTTP:80
Forward to target group
Project-1-1: 1 (100%)
Target group stickiness: Off
1 rule
ARN
Not applicable


Not secure
project-1-1-1267431449.us-east-1.elb.amazonaws.com
IntelliPaat
Name:
Email:
Submit