

## Create IAM user

**Create user | IAM | Global**

https://512692426603-zviz67sl.us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create

**User created successfully**

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

**Retrieve password**

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

**Console sign-in details**

**Console sign-in URL**

<https://512692426603.signin.aws.amazon.com/console>

**User name**

[shravani-admin](#)

**Console password**

[\\*\\*\\*\\*\\*](#) [Show](#)

[Email sign-in instructions](#)

[Cancel](#) [Download .csv file](#) [Return to users list](#)

## Create VPC

**VPC dashboard**

**You successfully created vpc-0e96a0d7b539dde35 / shravani-vpc**

**vpc-0e96a0d7b539dde35 / shravani-vpc**

**Details**

**VPC ID**

[vpc-0e96a0d7b539dde35](#)

**State**

[Available](#)

**Block Public Access**

[Off](#)

**DNS resolution**

[Enabled](#)

**Main network ACL**

[acl-0669c5022c5dcd0aa](#)

**IPv4 CIDR (Network border group)**

[10.0.0.0/16](#)

**Encryption control ID**

[-](#)

**Tenancy**

[default](#)

**Default VPC**

[No](#)

**Network Address Usage metrics**

[Disabled](#)

**Encryption control mode**

[-](#)

**DHCP option set**

[dopt-0ba5e0a40b5476f28](#)

**Route 53 Resolver DNS Firewall rule groups**

[-](#)

**DNS hostnames**

[Disabled](#)

**Main route table**

[rtb-06297d232b879469c](#)

**IPv6 pool**

[-](#)

**Owner ID**

[512692426603](#)

[Actions](#)

[Filter by VPC](#)

## Create subnet

**VPC dashboard**

**Subnets (2)**

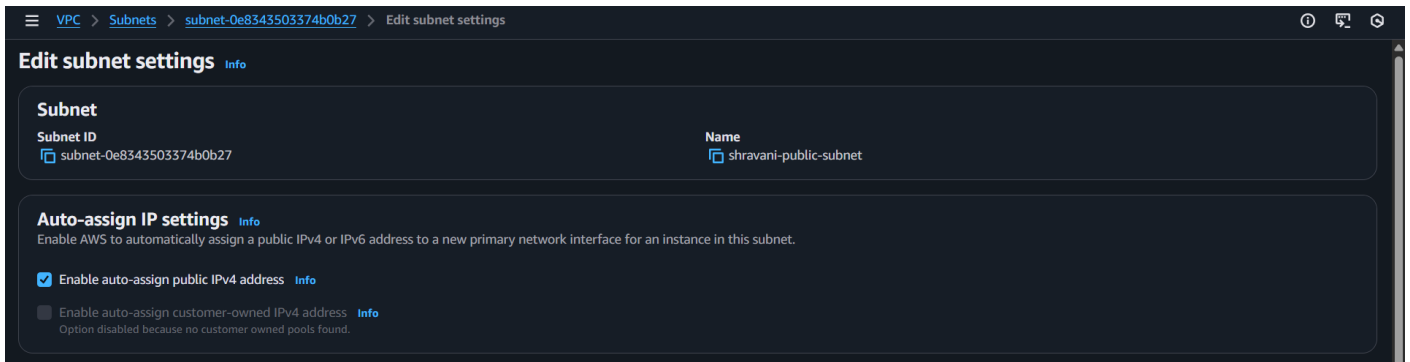
[Find subnets by attribute or tag](#)

[Last updated less than a minute ago](#)

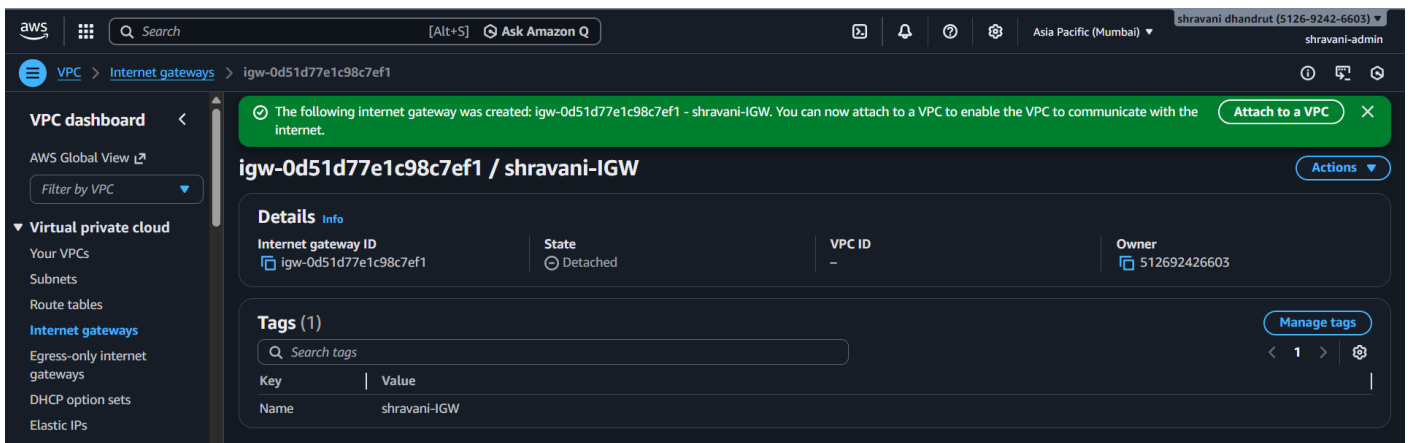
[Actions](#) [Create subnet](#)

<input type="checkbox"/>	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
<input type="checkbox"/>	shravani-private-subnet	subnet-0ed0ddeb19faf45c7	Available	vpc-0e96a0d7b539dde35   shra...	Off	10.0.2.0/24
<input type="checkbox"/>	shravani-public-subnet	subnet-0e8343503374b0b27	Available	vpc-0e96a0d7b539dde35   shra...	Off	10.0.1.0/24

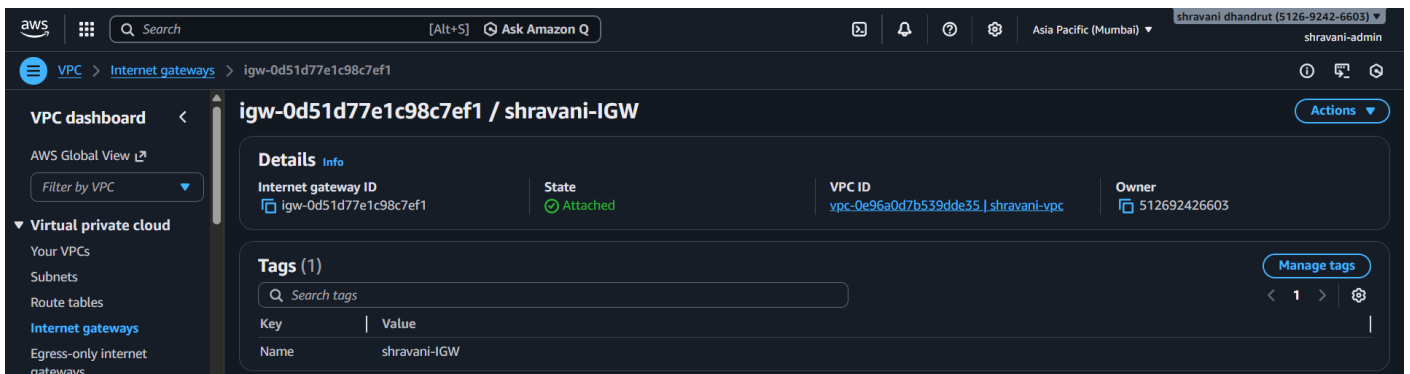
## Enabling auto-assign IP for Public subnet



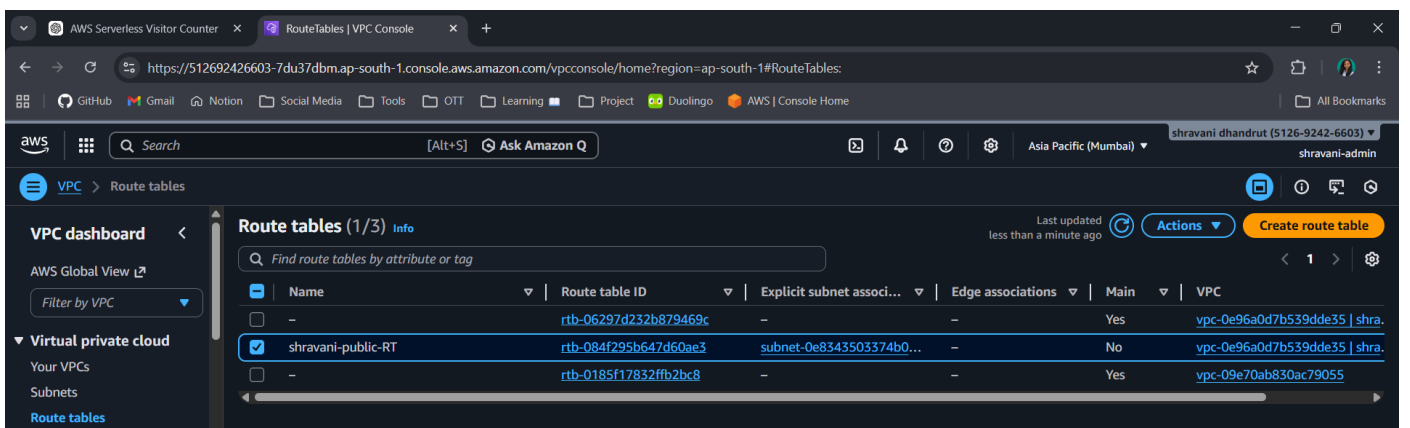
## Create Internet Gateway



## Attach Internet Gateway to VPC



## Create route table -> Edit route -> edit subnet association.



## Create an IAM role for EC2 with full S3 access

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for Identity and Access Management (IAM), Access Management, and Access reports. The main content area displays the details for the 'shra-ec2-role'. A green banner at the top indicates 'Role shra-ec2-role created.' with a 'View role' button. Below this, the role's summary is shown, including its creation date (February 14, 2026, 17:45 UTC+05:30), ARN (arn:aws:iam::512692426603:role/shra-ec2-role), and Instance profile ARN (arn:aws:iam::512692426603:instance-profile/shra-ec2-role). The 'Permissions' tab is selected, showing 'Permissions policies (1)' and a search bar. The bottom of the console shows the AWS logo, search bar, and various utility links like CloudShell, Feedback, and Console Mobile App.

**Role shra-ec2-role created.** [View role](#) [Delete](#)

### shra-ec2-role Info

Allows EC2 instances to call AWS services on your behalf. IAM role for EC2 to access S3 without storing credentials

**Summary** [Edit](#)

<b>Creation date</b> February 14, 2026, 17:45 (UTC+05:30)	<b>ARN</b> <a href="#">arn:aws:iam::512692426603:role/shra-ec2-role</a>	<b>Instance profile ARN</b> <a href="#">arn:aws:iam::512692426603:instance-profile/shra-ec2-role</a>
<b>Last activity</b> -	<b>Maximum session duration</b> 1 hour	

**Permissions** **Trust relationships** **Tags** **Last Accessed** **Revoke sessions**

**Permissions policies (1)** Info [Simulate](#) [Remove](#) [Add permissions](#)

You can attach up to 10 managed policies.

**Filter by Type** [All types](#) [1](#) [Settings](#)

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Launch EC2 Instance

The screenshot shows the AWS EC2 console interface. The left sidebar contains navigation links for EC2, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays the details for the 'shra-backend-server' instance. A green banner at the top indicates 'Instance shra-backend-server created.' with a 'Connect' button. Below this, the instance's summary is shown, including its ID (i-080f0df9b983c0331), Public IPv4 address (43.205.127.64), Private IPv4 address (10.0.1.205), and Instance state (Running). The 'Instance type' is t3.micro, and the 'VPC ID' is vpc-0e96a0d7b539dde35 (shravani-vpc). The 'Subnet ID' is subnet-0e8343503374b0b27 (shravani-public-subnet). The 'IAM role' is shra-ec2-role. The 'IMDSv2' is required. The 'Managed' section shows the instance's lifecycle state. The bottom of the console shows the AWS logo, search bar, and various utility links like CloudShell, Feedback, and Console Mobile App.

**Instance shra-backend-server created.** [Connect](#) [Instance state](#) [Actions](#)

### Instance summary for i-080f0df9b983c0331 (shra-backend-server) Info

Updated less than a minute ago

<b>Instance ID</b> <a href="#">i-080f0df9b983c0331</a>	<b>Public IPv4 address</b> <a href="#">43.205.127.64</a> <a href="#">open address</a>	<b>Private IPv4 addresses</b> <a href="#">10.0.1.205</a>
<b>IPv6 address</b> -	<b>Instance state</b> <a href="#">Running</a>	<b>Public DNS</b> -
<b>Hostname type</b> IP name: ip-10-0-1-205.ap-south-1.compute.internal	<b>Private IP DNS name (IPv4 only)</b> <a href="#">ip-10-0-1-205.ap-south-1.compute.internal</a>	<b>Elastic IP addresses</b> -
<b>Answer private resource DNS name</b> -	<b>Instance type</b> <a href="#">t3.micro</a>	<b>AWS Compute Optimizer finding</b> <a href="#">Opt-in to AWS Compute Optimizer for recommendation s.</a> <a href="#">Learn more</a>
<b>Auto-assigned IP address</b> <a href="#">43.205.127.64</a> [Public IP]	<b>VPC ID</b> <a href="#">vpc-0e96a0d7b539dde35</a> (shravani-vpc)	<b>Auto Scaling Group name</b> -
<b>IAM role</b> <a href="#">shra-ec2-role</a>	<b>Subnet ID</b> <a href="#">subnet-0e8343503374b0b27</a> (shravani-public-subnet)	<b>Managed</b> -
<b>IMDSv2</b> Required	<b>Instance ARN</b> <a href="#">arn:aws:ec2:ap-south-1:512692426603:instance/i-080f0df9b983c0331</a>	

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Configure Apache server over EC2

### Commands:

```
yum install httpd* -y
```

```
systemctl status httpd
```

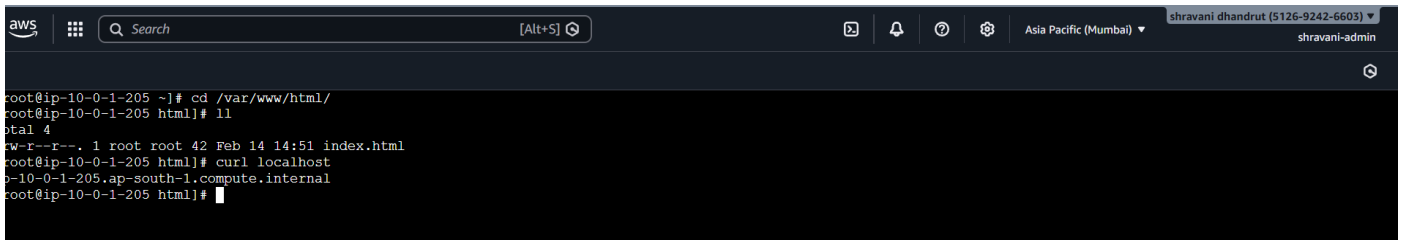
```
systemctl enable httpd
```

```
systemctl start httpd
```

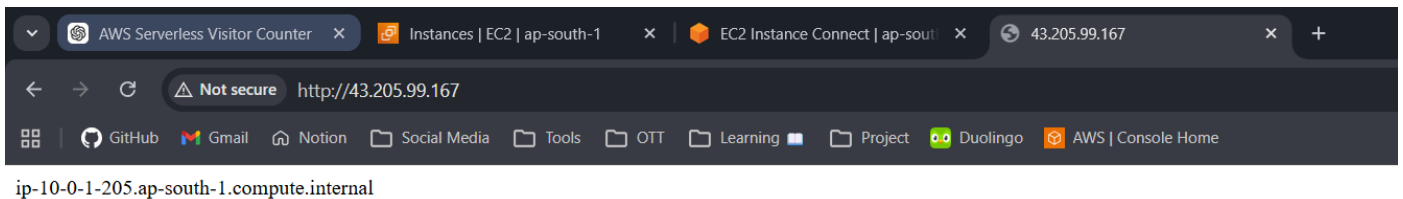
```
cd /var/www/html/
```

```
ls
```

```
echo $(hostname -f) > index.html
```

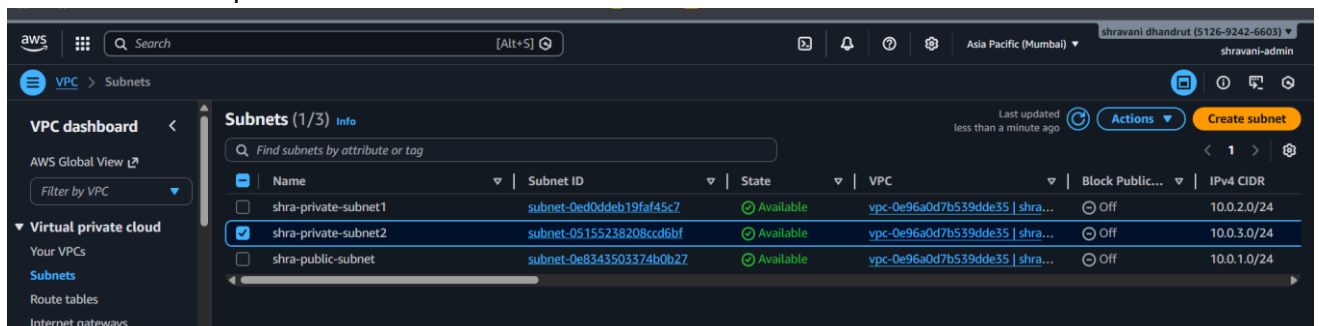


```
root@ip-10-0-1-205 ~]# cd /var/www/html/
root@ip-10-0-1-205 html]# ll
total 4
-rw-r--r-- 1 root root 42 Feb 14 14:51 index.html
root@ip-10-0-1-205 html]# curl localhost
ip-10-0-1-205.ap-south-1.compute.internal
root@ip-10-0-1-205 html]#
```

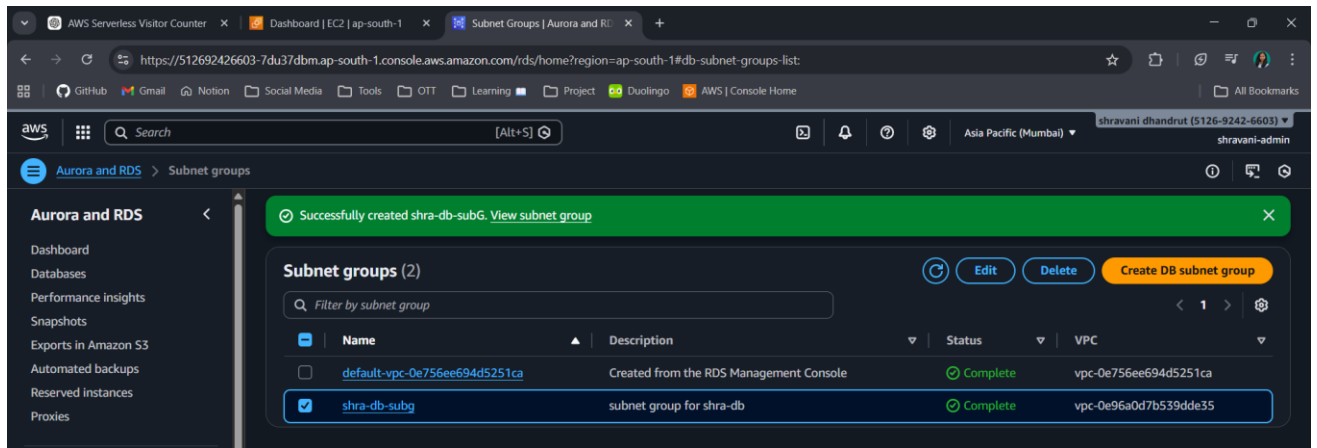


## Create a Mysql Database using RDS

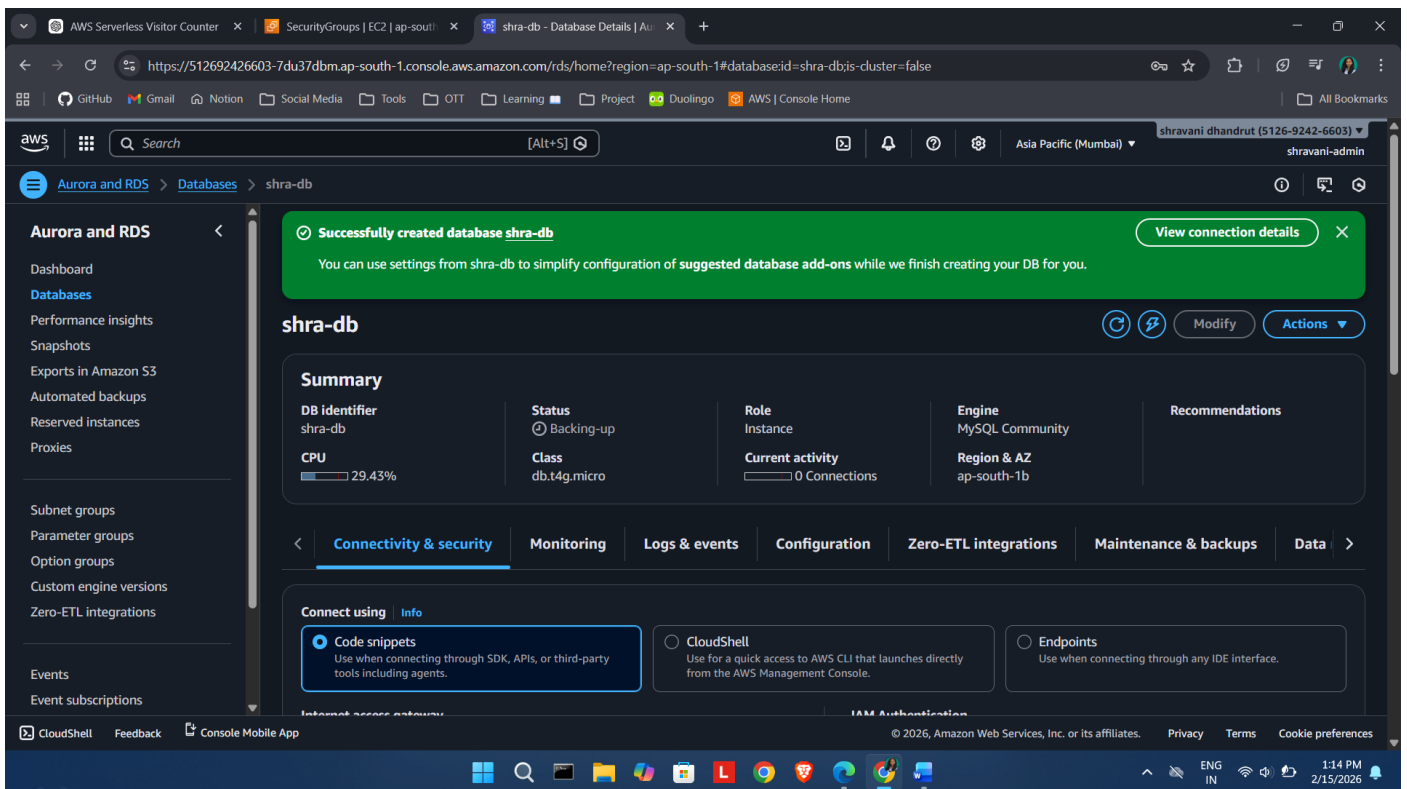
1. Create one more private subnet in another AZ



## 2. Create db subnet group



## Create Database



## Install and open mysql in EC2

- Yum install mysql -y OR yum install mariadb
- mysql -h shra-db.cd62samku30p.ap-south-1.rds.amazonaws.com -u admin -p

```
aws | [grid icon] | [search icon] | [terminal icon] | [bell icon] | [help icon] | [gear icon] Asia Pacific ▾ shravani dhandrut (5126-9242-6603) ▾ shravani-admin
```

```
#_
~\#####_ Amazon Linux 2023
~~\#####\
~~\###|
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~V~' '->
~~~~
~~~.-.
~~-/ -/
~/m/' -/
```

```
Last login: Sun Feb 15 08:03:15 2026 from 13.233.177.3
[ec2-user@ip-10-0-1-205 ~]$ sudo mysql -h shra-db.cd62samku30p.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 8.4.7 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
```

Database

```

aws  [Alt+S]
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| appdb    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.006 sec)

MySQL [(none)]> use appdb;
Database changed
MySQL [appdb]> CREATE TABLE visitors(
  -> id INT AUTO_INCREMENT PRIMARY KEY,
  -> message VARCHAR(255),
  -> created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
  -> );
Query OK, 0 rows affected (0.091 sec)

MySQL [appdb]> show tables;
+-----+
| Tables_in_appdb |
+-----+
| visitors         |
+-----+
1 row in set (0.009 sec)

MySQL [appdb]> insert into visitors(message)values('hello from RDS!');
Query OK, 1 row affected (0.020 sec)

MySQL [appdb]> insert into visitors(message)values('Hii, I am Shravani!');
Query OK, 1 row affected (0.006 sec)

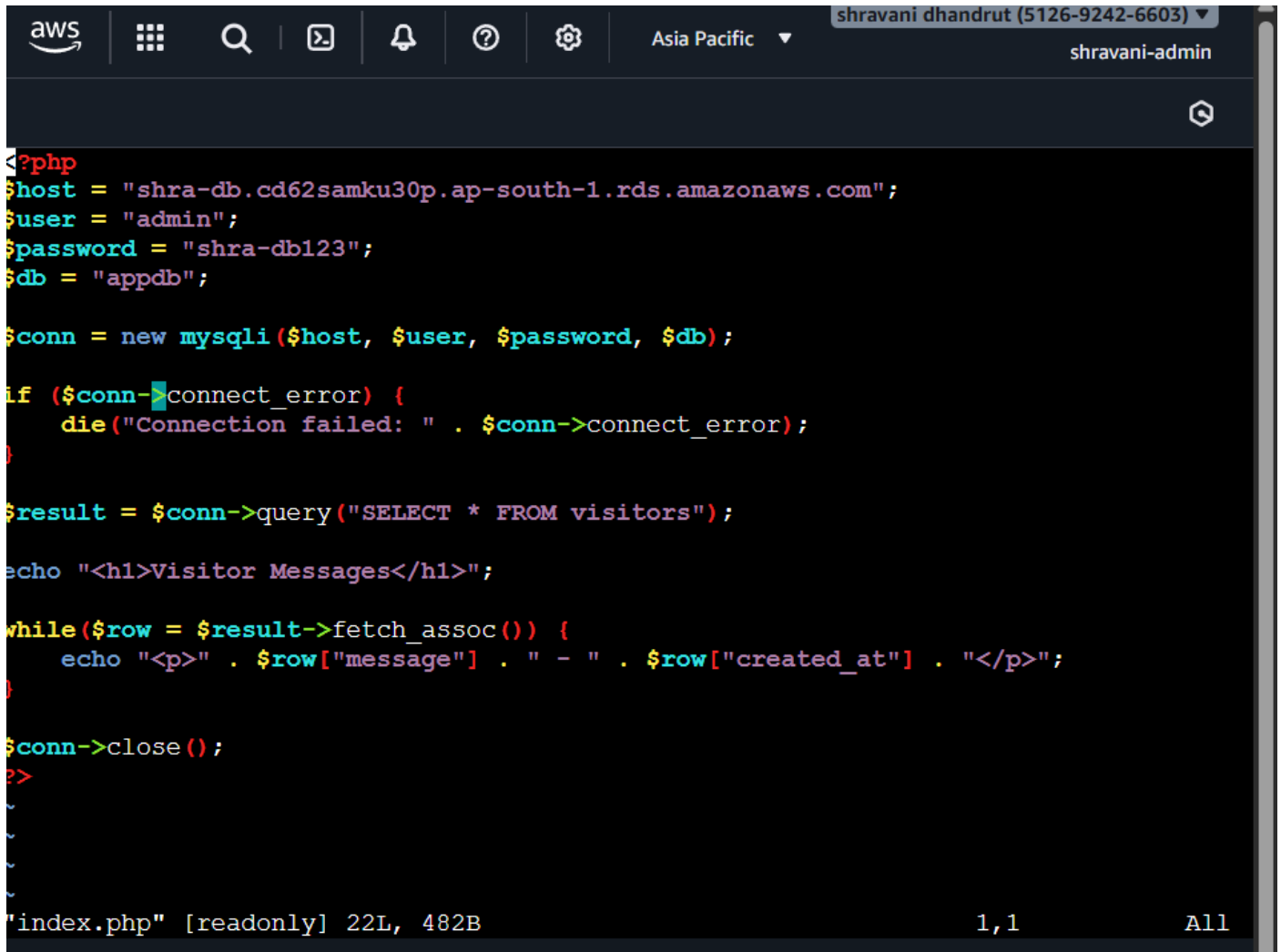
MySQL [appdb]> select * from visitors;
+-----+
| id | message | created_at |
+-----+
| 1 | hello from RDS! | 2026-02-15 08:28:28 |
| 2 | Hii, I am Shravani! | 2026-02-15 08:29:28 |
+-----+
2 rows in set (0.001 sec)

MySQL [appdb]> exit
Bye
[ec2-user@ip-10-0-1-205 ~]$

```

## Install php in ec2

- `dnf install php php-mysqldb -y`
- `cd /var/www/html`
- `rm -f index.html`
- `sudo nano index.php`
- write a php code inside index.php



```
?php
$host = "shra-db.cd62samku30p.ap-south-1.rds.amazonaws.com";
$user = "admin";
$password = "shra-db123";
$db = "appdb";

$conn = new mysqli($host, $user, $password, $db);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$result = $conn->query("SELECT * FROM visitors");

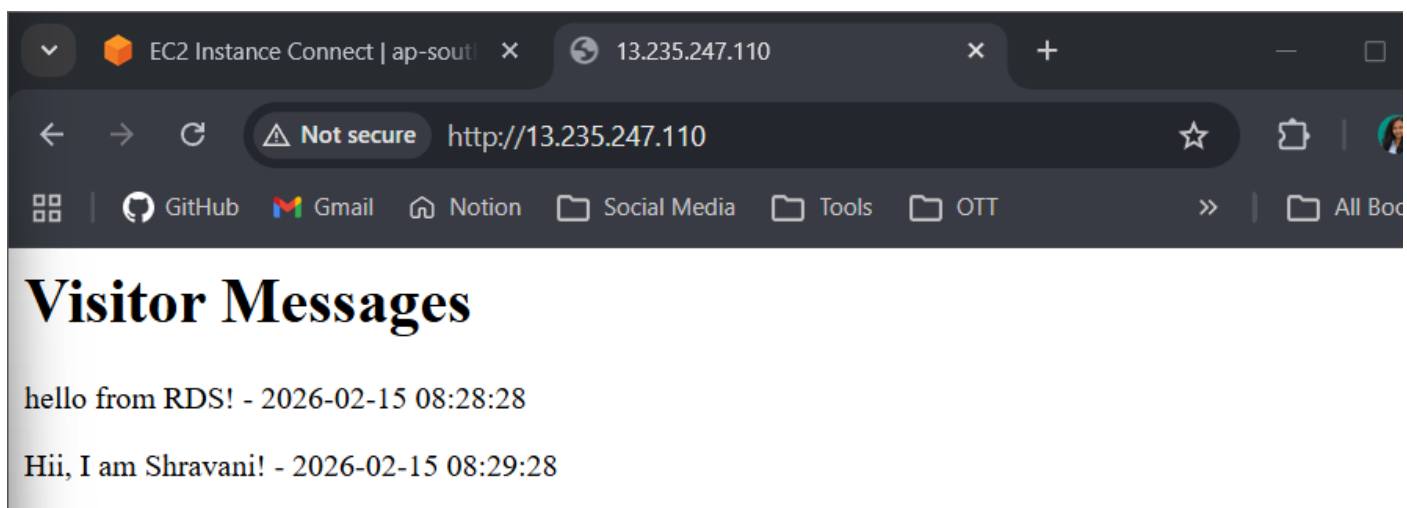
echo "<h1>Visitor Messages</h1>";

while($row = $result->fetch_assoc()) {
    echo "<p>" . $row["message"] . " - " . $row["created_at"] . "</p>";
}

$conn->close();
?>
```

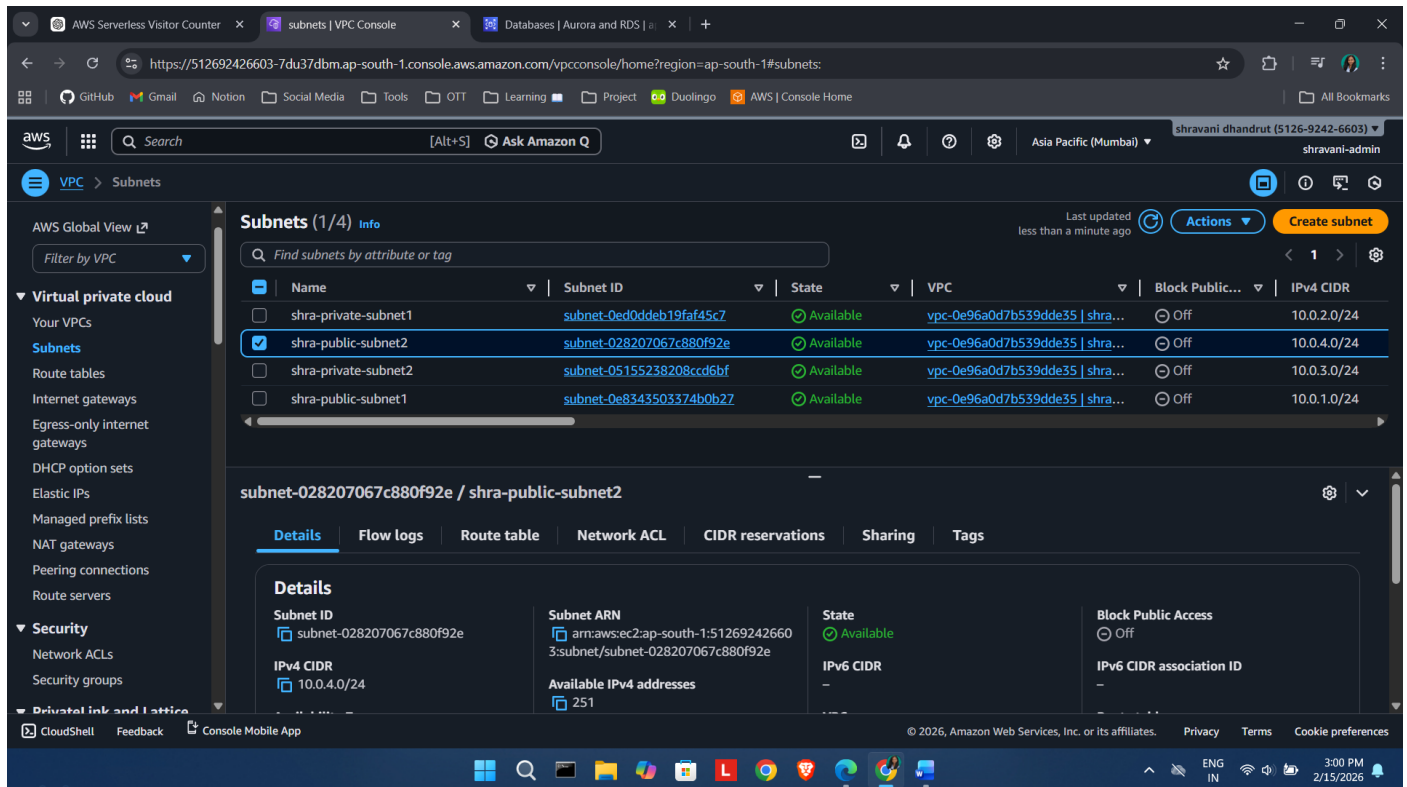
"index.php" [readonly] 22L, 482B 1,1 All

- check it works or not -> open browser -> enter EC2 public IP



## Scaling

Create a second public subnet 10.0.4.0/24



The screenshot shows the AWS VPC console interface. On the left, there's a navigation menu with options like 'Virtual private cloud', 'Subnets', 'Route tables', etc. The main area displays a list of subnets under the heading 'Subnets (1/4)'. The subnets listed are:

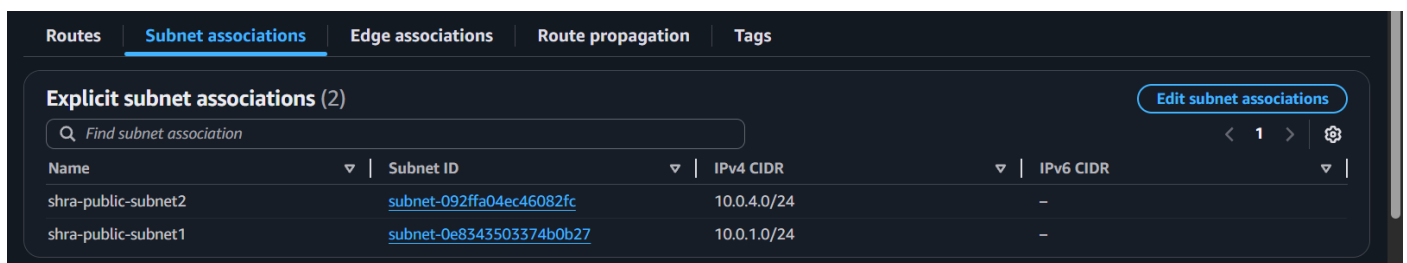
Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
shra-private-subnet1	subnet-0ed0deb19faf45c7	Available	vpc-0e96a0d7b539dde35   shra...	Off	10.0.2.0/24
shra-public-subnet2	subnet-028207067c880f92e	Available	vpc-0e96a0d7b539dde35   shra...	Off	10.0.4.0/24
shra-private-subnet2	subnet-05155238208ccd6bf	Available	vpc-0e96a0d7b539dde35   shra...	Off	10.0.3.0/24
shra-public-subnet1	subnet-0e8343503374b0b27	Available	vpc-0e96a0d7b539dde35   shra...	Off	10.0.1.0/24

The 'shra-public-subnet2' is selected, and its details are shown below the list. The details include:

- Subnet ID: subnet-028207067c880f92e
- Subnet ARN: arnaws:ec2:ap-south-1:512692426603:subnet/subnet-028207067c880f92e
- State: Available
- Block Public Access: Off
- IPv4 CIDR: 10.0.4.0/24
- Available IPv4 addresses: 251
- IPv6 CIDR: -
- IPv6 CIDR association ID: -

Enable Auto-assign IP

Route table -> Edit subnet association -> add shra-public-subnet2

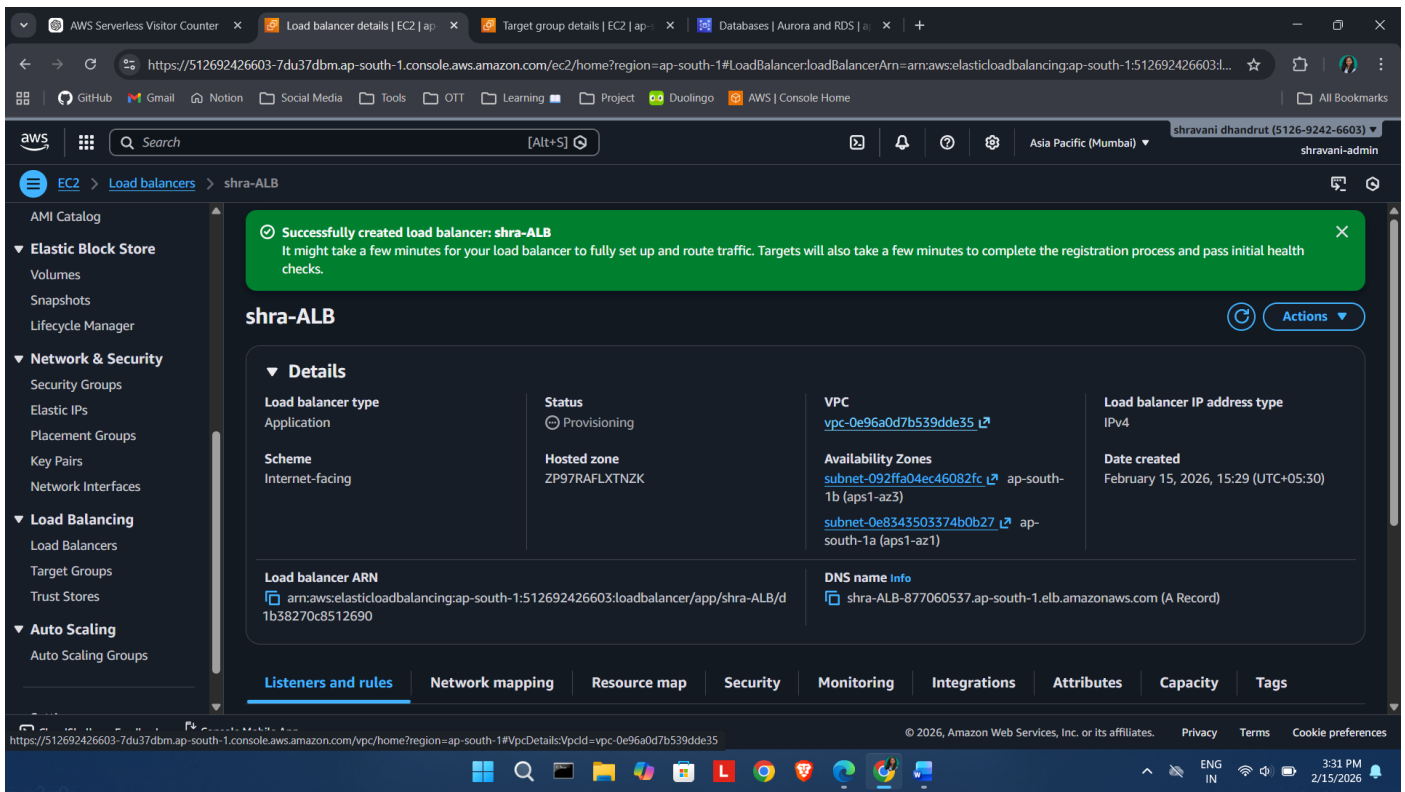


The screenshot shows the 'Subnet associations' tab in the AWS VPC console. It displays a table of explicit subnet associations for a route table. The table has the following columns: Name, Subnet ID, IPv4 CIDR, and IPv6 CIDR. The associations are:

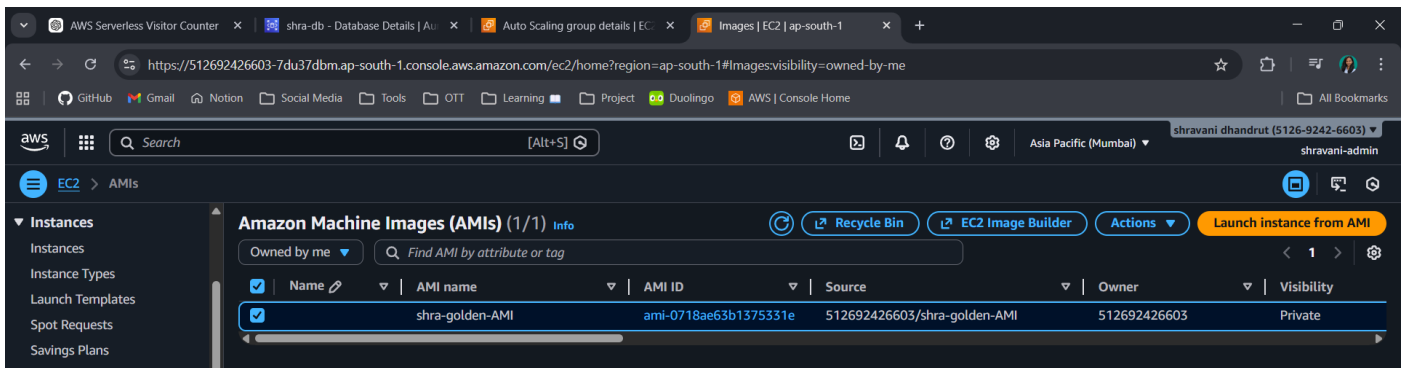
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
shra-public-subnet2	subnet-092ffa04ec46082fc	10.0.4.0/24	-
shra-public-subnet1	subnet-0e8343503374b0b27	10.0.1.0/24	-

Create ALB -> create SG for ALB ->

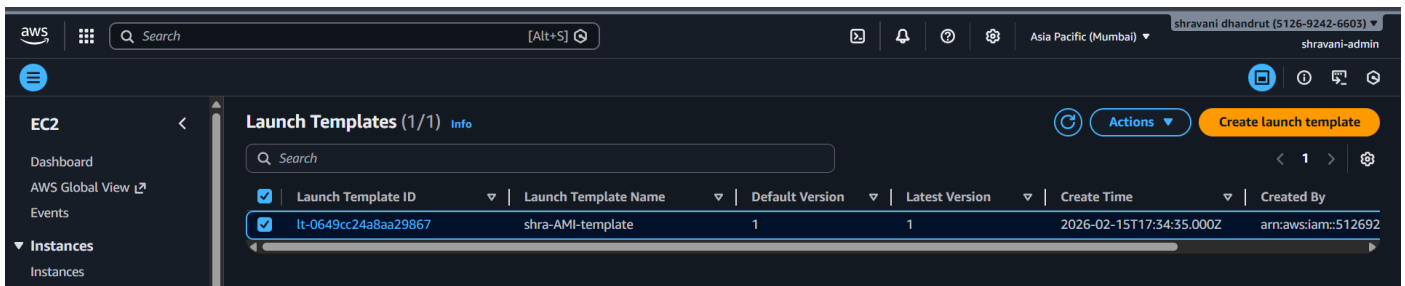




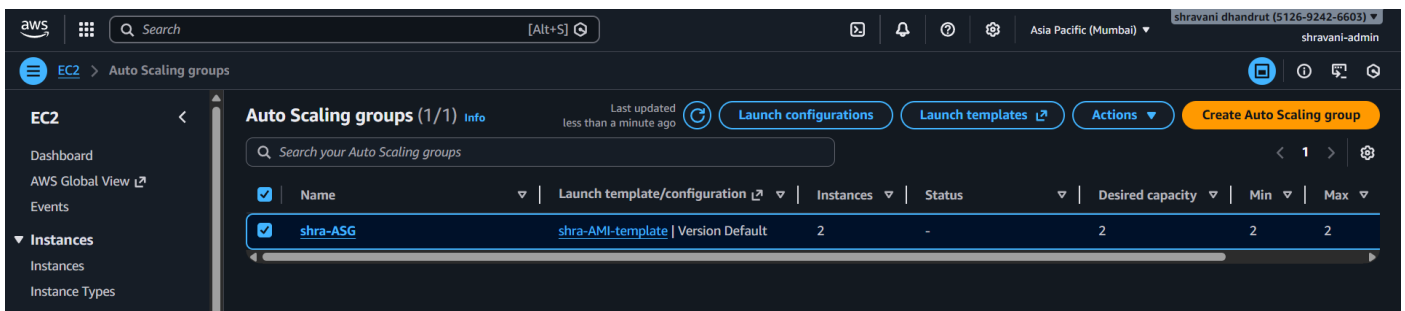
## Create Golden AMI from existing EC2 instance (ec2-backend-server)



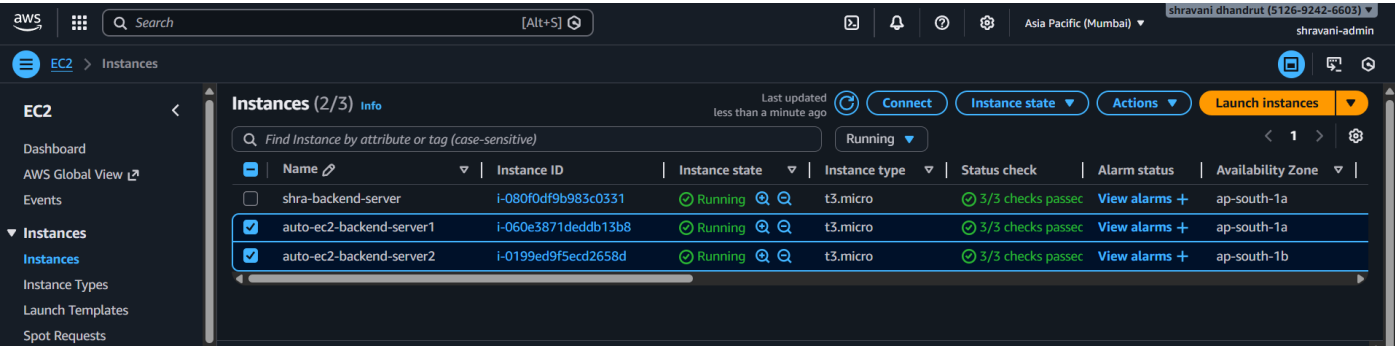
## Launch template using Golden AMI



## Create auto-scaling group

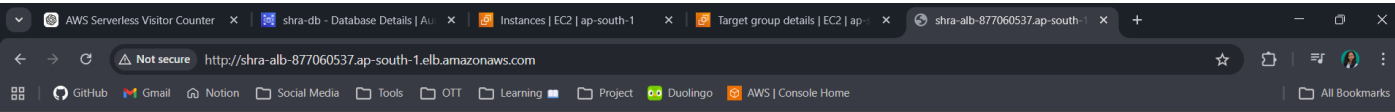


Instances created automatically



Go to RDS-> security group -> edit inbound rule -> add(custom=auto-scaling SG)

Check if it is working properly using ALB dns name



Visitor Messages

hello from RDS! - 2026-02-15 08:28:28

Hii, I am Shravani! - 2026-02-15 08:29:28