Project - 1: Deploying a Multi-Tier Website Using AWS EC2

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

Solution:

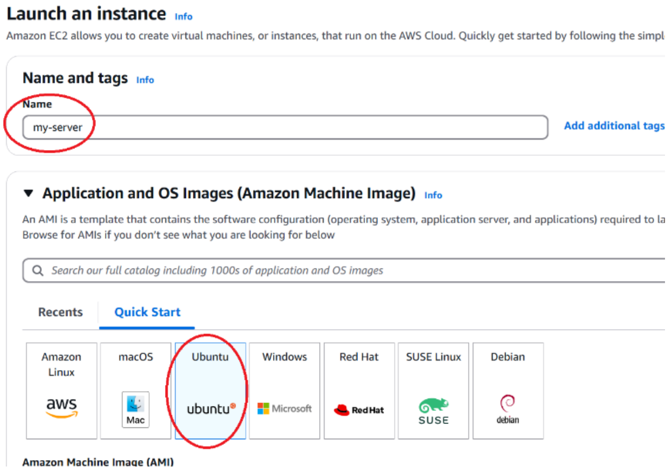
Steps To Solve: 1. Launch an EC2 Instance
2. Enable Auto Scaling on these instances (minimum 2)
3. Create an RDS Instance
4. Create Database & Table in RDS instance:
a. Database name: intel b. Table name: data c. Database password: intel123
5. Change hostname in website
6. Allow traffic from EC2 to RDS instance
7. Allow all-traffic to EC2 instance

Here's a step-by-step guide to move Company ABC's product to AWS, ensuring high availability with Auto Scaling for the PHP website and setting up an RDS MySQL database:

Step 1: Launch an EC2 Instance

1. **Log in to AWS Management Console**.
2. **Navigate to the EC2 Dashboard**.
3. **Click on "Launch Instance"**.

4. **Choose an Amazon Machine Image (AMI)**



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simpl

Name and tags Info

Name: my-server

Add additional tags

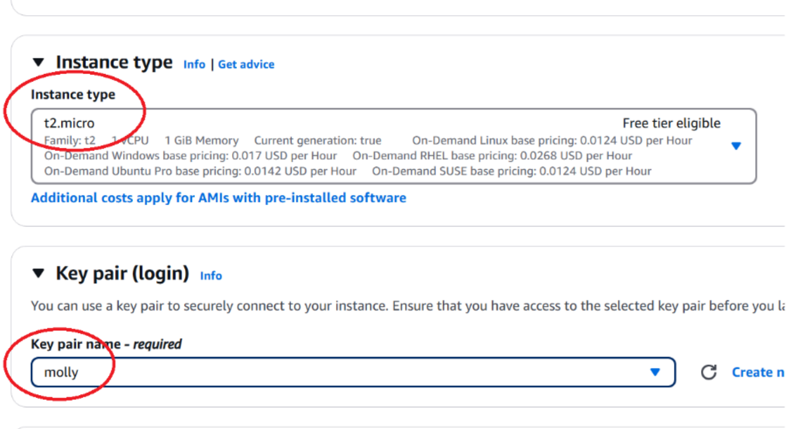
▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to la
Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

5. **Select an instance type**, e.g., `t2.micro`.



▼ Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

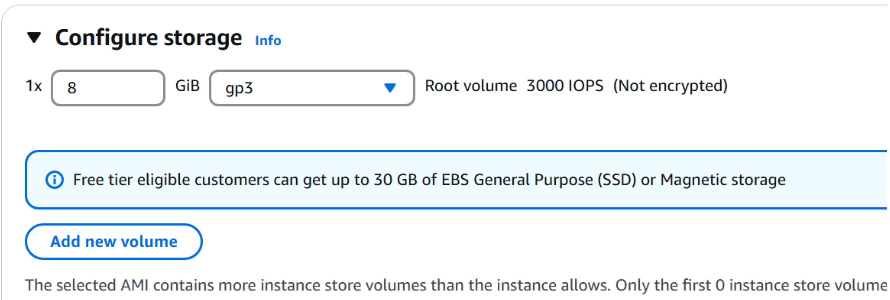
▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you la

Key pair name - *required*: molly

Create n

6. **Configure instance details**:
   - Add the instance to an Auto Scaling group later.
7. **Add Storage**:
   - Configure storage settings as required.



▼ Configure storage Info

1x 8 GiB gp3 Root volume 3000 IOPS (Not encrypted)

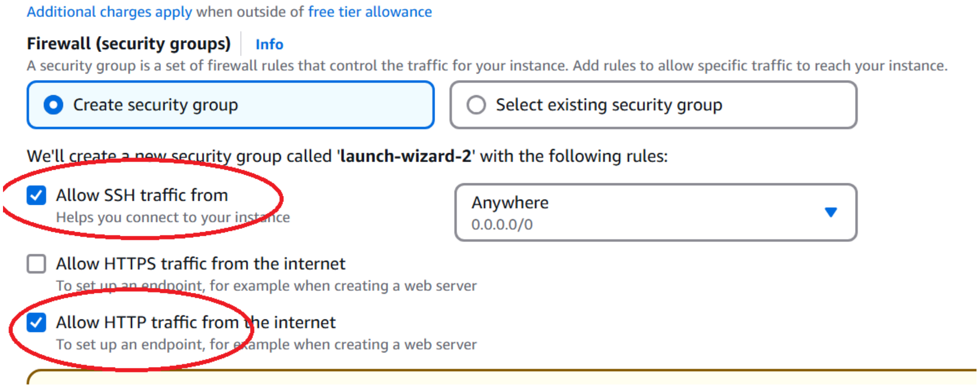
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volume

8. **Add Tags**:
   - Tag your instance for easier identification.
9. **Configure Security Group**:

- Allow HTTP (port 80) and SSH (port 22) from anywhere.



Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

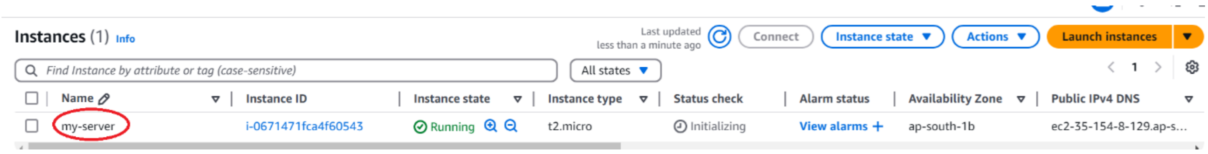
We'll create a new security group called '**launch-wizard-2**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

□ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

10. **Review and Launch**:
    - Select an existing key pair or create a new one for SSH access.
11. **Click "Launch Instances"**.

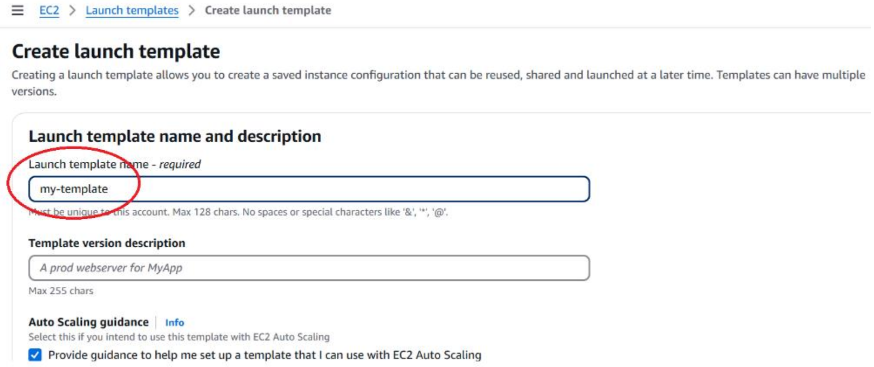


Instances (1) Info — Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
my-server	i-0671471fca4f60543	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-35-154-8-129.ap-s...

Step 2: Enable Auto Scaling

1. **Create a Launch Template**:
    - Go to the EC2 Dashboard > Launch template > Create Launch template.



EC2 > Launch templates > Create launch template

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*: my-template

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

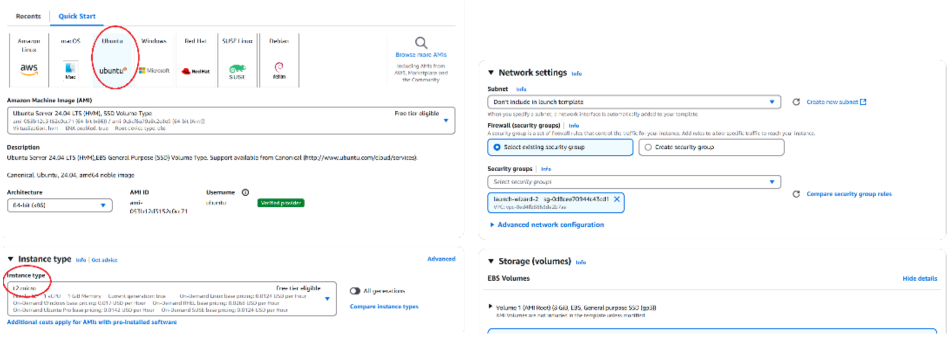
Template version description: *A prod webserver for MyApp*

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



Amazon Machine Image (AMI): Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible

Description: Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture: 64-bit (x86) — AMI ID: ami-053b12d3152c0cc71 — Username: ubuntu — Verified provider

Instance type: t2.micro — Free tier eligible

Additional costs apply for AMIs with pre-installed software

Network settings: Subnet: Don't include in launch template

Firewall (security groups): ● Select existing security group ○ Create security group

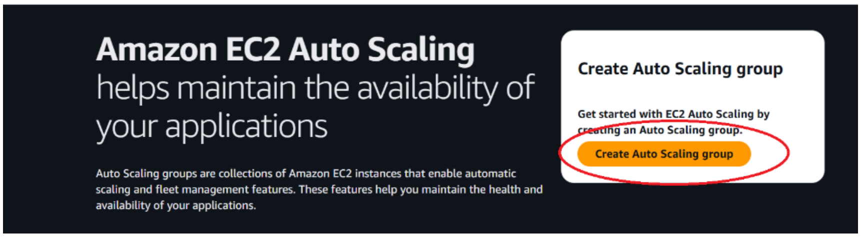
Security groups: launch-wizard-2 sg-0d8cee70944c43cd1

Storage (volumes): EBS Volumes — Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))

AMI Volumes are not included in the template unless modified

    - Use the same settings as your EC2 instance.
2. **Create an Auto Scaling Group**:

- Go to the EC2 Dashboard > Auto Scaling > Auto Scaling Groups > Create Auto Scaling Group.



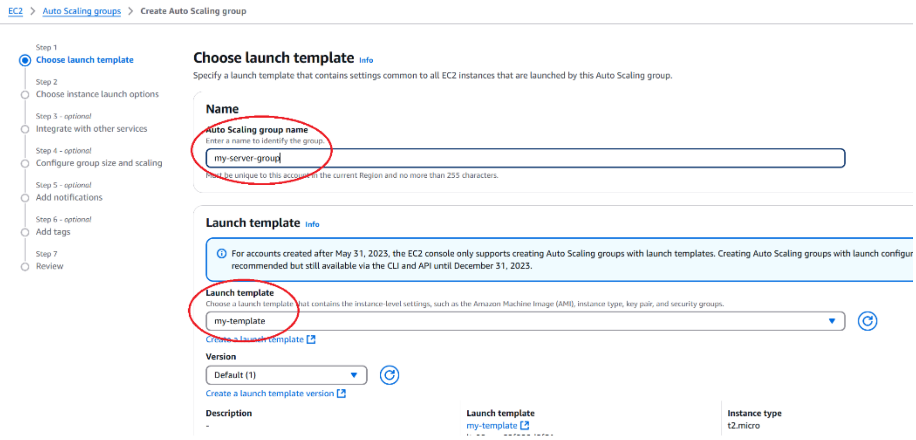
Amazon EC2 Auto Scaling helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

- Select the launch template created earlier.



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
Review

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.
my-server-group
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 configur recommended but still available via the CLI and API until December 31, 2023.

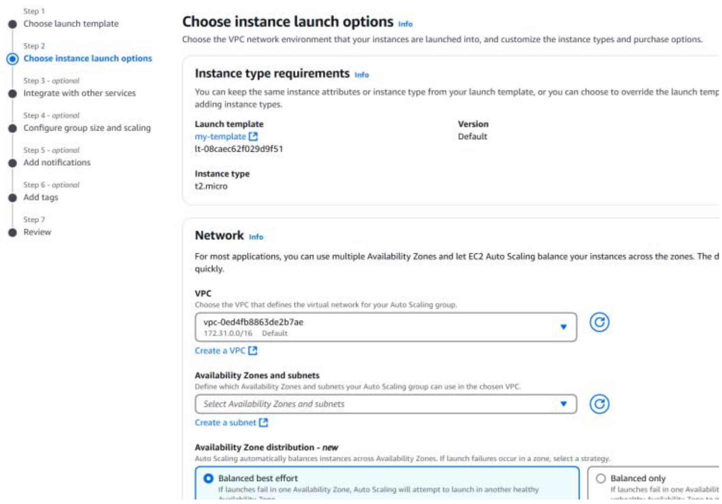
Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
my-template
Create a launch template

Version
Default (1)
Create a launch template version

Description
-

Launch template
my-template

Instance type
t2.micro



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
Review

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch temp adding instance types.

Launch template
my-template
lt-08caec62f029d9f51

Version
Default

Instance type
t2.micro

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The d quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-0ed4fb8863de2b7ae
172.31.0.0/16 Default
Create a VPC

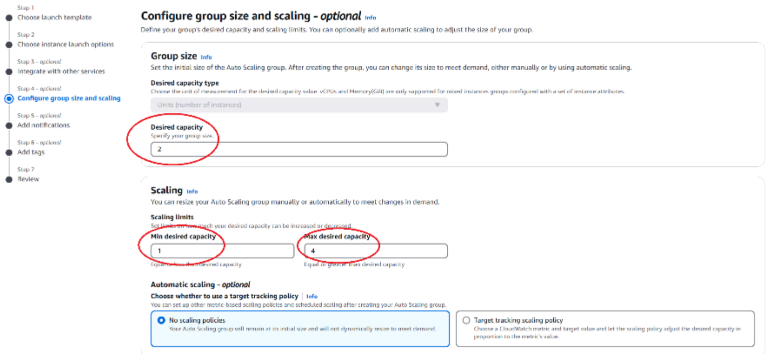
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
Create a subnet

Availability Zone distribution - *new*
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

● Balanced best effort
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy

○ Balanced only
If launches fail in one Availabilit

- Set the desired capacity, minimum, and maximum number of instances (e.g.,Min: 2, Max: 4).



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
Review

Configure group size and scaling - *optional* Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

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
1
Equal or less than desired capacity

Max desired capacity
4
Equal or greater than desired capacity

Automatic scaling - *optional*

Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

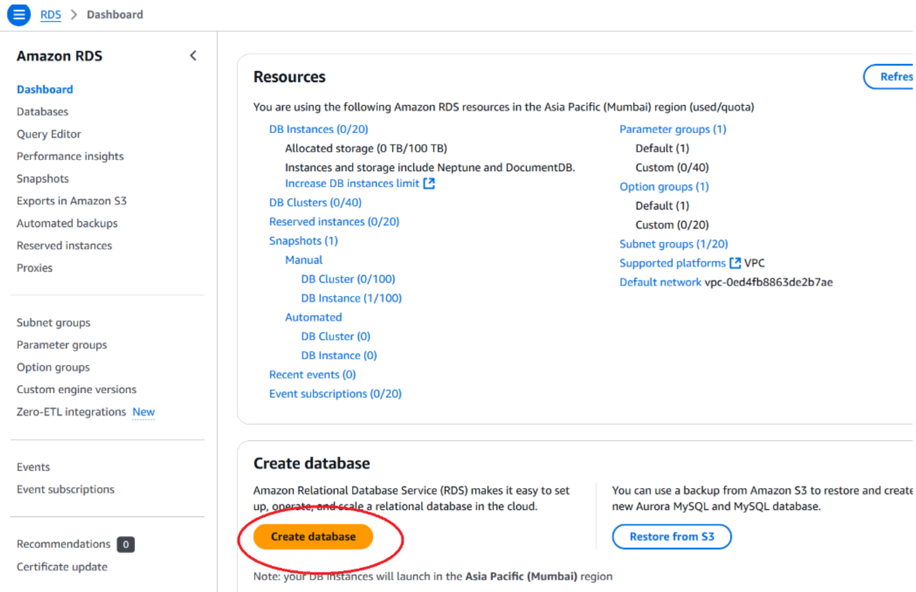
● No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

○ Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

- Configure scaling policies as needed.

Step 3: Create an RDS Instance

1. **Navigate to the RDS Dashboard**.
2. **Click on "Create Database"**.



RDS > Dashboard

Amazon RDS

Resources

You are using the following Amazon RDS resources in the Asia Pacific (Mumbai) region (used/quota)

- DB Instances (0/20)
  - Allocated storage (0 TB/100 TB)
  - Instances and storage include Neptune and DocumentDB.
  - Increase DB instances limit
- DB Clusters (0/40)
- Reserved instances (0/20)
- Snapshots (1)
  - Manual
    - DB Cluster (0/100)
    - DB Instance (1/100)
  - Automated
    - DB Cluster (0)
    - DB Instance (0)
- Recent events (0)
- Event subscriptions (0/20)
- Parameter groups (1)
  - Default (1)
  - Custom (0/40)
- Option groups (1)
  - Default (1)
  - Custom (0/20)
- Subnet groups (1/20)
- Supported platforms VPC
- Default network vpc-0ed4fb8863de2b7ae

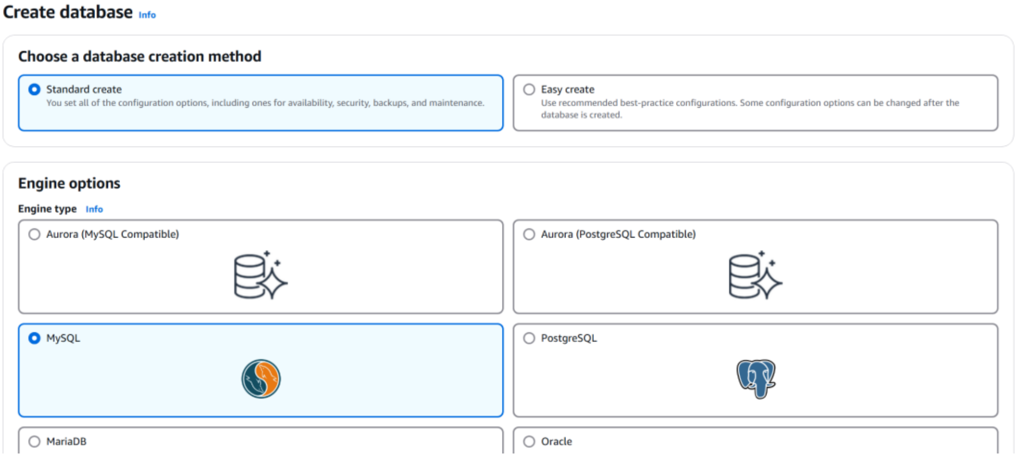
Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

You can use a backup from Amazon S3 to restore and create new Aurora MySQL and MySQL database.

Note: your DB instances will launch in the **Asia Pacific (Mumbai)** region

3. **Select MySQL as the database engine**.



Create database Info

Choose a database creation method

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

● MySQL

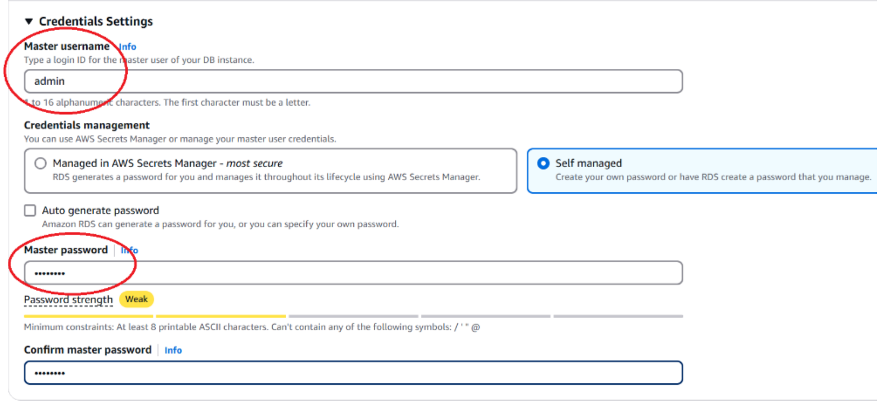
○ PostgreSQL

○ MariaDB

○ Oracle

4. **Configure the DB instance**:
   - Choose a DB instance class (e.g., `db.t2.micro`).
   - Set Master username as `admin` and password as `intel123`.

- Set the DB name as `intel`.



▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

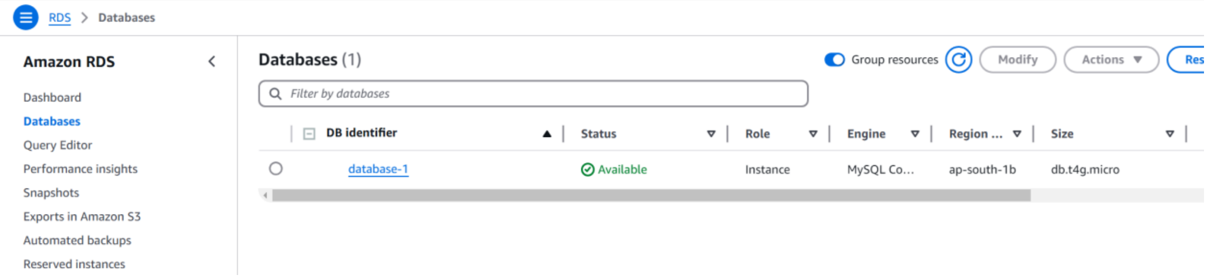
● Self managed
Create your own password or have RDS create a password that you manage.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password | Info
••••••••
Password strength Weak
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password | Info
••••••••

5. **Configure advanced settings**:
   - Ensure the DB is in the same VPC and subnets as your EC2 instances.
6. **Create the DB instance**.

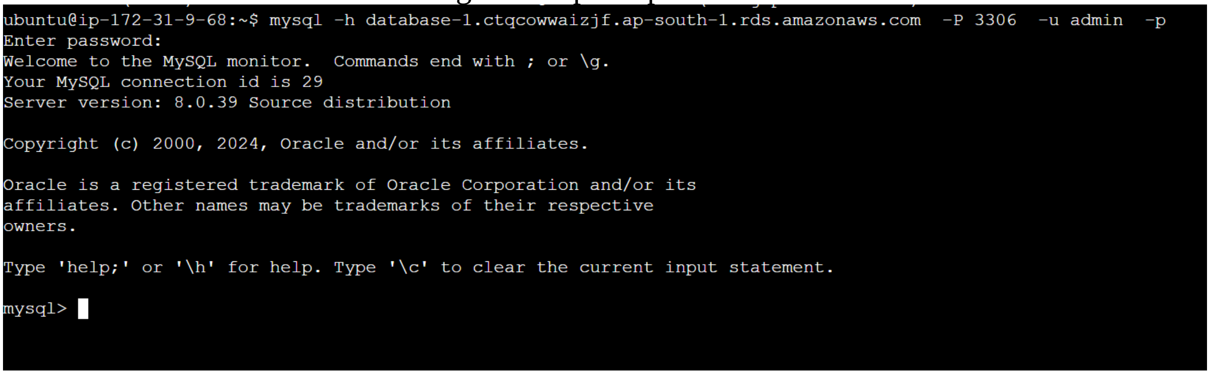


Databases (1)

DB identifier	Status	Role	Engine	Region ...	Size
database-1	✓ Available	Instance	MySQL Co...	ap-south-1b	db.t4g.micro

Step 4: Create Database & Table in RDS

1. **Connect to the RDS instance** using the endpoint provided in the RDS dashboard.

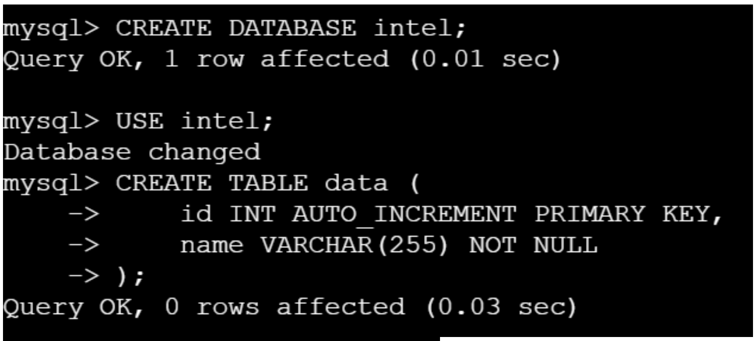


```
ubuntu@ip-172-31-9-68:~$ mysql -h database-1.ctqcowwaizjf.ap-south-1.rds.amazonaws.com  -P 3306  -u admin  -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> 
```

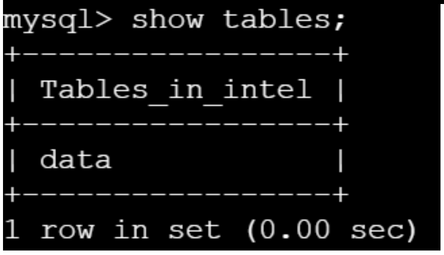
2. **Create the database and table**:

sql

```
CREATE DATABASE intel;
USE intel;
CREATE TABLE data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);
```



```
mysql> CREATE DATABASE intel;
Query OK, 1 row affected (0.01 sec)

mysql> USE intel;
Database changed
mysql> CREATE TABLE data (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)
```

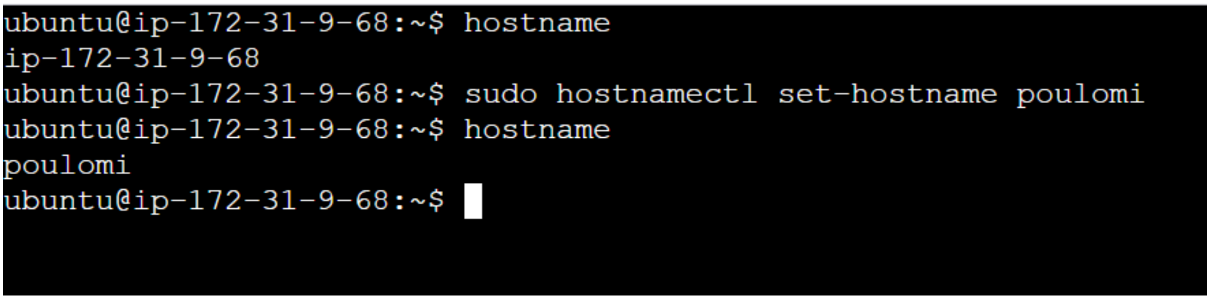


```
mysql> show tables;
+-----------------+
| Tables_in_intel |
+-----------------+
| data            |
+-----------------+
1 row in set (0.00 sec)
```

Step 5: Change Hostname in Website

1. **Connect to your EC2 instance** via SSH.
2. **Update the hostname**:

```
sudo hostnamectl set-hostname new-hostname
```

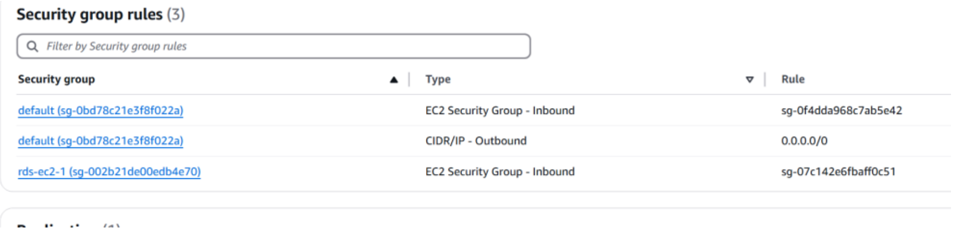


```
ubuntu@ip-172-31-9-68:~$ hostname
ip-172-31-9-68
ubuntu@ip-172-31-9-68:~$ sudo hostnamectl set-hostname poulomi
ubuntu@ip-172-31-9-68:~$ hostname
poulomi
ubuntu@ip-172-31-9-68:~$
```

3. **Update the website configuration** if needed.

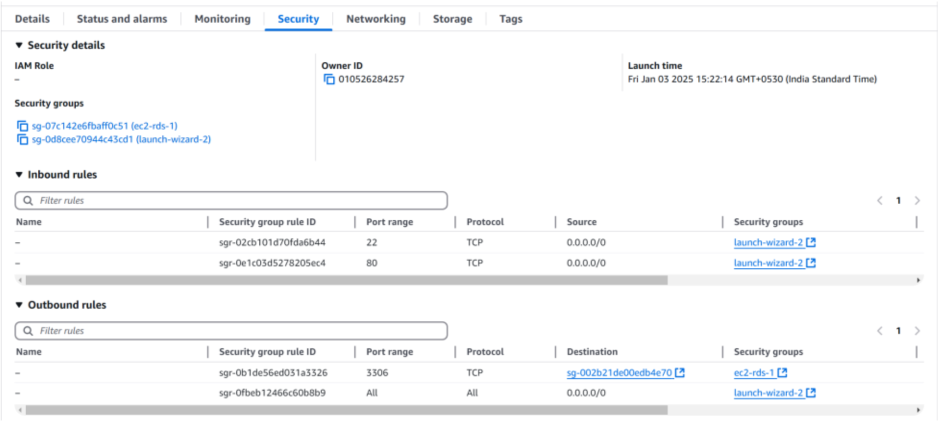
Step 6: Allow Traffic from EC2 to RDS Instance

1. **Modify the Security Group of the RDS instance**:
   - Allow inbound MySQL/Aurora traffic (port 3306) from the security group of your EC2 instances.



Security group rules (3)

Security group	Type	Rule
default (sg-0bd78c21e3f8f022a)	EC2 Security Group - Inbound	sg-0f4dda968c7ab5e42
default (sg-0bd78c21e3f8f022a)	CIDR/IP - Outbound	0.0.0.0/0
rds-ec2-1 (sg-002b21de00edb4e70)	EC2 Security Group - Inbound	sg-07c142e6fbaff0c51



▼ Security details

IAM Role
–

Owner ID
010526284257

Launch time
Fri Jan 03 2025 15:22:14 GMT+0530 (India Standard Time)

Security groups

sg-07c142e6fbaff0c51 (ec2-rds-1)
sg-0d8cee70944c43cd1 (launch-wizard-2)

▼ Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-02cb101d70fda6b44	22	TCP	0.0.0.0/0	launch-wizard-2
–	sgr-0e1c03d5278205ec4	80	TCP	0.0.0.0/0	launch-wizard-2

▼ Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
–	sgr-0b1de56ed031a3326	3306	TCP	sg-002b21de00edb4e70	ec2-rds-1
–	sgr-0fbeb12466c60b8b9	All	All	0.0.0.0/0	launch-wizard-2

Step 7: Allow All-Traffic to EC2 Instance

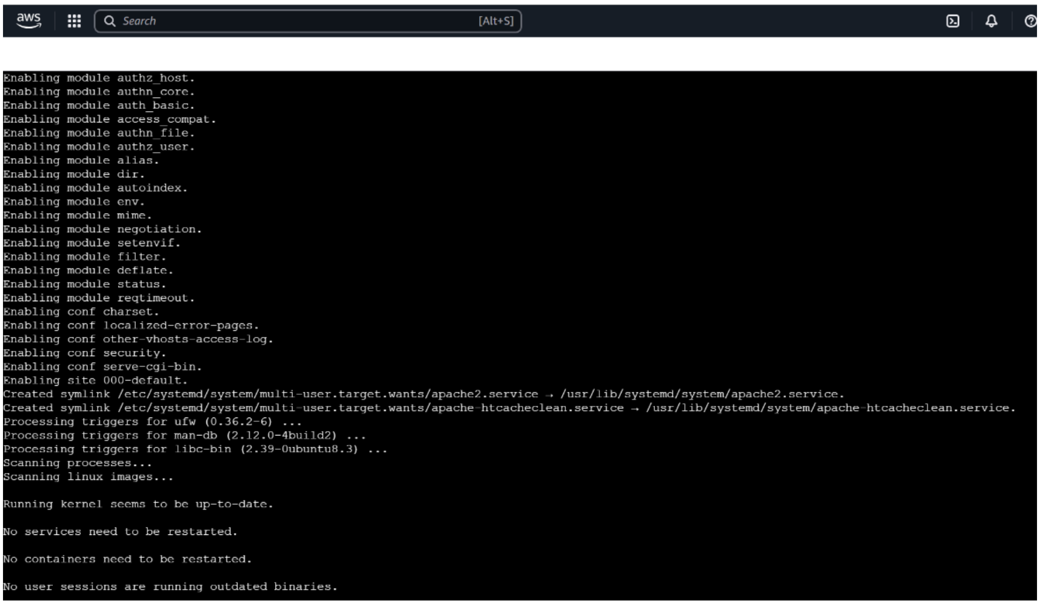
1. **Update the Security Group of the EC2 instances**:
   - Allow all necessary inbound traffic. For high availability, ensure HTTP (port 80) and HTTPS (port 443) are open.

Step 8: Download PHP file and enable connection between db and PHP file.

1. **Connect to your EC2 instance and update package and download apache**

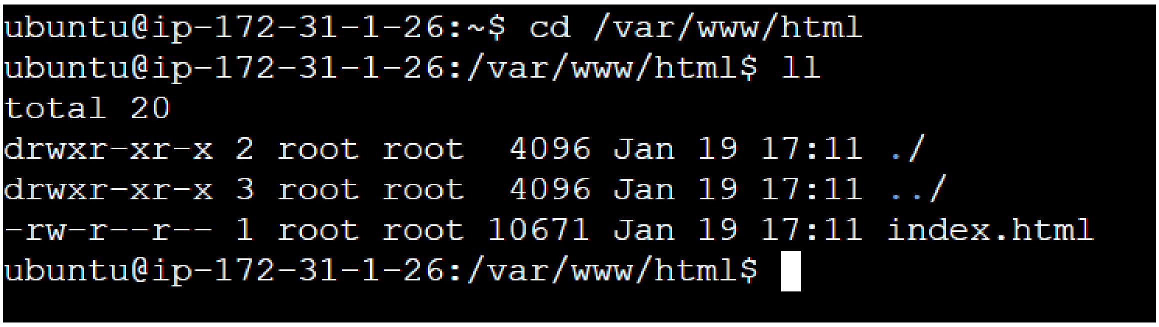
   sudo apt update

   sudo apt install apache2 -y

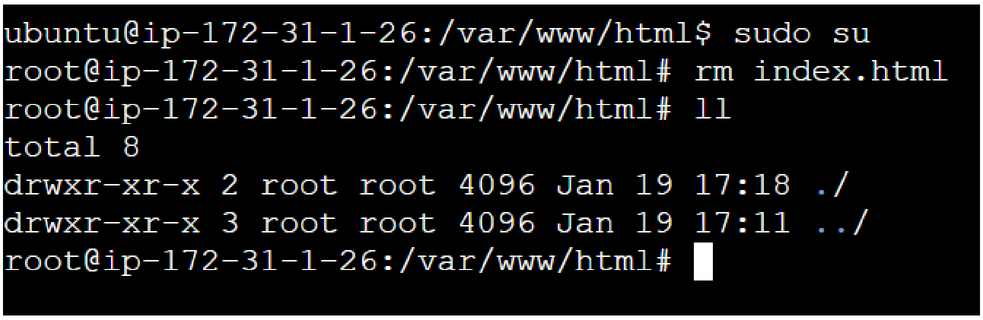


```
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
```

2. Go to cd /var/www/html and we can see a index file. We need to remove the file.
   rm index.html

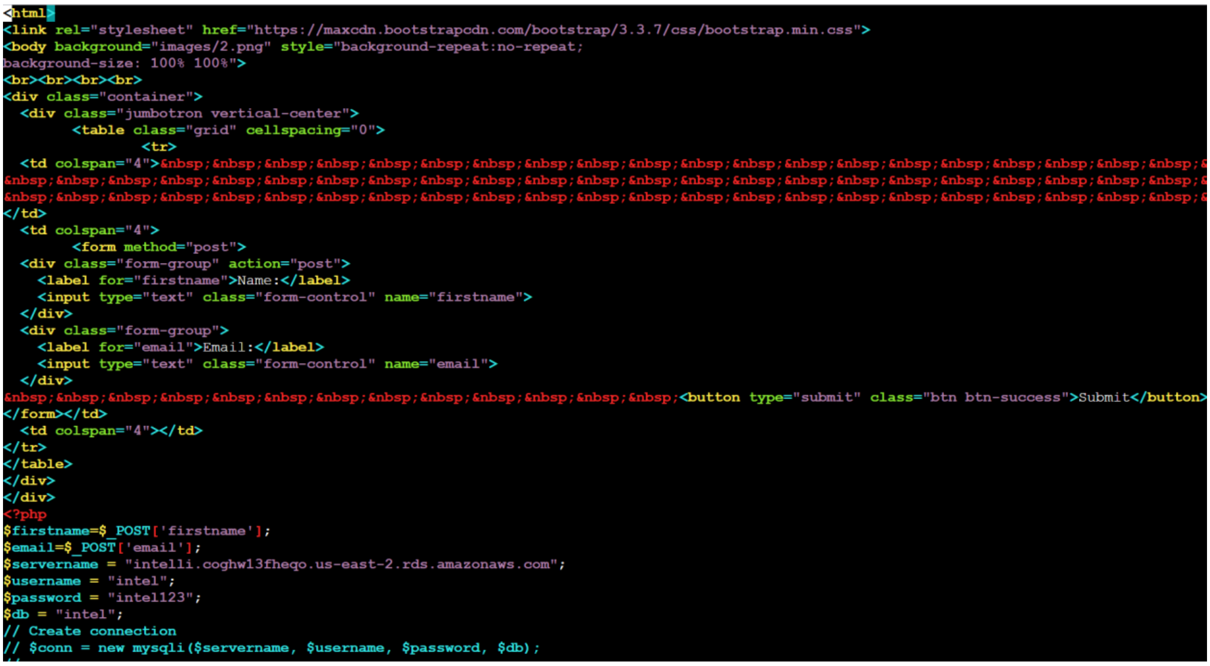


```
ubuntu@ip-172-31-1-26:~$ cd /var/www/html
ubuntu@ip-172-31-1-26:/var/www/html$ ll
total 20
drwxr-xr-x 2 root root  4096 Jan 19 17:11 ./
drwxr-xr-x 3 root root  4096 Jan 19 17:11 ../
-rw-r--r-- 1 root root 10671 Jan 19 17:11 index.html
ubuntu@ip-172-31-1-26:/var/www/html$ 
```



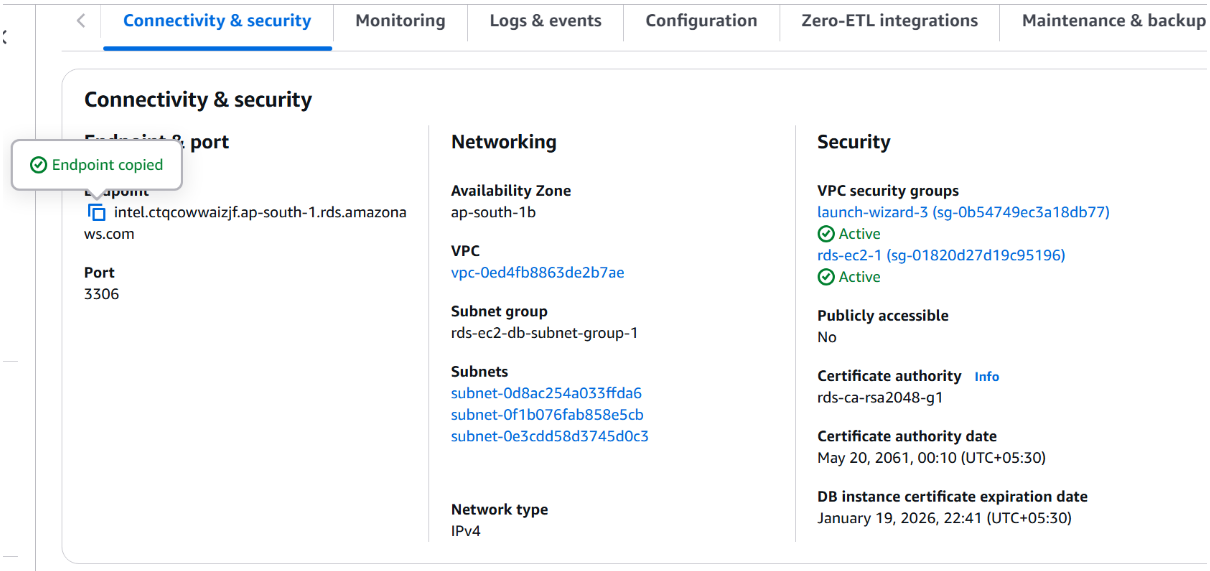
```
ubuntu@ip-172-31-1-26:/var/www/html$ sudo su
root@ip-172-31-1-26:/var/www/html# rm index.html
root@ip-172-31-1-26:/var/www/html# ll
total 8
drwxr-xr-x 2 root root 4096 Jan 19 17:18 ./
drwxr-xr-x 3 root root 4096 Jan 19 17:11 ../
root@ip-172-31-1-26:/var/www/html# 
```

3. Create a new index file and paste the PHP code in it.
   Vi index.php and paste the contents.



```
<html>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<body background="images/2.png" style="background-repeat:no-repeat;
background-size: 100% 100%">
<br><br><br><br>
<div class="container">
  <div class="jumbotron vertical-center">
        <table class="grid" cellspacing="0">
                <tr>
  <td colspan="4">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
</td>
  <td colspan="4">
        <form method="post">
  <div class="form-group" action="post">
    <label for="firstname">Name:</label>
    <input type="text" class="form-control" name="firstname">
  </div>
  <div class="form-group">
    <label for="email">Email:</label>
    <input type="text" class="form-control" name="email">
  </div>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="submit" class="btn btn-success">Submit</button>
</form></td>
  <td colspan="4"></td>
</tr>
</table>
</div>
</div>
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "intelli.coghw13fheqo.us-east-2.rds.amazonaws.com";
$username = "intel";
$password = "intel123";
$db = "intel";
// Create connection
// $conn = new mysqli($servername, $username, $password, $db);
```

4. **Go to your database and copy the endpoint and paste the endpoint in the server name provided in the php file. Save and exit**



Connectivity & security

Endpoint & port

Endpoint copied

Endpoint
intel.ctqcowwaizjf.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1b

VPC
vpc-0ed4fb8863de2b7ae

Subnet group
rds-ec2-db-subnet-group-1

Subnets
subnet-0d8ac254a033ffda6
subnet-0f1b076fab858e5cb
subnet-0e3cdd58d3745d0c3

Network type
IPv4

Security

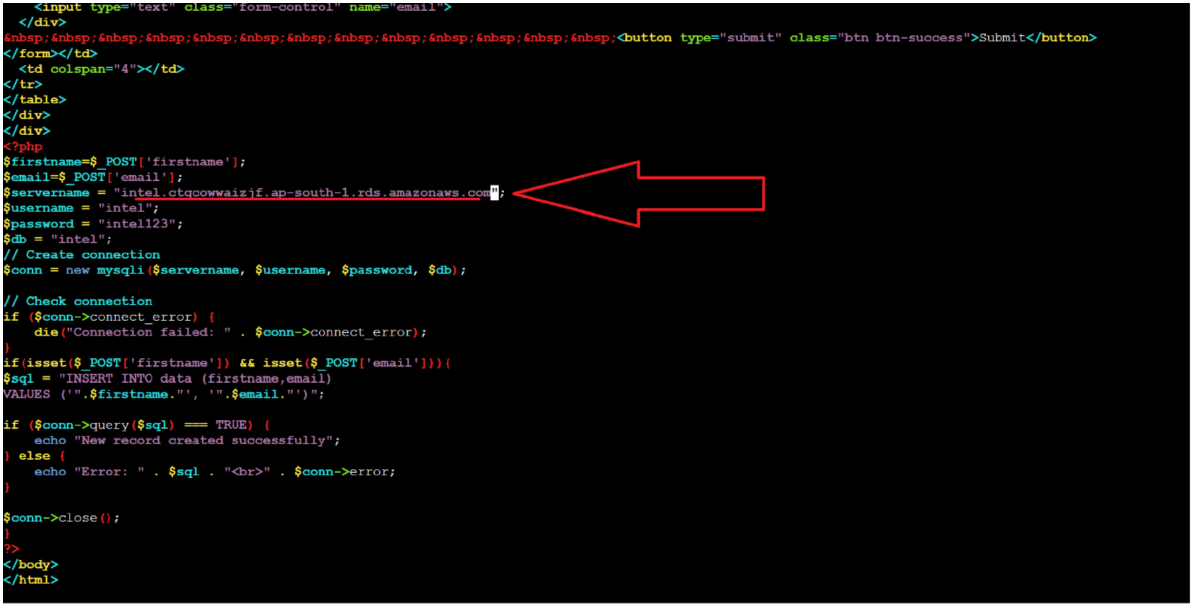
VPC security groups
launch-wizard-3 (sg-0b54749ec3a18db77)
Active
rds-ec2-1 (sg-01820d27d19c95196)
Active

Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 20, 2061, 00:10 (UTC+05:30)

DB instance certificate expiration date
January 19, 2026, 22:41 (UTC+05:30)

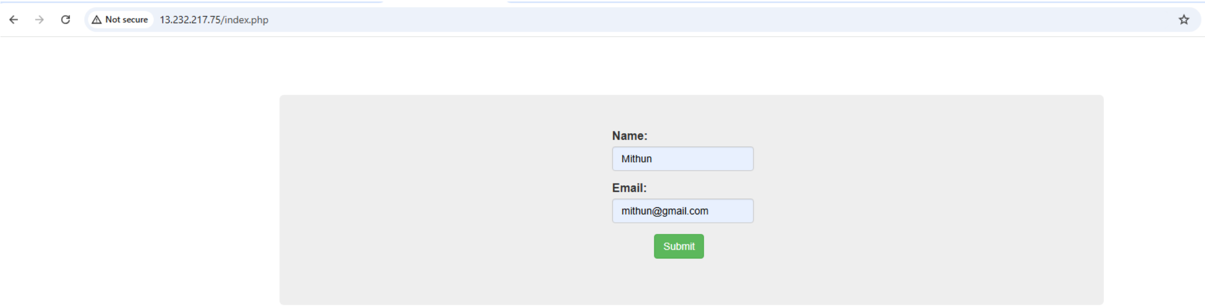


```
    <input type="text" class="form-control" name="email">
  </div>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<button type="submit" class="btn btn-success">Submit</button>
</form></td>
  <td colspan="4"></td>
</tr>
</table>
</div>
</div>
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "intel.ctqcowwaizjf.ap-south-1.rds.amazonaws.com";
$username = "intel";
$password = "intel123";
$db = "intel";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if(isset($_POST['firstname']) && isset($_POST['email'])){
$sql = "INSERT INTO data (firstname,email)
VALUES ('".$firstname."', '".$email."')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
?>
</body>
</html>
```

5. **Install mysql and PHP .**
   Apt install php -y

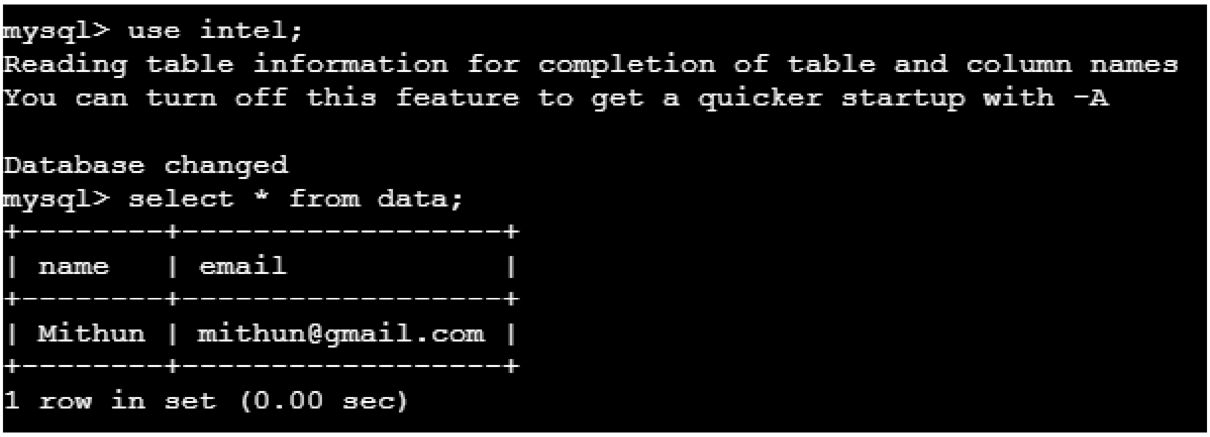
   sudo apt install mysql-server -y

6. As we give the public ip address of our instance we can see the php file running on apache and whatever data we add here I will be automatically save in the db and we can see it from our ec2 query.



Name: Mithun

Email: mithun@gmail.com



```
mysql> use intel;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from data;
+--------+------------------+
| name   | email            |
+--------+------------------+
| Mithun | mithun@gmail.com |
+--------+------------------+
1 row in set (0.00 sec)
```